IBM Cloud Pak for AIOps 4.8.0 - PDF 1



© Copyright IBM Corp. 2025.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Tables of Contents

IBM Cloud Pak for AIOps 4.8.0 1

Release notes 1
- What's new 1
  - Known issues and limitations 4
  - Deprecated and removed functions 25
  - Frequently Asked Questions 27

Overview 30
- Architecture 36
  - IBM Cloud Pak for AIOps architecture 36
  - Infrastructure Automation architecture 39
- Regulatory compliance 41
  - Federal Information Processing Standards (FIPS) 41
    - Enabling FIPS support for IBM Cloud Pak for AIOps 42
    - Enabling FIPS support for Infrastructure Automation 44
    - Enabling FIPS support for a Netcool integration 45
  - Audit logging 47
    - Logged actions and data 48
    - Audit log messages 51
    - Enabling Audit for Infrastructure Automation {{site.data.keyword.infra_mgmt_manageservices}} 55
  - Security and Privacy by Design (SPbD) 56
  - GDPR readiness 56
  - Data retention 59

Getting started 60
- Roles and archetypes 60
- IBM Cloud Pak for AIOps Workflows 61

Deploying on Linux 63
- Planning 63
  - Incremental adoption ({{site.data.keyword.cp4waiops_notm}}) 67
  - Configuring local volumes 68
- Online installation 70
- Offline installation 74
- Online upgrade 79
- Offline upgrade 81
- Administering 84
  - Adding a control plane node 84
  - Adding a worker node 85
  - Backup and restore 86
  - Configuring topology resources for production 86
  - Recovering from node failure 87
  - Removing a node 89
  - Resizing storage 90
  - Restarting the environment 90
  - Updating the deployment type 95
  - Using a custom certificate 96
  - MustGather tool 96
- Uninstalling 97

Deploying on OpenShift 97
- Planning 98
  - Incremental adoption ({{site.data.keyword.cp4waiops_notm}}) 98
  - Supported platforms 99
  - Hardware requirements 101
    - Performance considerations for metrics data collection 104
    - Performance considerations for logs data collection 105
    - Performance considerations for events data collection 105
    - Performance considerations for tickets data collection 105
    - Performance considerations for observers 106
  - Storage requirements 106
  - Network requirements 111
  - High availability considerations 113
    - Multi-zone HA 113
    - Cold standby DR 115
      - Backing up IBM Cloud Pak for AIOps 116
      - Restoring IBM Cloud Pak for AIOps 118
      - Troubleshooting 122
  - Security considerations 124
    - Verifying images 125
    - Downloading or listing images 126

Licensing 127
Passport Advantage part numbers 128
Supporting products and extensions 130
Installing 130
Starter installation 131
Production installation 138
Online (console) 138
Online (CLI) 146
Offline 155
Bastion host 157
Portable device 169
Installing on cloud platforms 182
Installing on ARO 182
Installing on GCP 185
Installing on ROKS 187
Installing on ROSA 187
Installing on AWS GovCloud 191
Upgrading 193
Online (console) 194
Online (CLI) 198
Offline 203
Administering 208
Backup and restore 209
Backup and restore with OADP 209
Installing the backup and restore tools 211
Configuring ODF as backup storage for OADP 216
Backing up Cloud Pak for AIOps 219
Restoring Cloud Pak for AIOps 224
Upgrading the backup and restore artifacts 236
Backup and restore with Fusion (Cold Standby DR) 115
Backing up IBM Cloud Pak for AIOps 116
Restoring IBM Cloud Pak for AIOps 118
Troubleshooting 122
Configuring topology resources for production 245
Health checks and monitoring 246
Installation and upgrade health checks 248
Proactive health checks 249
Reactive health checks 249
Self-monitoring with alerts 250
Self-monitoring with topology 255
Monitoring with Prometheus and Grafana 261
Cloud Pak for AIOps health dashboard 269
Cloud Pak for AIOps usage dashboard 294
Monitoring data processed 313
Circuit breaker for Kafka robustness and stability 316
Creating a cluster monitoring ConfigMap object 316
Creating a user-defined workload monitoring ConfigMap 316
Configuring core OpenShift monitoring components 317
Configuring components that monitor user-defined projects 318
Creating Prometheus rule to generate OpenShift alerts 318
Push alerts generated in OpenShift into {{site.data.keyword.cp4waiops_short}} 321
Creating a policy to promote alerts to an incident 323
Managing certificates 324
Using a custom certificate 324
Multi-zone rebalancing 325
Resizing storage 328
Restarting the environment 331
Secure Tunnels 336
Installing Secure Tunnel 338
Installing the Connector 342
Upgrading Secure Tunnel 346
Creating Secure Tunnel Connections 347
Viewing Secure Tunnel 363
Modifying Secure Tunnel 364
Auditing Secure Tunnel 365
Updating the deployment type 368
Uninstalling 369
Infrastructure Automation 381

Defining integrations 381
Available integrations 381
Netcool integrations 388

IBM Tivoli Netcool/Impact integration 389
Configuring an integration in IBM Tivoli Netcool/Impact 393
IBM Tivoli Netcool/OMNIbus integrations 395
Configuring launch-in-context from IBM Tivoli Netcool/OMNIbus 400
Adding incidents and alerts to the DASH navigation menu 400
Creating tools for launch-in-context from the Event Viewer 401
Creating a TLS-enabled integration to an OpenShift ObjectServer 401
Filter clause 402
Netcool event load management 403
Netcool connector troubleshooting 404
Netcool event load splitting calculation 404
Integrations 408
Integrations UI 409
Integration types 410
Outgoing integrations 411
Microsoft Teams integration 411
Configuring Microsoft Teams application for integration 415
Slack integration 416
Configuring Slack for integration 419
ServiceNow App integration 421
Creating a Change Risk Assessment 427
ServiceNow App privacy policy 427
SSH integrations 428
Installing new SSH key during migration 430
SSH algorithms 434
Email Notification integration 434
GitHub integration 439
Ansible integrations (bi-directional) 442
IBM Db2 integration 443
Incoming integrations 444
Appdynamics integrations 445
AWS CloudWatch integrations 450
Custom integrations 454
Creating custom integrations using the integration SDK 461
Creating dashboards using the dashboard extension toolkit 461
Dynatrace integrations 461
ELK integrations 472
Generating API Key for ELK integrations 478
EFK integrations 478
Falcon LogScale integrations 483
Generic webhook integration 487
Integrating with IBM SevOne Network Performance Management 490
Integrating with IBM Turbonomic 491
Integrating with Zabbix 493
Example: IBM SevOne NMS mappings 496
Example: Prometheus AlertManager mappings 499
Infrastructure Management integrations 502
Instana integrations 504
Jira integrations 509
Kafka integrations 511
Metric Manager integrations 517
Mezmo integrations 518
New Relic integrations 523
Observer integrations 528
PagerDuty integrations 528
ServiceNow integration 531
Mapping similar ticket data from ServiceNow 539
Splunk integrations 541
Zabbix integrations 547
Delay configurations in data integrations 553
Mapping data 553
Log data 554
Event data 560
Performance considerations for metric data collection 563
Loading data for AI training 564
Loading data from domain-specific log sources 564
Deduplication 565
HTTP headers for integration credentials 565
Remote deployment of integrations 566
Observer jobs 566

ALM observer jobs 570
Ansible AWX observer jobs 572
AppDynamics observer jobs 573
AWS observer jobs 574
Azure observer jobs 575
Big Cloud Fabric observer jobs 576
BigFix Inventory observer jobs 578
Ciena Blue Planet observer jobs 579
Cisco ACI observer jobs 580
Contrail observer jobs 583
Datadog observer jobs 585
DNS observer jobs 586
Docker observer jobs 587
Dynatrace observer jobs 589
File observer jobs 591
GoogleCloud observer jobs 594
HP NFVD observer jobs 595
IBM Cloud observer jobs 599
Instana observer jobs 600
Migrating from Instana connector to observer 601
Jenkins observer jobs 602
Configuring Jenkins observer job plug-in 603
More Jenkins plug-in customization tasks 605
Customizing the Jenkins topologies 606
Creating Jenkins topology templates 607
Juniper CSO observer jobs 607
Kubernetes observer jobs 609
Network Manager observer jobs 616
New Relic observer jobs 617
OpenStack observer jobs 619
Rancher observer jobs 622
REST observer jobs 623
SDC ONAP observer jobs 627
ServiceNow observer jobs 629
SevOne observer jobs 632
Tivoli Application Dependency Discovery Manager observer jobs 633
Viptela observer jobs 635
VMware NSX observer jobs 636
VMware vCenter observer jobs 637
Zabbix observer jobs 639
Configuring observer job security 640
Event Manager integrations 642
Resource requirements for running probes 642
Connecting with on-premises probes 643
Configuring the Cloud system 645
Configuring the on-premises system 647
IBM SevOne NPM probe integration 648
IBM SevOne NPM probe integration 648
IBM Turbonomic probe integration 650
Datatog probe integration 654
Probe Common Tasks 657
Individual probe integration PDF documentation 658

Viewing AIOps Insights 658
AIOps Insights UI 658
AIOps Insights Scopes and Filter 662

Training AI models 664
About AI models 665
Prerequisite AI models 669
Preparing the AI environment 670
Adjusting model training resources 670
Configuring alert expiry time 670
About Elasticsearch indices and shards 671
Determining your current shard consumption 671
Managing embedded JSON 672
Scaling log anomaly detection pods 673
Configuring log formats for WebSphere log anomaly detection 674
About AI algorithms 674
About change risk 675
About event grouping 675
About log anomaly detection algorithms 676
About natural language log anomaly detection 678
Workflow 678
Example 681
About statistical baseline log anomaly detection 685

Workflow 686
Example 687
About golden signals log anomaly detection 691
About shape correlation 695
About metric anomaly detection 696
About golden signal alert enrichment 697
About probable cause ranking 698
About similar tickets 700
Language support 700
Preparing AI algorithms 701
About algorithm types 701
About pre-trained algorithms 701
Configuring trainable AI algorithms 702
Setting up training for alert seasonality detection 702
Change risk 703
Setting up training for log anomaly detection - golden signals 708
Natural language log anomaly detection 709
Metric anomaly detection 714
Similar tickets 714
Temporal grouping 716
Managing AI algorithms 717
Generating AI models 717
Displaying log anomaly detection resources and models 718
Data quality messages 719
Managing unsupported languages 721
Determining which algorithm generated a log anomaly 722
Deploying models 723
Editing algorithm configurations 723
Deleting algorithm configurations 724
Managing AI model versions 725
Changing the deployed version 725
Deleting a version 726
Draft and archived policies 727
Displaying data assets 728

Managing applications (services) and viewing topologies 729
Resource management: Searching and filtering 730
The Resource management page 733
Defining and editing applications (services)) 735
Managing topologies 737
Rendering a topology 738
Building a custom topology 740
Viewing a topology 741
Viewing a resource on a map 749
Viewing topology history 750
Viewing topologies with URLs 751
Viewing topologies with API 753
Topology query syntax 755
Template types 757
Viewing resource group templates 758
Building resource group templates 758
Incidents 760
Defining business criticality 762
Running data administration routines 762
Configuring topology user preferences 763
Configuring topology elements 765
Configuring advanced settings 765
Configuring topology tools 768
Splunk integration example 772
Configuring resource types 773
Configuring relationship types 775
Configuring custom icons 776
Configuring rules 777
Defining merge rules 778
Defining match tokens rules 780
Defining tags rules 782
Defining history rules 784
Defining file enrichment rules 784
Defining events filter rules 787
Defining business criticality rules 787
Rules reference 788

Resolving incidents and alerts 789

About events, alerts and incidents 789
Incidents 790
Incident overview 792
Searching and filtering incidents 798
Creating views 802
Setting user preferences for the Incidents list 804
Managing multiple incidents 805
Alerts 806
Icons used in the Alert Viewer 808
Alert details 808
Working with alerts and menu customization 812
Searching and filtering alerts 817
Managing multiple alerts 825
Creating views 825
Creating quick filters 828
Using custom fields 829
Setting user preferences for the Alert Viewer 830
About alert groups 832
Displaying an alert group 832
Displaying alert topology 832
Displaying alert seasonality 833
Displaying temporal correlation 835
Log anomaly alerts 836
High risk alerts from change risk 837
Viewing metric anomaly details 838
Metric anomaly forecasts 841
Viewing multiple metrics on the metric anomaly chart 841
Using Metric search to create a timeseries chart 843
Blocking, unblocking, and purging metrics 845
Displaying analytics details for an alert group 845
Viewing incidents that are displayed in outgoing integrations 846
Alert and incident enrichment 847
Event mapping and alert severity 847
ChatOps insights 848
Managing Incidents in ChatOps 849
Probable cause in ChatOps 851
Incident Resources 852
Associated automations 852
Recommended actions 853
Viewing alerts in ChatOps 854
Change risk in ChatOps 857

Automating processes 858
Policies 859
Searching and filtering policies 860
Policy details 861
Seasonality policy details 862
Temporal policy details 863
Default incident creation policies 864
Default Netcool scope-based grouping 865
Incident Creation based on Persistence and Golden Signal 866
Creating policies 867
Suppress alerts 867
Suppress seasonal alerts 869
Group alerts based on scope 869
Assign a runbook to alerts 871
Promote alerts to an incident 872
Invoke an IBM Tivoli Netcool/Impact policy 874
Populate a dashboard or report 876
Policy editor 879
Supported Anomaly insights properties 883
Examples of policy conditions mapped to alert JSON 885
Best practices for condition sets 887
Refreshing the policies table 887
Managing multiple policies 887
Renaming policies 888
Archiving policies 888
Migrating Netcool correlations to Cloud Pak for AIOps 888
Runbooks 890
Runbooks and Actions 891
Runbooks Library 891
Run your first runbook 891
Load and reset example runbooks 892
Create a runbook 894
Edit a runbook 894

Runbook elements and editing actions 895
Adding actions 896
Adding nested runbooks 897
Adding parameters 898
Parameter types 899
Adding a GOTO element 900
Configuring runbook creation 900
Configuring runbook approval 901
Runbook approval process 901
Runbook versions 902
Save and publish a runbook 902
Filtering runbooks 903
Assigning runbooks to groups 903
Export and import runbooks 903
Delete a runbook 905
Activities 905
Monitor runbook history 906
Actions 906
Managing actions 906
Creating an action 907
Creating SSH actions 908
Single and multi target actions 910
Output format of multi target actions 910
Comparison of single and multi target actions 911
Creating HTTP actions 911
Creating Client-side actions 912
Deleting an action 913
Ansible actions 913
Harvesting Ansible actions 914
Ansible action interactions 914
Job Template parameters and surveys 914
Predefined actions 915
Action versions 916
Save and publish an action 916
Testing an action 916
Running an action 917
Activities 917

Visualizing analytics 918
Integrating IBM Cognos Analytics with {{site.data.keyword.cp4waiops_notm}} 919
Data flow configuration 920
Creating a Data Module 921
Creating a dashboard 921
IBM Cloud Pak for AIOps widgets 923
Alert list custom widget 923
Topology viewer custom widget 924
Incident list custom widget 924
Button custom widget 925
Example configurations 926

Cloud Pak administration 927
UI console 927
Supported browsers 927
Accessing a cluster with the console 928
Customizing the platform UI 928
Enabling and disabling tours 928
Disabling auto complete on the login page 929
Customizing the Platform UI (Zen) route 929
Updating default routes to custom hostname 930
Customizing the branding of the web client 931
Customizing the home page 932
Supported languages 933
Managing user access control 934
Identity Management (IM) 935
Multiple LDAP domains 936
Authentication types 936
Delegating authentication to OpenShift 938
OIDC registration by using client CRD 939
Auditing IM service 940
Identity Management for IBM Cloud Pak for AIOps platform users 942
Configuring single sign-on using SAML 942
Configuring single sign-on using IBM Cloud Pak for AIOps UI 943
Configuring single sign-on using OpenID Connect 945

Configuring LDAP connection 947
Custom search base support for LDAP group and user entity in SCIM group and user APIs 951
Using realName as a query filter 952
Enabling LDAP Nested Search 953
Changing LDAP cache settings 958
Changing LDAP search settings 959
Changing SCIM LDAP search limit settings 961
Adding custom SCIM attributes 962
Updating SCIM LDAP attributes mapping 965
Enhancing SCIM group and user API performance 968
SCIM pagination support for LDAP server 970
Changing LDAP contextPool settings 971
Cluster configurations 972
Allowing limited users to log in by using only LDAP 972
Updating custom hostname and TLS secret by using a configmap 975
Changing the cluster administrator access credentials 979
Customizing the cluster access URL 981
Adding a logout redirect URL 983
Configuring access and identity token validity 984
Changing the security role mappings refresh time interval that is used during authorization 985
Changing LDAP search cache variable values 986
Using a proxy server for outbound traffic 987
IM sample configuration 988
SCIM configuration by using IBM Cloud Pak for AIOps UI 988
IBM Cloud Pak SCIM OKTA integration 989
IBM Cloud Pak SCIM AZURE AD integration 996
IBM Cloud Pak JIT support for SCIM OKTA and Azure AD 1004
Integrating SAML with Azure Active Directory 1006
EDB postgresql database backup and restore 1009
Backing up EDB postgresql database 1009
Restoring EDB postgresql database 1010
Adding users 1011
Managing user groups 1013
Roles and permissions 1015
Managing profiles 1022
Creating restriction filters 1025
Creating redaction policies 1031
Managing profiles 1022
Creating restriction filters 1025
Creating redaction policies 1031
Setting your preferences 1050
Secret rotation 1051
Issue Resolution Core operator secret usage 1051
Issue Resolution Lifecycle operator secret usage 1058
Administering data storage 1059
Restricting the number of alerts returned by the data layer 1061
Disabling additional alert count queries 1062
Updating usage data collection preferences 1063
Clearing out topology data without reinstalling 1064
Setting the idle session timeout 1066
Configuring probable cause 1067

Technology preview 1068

Reference 1068
Operator cluster service versions (CSVs) 1068
Operator permissions (IBM Cloud Pak for AIOps) 1069
Operator permissions (Infrastructure Automation) 1100
Alert and Event schemas 1109
Alerts and Events schema fields 1111
Resource schemas 1117
EventType schemas 1118
Normalized Alert Mapping Rules 1118

Troubleshooting 1125
Install and upgrade 1126
Clusters and nodes 1133
Datastores 1139
Integrations 1140
Secure Tunnel 1151
AI model management 1154
Alert management 1159
Runbook Automation 1163
ChatOps 1163

Topology 1167
Viewing the topology logs 1169
Setting observer log levels 1169
Defining classifier log settings 1171
Troubleshooting Identity Management (IM) issues 1171
Login 1172
Internal server error while login Platform UI by using SAML with LDAP dependency 1172
Cannot authenticate Kubeadmin user as CloudPakAdministrator by default 1173
Cannot log in to the console by using the admin username 1173
Docker login results in unencrypted password warning 1173
Login too slow or times out, or invalid username or password error 1174
Cannot log in to the console by using {{site.data.keyword.open_s}} authentication 1174
Cannot log in to the console after reinstallation of {{site.data.keyword.cs}} 1175
Default admin cannot log in due to LDAP server issues 1176
Username shows `Undefined Undefined` after login to CloudPak dashboard using SAML login option 1177
Gateway timeout error while you log in to the console 1178
LDAP 1179
Enable debugging for user authentication issues 1179
Troubleshooting LDAP configuration 1180
Configuring LDAP over SSL 1185
Troubleshooting users and user groups search issues 1188
Timeout error while setting up an LDAP connection 1190
Timeout error after enabling LDAP Nested search for Microsoft Active Directory 1191
Client registration failure in Platform UI console 1191
Intermittent login failure in Platform UI console 1192
LDAP nested search for Microsoft Active Directory does not work correctly 1192
Pod issues 1193
auth-idp pod restarts multiple times 1193
Pods are not scheduled 1194
secret-watcher and security-onboarding pods fail to start 1194
Secret watcher pod shows CrashLoopBackOff status 1195
OIDC 1195
OIDC client creation is not successful 1196
OIDC registration fails to update 1196
API calls 1197
Error 403 as a response while calling `/iam-token/serviceids/` APIs 1197
Cannot generate IAM access tokens through API for external OIDC and SAML users 1198
Troubleshooting certificates 1198
Infrastructure Automation 1200
Troubleshooting installing and uninstalling Infrastructure Automation 1200
Troubleshooting Infrastructure Management 1200

Support 1202
Opening a support case 1202
Installing and running the MustGather tool (Linux) 96
Installing the MustGather tool (OpenShift) 1204
Running the MustGather tool 1205
Healthcheck 1208
Data collection modes 1214
Advanced tips 1216
Security bulletins and fixes 1221
Fixed Support originated issues 1225

Notices 1226

Glossary 1228

IBM Cloud Pak for AIOps

Learn more

Overview

Getting started

Installation

Planning

System requirements

Installing

Preparing to install

Installing the Cloud Pak

Installing supporting products and extensions

Features

Integrations

Application management

AI modelling and training

Topology viewing

User access control management

ChatOps insight management

Help and support

IBM Cloud Technology Slack channel

Troubleshooting

Support

IBM Cloud Pak® solutions closely align with other industry-leading software providers. While IBM values the use of inclusive language, terms that are outside of IBM's direct influence, for the sake of maintaining user understanding, are sometimes required. As other industry leaders join IBM in embracing the use of inclusive language, IBM will continue to update the documentation to reflect those changes.

Copyright IBM Corporation 2022. All Rights Reserved.

Release notes

Learn more about the IBM Cloud Pak® for AIOps documentation and what's new with IBM Cloud Pak for AIOps 4.8.0:

- What's new
- Known issues and limitations
- Regulatory compliance
- Deprecated and removed functions
- Frequently Asked Questions (FAQs)

What's new in IBM Cloud Pak for AIOps 4.8.0

Get a quick overview of what is new with IBM Cloud Pak® for AIOps.

IBM Cloud Pak for AIOps leverages AI to streamline IT operations, increase uptime, improve efficiency, and reduce costs. The solution supports Operations teams to deliver optimum performance and assurance by:

- Breaking down data silos
- Providing real-time insights into IT systems
- Augmenting incident management with AI
- Automating incident resolution

With Version 4.8.0, IBM Cloud Pak for AIOps delivers improved capabilities and new features for installing, runbooks, integrating with Instana, and more.

Overall, the following new features and changes are introduced with this IBM Cloud Pak version:

- Install and upgrade
- Managing user access control
- Defining integrations
- Automating processes (Policies, Runbooks, Actions)
- Resolving incidents and alerts
- Managing applications (services) and viewing topologies
- Infrastructure Automation
- Non-production (tech preview)
- Documentation changes
- Security and compliance

Install and upgrade

- **Elasticsearch is replaced by OpenSearch (IBM Cloud Pak for AIOps on Red Hat® OpenShift®)** Note that deployments of IBM Cloud Pak for AIOps on Linux already use OpenSearch. The following updates have been made to support this transition:
  - An OpenSearch filter is added to the *ibm-pak mirror-manifests* command in the offline installation and upgrade topics. Learn more
  - The upgrade topics are updated to migrate from Elasticsearch to OpenSearch. Learn more
  - The *Backup and restore with OADP* procedures create a new network policy, and add a new **`esBackupPVC`** parameter. Learn more
  - There are updates to various topics that reference Elasticsearch.
- **(Nonproduction) Multi-zone high availability and disaster recovery (HADR) (IBM Cloud Pak for AIOps on Red Hat OpenShift)**. This feature continues to be hardened, but remains a technology preview. Installation topics are updated with the ability to create an installation resource for a multi zone cluster. Learn more.
- **Resizing storage (IBM Cloud Pak for AIOps on Red Hat OpenShift)** The method for increasing PVC sizes for Cassandra, Postgres, Elasticsearch, and Kafka is updated in v4.8.0. Learn more.
- **Offline deployments no longer retrieve catalogs from the CASE file (IBM Cloud Pak for AIOps on Red Hat OpenShift)**. Offline installation and upgrade procedures are updated for this. Learn more: Offline install and Offline upgrade.
- **Changes to image mirroring (IBM Cloud Pak for AIOps on Red Hat OpenShift)** Image mirroring for offline deployments now uses the **`aiopsctl bastion <..>`** command instead of podman. For more information, see Offline installation: Mirror images and Offline upgrade: Mirror the images.
- **Renewed certificate (IBM Cloud Pak for AIOps on Red Hat OpenShift)** The certificate that is used to sign the IBM Cloud Pak for AIOps images has been renewed, and the new public key is listed in Verifying images. This only affects you if you want to verify the IBM Cloud Pak for AIOps images.
- **New minimum Netcool Operations Insight version requirement (IBM Cloud Pak for AIOps on Red Hat OpenShift)** If IBM® Netcool® Operations Insight® is installed on the same cluster, it must be at version 1.6.13 or higher. Learn more.
- **Incremental adoption (IBM Cloud Pak for AIOps on Linux)** Deployment mode is now configured with a command line argument instead of with a configuration file. Learn more

Defining integrations (connections)

- **Instana integration:** You can now use the Instana Observer instead of the Instana Connector to retrieve topology data from an integration with Instana. **Note** that the Instana Connector will continue to collect event and metric data, but not topology data.
  - See Migrating from the Instana Connector to the Observer for the steps to migrate from the Instana Connector to the Observer.
  - See Configuring Instana Observer jobs for information on using the Instana Observer.

Resolving incidents and alerts

- **Scope-based grouping explainability:** When viewing an incident's alert list, and the selected alert is part of an alert group, the **Alert details** side panel displays information about the underlying temporal, topological, or scope-based groups. For scope-based groupings, you can now click the link **Scope-based policy associated to this alert** to go directly to the associated policy in the Policies UI.

Managing user access control

- **New read-only permission:** The new *View operational data* permission provides users with read-only access to Cloud Pak for AIOps. The *Manage operational data* permission is required to make changes to alerts, incidents, filters, and views. For more information, see Basic operations permissions.

  Important: After upgrading to IBM Cloud Pak for AIOps 4.8.0, users will not be able to perform some tasks until the following permissions are added to existing roles:
  - Manage operational data
  - Use runbooks
  - View topologies

Managing applications (services) and viewing topologies

- **New Instana Observer:** For an integration with Instana, you can now use an Observer integration instead of the previously available integration (Connector type) integration. With this Observer integration, you can retrieve topology data from an integration with Instana.

  Note: The existing Instana (Connector) integration continues to collect event and metric data, but not topology data.

Automating processes (Policies, Runbooks, Actions)

- **Runbook improvements:**
  - You can now administer conditional branching for semi-automated runbooks, as well as fully automated runbooks. For more information, see Adding actions.
  - Automation HTTP actions and client-side actions can now reference unencrypted HTTP endpoints. For more information, see Creating HTTP actions and Creating Client-side actions.
  - There is a new example runbook for expanding Kafka broker storage. For more information, see Sample runbooks with special administrative use cases.
  - Users with the read-only permission `View runbooks` can browse runbooks, actions, and activities. They can not run any runbooks and actions.

Non-production (tech preview)

- **Visualizing analytics with IBM Cognos Analytics:**, the supported capabilities when integrating with IBM Cognos Analytics are improved:
  - You can integrate with IBM Db2 to store alert and incident data in the Db2 database. With this integration, you can create policies to send the alert or incident data from IBM Cloud Pak for AIOps to populate a dashboard or report in IBM Cognos® Analytics.

    For more information about creating this integration, see Creating an IBM Db2 integration. For more information about using policies to send data for a dashboard or report, see Populating a dashboard or report.
  - A new Incident list widget and Button widget are available for use in building a custom dashboard. For more information about the available widgets that you can use, see IBM Cloud Pak for AIOps widgets within IBM Cognos Analytics.
  - Example dashboards are now available to help you learn about the capabiltities of the available widgets and to help you create your own dashboard. For more information about creating a dashboard, see Creating a dashboard.

  Overall, this integration provides you with additional tools to create custom dashboards and reports to visualize and analyze your IBM Cloud Pak for AIOps data. When you set up this integration, you can open IBM Cognos Analytics from the Cloud Pak for AIOps console and begin using IBM Cognos Analytics dashboards to build visualizations of your IBM Cloud Pak for AIOps data, such as alert and topology data.
- **Additional feature previews:** In addition to the integration with IBM Cognos Analytics, IBM Cloud Pak for AIOps includes the following components and features for non-production use as a technology preview:
  - Incident (Story) prioritization from persistence analysis and golden signals extraction
  - Multi-zone high availability and disaster recovery (HADR)
  - New data collection functionality for Dynatrace integration

For more information about these feature previews, see Non-production features (Technology preview).

Infrastructure Automation

- **IBM Cloud Pak for AIOps integration improvements**: The integration between Infrastructure Automation and IBM Cloud Pak for AIOps is improved so that metric data can be sent to IBM Cloud Pak for AIOps in addition to the existing sending of event and inventory data. Metric data syndication is also available through the integration in addition to the existing event and inventory data syndication.
- **Upgrade capability:** You can now upgrade from Version 4.4.x to Version 4.8.0.
- When creating a Terraform Service Catalog item, you now have two options to create a Service Dialog. You can use an existing Service Dialog or create a new Service Dialog. For more information, see Running a Terraform Template from a Service Catalog

Security and compliance

- **Fixed security vulnerabilities:** Fixes and updates are included to address reported security-related vulnerabilities with IBM Cloud Pak for AIOps. For more information about the included fixes, see Security bulletins and fixes.

Documentation changes

In addition to the documentation for the preceding new and changed features, the following documentation is updated:

- **Enrich incidents by using IBM Tivoli Netcool/Impact policy**: If needed, you can enrich Cloud Pak for AIOps incidents with data from external systems. For more information, see Enriching an incident.
- **Maintaining and debugging automation policies:** Effectively maintain your IBM Cloud Pak for AIOps policies and debug issues that might arise. Determine what policies are firing for alerts, find policies that are related to incidents, and use the policy table to discover policy usage and issues. For more information, see Maintaining and debugging automation policies.

Previously introduced features and changes

For more information about new features and changes that were introduced in previous versions of IBM Cloud Pak for AIOps, see:

- What's new in Version 4.7.1
- What's new in Version 4.7.0
- What's new in Version 4.6.1

- What's new in Version 4.6.0
- What's new in Version 4.5.1
- What's new in Version 4.5.0
- What's new in Version 4.4.1
- What's new in Version 4.4.0
- What's new in Version 4.3.0
- What's new in Version 4.2.1
- What's new in Version 4.2.0
- What's new in Version 4.1.2
- What's new in Version 4.1.1
- What's new in Version 4.1.0

Known issues and limitations

Review the known issues for IBM Cloud Pak® for AIOps.

Also, review the troubleshooting documentation for more information on common issues. For more information, see troubleshooting.

- Install and upgrade
- Backup and restore
- Access control
- Identity Management
- Observers and integrations
- IBM Cloud Pak for AIOps and IBM Netcool Operations Insight
- Applications and topologies
- Infrastructure automation
- UI console
- AI model management and training
- ChatOps
- Incidents and alerts
- Policies
- Secure Tunnel
- Runbook Automation
- AIOps Insights
- Ticketing

Install and upgrade

- Unable to reach inventory service
- Limitation on number of instances
- Manual resource adjustments are not persisted
- Services fail to connect to Cassandra
- The `ibm-aiops-orchestrator` pod throws an `OOMKilled` error
- Automatic approval required for installation
- Elasticsearch health status yellow after restoring from a backup
- ChatOps Microsoft Teams integration does not work with a proxy server
- Kafka topics cannot be listed with oc get kafkatopics

Also, review the troubleshooting documentation for more information on common issues. For more information, see troubleshooting.

Unable to reach inventory service

If your system is installed in the same namespace as IBM Cloud Pak® for Network Automation, the connection to the inventory service breaks. This is caused by a clash with another service with the same name.

Workaround: Use the config.yaml file to enforce the correct values for INVENTORY_SERVICE_HOST and INVENTORY_SERVICE_PORT.

Example:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: noi-topology-sizing
  namespace: namespace
data:
  asm: |
    ui-api:
      containers:
        ui-api:
          env:
            - name: INVENTORY_SERVICE_HOST
              value: noi-topology-inventory.namespace.svc
            - name: INVENTORY_SERVICE_HOST
              value: "9178"
```

Limitation on number of instances

IBM Cloud Pak for AIOps and Infrastructure Automation can co-exist on the same cluster, but you cannot have multiple instances of IBM Cloud Pak for AIOps or Infrastructure Automation on the same cluster.

Manual adjustments are not persisted

Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources (such as increased CPU and memory values) are lost when an event such as upgrade, pod restart, resource editing, or node restart triggers a reconciliation. Reconciliation causes any manually implemented adjustments to be reverted to their original default values. Depending on the parameters that you want to adjust, you might be able to use a custom profile to persist your changes. For more information about custom profiles, see Custom profiles.

Services fail to connect to Cassandra

After you install IBM Cloud Pak for AIOps for a production environment deployment, various services might not be available due to connection issues with Cassandra. To resolve this issue if it occurs, restart Cassandra and the schema creation pods.

The `ibm-aiops-orchestrator` pod throws an `OOMKilled` error

If your environment has many secrets and ConfigMaps, when the **`ibm-aiops-orchestrator`** (lead operator) attempts to build its cache, the operator can exceed its memory allocation and cause a Kubernetes out-of-memory error for the container. This error can prevent the IBM Cloud Pak for AIOps installation from reconciling, blocking the installation from completing.

If you encounter this issue, the operator requires more memory resources to build its cache. Override the subscription resource to increase the memory limits for the pod and avoid the out-of-memory issue.

Automatic approval required for installation

The use of manual approval strategies for InstallPlans in a project (namespace) can affect the IBM Cloud Pak for AIOps installation.

For instance, if you use manual approval for any of your InstallPlans to install operators in **`All Namespaces`** mode (**`cluster`** scope), the manual approval can affect your install. The installation of IBM Cloud Pak for AIOps requires automatic approval to be used.

Elasticsearch health status yellow after restoring from a backup

When you are restoring an Elasticsearch backup to a new single node Elasticsearch cluster, the Elasticsearch database might not work as expected. Instead, the Elasticsearch cluster health shows that a yellow status after the restore completes.

ChatOps Microsoft Teams integration does not work with a proxy server

If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps or an environment that uses a proxy server, then you cannot use the ChatOps Microsoft Teams integration. The use of a proxy with the Chatops Microsoft team integration is not supported.

Kafka topics cannot be listed with oc get kafkatopics

Kafka topics are created by using the KafkaTopic custom resource or API. Only topics that are created by custom resource are shown when running **`oc get kafkatopics`**. To see a full list of all the Kafka topics use the following steps:

1. Install kcat.
2. Get the **`waiops-mustgather.sh`** script. For more information, see Installing the IBM Cloud Pak for AIOps MustGather tool.
3. Run **`waiops-mustgather.sh -V kafka-topics`**

Backup and restore

- Backup and restore with Fusion

Backup and restore of IBM Cloud Pak for AIOps with older versions of IBM Fusion does not work with Portworx

If you are using Portworx as your storage provider, then IBM Fusion must be at v2.9.0 or higher, or backup and restore will fail.

Access control

- Automation Analyst role unused in IBM Cloud Pak for AIOps
- Users from a user group remain after user group is deleted
- UI redirects to unexpected page when logging in after a session timeout
- Users in a user group are not listed under the Manage assignees pane of the Incidents and alerts page

Automation Analyst role unused in IBM Cloud Pak for AIOps

By default, an Automation Analyst role is displayed within the IBM Cloud Pak for AIOps console Access control page when you are assigning a role to a user. This default role is used within the IBM® Automation family of offerings, which includes IBM Cloud Pak for AIOps, however, this role is not used within IBM Cloud Pak for AIOps.

This role does not include or provide any permissions within IBM Cloud Pak for AIOps and should not be assigned to users within IBM Cloud Pak for AIOps.

Users from a user group remain after user group is deleted

When you delete a user group, the users that were included in the group remain in your list of users. Any role that is inherited through the deleted user group is removed from the users. If the users were assigned roles individually, they continue to have those roles and can continue to log in to the UI console and complete tasks. If the users that were in the deleted user group need to be removed completely, an administrator needs to manually remove the users. Users can be removed by clicking the *Delete* icon for the user's entry within the list of users on the *Access control Users* tab.

UI redirects to unexpected page when logging in after a session timeout

After a session timeout occurs and a user logs in to the UI console again, the user can be redirected to a different page than the page that they were on when the timeout occurred. For instance, a user that was working on the AI Model Management training page when their session timed out might be redirected to a graphql playground page after logging back in. This redirect occurs because the UI uses the last request URL that included the expired token to identify where to redirect the user when the user logs back in. If this redirect occurs, the user needs to manually return to the expected page in the UI to continue working.

Users in a user group are not listed under the Manage assignees pane of the Incidents and alerts page

When you have users within a user group and view the *Manage assignees* pane of the Incidents and alerts page, you might not see some users who are listed. This error can occur when the users from the LDAP user group are not individually onboarded. To verify whether a user is onboarded, go to the **Access control** > **Users** tab and check whether the user is listed. If the user is not listed, that user must first log in to the console, which validates their roles and permissions. After logging in, the user can display in the list of users and on the *Manage assignees* pane.

The *Manage assignees* pane is viewable from the list of all Incidents. Select an incident and then click *Manage assignees*. After you select an existing user group, you should see the included users who are listed.

Identity Management

The IBM Cloud Pak foundational services Identity Management (IM) service is used by IBM Cloud Pak for AIOps. This service includes the following known issues and limitations:

Note: Some descriptions of the listed known issues in Identity Management are shortened.

- Login failure in Platform UI console while upgrading foundational services version 3.22 or version 3.23 to foundational services version 4.x.x.
- LDAP user names are case-sensitive.
- The OpenShift group does not synchronize when a user is added or removed from an LDAP group.
- The OpenShift users are not removed when you remove them from the LDAP group.
- You cannot onboard OpenShift user group to Identity Management as the groups property of the `user.openshift.io` API is deprecated.

For more information about all the known issues and limitations that are related to Identity Management, see Known issues in foundational services.

Observers and integrations

- Kubernetes Observer job fails to restart after OOM
- ServiceNow Observer UI displays superfluous character
- 'Failed to read certificate' error
- Duplication of resources in the topology if certain observer job parameters are changed after a job has been run
- Incomplete historical data processing in the event of integration pods restarting
- File and Rest observer topology service location URL not accessible
- Integration console displays special characters incorrectly
- Turbonomic integration with IBM Cloud Pak for AIOps affects other integrations in Turbonomic
- No notification in IBM Cloud Pak for AIOps on Turbonomic actions closed without execution
- New Relic observer does not support dashboard tokens for new users
- All dates and time are in US-en format
- Appdynamic historical start date and time cannot be older than 4 hours
- Appdynamic live mode aggregation interval is 1 minute
- The observer-service pod is in crash back loop due to ghost vertex
- Alerts for Instana without associated topologies
- Alerts for Instana topology not mapping properly
- Instana metric collection API rate limit exceeded error
- IBM Cloud Pak for AIOps cannot close some alerts when an Instana integration exists
- Delay query time for integrations
- When creating a kubernetes observer job that is named 'weave_scope', 'load', 'kubeconfig', or 'local', the job fails to run
- Log Anomaly 8k limit on field mapping in details field of the alert schema
- Log integration does not start when multiple log integrations are active
- After a restore, data from integrations is not processed
- Historical data from ServiceNow instance gets collected only when the historical data flow is reenabled
- Dynatrace integration pod restarted and does not retrieve all historical data
- ServiceNow user account gets locked out after a few hours
- Scale resources when running log anomaly training on large data
- The integration status for Elk, Custom Logs, Mezmo, and Falcon LogScale sometimes show 'not running' even though the Flink job and gRPC pod are running correctly
- Log data integrations status is "Done" even though historical data is still loading
- IBM Tivoli Netcool/Impact integration stops event processing with exceptions
- Impact integration fails for Impact server with non-default cluster name
- Connector experienced a failure due to a Bad Request for connection when data flow was enabled
- GitHub connector issues with similar tickets and adding assignees to mappings
- ServiceNow ticket contains too much text
- ServiceNow user account locked out
- ServiceNow observer status given as `Unknown` for an extended period
- Integrations unable to pause and resume both inbound and outbound data flow
- Dynatrace topology does not support proxy target system
- Error messages in the agent log when the Dynatrace integration is disabled
- Dynatrace topology unable to create a second instance on the same cluster

- Output to Db2 via the connector is not keeping up
- When using the Db2 integration, data may not be in sync when using an update trigger in a policy
- Db2 integration shows a status of Running regardless of whether correct login details have been specified
- Integrations stuck in Initializing state
- New Relic observer job fails with an authorization error

Kubernetes Observer job fails to restart after OOM

Kubernetes Observer jobs with very large payloads can encounter an OOM (out-of-memory) error, after which they may fail to restart. The observer appears offline, but a health check fails to flag any errors.

Workaround: Restart the observer if it appears as offline in the UI.

ServiceNow Observer UI displays superfluous characters

If the Entity_mapping field is updated from the Observer UI, superfluous curly brackets and quotations are displayed in the Resource type-mapping field.

Workaround: This is a cosmetic issue and can be ignored.

'Failed to read certificate' error

This error can occur when an observer attempts to create an SSL certificate and the endpoint server does not respond.

In the example error message below, a vmvcenter.crt certificate error occurs because the endpoint server does not respond.

```
Failed to read certificate for field [certificate].
The file 'vmvcenter.crt' could not be found under /opt/ibm/netcool/asm/security/.
```

Workaround: Ensure that the endpoint server is running correctly.

Duplication of resources in the topology if certain observer job parameters are changed after a job has been run

Certain resource parameters are used to uniquely identify a resource. If one of these parameters is changed after the initial job run, then any subsequent job run will result in duplicate records. For example, if the parameter of 'hostname' is replaced with 'Ipaddress' after a topology has been created, a subsequent discovery will consider the resource as new, and create a duplicate record.

The following resource parameters uniquely identify a resource. Changing them after the initial job has been run will result in duplicate records.

Workaround: If you need to modify these values, do not modify the existing job. Instead, create a new job.

Table. Observer job parameters

Observer	Job parameter
ALM	n/a
AppDynamics	`account`
AWS	`region`, `dataTenant`
Ansible AWX	`host`, `user`
Azure	`data_center`
BigFix Inventory	`data_center`
Big Cloud Fabric	`proxy-hostname`, `proxy-username`, `bcf-controllers`
Ciena Blue Planet	`data_center`, `tenant`
Cisco ACI	`tenant_name`
DNS	`addressTypes`, `server`, `port`, `recurse`
Docker	`endPoint.port`
Dynatrace	`datatenant`, `hostname`
File	`provider`, `file`
GitLab	`datatenant`, `hostname`
GoogleCloud	`project_id`
HPNFVD	`datacenter`, `username`, `cnf_job`
IBM Cloud	`instance`, `username`, `region`
ITNM	`instance`
Jenkins	`jenkins_observation_namespace`
Juniper CSO	`cso_central_ms_url`, `user_domain_name`, `domain_project_tenant_name`
Juniper Contrail	`api_server_url`, `os_project_name`, `os_tenant_name`
Kubernetes	`data_center`, `namespace`
NewRelic	`accountName`, `accountId`
OpenStack	`data_center`, `os_project_name`
Rancher	`accessKey`, `clusterId`
REST	`provider`
SDC ONAP	`host`, `username`
ServiceNow	`instance_url`, `username`
SevOne	`datatenant`, `hostname`
TADDM	`api_url`, `username`
Viptela	`data_center`
VMware NSX	`data_center`
VMware vCenter	`data_center`

Observer	Job parameter
Zabbix	`data_center`

Incomplete historical data processing in the event of integration pods restarting

If you create an integration to collect historical data for Metric Anomaly AI Training, you might come across an issue where the integration pod restarts, but does not retrieve all historical data for training. As a result, you might suffer data loss.

An integration pod can restart due to outages, the target system crashing, or pod crashes in the environment. This issue can occur intermittently, depending on the number of metrics that are selected for the integration and the amount of data to be retrieved.

File and Rest observer topology service location URL not accessible

When creating an edge via either the File or Rest observers, the POST request returns a Topology service location URL that is not accessible. The URL cannot be used to manage the edge because the relevant API is not exposed. **Workaround:** None

Integration console displays special characters incorrectly

If you use special characters in the **Name** and **Description** fields while creating or editing an integration, the console *Integrations* page might display the special characters incorrectly. Nevertheless, the integration is saved.

Turbonomic integration affects other integrations in Turbonomic

The Turbonomic integration with IBM Cloud Pak for AIOps enables actions that are created or executed in Turbonomic to be notified in IBM Cloud Pak for AIOps through the enabled webhook workflow. However, Turbonomic allows only one webhook workflow per action. Therefore, other integrations that are enabled in Turbonomic, like ServiceNow, might not get any notification when actions are created or executed in Turbonomic.

No notification in IBM Cloud Pak for AIOps on Turbonomic actions closed without execution

IBM Cloud Pak for AIOps does not receive any notification from Turbonomic for actions that are closed without being executed. For example, an action related to an erroneous condition that is no longer occurring gets automatically closed in Turbonomic. But its corresponding IBM Cloud Pak for AIOps alert remains open indefinitely and must be cleared manually from the console.

New Relic observer does not support dashboard tokens for new users

For *new users* of the New Relic observer, the observer does not work as it no longer supports the New Relic One dashboard token. However, it will continue to work for *existing users* who are using the old token that was generated previously through the old dashboard.

All dates and time are in US-en format

When you are scheduling data collection for a integration, all dates and times are presented in the US-en formats:

- All dates are configured and presented in the `mm/dd/yyyy` format.
- All times are configured and presented in the `hh:mm AM/PM` 12-hour clock format.

You cannot switch the date or time format.

Appdynamic historical start date and time cannot be older then 4 hours

Historical start date and time is configurable, but if you set it to beyond the past 4 hours, then the integration will ignore it and only retrieve the past 4 hours of data.

Appdynamic live mode aggregation interval is 1 minute

In live mode, the new aggregation interval allowed is only 1 minute.

The observer-service pod is in a crash back loop due to a ghost vertex

If you notice that the topology observer-service pod is not functioning correctly and that restarting the pod does not correct the issue, a ghost vertex might need to be removed. To remove the vertex, you need to traverse an edge to the vertex, and then delete the vertex. To traverse to the vertex, use the type vertex and definesType edge.

1. Run the following command to find the ID for the type vertex.

   ```
   oc exec -it <topology pod> -- curl -X GET --header 'Accept: application/json' --header 'X-TenantID: 01abea99-8dff-7f71-bef3-09136b6a4ff0' 'https://localhost:8080/1.0/topology/types?_filter=keyIndexName=ASM::entityType::mgmtArtifact::ASM_OBSERVER_JOB' -u <username><password> --insecure
   ```

   Where

   - `<username>` - Your Topology IBM Cloud Pak for AIOps API username
   - `<password>` - Your Topology IBM Cloud Pak for AIOps API password
2. Run the following command to use the definesType edge to get the ID for the vertex that is causing the issue.

   ```
   oc exec -it <topology pod> -- curl -X GET --header 'Accept: application/json' --header 'X-TenantID: 01abea99-8dff-7f71-bef3-09136b6a4ff0'  'https://localhost:8080/1.0/topology/resources/<type vertex ID>/references/out/definesType' <username><password> --insecure
   ```

   Where

   - `<type vertex ID>` - The ID for the type vertex that you retrieved in step 1.
   - `<username>` - Your Topology IBM Cloud Pak for AIOps API username
   - `<password>` - Your Topology IBM Cloud Pak for AIOps API password
3. Run the following command to delete the vertex.

```
oc exec -it <topology pod> -- curl -X DELETE --header 'Accept: application/json' --header 'X-TenantID: 01abea99-8dff-7f71-bef3-09136b6a4ff0' 'https://localhost:8080/1.0/topology/resources/<type vertex ID>/references/out/definesType?_delete=nodes&_delete_self=false&_node_label=vertex' -u <username><password> --insecure
```

Where

- `<type vertex ID>` - The ID for the type vertex that you retrieved in step 1.
- `<username>` - Your Topology IBM Cloud Pak for AIOps API username
- `<password>` - Your Topology IBM Cloud Pak for AIOps API password

Delay query time for integrations

If you set an integration to retrieve *Live data for continuous AI training and anomaly detection* or *Live data for initial AI training*, you might need to configure a delay to offset the query time window to provide a time buffer for preventing the partial retrieval of real-time data. You cannot configure this delay within the UI console. You must use a command line to configure the delay. For more information about configuring this delay, see Delay configuration in data integrations.

Alerts for Instana without associated topologies

In some cases Instana alerts will not have associated topologies. In most cases this happens because the resource that originated the event is no longer available in Instana. For example, a pod that triggers an Instana event can be redeployed by the underlying kubernetes engine.

Alerts for Instana topology not mapping properly

In some cases alerts are not mapped correctly to a corresponding Instana topology node. For example, alerts generated from log anomaly detection or metric anomaly detection (or other sources) might not show as associated with a Instana topology node.

As a workaround, you need to define your own match rules to correlate with the source data. To define a match rule, click on **Resource Management**, then click **Settings**, **Topology Configuration**, and finally **Configure** on the *Rules* Tile. When you are configuring the match token values to use, the values depend on the data that you are sending to Instana.

Instana metric collection API rate limit exceeded error

The recommended rate limit will be double the number of resources. There might be situations where the limits need to be increased. When a more precise limit is required, use the following formula to determine the limit to use:

```
number-of-metric-API-calls-per-hour ~= (number-of-selected-technologies x 2) x (snapshots-for-selected-technologies / 30) x (60 / collection-interval)
number-of-topology-API-calls-per-hour ~= (number-of-application-perspectives x (60 / collection-interval)) +(number-of-services x (60 / collection-interval))
number-of-events-API-calls-per-hour = 60

total= number-of-metric-API-calls-per-hour + number-of-topology-API-calls-per-hour + number-of-events-API-calls-per-hour
```

Note: Each plugin can have a different number of metrics collected. The mean value across these is used, which is 2 collection cycles per plugin. If the environment is unbalanced, for instance you have mostly hosts that define most metrics, then the formula might underestimate the required limit.

To determine the number of resources (snapshots) for each infrastructure pluginAPI, use the following:

```
api/infrastructure-monitoring/snapshots?plugin=technology_name
```

Example:

```
api/infrastructure-monitoring/snapshots?plugin=host
```

For more information about Instana APIs, see Instana API.

The following example curl commands allow you to retrieve the number of:

- snapshots-for-selected-technologies (such as host)

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/infrastructure-monitoring/snapshots?plugin=host' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

- number-of-application-perspectives

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/application-monitoring/applications' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

- number-of-services

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/application-monitoring/services' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

IBM Cloud Pak for AIOps cannot close some alerts when an Instana integration exists

If you have an Instana integration created and have Instana 221 (Saas or Self-Hosted), you might encounter an issue where IBM Cloud Pak for AIOps might not be able to close some alerts. Instead, you might need to check the status for the associated event within the Instana dashboard and clear the alert manually. For more information, see Troubleshooting integrations: Instana Event integration.

When creating a Kubernetes observer job that is named 'weave_scope', 'load', 'kubeconfig', or 'local', the job fails to run

If you create a Kubernetes observer job with the name `weave_scope`, `load`, `kubeconfig`, or `local`, the job always fails to run. When this error occurs, you can view an error icon in the schedule column for the job. To avoid this issue, do not use these names for the observer job.

Log Anomaly 8k limit on field mapping in details field of the alert schema

The limitation is that the datalayer imposes an 8 kb size limit on the details field in the alert schema. The details field is populated by the log anomaly event, which provides the relevant information to display in slack when trying to view alerts in the chatops. Whenever the details field size exceeds 8 kb, the returned json object is truncated and therefore when the user clicks **view alerts** to retrieve the alerts related to an incident, expected results are not seen and an error is recorded.

The current fields under the details objects are:

```
end_timestamp: int
original_group_id: str
causality: dict
detected_at: float
source_application_id: str
log_anomaly_confidence: float
log_anomaly_model: List[str]
prediction_error: dict
error_templates: List[int]
count_vector: List[int]
text_dict: dict
application_group_id: str
application_id: str
model_version: str
severity_from_model: int
description: str
```

Log integration does not start when multiple log integrations are active

If you have many active log integrations, such multiple Kafka, ELK, Splunk, and Falcon LogScale integrations, and you create and enable another Falcon LogScale integration, you might notice that the integration status is stuck in an error or restarting state. This state can even occur after the integration is operating as expected.

This can occur if you exceed the limit for the number of jobs that can run on the underlying service, which results in insufficent resource available to start the integration. To resolve this issue, complete one or more of the following tasks:

- Increase the replica count of your task managers.
- Increase the task manager count per replica.
- Change the parallelism of your integrations.
- Cancel other integrations

After a restore, data from integrations are not processed

If you have an integration that you are restoring, the status for these integrations can be in error after the restore process completes. To resolve this status, you need to edit and save your integrations with the Integrations in the IBM Cloud Pak for AIOps IBM Cloud Pak for AIOps console. Editing the integration regenerates the associated Flink job for the integration, which updates the status.

Historical data from ServiceNow instance gets collected only when the historical data flow is reenabled

If you enable historical data flow for a ServiceNow integration, you might notice that the historical data is not collected from ServiceNow. For instance, when you check the **`grpc-snow`** pod, you can see ticket data available, but when you check the Flink job or in Elasticsearch, you can notice that no data was collected. If this issue occurs, turning off the historical data flow and turning it back on can cause the data to begin to be collected.

Dynatrace integration pod restarted and does not retrieve all historical data

If you have a Dynatrace integration created and pull historical data with multiple metrics for Metric Anomaly AI Training, you can encounter an issue where the Dynatrace pod restarts, but does not complete retrieving the expected historical data for training. This issue can occur intermittently, depending on the number of metrics that are selected for the integration and the amount of data to be retrieved.

If this potential out-of-memory or out-of-resources issue occurs, consider creating separate integrations to monitor different and smaller sets of metrics. By splitting the integrations, you can reduce the amount of data to be retrieved through the initial integration that can cause this issue.

ServiceNow user account gets locked out after a few hours

If there is an active ServiceNow integration with data collection enabled and the ServiceNow credentials change, the ServiceNow user account can get locked out. ServiceNow has an automatic login locking script called "SNC User Lockout Check", which locks users out after more than 5 failed attempts (including any failed API calls).

If you check the Incidents and alert page, you will see also an alert saying "ServiceNow instance authentication failed".

When this problem occurs, unlock the user in ServiceNow. Then change the password in the ServiceNow integration and save. When authentication fails in the ServiceNow integration, there is a 1-minute wait time before you can access it, to prevent a lockout from occuring quickly.

Scale resources when running log anomaly training on large data

In some cases it is observed that log anomaly training fails on large data due to being Out Of Memory (OOM) or if there is a problem with ES shards. The solution is to scale up the resources to handle large data training.

For more information about shard management, see About indices and shards. For more information about increasing ES Resources, see Log anomaly training pods CPU and Memory resource management.

The integration status for Elk, Custom Logs, Mezmo, and Falcon LogScale sometimes shows 'not running' even though the Flink job and gRPC pod are running correctly

On creating a integration, the Flink job retrieves data normally and the gRPC pod is running without error. However, the console shows that the integration status is 'not running'.

Log data integrations status is "Done" even though historical data is still loading

When a log data integration (Falcon Logscale, ELK, Mezmo, Custom, Splunk) is running in **Historical data for initial AI training** mode, and a custom regex is added in the `field_mapping` section, the data processing can take a long time. Although the **Data collection status** might be shown on the UI as **Done**, data could still be being processed and written to Elastic in the background.

To speed up this process, you can increase the **Base parallelism** number that is associated with that integration. For more information, see Increasing data streaming capacity.

IBM Tivoli Netcool/Impact integration stops event processing with exceptions

If you have an IBM Tivoli Netcool/Impact integration, you can encounter an issue where the integration temporarily stops processing during the sending of an event stream to IBM Cloud Pak for AIOps.

This issue can occur when you have an IBM Cloud Pak for AIOps policy that triggers an IBM Tivoli Netcool/Impact policy when certain types of events are received. If this issue occurs and stops the event processing, the Impact integration logs or Impact policylogger logs can include messages that are similar to the following example exceptions:

```
[6/14/23, 11:38:45:816 UTC] 0000005d ConnectorMana W failed to send status update
...
[6/14/23, 11:38:45:815 UTC] 000023ca StandardConne W configuration stream terminated with an error
...
[6/14/23, 11:38:45:816 UTC] 000023cc GRPCCloudEven W consume stream terminated with an error: channel=cp4waiops-cartridge.lifecycle.output.connector-requests
```

If you encounter this issue, you might need to restart the `impact-connector` pod to begin the processing of the event stream again.

IBM Tivoli Netcool/Impact integration fails for IBM Tivoli Netcool/Impact server with non-default cluster name

If the IBM Tivoli Netcool/Impact cluster uses a non-default cluster name ("NCICLUSTER"), the integration can fail to validate the integration. The IBM Tivoli Netcool/Impact server may report `DynamicBindingException` errors in the impactgui.log:

```
com.micromuse.common.nameserver.DynamicBindingException: DynamicBindingException: Service [NCICLUSTER] not in nameserver.
```

To resolve the issue, wait for the backend IBM Tivoli Netcool/Impact server to finish initializing before starting or restarting the Impact GUI server.

If IBM Tivoli Netcool/Impact is running fix pack 7.1.0.26 or later, you can also resolve the issue by setting the `nameserver.defaultcluster` property in the GUI server. Add the following line to *$IMPACT_HOME/etc/nameserver.props*:

```
impact.nameserver.defaultcluster=CLUSTERNAME
```

where *CLUSTERNAME* is the name of the IBM Tivoli Netcool/Impact cluster.

IBM Cloud Pak for AIOps and IBM Netcool Operations Insight

- IBM Cloud Pak for AIOps Strimzi Kafka topics created without replication
- IBM Cloud Pak for AIOps pods not starting after a cluster restart
- NOIHybrid incorrectly listed in provided APIs for IBM Netcool Operations Insight operator
- Rare issue: unable to deploy log anomaly detection model

IBM Cloud Pak for AIOps Strimzi Kafka topics created without replication

IBM Cloud Pak for AIOps supports multiple replications of Kafka topics for large production installations, such as for data redundancy. The IBM Cloud Pak for AIOps console can automatically create Kafka topics when integrations are created. When a topic is dynamically created by the IBM Cloud Pak for AIOps console, the replication is set to `1` in the controller. As such, Kafka topics the are created during installation can have multiple replicates, but those topics that are created dynamically do not.

If you are implementing a production (large) deployment of IBM Cloud Pak for AIOps, you might lose data if your Kafka pods fail or restart. If the data flow is enabled in your Kafka integration when the Kafka pods go down, you might experience a gap in the data that your integration generated during that down period. Upgrades or updates to workers can cause a Kafka broker restart.

You can manually modify the Kafka topic replication inside the Kafka container from a value of `1` to `3` to mitigate any potential data loss from this issue.

IBM Cloud Pak for AIOps pods not starting after a cluster restart

When the cluster is restarted, all nodes have a *STATUS* of *Ready*. Most pods return to a *STATUS* of *Running* except for some IBM Cloud Pak for AIOps pods.

One potential cause is that Elasticsearch must be up and running before the IBM Cloud Pak for AIOps pods can start.

Restart the Elasticsearch pod to get all pods back to a *STATUS* of *Running*.

NOIHybrid incorrectly listed in provided APIs for IBM Netcool Operations Insight operator

NOIHybrid is incorrectly included in the **Provided APIs** list for the IBM Netcool Operations Insight operator. This list is displayed in the Red Hat OpenShift Container Platform web console under **Installed Operators** > **Netcool Operations Insight** > **Operator Details**. Do not use *NOIHybrid* APIs.

Rare issue: unable to deploy log anomaly detection model

Very occasionally, following successful completion of log anomaly detection training, an error similar to the following is displayed in the AI management UI training page following an attempt to deploy the model.

```
Error
Model deployment failed
```

Within the error textbox, you will also see the text "**Forbidden**".

If you investigate the **aiops-ai-model-ui** pod logs, you will also see the following error.

```
ForbiddenError: invalid csrf token
```

If this occurs, first refresh the browser and try to deploy again.

If that does not remedy the situation, then log out and log back in, and then try to deploy the model again.

Connector experienced a failure due to a Bad Request for connection when data flow was enabled

When data flow is enabled for the Splunk connector, the connection fails with a `Bad Request for connection` error.

Workaround: Check that the models are deployed correctly. Then deploy the models in the AI hub UI.

GitHub connector issues with similar tickets and adding assignees to mappings

The following known issues have been obeserved with the GitHub connector:

- In the Incident **Overview** > **Add tickets to this incident** panel, GitHub issues are missing the *Updated by* information. Additionally, searching for GitHub similar past resolution tickets from the **Source** drop-down menu in this panel will not display any tickets.
- The GitHub connector might be missing from the list of integrations in training modules such as Similar tickets and Change risk.
- The default issue mappings in a GitHub integration does not have assignees. However, if assignees are added to the mappings issues are not created in GitHub.

ServiceNow ticket contains too much text

If the ServiceNow change request, incident, or problem contains a large amount of text, such as work notes with close to 150,000 characters, the ticket is dropped and a warning is logged in the pod log. Dropping the ticket affects change risk for that ticket, as it will not occur or that ticket will not be used for similar ticket detection.

ServiceNow user account locked out

If there is an active ServiceNow integration with data collection enabled and the ServiceNow credentials change, the ServiceNow user account can get locked out. ServiceNow has an automatic login locking script that is called "SNC User Lockout Check", which locks users out after more than five failed attempts (including any failed API calls).

If you check the *Incidents and alert* page, you see also an alert saying "ServiceNow instance authentication failed".

When this problem occurs, unlock the user in ServiceNow. Then, change the password in the ServiceNow integration and save. When authentication fails in the ServiceNow integration, there is a 1-minute wait time before you can access it, to prevent a lockout from occurring quickly.

ServiceNow observer status given as `Unknown` for an extended period

There is currently a known issue whereby the status of the ServiceNow observer in the UI may be shown as `Unknown` for an extended period (up to 15 minutes) following a successful initialization before finally changing to `Running`. This issue is intermittent. It does not happen every time the ServiceNow observer is started, and it does not affect data collection.

Integrations unable to pause and resume both inbound and outbound data flow

Integrations with bi-directional data flow does not completely pause the flow of data both inbound to AIOPs and outbound from AIOps when the data flow is disabled in the UI.

Currently, when data flow is disabled, only the inbound data flow from the event source to IBM Cloud Pak for AIOps is disabled. Outbound data (in the form of actions) are still being pushed from IBM Cloud Pak for AIOps to the event source. This could cause some alerts being out of sync between the two systems.

Workaround: As a workaround you can resynchronize the alerts from Netcool and AIOps using the following steps:

1. Disable the existing Netcool connector dataflow in the UI. This is to ensure that the connector releases any file locks before deletion.
2. Get the existing Netcool connection ID.
3. Delete the Netcool connection from the UI.
4. Clear the AIOpsAlertId and AIOpsState columns in Netcool alerts.status table.
5. Close AIOps Alerts using the connectionId as a filter.
6. Install a new Netcool connection.

Getting the existing Netcool connection ID

1. Get the connector name.

   ```
   oc get connectorconfiguration -l aiops.connector.type=netcool-connector
   ```

2. Set the `connectionname` variable with the connection name and `namespace` variable with the AIOps namespace.

   ```
   connectionname="netcool"
   namespace="aiops"

   oc project $namespace
   connconfig=$(oc get connectorconfiguration --no-headers | grep "$connectionname" | awk '{print $1}')
   connconfiguid=$(oc get connectorconfiguration $connconfig -o jsonpath='{.metadata.uid}')
   ```

3. Make a note of the connection ID value which will be used in the subsequent steps. Make sure the variable is not empty before proceeding to the next step.

   ```
   echo $connconfiguid
   ```

Clearing the `AIOpsAlertId` and `AIOpsState` columns in Netcool `alerts.status` table

1. Login to the ObjectServers using the NCO_SQL **($OMNIHOME/bin/nco_sql)** utility and execute the following command:

```
-- Clear AIOps columns
update alerts.status set AIOpsAlertId='',AIOpsState='';
go
```

2. Exit the NCO_SQL utility.

Closing AIOPs Alerts using the `connectionId` as a filter

1. Find one of the **ir-core-ncodl-std** pods.

```
$irCorePod=$(oc get pods --no-headers | grep 'ir-core-ncodl-std' | awk '{print $1}' | head -n 1)
```

2. Get into the pod terminal.

```
oc exec -it $irCorePod -- /bin/bash
```

3. Update the command and replace **<connectionId>** with the value from the previous command. This command will list all alerts matching the filter.

```
curl --insecure -X GET -H "Content-Type:application/json" -H "Accept:application/json" \
--user "${API_AUTHSCHEME_STATICKEY_USERNAME}:$(cat ${API_AUTHSCHEME_STATICKEY_PASSWORD_FILE})" \
"https://localhost:10011/irdatalayer.aiops.io/v1/cfd95b7e-3bc7-4006-a4a8-a73a79c71255/alerts?
filter=sender.connectionId%3D%27<connectionId>%27"
```

Example command with **a4e3c212-5f84-4be3-989b-d3f293f0183e** as the **sender.connectionId** filter.

```
curl --insecure -X GET -H "Content-Type:application/json" -H "Accept:application/json" \
--user "${API_AUTHSCHEME_STATICKEY_USERNAME}:$(cat ${API_AUTHSCHEME_STATICKEY_PASSWORD_FILE})" \
"https://localhost:10011/irdatalayer.aiops.io/v1/cfd95b7e-3bc7-4006-a4a8-a73a79c71255/alerts?
filter=sender.connectionId%3D%27a4e3c212-5f84-4be3-989b-d3f293f0183e%27"
```

4. Execute the following command to close alerts matching the **sender.connectionId** filter.

```
curl --insecure -X PATCH -H "Content-Type:application/json" -H "Accept:application/json" \
--user "${API_AUTHSCHEME_STATICKEY_USERNAME}:$(cat ${API_AUTHSCHEME_STATICKEY_PASSWORD_FILE})" \
"https://localhost:10011/irdatalayer.aiops.io/v1/cfd95b7e-3bc7-4006-a4a8-a73a79c71255/alerts?
filter=sender.connectionId%3D%27a4e3c212-5f84-4be3-989b-d3f293f0183e%27" -d '{"state": "closed"}'
```

5. Alerts from the old connection should now be closed.

Dynatrace topology does not support proxy target system

The Dynatrace topology feature in the Dynatrace metrics, events, and topology integration does not currently support a proxy target system.

Error messages in the agent log when the Dynatrace integration is disabled

There is currently a known issue whereby you may see the following error messages if you disable the Dynatrace integration:

```
java.lang.ArrayIndexOutOfBoundsException
Skipping concurrent rule based metrics collection
```

Dynatrace topology unable to create a second instance on the same cluster

Currently Dynatrace topology only allows you to create a single local deployment in a cluster. If you attempt to create a second Dynatrace topology instance using another name, the attempt will fail and no new pods will be created in the cluster.

Output to Db2 via the connector is not keeping up

There is currently a known issue with the IBM Db2 integration whereby even at a very low alert rate, the output to the Db2 instance using the connector is not able to keep up. If you are expecting an alert rate of around 700/s, you will see the data in your DB2 instance appear with a delay in seconds.

Workaround: Increase the resource management and HPA on both the **IntegrationRuntime** and **Db2** connector pods.

When using the Db2 integration, data may not be in sync when using an update trigger in a policy

There is currently a known issue with the IBM Db2 integration whereby the data in the INCIDENTS_REPORTER_STATUS table may not be in sync with AIOps updates if you are using an update trigger in the policy.

When there is just a create trigger in the policy, the data is valid. But if you are using an update trigger, the updates are happening asynchronously, so the latest update can be overridden with the previous update if they are happening at the same time.

There is no workaround currently available.

Db2 integration shows a status of Running regardless of whether correct login details have been specified

There is currently a known issue with the IBM Db2 integration whereby the **Data collection status** of the Db2 integration always displays a green tick to indicate that it is running even if incorrect login details have been specified when creating or updating the IBM Db2 integration.

There is no workaround currently available. If the Db2 connector does not return any table or the Db2 database appears not to be receiving incident or alert data, check the log files to see whether you have specified the Db2 login details correctly, and update the **Username** and **Password** fields in the IBM Db2 integration accordingly.

Integrations stuck in Initializing state

There is currently a known issue whereby integrations occasionally appear to be stuck in the Initializing state.

This issue will resolve itself after a delay.

New Relic observer job fails with an authorization error

You might notice that the New Relic observer job fails due to an authorization error. The issue occurs due to changes in the New Relic API.

The error can resemble the following example:

```
ERROR  [2025-02-06 16:02:51,209] [pool-12-thread-1] c.i.i.t.o.n.j.NewRelicLoadJob -  Failed to validate connection:
javax.ws.rs.NotAuthorizedException: HTTP 401 Unauthorized
```

There is no workaround currently available.

Applications and topologies

- Composite resources with differing geolocation markers are plotted separately in the Resource map
- Resources with a _compositeId value have 'Related Services' and 'Related resource groups' tabs disabled
- JVM heap out-of-memory (OOM) failures when loading large number of resources
- Critical error message displayed when attempting to render an application
- Different date and time in Cloud Pak for AIOps console and ChatOps between users
- Deleting a tag template can cause out-of-memory errors
- The Find Path tool ignores filters
- Fault localization and blast radius are not producing accurate results
- High volumes of data can cause Spark workers to run out of memory
- Probable cause is not producing accurate results
- Azure observer missing subnet relationship in topology
- Topologies not visible on Incident Topology page after resource merge
- Openstack observer missing edge-runsOn connectivity in topology
- Topology viewer user interface crashes after the update manager is displayed

Composite resources with differing geolocation markers are plotted separately in the Resource map

On rare occasions a composite resource may contain more than one geolocation marker. All of these will be plotted on the Resource map. If one of these locations falls outside the displayed map area, its status is not displayed.

Workaround: None. Be aware of this quirk when viewing composite resource on the Resource map.

Resources with a _compositeId value have 'Related Services' and 'Related resource groups' tabs disabled

Group and service information cannot be fetched for resources that are part of a composite.

This defect is only encountered when viewing the related service or related resource group details of a resource that is part of a composite. This resource will have the 'Related services' and 'Related resource groups' tabs disabled.

Workaround: none

JVM heap out-of-memory (OOM) failures when loading large number of resources

When running topology loads in quick succession, it is possible to experience some OOM errors and undesired topology pod restarts, even though the pods will continue the processing after restarting.

This error can occur when running resource loads of several millions in a large deployment, and could slow down the loading process. The following type of error message can be seen in the pod logs:

```
WARN [2022-10-25 15:43:31,906] [JanusGraph Session-io-4] c.d.o.d.i.c.c.CqlRequestHandler - Query '[4 values] SELECT
column1,value FROM janusgraph.graphindex WHERE key=:key AND column1>=:slicestart AND column1<:sliceend LIMIT :maxrows
[key=0x02168910cfd95b7e3bc74006a4a8a73a79c71255a0726573...<truncated>, slicestart=0x00, sliceend=0xff,
maxrows=2147483647]' generated server side warning(s): Read 5000 live rows and 1711 tombstone cells for query SELECT
value FROM janusgraph.graphindex WHERE key = 02168910cfd95b7e3bc74006a4a8a73a79c71255a07265736f757263e5 AND column1 >
0039247011800128715902900128715002900 AND column1 < ff LIMIT 5000; token 9157578746393928897 (see
tombstone_warn_threshold) JVMDUMP039I Processing dump event "systhrow", detail "java/lang/OutOfMemoryError" at 2022/10/25
15:43:32 - please wait. JVMDUMP032I JVM requested System dump using '/tmp/cassandra-
certs/core.20221025.154332.1.0001.dmp' in response to an event
```

Cause: Not enough headroom exists between JVM memory limit and the pod memory limit, usually because one was increased without also increasing the other.

Workaround: Ensure that any changes in heap size maintain enough headroom between these settings.

Example: In this example (for a topology size1) the pod limits are set to 3.6 GB while the maximum memory for the JVM (`-Xmx`) is set to 3 GB, thereby leaving 0.6 GB of headroom free for use by the OS.

```
size1:
   enableHPA: false
   replicas: 2
   jvmArgs: "-Dcom.ibm.jsse2.overrideDefaultTLS=true -Xms1G -Xmx3G"
   resources:
      requests:
         memory: "1200Mi"
         cpu: "2.0"
      limits:
         memory: "3600Mi"
         cpu: "3.0"
```

Critical error message displayed when attempting to render an application

This problem occurs when all of the groups, within the application you are attempting to render, have no members. When the application is selected in **Application management**, it does not render and a critical error message is displayed on the UI.

Avoid creating applications with no members. If an application with no members was created for test purposes only, then ignore this error.

Different date and time in Cloud Pak for AIOps console and ChatOps between users

The date and time format for an Incident in the IBM Cloud Pak for AIOps console *Application management* tool and the associated ChatOps notification can be different between users. The format and time zone that is used in the Cloud Pak for AIOps console and ChatOps notification is set to the user's locale. If different users are in different time zones, the displayed date and time are different in the Cloud Pak for AIOps console and ChatOps notification.

Deleting a tag template can cause out-of-memory errors

If a tag is applied to a large number (that is, thousands) of topology resources, then deleting the tag template can cause out-of-memory errors with the **`topology-merge`** pod.

Avoid creating tag templates that use tags that occur with such frequency. Later, do not tag thousands of resources with the same tag, and avoid them being used in a group.

The Find Path tool ignores filters

The topology path tool fails to launch with filters applied.

Launch the path tool without filters, then manually apply the filter settings on the path page.

Probable cause is not producing accurate results

The correlation algorithms for probable cause currently require the use of a Kubernetes model with service-to-service relationships, or the use of dependency relationships between non-Kubernetes resources.

Complete the following steps to create the required relationships for Kubernetes. This procedure configures Topology Manager to overlay relationships provided by the File observer onto the Kubernetes topology.

Note: The Kubernetes observer must be configured and loading data.

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.
3. From the topology navigation toolbar, expand **Settings**, click **Topology configuration**.
4. On the **Rules** tile, click **Configure** to navigate to the *Rules administration* page.
5. On the **Merge** tab, click **New** to create a **New merge rule**.

   In this scenario, data that is provided by the File observer will be used to add the relationships.
6. Specify the following information on the **New merge rule** page:
   1. Rule name: **`k8-file-service`**.
   2. Set the rule Status to **Enabled**.
   3. Add the **`uniqueId`** property to the set of Tokens.
   4. Expand the **Conditions** section and select **File** and **Kubernetes** from the set of available Observers and click **Add**.
   5. Specify **`service`** for **Resource types** and click **Add**.
   6. Click **Save** to save the new Merge rule.
7. Locate the services that you want to relate together and make a note of their source-specific uniqueId, such as **`05f337a1-5783-43bb-9323-dfba941455c7`** (shipping) and **`ae076382-3df9-46cb-97e9-a0342d219efb`** (web).
8. Create a file for the File Observer that contains the **`service-dependsOn-service`** relationships necessary for the correlation algorithms to work.

   The following example creates two services, web and shipping, and states that web **`dependsOn`** shipping. Repeat this as required to relate your services together.

   ```
   V:{"uniqueId": "05f337a1-5783-43bb-9323-dfba941455c7", "name": "shipping", "entityTypes": ["service"]}
   V:{"uniqueId": "ae076382-3df9-46cb-97e9-a0342d219efb", "name": "web", "entityTypes": ["service"]}
   E:{"_fromUniqueId":"ae076382-3df9-46cb-97e9-a0342d219efb", "_edgeType":"dependsOn", "_toUniqueId":"05f337a1-5783-43bb-9323-dfba941455c7”}
   ```
9. Load this file into Topology Manager to relate the services. For more information, see Configuring File Observer jobs.

   If your topology changes, then re-create and reload the file as required. A similar process can be followed for non-Kubernetes sources.

High volumes of data can cause Spark workers to run out of memory

If your environment handles high (10+ millions) workloads of alerts or event, your Spark workers can run out of storage (ephemeral storage). If you encounter this issue, restart the affected Spark workers. This issue can also occur if you are running multiple jobs, which can cause the file system to fill, such as with log or JAR files.

Azure observer missing subnet relationship in topology

For the Azure Observer, a subnet can be intermittently missing the relationship with an IP address in the topology for a resource. While the relationship can be intermittently missing, both the subnet and IP address verticies remain available in the topology.

Topologies not visible on Incident Topology page after resource merge

When resources from two observer sources have been merged using the topology merge functionality, the topology is no longer displayed in the Incident view. This known issue affects **only** the Incident view, and the topology is still present in all other views.

Openstack observer missing edge-runsOn connectivity in topology

After running an Openstack observer job, the **`edge-runsOn`** connectivity between **`ComputeHost`** and **`Hypervisor`** elements is not shown in **`Resource Managenent -> Resources`** when it should be.

Topology viewer user interface crashes after the update manager is displayed

The issue is that you cannot use the update manager feature in topology viewer. The solution is to change your topology viewer user preferences to auto render changes on refresh, which prevents the update manager from appearing.

Infrastructure Automation

- Kubernetes permissions are missing for user roles for using Managed services and the Service catalog
- Non-LDAP users cannot access Infrastructure Management
- Red Hat Advanced Cluster Management and IBM Cloud Pak for Multicloud Management core are not supported
- Users are redirected to the Administration panel when logging back into the UI
- Managed services secrets and certificates cannot be customized during installation
- Database fails to reset when error occurs during database creation for Infrastructure Management
- The **`cam-tenant-api`** pod is not in a ready state after installing the iaconfig CR
- Print or export as PDF entire tables does not work as expected
- Infrastructure Management log display in the UI is removed
- Infrastructure Automation Test deploy fails
- On Infrastructure Management appliances, an Ansible playbook deployment fails
- After restoring from a backup, the Managed services deployment fails
- Infrastructure Management fails to save container provider after changing to a new token
- Infrastructure Automation install fails on FIPS enabled Power cluster
- Embedded Terraform feature not supported with FIPS enabled OpenShift Container Platform cluster

Kubernetes permissions are missing for user roles for using Managed services and the Service catalog

If you install Infrastructure Automation, you, or an administrator, must add the required Kubernetes permissions to user roles before your users can begin to access and use Managed services or the Service catalog.

As an administrator, add the following permissions to your use roles:

Table. Required permissions

Role	Required permission for Infrastructure Automation
Automation Administrator	Administer Kubernetes resources
Automation Operator	Manage Kubernetes resources
Automation Developer	Edit Kubernetes resources
Automation Analyst	View kubernetes resources

For more information about how to add permissions to a role, see Managing roles for Infrastructure Automation.

Non-LDAP users cannot access Infrastructure Management

Non-LDAP authenticated users cannot be used with Single Sign-On for Infrastructure Management. If you attempt to use Infrastructure Management with a non-LDAP authenticated user, you can encounter the following error:

While logged in to the Infrastructure Automation UI console with a non-LDAP user, attempting to start Infrastructure Management fails with an error. This is a limitation.

The error states:

```
OpenID Connect Provider error: Error in handling response type.
```

Red Hat Advanced Cluster Management and IBM Cloud Pak for Multicloud Management core are not supported

Installation of Infrastructure Management in IBM Cloud Pak for AIOps does not support Red Hat Advanced Cluster Management and IBM Cloud Pak for Multicloud Management core. You can continue to use the Kubernetes cluster life-cycle templates and services to create a Kubernetes cluster and import the cluster to an existing installation of Red Hat Advanced Cluster Management, if an installation is available. Deploying hybrid applications are also not supported by Infrastructure Automation.

Users are redirected to the Administration panel when logging back into the UI

When you are working within Infrastructure Automation and log out and then log back in, you can be redirected to the Administration panel instead of the Infrastructure Automation home page or other page that you were previously using. If this occurs, you can use the Cloud Pak switcher in the upper right of the UI console to switch to the Infrastructure Automation home page and then return to the page that you were previously using.

Database fails to reset when error occurs during database creation for Infrastructure Management

If you are creating the database for the Infrastructure Management appliance and you encounter an error, such as the database creation failing to complete successfully, you might not be able to continue with your setup without redeploying. For instance, if the creation fails, resetting the database to clean up your database and deployment can also fail. To resolve this issue, you need to redeploy the Infrastructure Management appliance image before reattempting to create the database.

The `cam-tenant-api` pod is not in a ready state after installing the iaconfig CR

After you install Infrastructure Automation, you can encounter an error where the **`cam-tenant-api`** pod displays as running, but not in a ready state. When this error occurs, you can see the following message:

```
[ERROR] init-platform-security - >>>>>>>>>> Failed to configure Platform Security. Will retry in 60 seconds <<<<<<<<<<<<
OperationalError: [object Object]
```

If this error occurs, delete the **`cam-tenant-api`** pod to cause the pod to restart and attempt to enter a ready state.

Print or export as PDF entire tables does not work as expected

If you are using the Firefox browser, and you select **Print or export as PDF** on the **Compute** > **Containers** > **Projects** page to print or export the entire table of data, the print, or export might not work as expected. For instance, some data, such as table rows might be missing. If you encounter this issue, try a different browser for printing or exporting the data.

Infrastructure Management log display in the UI is removed

Log display support on the UI is removed for Infrastructure Management. As an alternative for viewing these logs, use Kubernetes standard methods such as **`oc log`** commands, viewing the output in Red Hat OpenShift Container Platform or Kubernetes, or setting up a log aggregator for your cluster.

You can still see the log tabs (**Collect Logs**, **IA:IM Log**, **Audit Log**, and **Production Log**) on the **Settings** > **Application Settings Diagnostic** page. However, instead of displaying the log information, the following message is displayed: **`Logs for this IA:IM Server are not available for viewing`**.

Infrastructure Automation Test deploy fails

Infrastructure Automation Test deploy from a **Service Overview** page fails to deploy.

On Infrastructure Management appliances, an Ansible playbook deployment fails

When you attempt to deploy an Ansible playbook on an Infrastructure Management appliance through an embedded Ansible deployment, the playbook deployment can fail with the following error:

```
<35.237.119.31> ESTABLISH SSH CONNECTION FOR USER: ubuntu
fatal: [35.237.119.31]: FAILED! => {
"msg": "Unable to create local directories(/home/manageiq/.ansible/cp): [Errno 13] Permission denied: b'/home/manageiq'"
}
```

If you encounter this error, log in to the appliance as the root user and then deploy the playbook again:

1. Run the command:

   ```
   mkdir -p /home/manageiq
   ```

2. Run the command:

   ```
   chown manageiq:manageiq /home/manageiq
   ```

3. Deploy the Ansible playbook again.

After restoring Managed services from a backup, the Managed services deployment fails

After you restore Managed services (**`cam`**) from a backup, the deployment instance fails with a **`socket hang up`** error.

If this error occurs, restart the **`cam-iaas`** pod by running the following command:

```
oc delete pod <cam-iaas-xxxx> -n <namespace>
```

Where **`<namespace>`** is the project (namespace) where Infrastructure Automation is installed, and **`<cam-iaas-xxxx>`** is the name of the **`cam-iaas`** pod to restart.

With this restart, the service deployment can complete successfully.

Infrastructure Management fails to save container provider after changing to a new token

Infrastructure Management fails to save changes to a container provider after updating the access token when the Metrics collection is enabled.

Workaround: After updating and validating the new token in the *Edit provider* dialog box, switch to the **Metrics** tab and validate the existing endpoint. The **Save** button is now enabled.

Infrastructure Automation install fails on FIPS enabled Power cluster

There is an intermittent issue with the installation of standalone Infrastructure Automation on a FIPS enabled Linux on Power (ppc64le) cluster. This occurs when AllNamespace or OwnNamespace mode is used. A problem with the events operator pod causes the installation to fail.

Embedded Terraform feature not supported with FIPS enabled OpenShift Container Platform cluster

The Embedded Terraform feature in Infrastructure Automation is not yet supported in a FIPS enabled OpenShift Container Platform cluster.

UI console

- Tour icon disappears when browsing to another console page while a guided tour is running
- 'Data cannot be displayed': error given for 'Defined Applications' and 'Favorite Applications' tiles on the Home page
- User credential timeout starts in the backend
- Unable to access Identity Providers link in console
- Slow loading pages
- Alert Viewer page shows 400 Error

Tour icon disappears when browsing to another console page while a guided tour is running

When you start a guided tour and navigate away from the page to the IBM Automation *Home* page, the *Tour* icon might not display on the toolbar. This behavior can occur when an IBM Cloud Pak for AIOps tour is still running. Only one guided tour can run at a time. To resolve this issue, return to an IBM Cloud Pak for AIOps page, click on the *Tour* icon and close a tour. When you return to the IBM Automation *Home* page, the *Tour* icon reappears.

'Data cannot be displayed': error given for 'Defined Applications' and 'Favorite Applications' tiles on the Home page

On opening the Console home page neither 'Defined Applications' nor 'Favourite Applications' is listed in their tiles. However, they do exist as they can be viewed under the 'Resource Management' section. The fix for this is to restart the aiops-base-ui pods.

User credential timeout starts in the backend

The first indications of a user credential timeout might be backend failures. For example, failure to load incident, alert, or policy lists. To resolve this issue if it occurs, log out and log back in again, or wait for the frontend logout to occur.

Unable to access Identity Providers link in console

You might notice that you are unable to access the **Identity Providers** link located in the UI console. You might see an error such as *HTTP ERROR 431*. If this issue occurs, configure the LDAP connection. For more information, see Identity Management (IM).

Slow loading pages

If you are experiencing slow loading of Cloud Pak for AIOps pages, it might be because the server TLS certificate is not trusted by your browser. Some browsers (for example, Microsoft Edge and Google Chrome) prevent caching of resources when the server's certificate is untrusted. This means all static resources associated with the page are fully reloaded on every refresh, significantly slowing down page loads.

To resolve the issue, a certificate signed by a certificate authority that is trusted by your client devices should be used. Using a custom certificate is described in the Using a custom certificate page. This may be a certificate signed by a well known certificate authority, or an internal certificate authority pre-configured on your client devices.

Alert Viewer page shows 400 Error

You might notice that you are unable to access the Alert Viewer page. You might see a **400** error. This error is caused when the cookies for the IBM Cloud Pak for AIOps domain in your browser get too large for the data layer service to consume.

To resolve this issue, clear the cookies in your browser and then reload the page.

AI Model management and training

- Log parsing assigns messages to catch-all template instead of generating expected template
- Elasticsearch record count does not match record count published to Kafka topic
- Log anomaly detection EXPIRY_SECONDS environment variable not retained after an upgrade
- Log anomalies are not detected by natural language log anomaly detection algorithm
- Metric anomaly detection training does not run on schedule
- In Change risk training, Precheck indicates "Good data" but models fail to create
- Alerts for the Log Anomaly - Golden Signals algorithm are not generated when inference log data contains name and value pairs
- Counts against template patterns are not updated in the training UI
- Similar tickets training in IBM Cloud Pak for AIOps on Linux
- Change risk is unable to load the models
- **`aiops-lad-flink-taskmanager-xxx`** in CrashLoopBackOff status

Log parsing assigns messages to catch-all template instead of generating expected template

If you use catch-all templates for mapping uncategorized messages during AI model training, you can encounter an issue where the log parsing assigns messages for an error to the catch-all templates instead of generating an expected template for that error. If this issue occurs, you might not see expected anomalies.

If you suspect this issue is occurring and you do not see expected anomalies, complete the following steps to manually verify your training templates, and remove any catch-all templates that incorrectly generated.

1. Retrieve the normalized logs from your logtrain indices.
2. Identify the logs that are error logs. Review those logs to determine the template mappings.
3. Retrieve the identified templates from Elasticsearch.
4. Use the error log contents and the template ID from the retrieved normalized logs to identify the template string within the retrieved templates.
5. If the template string is comprised entirely of parameters, or a single word and parameters, the template might be a catch-all template. For example, the following string is an example of a catch-all template:

   ```
   <>to <><><><><>
   <> <> <> <>-<>-<> <> <> <> <> <> <> <> <> <> <>
   ```

6. Manually delete any catch-all templates.

Elasticsearch record count does not match record count published to Kafka topic

When you push a large training file (for example 60 M records, such as logs or events) to Kafka through your configured integration, the number of records that are ingested and displayed on Elasticsearch might not match. Elasticsearch record count might be lower than Kafka count due to deduplication. If you encounter this issue, split large files into smaller batches and send them individually to Kafka (for example, 5 M records each). When you are pushing a batch, ensure that you wait for an ingest to complete and the associated records display on Elasticsearch before you push the next batch of records.

Log anomaly detection EXPIRY_SECONDS environment variable not retained after an upgrade

If you set a value for the *EXPIRY_SECONDS* environment variable and upgrade, the environment variable is not retained after the upgrade.

After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly detection alerts.

Log anomalies are not detected by natural language log anomaly detection algorithm

In some cases a model that has been trained successfully is unable to detect certain log anomalies. The quality of the model is independent of whether it trained successfully, and model quality tends to improve as more training data is available. If the model is not detecting anomalies in your logs, consider training the model again but using additional days of training data to improve the model quality.

Metric anomaly detection training does not run on schedule

If you have metric anomaly detection training scheduled to run, such as daily, you can encounter an issue where the training does not run as scheduled. If the training job does not run on schedule, log in to the IBM Cloud Pak for AIOps console and click the *Metric anomaly detection* algorithm tile and then *Train models*.

In Change risk training, Precheck indicates “Good data” but models fail to create

On rare occasions, a Change risk model fails to create, even though *Precheck data* indicates that the data is good. This failure is caused by an insufficient number of problematic change risk tickets being available to create a good model. This problem resolves itself when enough tickets become available for the model. (For more information, see Closed change ticket count requirements).

To confirm that insufficient problem tickets is causing the failure, view the Change risk logs on the training pods.

To retrieve the pod:

```
oc get pod | grep training-cr
```

View the logs for training Change risk models.

```
oc logs <pod-name> # Ex: training-cr-1b5ef57f-9053-4037-95ca-c1e8b8748fc5
```

Check whether the log contains the following

```
size of the problematic (aka labels) tickets is insufficient
```

If confirmed, ensure that enough problematic change tickets are available before training the model again.

Alerts for the Log Anomaly - Golden Signals algorithm are not generated when inference log data contains name and value pairs

In IBM Cloud Pak for AIOps 4.8.0, alerts might not be generated for the Log Anomaly - Golden Signals algorithm when inference log data contains name and value pairs. These pairs are tokens with the **`key=value`** pattern. They might prevent anomalies from being matched to their respective templates.

For example, training generates the following log template:

```
<> exe="/usr/bin/dbus-daemon" sauid=UNKNOWN_VAR hostname=? addr=? terminal=?'
```

Then, during inference, incoming log data is matched against the template from training. If the incoming log data contains tokens with the **`key=value`** pattern like in the following example, the logs are classified as unmatched.

```
[3557470.922719]  exe=\"/usr/bin/dbus-daemon\" sauid=103 hostname=? addr=? terminal=?'
[3557971.107893]  exe=\"/usr/bin/dbus-daemon\" sauid=103 hostname=? addr=? terminal=?'
```

Alerts from this set of logs are not displayed.

Counts against template patterns are not updated in the training UI

When the log anomaly detection - golden signals algorithm generates alerts in IBM Cloud Pak for AIOps 4.8.0, the counts against those template patterns are not updated in the training UI table. Enable historic alert storing in Elasticsearch to access alert counts.

Workaround: Enable the components by editing the installation. Use the following commands to access the installation through the command line. Replace **`<installation-name>`** with the name of the installation:

```
oc get installation
oc edit installation <installation-name>
```

Or access the installation with the Red Hat® OpenShift® console. Go to **Operators** > **Installed Operators** > **IBM Cloud Pak for AIOps** > **IBM Cloud Pak for AIOps** and edit the YAML file for **`aiops-installation`**.

After you access the installation, edit it to include the following values:

```
spec:
  automationFoundation: {}
  license:
    accept: true
  pakModules:
    - config:
```

```
    - name: ir-core-operator   # Find the config item with this name, or add this item if it does not exist
      spec:
        issueresolutioncore:
          customSizing:
            deployments:
            - name: datarouting
              replicas: 1   # Use 3 for large deployment
            - name: esarchiving
              replicas: 1
    enabled: true
    name: applicationManager  # Find the pakModules item with this name, or add this item if it does not exist
```

After the components are enabled, new alerts are stored to Elasticsearch indices, and the alert counts for subsequent alerts are updated correctly in the training UI table.

Similar tickets training in IBM Cloud Pak for AIOps on Linux

Similar tickets training is not available in IBM Cloud Pak for AIOps on Linux. You can manually add tickets from your ticketing integrations in the incident overview.

Change risk is unable to load the models

Change risk is unable to detect the usage of OpenSearch analytical engine. The algorithm relies on reading from a mounted file, but in this case, the file was not mounted correctly. As a result, change risk is unable to function properly and cannot write data to the database. Use the following command to add the missing **volume** and **volumeMounts** as a workaround:

```
CR_DEPLOY=$(oc get deployments | grep change-risk | awk '{print $1}')
# get es secret name
oc get AIOpsEdge aiopsedge -o jsonpath='{.spec.elasticsearchSecret}'
# copy that secret name, something like `aiops-installation-es-admin-user-connection-secret`,
# and run the following oc patch command by replacing the ES_SECRET_NAME
oc patch deployment $CR_DEPLOY --type=json -p '[
  {
    "op": "add",
    "path": "/spec/template/spec/volumes/-",
    "value": {
      "name": "elastic-volume",
      "secret": {
        "secretName": "ES_SECRET_NAME",
        "defaultMode": 420
      }
    }
  },
  {
    "op": "add",
    "path": "/spec/template/spec/containers/0/volumeMounts/-",
    "value": {
      "name": "elastic-volume",
      "mountPath": "/etc/elastic"
    }
  }
]'
```

`aiops-lad-flink-taskmanager-xxx` in CrashLoopBackOff status

You might notice that **aiops-lad-flink-taskmanager-xxx** pod is in **CrashLoopBackOff** status approximately 60 days after the pod deployment. You might notice the following errors in pod logs:

```
2025-02-10 15:34:12,183 INFO  org.apache.flink.runtime.taskexecutor.DefaultJobLeaderService [] - Start job leader
service.
2025-02-10 15:34:12,184 INFO  org.apache.flink.runtime.filecache.FileCache                  [] - User file cache uses
directory /tmp/flink-dist-cache-40ec62bb-f436-4ecf-ae5d-2b1a2e5a4d61
2025-02-10 15:34:12,188 INFO  org.apache.flink.runtime.taskexecutor.TaskExecutor           [] - Connecting to
ResourceManager pekko.ssl.tcp://flink@10.255.255.155:6123/user/rpc/resourcemanager_4(b243c492b5fcd94aecc216000b7946c5).
2025-02-10 15:34:12,346 INFO  org.apache.flink.runtime.taskexecutor.TaskExecutor           [] - Could not resolve
ResourceManager address pekko.ssl.tcp://flink@10.254.43.255:6123/user/rpc/resourcemanager_4, retrying in 10000 ms: Could
not connect to rpc endpoint under address pekko.ssl.tcp://flink@10.254.43.155:6123/user/rpc/resourcemanager_4.
```

If the issue is related to certificate rotation, a possible workaround might be manually rotating the certificates or restarting the pods.

ChatOps

- In Slack, a ChatOps communication to IBM Cloud Pak for AIOps times out without establishing a connection
- In a Microsoft Teams ChatOps, the attach template logs feature does not work
- Incidents cannot be reopened or restored in ChatOps
- No Recommended runbooks found in incident overview
- No incident content viewable in Microsoft Teams on mobile devices
- During a ChatOps secure tunnel creation an 'installation failed' message displays

In Slack, a ChatOps communication to IBM Cloud Pak for AIOps times out without establishing a connection

When sending a ChatOps communication from Slack to IBM Cloud Pak for AIOps, a known intermittent issue can occur. The communication between Slack and IBM Cloud Pak for AIOps can time out after 3 seconds of no response. A potential solution is to reconfigure your connection in the connection onboarding. Ensure that your Slack Bot can access your IBM Cloud Pak for AIOps instance in a prompt fashion. A more permanent and robust solution to this issue is being devised.

In a Microsoft Teams ChatOps, the attach template logs feature does not work

If you have a Microsoft Teams ChatOps, clicking the **Attach template logs** button does not work and the logs are not sent to your Microsoft Teams channel for review. As an alternative, use the **Preview logs** button to view the template logs.

Incidents cannot be reopened or restored in ChatOps

When an incident is closed it is archived and can no longer be modified. If a new alert occurs that is related to the archived incident, a new incident is created instead of reopening the archived incident.

No Recommended runbooks found in incident overview

If a runbook recommends to remediate an incident is deleted from the runbook Library, it does not remove the **Recommended runbooks** link in a ChatOps notification. This can result in the ChatOps runbook section linking out to an empty runbooks page in the incident overview.

No incident content viewable in Microsoft Teams on mobile devices

On mobile devices, when viewed in Microsoft Teams, incidents can appear with no viewable data. Where this happens, switch to using a computer to see the full incident data.

During a ChatOps secure tunnel creation an 'installation failed' message displays

When you create a ChatOps integration and it fails, wait for a few minutes to see if the installation retries, and if it does not simply create the integration again.

Incidents and alerts

- Active incident count is wrong on the Resource management page
- "An error occurred while fetching data from the server"
- Alerts tab shows "An unknown error occurred" error when all alerts are closed
- Closed incidents are missing details and displaying a critical error
- Metric anomaly chart unexpectedly changes from zoom view to normal view
- Unable to add metric anomaly in Related alerts to chart
- Limitation of preview text for default recommended action
- Metric search page chart lines are disjointed
- Some alerts not cleared even with a resolution event
- Alert views are unusable without ID and SUMMARY columns
- Filter on short ID in the incident table doesn't work

Active incident count is wrong on the Resource management page

The Resource management page displays a number of incidents in the Active incidents column instead of displaying one incident with a number of alerts. The Application viewer displays the correct information, however.

This error can occur when related groups or applications are erroneously linked to the incident count.

Workaround: Verify the correct number of incidents and alerts on the Application viewer.

"An error occurred while fetching data from the server"

When viewing the Incidents UI or when creating a policy to assign runbooks to alerts, you might see the message "An error occurred while fetching data from the server". Or in the Alert Viewer, you might see "An unknown error occurred". If you encounter these error messages, complete the following steps to delete the `aiops-ir-ui-api-graphql` pod. The pod is then automatically re-created, which should resolve the error.

1. Log in to your cluster by running the oc login command.

   ```
   oc login -u kubeadmin -p <password>
   ```

2. Delete the aiops-ir-ui-api-graphql pod.

   ```
   oc delete pod -l component=aiops-ir-ui-api-graphql -n <cp4aiops_namespace>
   ```

3. Wait for the pod to restart.

Alerts tab shows "An unknown error occurred" error when all alerts are closed

If you are viewing a closed incident that has all associated alerts resolved, you can encounter an error when you view the Alerts tab. This "An unknown error occurred" error displays when there are no associated alerts. You can ignore this error message as the incident and alerts are resolved and closed.

Same runbook status for multiple alerts in an incident

If more than one alert meets a runbook policy's conditions, the same runbook can be assigned to multiple alerts in an incident. From the incident overview page, you can select an alert and run an associated runbook. The runbook **Status** of the selected alert will be updated on the UI. However, the runbook status of other alerts in the incident might be updated with the same status. This is a known issue.

Closed incidents are missing details and displaying a critical error

Incidents with status of "Closed" are missing topology information on the incident **Overview** tab. The associated alerts of the closed incidents are also missing from **Alerts** tab. The following critical error is displayed on the **Topology** tab for closed incidents: "No resource exists with the specified identifier and time point".

Metric anomaly chart unexpectedly changes from zoom view to normal view

This can occur when a related alert is selected and added to a chart you are zoomed in on. The chart resets to normal view. To resolve, zoom in again after related alert is added to chart.

Unable to add metric anomaly in Related alerts to chart

In some cases, when you click the checkbox in the *Related alerts*, it does not add the related anomaly to the metric anomaly chart.

Limitation of preview text for default recommended action

When an alert is generated from any default log anomaly detection models, the preview of a recommended action might contain partial texts and does not reflect the full view of the recommended action. The first 4000 characters are extracted from the original resolution or action document webpage where possible, from which nonreadable texts such as URLs are excluded to form the content of the preview text.

Metric search page chart lines are disjointed

In the Metric search page chart, the normalised forecast line is disjointed from the baseline data line. This is because the forecast data is normalised independently from the baseline data. Although the lines might not match up, the values shown in the tooltips are correct.

Some alerts not cleared even with a resolution event

In scenarios where large amounts of historical event data are ingested into the system, it's possible that problems and resolutions can be processed out of order, sometimes resulting in alerts not clearing as expected. To avoid this issue, try ingesting smaller batches of event data into the system.

Alert views are unusable without ID and SUMMARY columns

When creating views in the Alert Viewer, you must include the ID and SUMMARY columns in the view. Otherwise, the view will be unusable and can only be deleted by using the API.

Filter on short ID in the incident table doesn't work

You cannot use the short **ID** from the incident table as a filter condition under **Other properties**. Instead, use the full incident `"id"` which can be found in the incident details side panel > raw data format.

Policies

- Condition **Value** field changes `String:alert` to `Value of:alert`
- Breadcrumb navigation missing in policy editor
- Condition **Matches** field for numeric **Operator** selections
- **Last run** time and **Matched** count is updated in policies other than the trigger policy
- Double scroll bars on browser window
- Policy processing failing due to long policy conditions
- No ServiceNow ticket created by incident creation policy
- Datarouting pods can fail and need to be restarted
- Using default values in automatically run runbooks not working
- Netcool alert not suppressed when X in Y suppression policy conditions are met
- Cloud Pak for AIOps Impact policy fails to run when 'No input' is selected
- Policy triggered based on 'incident-updated' runs more times than expected
- Parameter mapping issues in IBM Tivoli Netcool/Impact policies
- Can't use alert.details.name with "any of" in a policy condition
- Policy is triggered against alerts that don't match the condition
- WebSphere resolution action recommendation policy showing a failed status

Condition Values field changes "`String:`" to "`Value of:`"

For example, in a policy condition if the string "`alert.id`" is typed in the **Values** field, and then "`String:alert.id`" is selected, it is changed to "`Value of:alert.id`".

To prevent this, avoid a string that exactly matches the keyword. In this example, use the following condition instead:

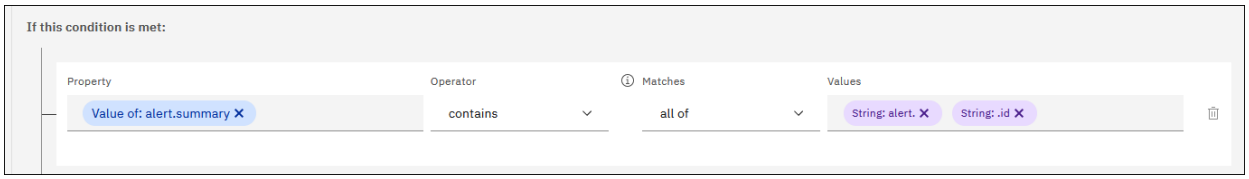


Figure. Policy condition

Note: This example is not an exact match of the string **alert.id**. This workaround finds all summaries containing **alert.** and **.id**.

Breadcrumb navigation missing in policy editor

In some cases where a policy name is long, the breadcrumb navigation in the top left of the policy edit session can be abbreviated. Clicking the breadcrumb still returns you to the Policy UI.

Condition "Matches" field for numeric "Operator" selections

When using a numeric **Operator** in a policy condition set (for example, **greater than**, **less than**, **greater or equal**) all options can be selected under **Matches**. However, always select **Only** for use with a numeric operator.

"Last run" time and "Matched" count updated in policies other than the trigger policy

In a case where an alert meets the incident-creation conditions of multiple policies, only one incident is created. However, all policies that proposed an incident, and the system incident creation policy has the same **Last run** times on the **Policies** hub. Each of these policies also increment their **Matched** counts by 1 in the **Details** tab of the side panel.

Double scroll bars on browser window

An extra scroll bar might appear on the right side of the browser window.

Policy processing failing due to long policy conditions

Policies that have conditions that are long (100,000+ characters) can cause policy processing to fail, resulting in no alerts or incidents being created. Failure can occur when temporal correlation training generates groups containing many alerts, and those alerts have long resource or type fields.

If this problem occurs, such policies should be disabled from the automation policy UI. Identify the policies by

1. Filtering for analytics policies.
2. Sorting by last run time (to identify those that have been processed recently and are likely triggering the problem).
3. Viewing the specification for each to see whether any have a long condition.

Any policies that show up by implementing these steps should be disabled.

No ServiceNow ticket created by incident creation policy

An alert meets the incident-creation conditions of a policy, but no ServiceNow ticket is associated to the incident.

To avoid this, complete the following steps under **actions** in the policy editor:

1. Click **Assign and notify**.
2. Select **In an existing channel**.

Datarouting pods can fail and need to be restarted

After installation, data for display in the UI, such as in the policy list, can be missing or stale.

If this issue occurs, restart the datarouting pods. You can identify these pods by using the following command:

```
oc get pods | grep datarouting.
```

Using default values in automatically run runbooks not working

This problem can occur by selecting a **Default parameter value** when creating a policy to assign runbooks to alerts. The `useDefault` value is not passed during automatic execution but is passed during manual execution.

You can execute the runbook manually from the **Runbooks** page.

Netcool alert not suppressed when X in Y suppression policy conditions are met

If you create an X in Y suppression policy that matches an alert originating from an IBM Tivoli Netcool/OMNIbus environment, the alert will not be suppressed.

Cloud Pak for AIOps Impact policy fails to run when 'No input' is selected

When enabling a Cloud Pak for AIOps Impact policy to Invoke an IBM Tivoli Netcool/Impact policy, the policy fails to run when **No input** is selected under the policy parameter mapping options. To avoid this, select one of the other mapping options. For example, select **Send the full alert**.

Policy triggered based on 'incident-updated' runs more times than expected

If you have a Cloud Pak for AIOps policy that is triggered based on incident-updated, in some cases, the policy might run more times than expected. This is because not all updates can always be processed atomically and the incident might then be updated multiple times. In turn, the incident-updated trigger will be activated more than once.

Parameter mapping issues in Netcool/Impact policies

The following known issues have been obeserved when creating policies to invoke IBM Tivoli Netcool/Impact:

- The policy cannot be created with a parameter mapping option of **No Input** selected. The policy is instead saved with a **Send the full alert** mapping.
- A Netcool/Impact policy cannot be edited to change the parameter mapping option. The change is not persisted. To workaround this problem, create a new Netcool/Impact policy with the new parameter mapping option and disable the current policy.
- When a Netcool/Impact policy is created with a trigger entity of **Incident** and a parameter mapping option of **Send the full incident**, the mapping changes to the **Customize** option if the policy is edited.

Can't use alert.details.name with "any of" in a policy condition

When alert.details is selected in the property field or the value field, the **Details name** field is an optional input where you can minimize the scope to a singular key within the alert's details. However, **Details name** cannot be used in a policy condition with the Matches option of "any of".

Policy is triggered against alerts that don't match the condition

This can occur when you try to compare a NULL value in the policy condition. The policy is triggered because you can have a condition of NULL = NULL matched when the parameters referenced in the policy are not present.

To avoid this problem, you can ensure that the alert property is not equal to NULL in the condition set. For example, see the following policy conditions for **alert.details.name**:

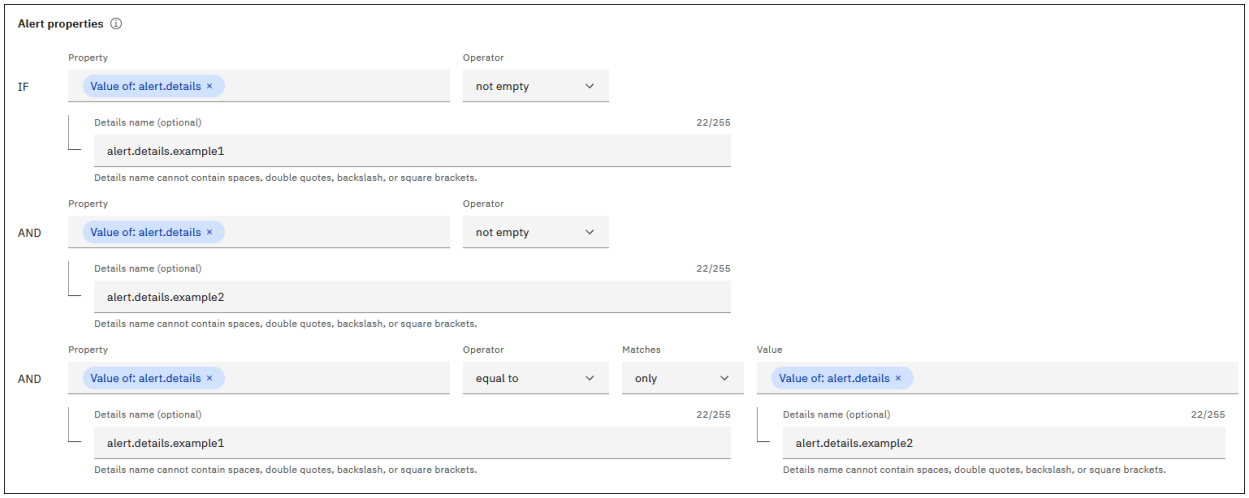


Figure. Policy conditions to avoid comparing a NULL value

WebSphere resolution action recommendation policy showing a failed status

The preset policy "WebSphere resolution action recommendation policy" might initially show a failed status in the policy table. If you encounter this, the policy status should self correct after a period of time.

Secure Tunnel

- Secure Tunnel connector is not running after restart with Podman

Secure Tunnel connector is not running after a restart with Podman

This issue can occur when you install the Secure Tunnel connector to a host machine on which Podman is installed. When the host machine is rebooted and the Secure Tunnel connector is checked by using the **`podman ps -a`** command, the Secure Tunnel connector container does not display **`running`** status.

If this issue occurs, the **`podman-restart`** service must be activated by using the **`systemctl`** command:

```
systemctl start podman-restart
systemctl enable podman-restart
```

After entering the command, check **`podman-restart`** worked by using the following command:

```
systemctl status podman-restart
```

If the Connector is still not running, try restarting the host machine.

Runbook Automation

- When **`alert.suppressed`** value is used, runbook does not automatically run

When `alert.suppressed` value is used, runbook does not automatically run

Normally, you can select a runbook and configure it to run automatically: when an alert is converted to an incident, the runbook is assigned and runs automatically. However, if the parameter value **`alert.suppressed`** is used, the runbook does not run automatically as it reads this as a Boolean value rather than a string value. Therefore, it is necessary to manually run the runbook.

AIOps Insights

- Number of Incidents reported in *Noise reduction* chart is inconsistent with number of Alerts and Events
- AIOps Insights dashboard fails to load even when data is available
- Events not showing up on *Noise reduction* chart

Number of Incidents reported in Noise reduction chart is inconsistent with number of Alerts and Events

In the **AIOps Insights** dashboard, the *Noise reduction* chart normally indicates the number of alerts, events, and incidents reported over a specific time period. However, the inclusion of historical data – containing longstanding, unresolved alerts – can result in a skewering of the data that is presented on the chart: the number of incidents that are presented can outnumber the number of alerts and events. Normally, this number is less than either – alerts reduce to a smaller number of events and events reduce to a smaller number of incidents.

The anomalous incident number happens because the reduction time frame covers alerts and events that are generated in the time period selected (for example, 7 Days). However, the incidents are generated from all outstanding alerts, including alerts that are not resolved, historically: alerts that occurred before the selected time period. So, in these circumstances, while the number of alerts and events is correct, the number of incidents is not.

AIOps Insights dashboard fails to load even when data is available

Large amounts of data can cause the dashboard to fail to load or time out with the message **`Error – Metrics unavailable`** displayed for each chart. The problem is a scaling issue. The **AIOps Insights** dashboard is not yet developed enough to handle huge amounts of data. A possible workaround is to increase resources for **`insights-api`** and **`elasticsearch`** pods. However, this approach might not be successful.

Events not showing up on *Noise reduction* chart

The charts in AIOps Insights cover a timeline no greater than 30 days. The dashboard reads the **`firstOccurenceTime`** value from only within that period. If an alert was created outside of that timeline, and deduplicated, it is not added to the **`eventCount`** in the AIOps Insights Noise reduction chart. In this scenario, the **`eventCount`** for the alert increments in IBM Cloud Pak for AIOps, but not in the *Events* segment of the Noise reduction chart.

Ticketing

- IBM Change Risk Assessment tab in ServiceNow not displaying change risk assessment details
- Select incident state transitions are not permitted in ServiceNow
- Data synchronization in ServiceNow is not working consistently

IBM Change Risk Assessment tab in ServiceNow not displaying change risk assessment details

In a proactive ChatOps channel, you can click the Change request ticket URL to view its details in a ServiceNow instance. However, in some cases in version 4.1.1 of the ServiceNow App, details might not be displayed in the **IBM Change Risk Assessment** tab.

To avoid this issue, update the ServiceNow App to version 4.2.1 or higher.

Select incident state transitions are not permitted in ServiceNow

Cloud Pak for AIOps enforces certain state transitions for its incidents. To synchronize incident data with ServiceNow, the following state transition restrictions must be enforced in ServiceNow as well:

- New -> On Hold
- On Hold -> New
- On Hold -> Resolved
- On Hold -> Closed
- On Hold -> Cancelled

An incident must be set to In Progress before it can go On Hold. An On Hold incident can only transition to In Progress.

Data synchronization in ServiceNow is not working consistently

If you have a ServiceNow integration, you can encounter an issue where updates to records in ServiceNow are not displaying for those incidents and alerts within Cloud Pak for AIOps.

To address this issue, run the following command from the namespace where Cloud Pak for AIOps is installed:

```
oc set env deployment/$(oc get deploy -l app.kubernetes.io/component=chatops-orchestrator -o
jsonpath='{.items[*].metadata.name }') GUNICORN_TOTAL_WORKERS=1
```

If this issue occurs, review the following tasks to ensure your integration and processes are set up correctly:

- Setting up the ServiceNow instance by configuring the ServiceNow App.
- Creating and configuring the ServiceNow integration.
- Configuring a policy to create ServiceNow tickets.

Deprecated and removed functions

Frequent updates to the product and changes in technology require that some function is deprecated or removed from support. See the following sections for definitions of stabilized, deprecated, and removed function and tables that identify items from these categories.

Deprecated functions

A *deprecated* item is supported, but no longer recommended for use and might become obsolete. Consider the alternative actions in the *Recommended action* and details that are provided in the following table:

Table. Deprecated function

Product or category	Affected item	Status	Version	Recommended action	More details and links
Microsoft System Center Virtual Machine Manager (SCVMM)	Infrastructure Automation - Infrastructure Management	Deprecated	3.6.0	No change in version 3.6.0	Microsoft System Center Virtual Machine Manager (SCVMM) might be removed as a supported platform for Infrastructure Management in a future release.
Advanced Content Runtime	Infrastructure Automation - Managed services	Deprecated	3.3.0	No change in version 3.6.0	Advanced Content Runtime might be removed in a future release. If removed from Infrastructure Management, you will need to import Advanced Content Runtime templates yourself.

Product or category	Affected item	Status	Version	Recommended action	More details and links
PostgreSQL 10 (Infrastructure Automation)	Infrastructure Automation - Infrastructure Management	Deprecated	3.6.0	Upgrade to PostgreSQL 13	• If you need to upgrade for appliances, see Migrating IBM Red Hat CloudForms and IBM Cloud Pak for Multicloud Management appliances to Infrastructure Automation • If need to upgrade for a podified installation, see Upgrading PostgreSQL for a podified Infrastructure Management installation.
GitOps	IBM Cloud Pak for AIOps	Deprecated	4.4.0	No change in version 4.4.0	Support for using the IBM GitOps repository method to install IBM Cloud Pak for AIOps is deprecated and will be removed in a future release.
Natural language log anomaly detection algorithm	IBM Cloud Pak for AIOps	Deprecated	4.4.0	Use log anomaly detection - golden signals	As a replacement, use log anomaly detection - golden signals.
Statistical baseline log anomaly detection algorithm	IBM Cloud Pak for AIOps	Deprecated	4.4.0	Use log anomaly detection - golden signals	As a replacement, use log anomaly detection - golden signals.

Removed functions

A *removed* item is typically function that was deprecated in previous releases and is no longer available in the product. You must use alternative features as a replacement for the removed function. Consider the alternative actions in the *Replacement* and details that are provided in the following table:

Table. Removed function

Product or category	Affected item	Version	Replacement	More details and links
GitOps	IBM Cloud Pak for AIOps	4.5.0	Use one of the other options in Installing.	Support for using the IBM GitOps repository method to install IBM Cloud Pak for AIOps is removed.
Kubernetes integration (connector) and documentation	IBM Cloud Pak for AIOps	4.1.0	Configure a Kubernetes observer job as a replacement.	
VMWare integration (connector) and documentation	IBM Cloud Pak for AIOps	4.1.0	Configure a VMware NSX observer job or VMware vCenter observer job as a replacement.	
Red Hat® OpenShift® Container Platform 4.8	IBM Cloud Pak for AIOps	3.7.1	Use Red Hat OpenShift Container Platform 4.10 or 4.12	Supported Red Hat OpenShift Container Platform versions
Red Hat OpenShift Container Platform 4.10	IBM Cloud Pak for AIOps	4.2.0	Use Red Hat OpenShift Container Platform 4.12	Supported Red Hat OpenShift Container Platform versions
Red Hat OpenShift Container Platform 4.11	IBM Cloud Pak for AIOps	4.4.1	Use Red Hat OpenShift Container Platform 4.12, 4.13, or 4.14	Supported Red Hat OpenShift Container Platform versions
Red Hat OpenShift Container Platform 4.12	IBM Cloud Pak for AIOps	4.7.1	Use Red Hat OpenShift Container Platform 4.14, 4.15, 4.16, or 4.17	Supported Red Hat OpenShift Container Platform versions
Red Hat OpenShift Container Platform 4.13	IBM Cloud Pak for AIOps	4.7.1	Use Red Hat OpenShift Container Platform 4.14, 4.15, 4.16, or 4.17	Supported Red Hat OpenShift Container Platform versions
GitHub integration (connector)	Integrations	3.6.0	--	--
IBM Cloud Pak CLI (`cloudctl`)	IBM Cloud Pak for AIOps, for offline install and upgrade	3.6.0	Use the `ibm-pak` plugin for the OpenShift `oc` CLI tool instead of `cloudctl` commands.	Install and upgrade instructions for offline deployments now use `oc ibm-pak` commands instead of `cloudctl` commands.
Event Manager Gateway	IBM Cloud Pak for AIOps and IBM Netcool Operations Insight deployment scenario	3.6.0	Use the Netcool integration (connector)	Installing IBM Cloud Pak for AIOps with IBM Netcool Operations Insight or on-premises Netcool Operations Insight in version 3.6.0
Log display support	Infrastructure Management	3.6.0	--	Log display support on the UI is removed
Terraform 0.11.15	Infrastructure Automation - Managed services	3.6.0	--	You might be able to still use Terraform 0.11.15, but you will need to upload the required engine, providers, or content yourself.
Terraform 0.12.31	Infrastructure Automation - Managed services	Deprecated	3.7.0	No change in version 3.7.0
Terraform 0.13.7	Infrastructure Automation - Managed services	Deprecated	3.7.0	No change in version 3.7.0

Product or category	Affected item	Version	Replacement	More details and links
Terraform 0.14.11	Infrastructure Automation - Managed services	Deprecated	3.7.0	No change in version 3.7.0
Terraform 1.1.9	Infrastructure Automation - Managed services	Deprecated	3.7.0	No change in version 3.7.0
New installs - certificates provided by IBM Automation foundation	IBM Cloud Pak for AIOps and Infrastructure Automation	4.1.0	Red Hat OpenShift Container Platform cluster certificates or user created custom certificates	What's new - install and upgrade in version 4.1.0
User interface using a different storage class	IBM Cloud Pak for AIOps	4.1.0		
Infrastructure Automation	Configure Network option from appliance_console tool	4.5.0	Use the command line to configure networking	To configure networking with the command line, **nmtui** is available for a text user interface or **nmcli** for scripting. Networking can also be configured through the *Cockpit Web interface* (https://:9090) as a root user.

frequently asked questions (FAQ)

Review the following frequently asked questions for IBM Cloud Pak for AIOps

About IBM Cloud Pak for AIOps

- What is IBM Cloud Pak for AIOps?
- What are the Capabilities of IBM Cloud Pak for AIOps?
- How can Cloud Pak for AIOps assist an Ops engineer?
- How can Cloud Pak for AIOps assist a Line of Business (LOB) engineer/CTO?
- How can Cloud Pak for AIOps assist a Site Reliability Engineer (SRE)?
- What are the benefits of using IBM Cloud Pak for AIOps?

What is IBM Cloud Pak for AIOps?

IBM Cloud Pak for AIOps is an AIOps platform developed by IBM to help organizations simplify complex IT operations and improve efficiency. The platform utilizes advanced, explainable AI to assess, diagnose, and resolve incidents across mission-critical workloads, easing the path to adopting AI for ITOps and decreasing operational costs. By automating routine tasks and providing advanced diagnostic capabilities, Cloud Pak for AIOps enables organizations to save time and resources, focusing on innovation and strategic initiatives. Its explainable AI approach ensures transparency and trust in the decision-making process, making it an ideal solution for high-stakes situations requiring rapid and accurate decision-making. To get a full overview of AIOps go to Overview.

What are the Capabilities of IBM Cloud Pak for AIOps?

Cloud Pak for AIOps capabilities include:

- Cross-domain data ingestion and integration
- Topology generation
- Event correlation and analytics
- Incident and pattern recognition
- Augmented remediation

How can Cloud Pak for AIOps assist an Ops engineer?

An Ops engineer can leverage the powerful AI and analytics capabilities of the Cloud Pak for rapid incident remediation, augmented with tailored context, and recommended actions. Use cases include incident remediation, and service assurance.

How can Cloud Pak for AIOps assist an Ops Manager?

Operations Managers can more effectively understand the types of incidents that most frequently occur and create policies and runbooks to improve their operations responsiveness and maximize uptime. Use cases include policy administration, and incident triage management.

How can Cloud Pak for AIOps assist a Line of Business (LOB) engineer/CTO?

Line of Business Managers and CTO's can take advantage of the AIOps Insights Dashboard to understand how their applications are performing over time and the benefits their AIOps Platform is providing to the business. A use case is application or service reporting.

How can Cloud Pak for AIOps assist a Site Reliability Engineer (SRE)?

A SRE can ensure that their applications are running smoothly, even when making changes or upgrades with Change Risk Assessment and the Topology timeline view. Use cases include incident resolution and change risk mitigation.

What are the benefits of using IBM Cloud Pak for AIOps?

The main benefit is the automation of every aspect of incident resolution, including:

- **Diagnose problems faster** and correlate a vast amount of unstructured and structured data in real time
- **Automate with confidence** and empower teams to automate tasks with transparent AI decision-making
- **Gain insights where you work** and keep teams focused, incorporating insights and recommendations into existing workflows to make informed decisions
- **Build and manages securely** – build policy at the microservice level and automate across application components

- **Integrate seamlessly** with monitoring tools and pretrained AI models to gain new insights

About the documentation

- Where can I learn about new or changed features for each release?
- Where can I access known issues for each release?
- Where can I find terms and definitions for IBM Cloud Pak for AIOps?
- Where can I find out about deprecated and removed functions?

Where can I learn about new or changed features for each release?

To learn about new features for the release, review the What's new.

Where can I access known issues for each release?

To review known issues and limitations, go to Known issues.

Where can I find terms and definitions for IBM Cloud Pak for AIOps?

Terms and definitions for IBM Cloud Pak for AIOps can be found in the Glossary.

Where can I find out about deprecated and removed functions?

To find out more about deprecated and removed functions, go to Deprecated and removed functions.

About installing the Cloud Pak

- Where can I find Cloud Pak for AIOps installation documentation?
- Where can I learn about the Hardware requirements for Cloud Pak for AIOps?

Where can I find Cloud Pak for AIOps installation documentation?

AIOps installation documentation can be found here.

Where can I learn about the Hardware requirements for Cloud Pak for AIOps?

Hardware requirements for Cloud Pak for AIOps can be found at Hardware requirements.

About administering and setting up the Cloud Pak

- Where can I learn about administering the Cloud Pak?

Where can I learn about administering the Cloud Pak?

To find out more about administration tasks such as adding users, backing up and restoring, and more, go to Cloud Pak administration.

About integrating with the Cloud Pak

Integrations in Cloud Pak for AIOps

There are over 160 industry standard integrations in Cloud Pak for AIOps, which ingest Events and Alerts, Metrics, Topology, and Logs from across your estate and tooling. You can create your own custom integrations using generic integrations and SDKs. You can also leverage your existing Netcool Probes. Cloud Pak for AIOps provides integration configuration and management.

Which types of data are collected by Cloud Pak for AIOps?

There are two types of data collected: structured and unstructured. Structured data includes logs, tickets, and CI/CD, while unstructured data includes events (alerts), metrics, and topology changes.

IBM Tivoli Network Manager deployment and integration

Where IBM Tivoli Network Manager is present, it should be deployed in a resilient manner and connected to a Netcool/OMNIbus ObjectServer. Any events it generates flows into AIOps through the Netcool Connector. An instance of DASH and Netcool/OMNIbus WebGUI with Topoviz installed is needed to access Structure Browser views. Its topology can be brought into the Cloud Pak for AIOps through the IBM Tivoli Network Manager Observer.

What are best practice guidelines when including Netcool components with the Cloud Pak?

The necessity to include Netcool components in an AIOps deployment depends on your requirements when planning. Where Netcool components are needed, their deployment should be governed by best practices. Some guidelines include:

- All Netcool components should be deployed with the latest fix pack applied
- All components should be deployed in a resilient manner
- All components should be configured to auto-start on machine boot

How do I initiate custom integrations, automations and workflows with IBM Netcool/Impact

Any custom integrations, automations, or workflows that are done by using Netcool/Impact can either be automatically initiated from the Netcool/OMNIbus Aggregation ObjectServer pair through an Event Reader, or from AIOps directly with the Netcool/Impact Connector.

How does Infrastructure Automation (IA) work with AIOps?

You can manage your environment with infrastructure automation and take incident-resolving actions directly from Cloud Pak for AIOps. Infrastructure Automation that are used along with Cloud Pak for AIOps allows for:

- Discovery of existing VMs
- Inventory and lifecycle management of infrastructure
- Compliance and policy Management of VMs
- Chargeback and reporting
- Day 2 provisioning of VMs
- Running of playbooks to automate VM and middleware configuration
- Configuration actions to remediate policy violations with Ansible

About the features of Cloud Pak for AIOps

How does Cloud Pak for AIOps calculate change risk?

Cloud Pak for AIOps discovers the relationship between changes and incidents for risk prediction. It uses state-of-the-art deep learning models for change request prediction risk, and provides explainability by presenting prior, related changes that were problematic, and uses state-of-the-art deep learning models for change request.

How are incidents handled in Cloud Pak for AIOps?

Cloud Pak for AIOps provides a complete overview/integration of your existing operational data: Events, Metrics, Logs, Topology across your silos. No change is needed in instrumentation.

It provides event processing, compression, and correlation to identify service-impacting events, and early detection of anomalies across events, metrics, and logs to enable proactive issue awareness and increased context for an incident.

It also gives probable cause localization and blast-radius to know what caused the problem and what is affected.

How does Cloud Pak for AIOps help you resolves issues?

It applies recommended actions to remediate the issue right away, and allows you choose the level of automation right for the task, from manual step-by-step guidance to semi-automated with guidance and automated commands to full auto-triggered automation. It also allows you collaborate and align across teams by using automatically generated enriched Ops information, analysis and guidance.

How does Cloud Pak for AIOps help you understand your managed environment?

Cloud Pak for AIOps help you visualize and understand your heterogeneous environment, including key metrics from a single screen. It offers a comprehensive application representation, even when “truth” spans multiple sources of insight. It gives a full view of changes to your environment using an historic timeline.

What are the different types of event correlation in Cloud Pak for AIOps?

Cloud Pak for AIOps uses **three different event correlation methods** that work together to correlate events.

- **Temporal**

  Event analytics groups events that have always suspiciously occurred together in the past.

- **Scope-based**

  Operations can define their own groupings based on local knowledge.

- **Topology-based**

  Defines templates for groups of related resources. Cloud Pak for AIOps automatically finds all examples that match the recipe.

What is log anomaly detection?

It is the detection of anomalies from log messages and includes:

- Anomalous time period prediction
- Entity mentions in error logs
- Explanation and pointer to log messages from anomalous time periods

What is metric anomaly detection?

It is the detection of anomalies from time series metrics including:

- Deviation from normal operating range
- Change from variable to flat
- High and low range changes
- Exceed previous range
- Exceed normal range variance

What is probable cause?

You can derive probable fault components using vertex-weighted topology graph traversal and a reasoning engine to understand the meaning of the topology relationships. It allows for localization to the most likely source of an issue within the application topology.

What is incident similarity?

For a given problem description, it helps find top 'k' ranked similar incidents from the past. It helps you understand the current issue and previous successful resolved actions. it consumes tickets and any data from the ticket progression to closure, including human-written investigation and resolution actions. It uses entity-action extraction and action sequence mining to understand tickets and summarize what was done.

What is fault localization and blast radius?

You can derive the full scope of components by using vertex-weighted topology graph traversal and a reasoning engine to understand the meaning of the topology relationships. It allows for blast-radius through directional dependency analysis of the related components that interact with the localized source of the issue.

What is event seasonality?

With event seasonality you can automatically discover events that occur with a regular pattern, which helps you identify chronic issues that can go undetected, provides valuable insights into problem-solving, and also continuous learning over days, weeks, months, and years.

What is change risk prediction?

It allows you assess the risk for each proposed change based on issues that are caused by historical changes. You can harvest and analyze the change ticket history to identify changes that implicitly failed when applied. You can also identify changes that resulted in subsequent issues if they rolled out.

What is event grouping with entity linking?

It groups events, alerts, anomalies to reduce tickets. These can be:

- Topological - which are group events that are related or connected (or both)
- Temporal - which is how you automatically discover events that tend to co-occur
- Scope - which automatically groups events based on scope
- Supergroup - which is a group of groups

What are the characteristics of Cloud Pak for AIOps algorithms?

- **Preprocessing**

  Each algorithm has its own preprocessing chain, which performs a chained set of transforms on the data, and Cloud Pak for AIOps validates the data fits the algorithm and will result in successful training.

- **Unsupervised**

  Algorithms built by IBM using tested data to validate the algorithms. The algorithm builds models without labels. Cloud Pak for AIOps algorithms create models of “normal” based on your data.

- **Validation**

  Understand if the model is not overfitting or underfitting the data. The algorithms self-adjust the model over time.

What are the advantages of AI-powered over traditional incident management?

IT teams rely on incident management to find and fix any number of unplanned events, from the mundane to the profound. The question to ask is as your systems have evolved, has your approach to incident management kept pace? An AI-powered solution can help you stay competitive and keep your business operations running smoothly. The advantages include:

- **Reaction time**

  AI-powered incident management offers teams proactivity and efficiency. It automatically analyzes information from the environment to discover, predict, and prevent incidents before they can impact business processes, which can result in faster time to identify incident causes, and helps maximize uptime and enhance customer satisfaction.

- **Data management**

  IT ingests and correlates data from various sources to provide a comprehensive view of the entire IT landscape. It helps IT operation teams quickly identify the underlying causes of incidents. It promotes collaboration between teams, subject matter experts, and stakeholders. It expedites incident resolution, reduces Mean Time To Repair (MTTR) and improves overall operational efficiency.

- **Scalability**

  AI and automation enable scalability. AI can analyze vast amounts of data in real time to handle increases in the volume and variety of incidents. Teams can more effectively manage incidents, regardless of the size and sophistication of their IT environment.

- **Predictability**

  AI anticipates issues and prevents them proactively. It can detect thresholds automatically and identify patterns, trends, and potential risks that can lead to incidents. It enhances operational efficiency by empowering preventive actions, such as proactive maintenance and capacity adjustments, that reduce the likelihood of incidents occurring.

What is Similar ticket analysis?

Similar tickets is an unsupervised learning algorithm that aggregates information about similar messages, anomalies, and events for a component or a service. It can also extract the steps used to fix previous incidents, if documented. It does this by connecting to your ServiceNow instance and analyzes the existing tickets therein. After training has completed, AIOps will automatically alert users working on an incident to any tickets raised in the past that are similar.

Overview of IBM Cloud Pak for AIOps

IBM Cloud Pak® for AIOps is an AIOps platform that deploys advanced, explainable AI using the IT Operations (ITOps) toolchain data so that you can confidently assess, diagnose, and resolve incidents across mission-critical workloads.

IBM Cloud Pak for AIOps eases the path to adopting advanced AI for ITOps to decrease your operational costs. With this Cloud Pak, you can increase your customer satisfaction by proactively avoiding incidents and accelerating your time to resolution.

The scale of IT systems and their complexity is continually increasing over the last few years because of digital transformation, containerization, and hybrid cloud adoption. IT teams are being inundated with routine maintenance activities and expanding cloud services, leaving them little or no time to contribute toward innovation. To accelerate business automation, reduce complexity, save costs, and automate regular tasks, companies must use the power of AI.

IBM Cloud Pak for AIOps powers automation by using diverse data sets from an entire range of hybrid environments from cloud to on-premises, and bringing the information together across ITOps. With this Cloud Pak, you can tap into shared automation services to get insight into how your processes run. You can also visualize hotspots and bottlenecks, and pinpoint what to fix with event detection to prioritize which issues to address first.

IBM Cloud Pak for AIOps helps you uncover hidden insights from multiple sources of data, such as logs, metrics, and events. The Cloud Pak delivers those insights directly into the tools that your teams already use, such as Slack or Microsoft Teams, in near real-time. Included AI management tools provides you with unprecedented visibility into your organization's infrastructure so that you can predict failures and facilitate problem resolution.

For information about the IBM Cloud Pak for AIOps architecture, see Architecture.

Concepts

IBM Cloud Pak for AIOps represents the state of managed entities by using the following three concepts:

- Events, which are a record containing structured data that summarizes key attributes of an occurrence on a managed entity. Events indicate a change in the state of a service or a system. Events are a category that includes the raw data, alerts, and incidents.
- Alerts, which represent an ongoing anomalous condition against a single managed entity. An alert might evolve over time as the condition changes, and extra context, and insights can be associated with it. An alert presents a single symptom, typically composed of one of more signals, which are immutable, and point-in-time statements about the entity in question.

  For example, an alert which indicates that a given resource is using 100% of its available storage is made up of multiple incoming signals of the disk usage at different times. The alert represents the ongoing time span that the resource in question is in this state.
- Incidents, which represent the context around an issue currently impacting operations. An incident is a collection of insights that come from different data sources, such as events, topology, and logs. Incidents bring together key elements to provide a holistic view of the problem. These elements include alerts, information about how the affected resources are related, guidance on possible resolutions, and automations to remediate the issue. A key difference from an alert is that while an alert has context and insights on a single entity, and a single symptom (alert type), an incident brings together all the symptoms for a more comprehensive understanding. The creation and evolution of incidents are informed by alerts.

  For example, an incident might bring together alerts on a full disk, alerts on increased request failures, and alerts on application failures. These lists of alerts can be paired with guidance on how to address each separate problem and automations, which can perform the solutions.

Features and benefits

With IBM Cloud Pak for AIOps, you can accelerate your journey toward AI powered operations. You can use the Cloud Pak to set up an integrated, AI-driven set of tools to pull, and analyze data from multiple IT platforms to help with ITOps management. IBM Cloud Pak for AIOps combines a set of capabilities to provide a single solution that facilitates predicting, communicating, and resolving events before they become serious problems. Issues are inevitable in any IT landscape, it's how and when you respond that can make a world of difference to your organization.

IBM Cloud Pak for AIOps includes the following key features:

- Guided tours
- Integrations and data ingestion
- Integrations and data ingestion
- AI modeling and training
- Resource management
- Topology
- Log anomaly detection
- Metric anomaly
- Event grouping
- Similar ticket comparison
- Incidents and alerts
- ChatOps
- Automations
- Change risk
- Additional capabilities
  - Infrastructure automation

These capabilities are supported by a network of integrations and capabilities that manage all facets of the AIOps lifecycle from model training to execution.

IBM Cloud Pak for AIOps collects information from all of your IT assets, such as applications, the infrastructure that they run on, and the networking systems that support them. It then uses that data to uncover hidden insights and identify root causes of events. Using this data to train your models lead to better event discovery and a more accurate understanding of your topology. With this advanced understanding of your topology, you can pinpoint where events occur and how far-reaching their impact is. When IBM Cloud Pak for AIOps detects potential incidents, it creates an incident about the incident. Then, it uses a ChatOps environment, such as Slack or Microsoft Teams, to notify, update, and provide your team with near real-time potential remedies, which empowers your teams to automate tasks with clear and open AI decision-making.

The user experience is simplified around AI lifecycle management. The modern experience breaks the complex task of AI model training into a simpler milestone-based experience. This makes it easier for any user to select, filter data, train the models, and view the progress and its output. Through this new experience, you can train the built-in AI models of change risk, event grouping, log anomaly detection, and similar tickets with your data with just a few clicks.

Guided tours

First-time users are presented with a guided tour upon login. This tour is customized for IBM Cloud Pak for AIOps to introduce users to the various home page components, shows how to access tools and capabilities, and where to complete key tasks across the platform.

The home page content is customized based on the user's role-based access. The initial tour shows the user the following components:

- Insights-related summary cards that provide a high-level glance at important metrics in the environment. These insights are related to only the components that the user has access to view or manage.
- The *Quick navigation* and *Support* links for easily accessing commonly used tools, product documentation, IBM Support, and more.
- The main navigation menu for the IBM Cloud Pak for AIOps console, which lists the tools and pages that are accessible by the user.
- The *Tours* menu for accessing more tours and guidance on how to use the Cloud Pak for AIOps console and completing essential tasks. The tour menu includes links for the welcome tour, and for tours to help get started with integrations and AI modeling.
- The getting started links for setting up integrations, learning more about IBM Cloud Pak, and managing user access control.

More tours can be accessed at any time by the user to get a better understanding of how to get started with different tasks and components. Users can select a tour by clicking the **Tours** icon in the Cloud Pak for AIOps console toolbar, and clicking their preferred tour. The following overview tours are available for IBM Cloud Pak 4.8.0:

- **Welcome to IBM Cloud Pak for AIOps**

  A repeat of the initial *Welcome* guided tour.

- **Discover AIOps insights**

  Overview of the AIOps Insights dashboard, which summarizes how IBM Cloud Pak for AIOps helps organizations anticipate, troubleshoot, and resolve IT incidents. The tour loads sample data to walk the user through available toolbars and charts, including Mean time to restore, Noise reduction, Incident activity, and Runbook usage.

- **Start with integrations**

  Available from the Integrations and the home page. This tour guides users through the Integrations to show how to add integrations between IBM Cloud Pak for AIOps and other tools.

- **Introduction to resource management**

  Available from the *Resource management* tool. This tour identifies the resource management capabilities, which you can use to gain an overview of applications, resource groups, and resources and select them to view their details and topology. The tour introduces you to resource groups, which you can use to build application definitions. These resource groups are created automatically when you connect tools, however, you can add groups by using the template page.

- **Understand applications and resource groups**

  Available from the *Resource management* tool when a user selects an application or a resource group to view details. This tour shows a user how to view incident details, view affected resources, view and navigate the timeline, and how to get insights from changes to a resource over the time.

- **Learn about the timeline**

  Available from the *Resource management* tool. This tour is available when a user selects to view details for an application that has an associated incident. This tour walks the user through the details of the timeline widget, showing how to interact with the data and see changes that have occurred for alerts.

- **Explore AI modeling**

  Available from the AI Model Management and the home page. This tour helps you to understand the two types of AI algorithms that are available to enable AI modeling. This tour uses the *Change risk model* summary card to show you how to set up and monitor the training of model-generating algorithms.

- **Introduction to automations**

  Available from the *Automations* tool, which the user can use to proactively set up actions and policies to detect and remediate events, both current and future. With this tour, a user learns how to create policies, view policy details, create and view runbooks, and more.

- **Create a new policy**

  This step-by-step tutorial is available from the *Automations* tool to guide users in the creation of a new policy. This step-by-step tutorial uses the *Promote alerts to an incident* policy template to help you understand the basics of setting up a new automation policy.

- **Introduction to incidents and alerts**

  Available from the *Incidents and alerts* tool. This tour provides an overview of how incidents bring insights together to present the holistic view of an incident. The tour guides you through the different tabs, options, and views of the *Incidents and alerts* tool to show you how to view your existing incidents and alerts, including how to filter the incident list, drill down to view alert details, clear alerts, and more.

- **Investigate an incident**

  Available from the *Incidents and alerts* tool when a user views an incident in detail. This tour guides a user through the process to investigate an incident. A user is introduced to how to view probable causes and top alerts, review associated automations, change incident settings, investigate incident alerts, and how to visualize an incident. The tour also introduces details on incidents, trigger alerts, and the legend and timeline for visualizing an incident.

Integration and data ingestion

Data is the life blood of AI but configuring data feeds can be complex. You can use the Cloud Pak for AIOps console tools to set up integrations to a wide variety of systems for data ingestion. IBM Cloud Pak for AIOps supports integrations to large system solutions including log management, event management, and service management tools within an IT environment, such as Splunk, Falcon LogScale, Mezmo, PagerDuty, and ServiceNow. You can also use the Netcool probes to ingest events into IBM Cloud Pak for AIOps.

A gRPC framework is used for some integrations to support local or remote deployment of a bidirectional data flow. These integrations, such as the Instana and Slack integrations, can be deployed locally in the same project (namespace) as IBM Cloud Pak for AIOps, or deployed remotely.

IBM Cloud Pak for AIOps provides many connectors, probes, and observers by default for setting up integrations with common IBM and third-party systems and services. In addition, if a default integration is not available for connecting to a system or service that you want to use with IBM Cloud Pak for AIOps, you can set up a custom integration to connect to that system or service.

For more information about setting up integrations, see the following sections:

- Defining integrations
- Available integrations

Netcool integrations

Integrations are available for connecting IBM Tivoli Netcool/OMNIbus and IBM Tivoli Netcool/Impact.

A *Netcool* integration is used to define an integration to IBM Tivoli Netcool/OMNIbus. An IBM Tivoli Netcool/Impact integration is used to define an integration to IBM Tivoli Netcool/Impact. For more information about these integrations and integrating with Netcool, see Netcool integrations.

Instana integrations

An Instana integration is available for connecting with Instana to enable connectivity to Instana events, metrics, and topology. Instana assets can now be combined with IBM Cloud Pak for AIOps AI, such as runbooks, metric anomaly detection, and log anomaly detection to help you gain insights. For more information about setting up this integration, see Creating Instana integrations.

Ansible integrations

IBM Cloud Pak for AIOps also includes Ansible integration for connecting to Ansible and an SSH integration to connect with another system where scripts and commands can run.

The Ansible integration provides you with the ability to set up a integration to an Ansible automation platform controller server to gain benefits from the configuration and orchestration capabilities of Ansible. For more information about setting up the integration, see Ansible automation provider.

The SSH integration is agentless and connects directly to the target machine. It authenticates by using SSH public key authentication. You can only create one SSH integration. For more information, Creating SSH integrations.

AI modeling

You can train AI models to hone its ability to derive insights from your incoming data integrations. To get the most out of your AI models, you can manually map your log data to the JSON training format and train your models with it. For more information about training specific types of models, including suggested mappings for events and logs, see the following training topics:

- About AI algorithms
- Configuring AI models

For more information about what you can do to gain further insight into your IT infrastructure, see the following topics:

- Navigating the ChatOps interface
- Managing incident

For guidance on how to plan data integrations and AI model training, see the following topic: Planning data loading and training.

Resource management

IBM Cloud Pak for AIOps brings together the capability to group resources from different data types into applications (services). You can flexibly define an application (service) to meet your business needs. With applications (services), you can obtain an integrated view of your resources to understand inter-dependencies.

IBM Cloud Pak for AIOps focuses on the user experience and health of your applications (services). This application-centric approach uses a built-in application template model to dynamically aggregate and group tagged resources to enable rapid visualization of different application components.

Topology

An integrated service and topology management and topology viewer provides complete up-to-date visibility of your applications and resources, and provides you with control over dynamic infrastructure and services. You can configure your topology visualizations to also show real-time or historical data.

In addition, to help you pinpoint incidents within applications and resource topologies, IBM Cloud Pak for AIOps tools work together to correlate events.

Log anomaly detection

IBM Cloud Pak for AIOps includes a pair of AI algorithms to gather log data from one or multiple components in the application architecture to identify a baseline of expected log message types. IBM Cloud Pak for AIOps uses this baseline to discover abnormal behavior in your live log data. These AI algorithms can run independently of the other. If both algorithms are enabled, then any log anomalies discovered by both are reconciled, so that only one alert is generated. In this case, the severity of the combined alert is equal to the highest severity of the two alerts. Each algorithm is used to gather data differently:

- Natural language log anomaly detection uses natural language techniques on a subset of your log data to discover abnormal behavior. This log anomaly detection takes large amounts of log data and trains on it to learn what is normal behavior for a particular component. This model goes beyond just looking

at error states or frequency of metadata around log messages. Instead, it determines when something becomes an anomaly compared to what patterns it typically exhibits.

- Statistical baseline log anomaly detection uses a statistical moving average on all of your log data to discover abnormal behavior. This algorithm automatically detects unusual patterns in logs and notifies you when they occur. Data that is used for analysis is updated every 30 minutes so this algorithm provides value quickly.

Note: This feature is only available for extended deployments. For more information about deployment types, see Incremental adoption.

For more information, see Log anomaly training definition.

Metric anomaly

Metric anomaly detection provides the ability to analyze any time series metric automatically using AI without the need for configuration or tuning. This AI leverages the Instana Integration, automations, and policies. You can group metric anomalies with log anomalies and other Alerts.

Metric anomaly detection can help you measure key performance indicators, like memory availability or latency, to identify trends and deviations in these areas early.

For more information, see About metric anomaly detection.

Event grouping

AI algorithms are provided with IBM Cloud Pak for AIOps to group events together and present the events within a single incident.

- Temporal grouping is used to group events that are discovered to co-occur over time. When a problem arises, typically multiple parts of a system or environment are impacted. When events in different areas co-occur, it makes sense to look at them together and treat them as one problem to try to determine what might have occurred. Grouping co-occurring events together reduces the number of tickets and incidents that are opened and the number of people looking at the same problem, thereby significantly reducing noise in your monitoring systems. It helps you to understand the context of an issue so you can prioritize, triage, and resolve it more quickly.
- Topological grouping is used to group events based on the resource groups in which events occur. For example, if you have a resource group that is made up of all the resources within a given Kubernetes namespace, then any events on pods, microservices, or other resources in that namespace are grouped together in a single topological group. Topological grouping helps you understand when events are connected based on their topology, providing valuable context for why related events might occur together.
- Scope-based grouping automatically groups events relating to an incident if they have the same defined scope and occur during the same period. A scope can be used to identify where events originate based on a common attribute, for example, the location of a server room. By understanding when events are related based on both time and location, you can more quickly diagnose incidents.

For more information, see Event grouping training definition.

Similar ticket comparison

The similar ticket AI algorithm in IBM Cloud Pak for AIOps can be used to define an AI model so that when incidents occur, the model can review details for similar tickets from the past to help determine a resolution to current problems. This model aggregates information from past similar tickets. It can also extract the steps that are used to fix previous incidents, if documented. This AI algorithm discovers details about similar messages, anomalies, and events within your tickets that occurred in the past and are impacting the current application. For more information, see Setting up training for similar tickets.

Note: This feature is only available for extended deployments. For more information about deployment types, see Incremental adoption.

Incidents and alerts

IBM Cloud Pak for AIOps enables you to identify health issues across your application, services, and network infrastructure on a single management console. It provides a Incidents and alerts tool that brings together incidents, alert details, alert severity, and alert filtering. This capability helps you to analyze the context of incidents across the full enterprise estate. This viewer acts as a single pane of glass incident dashboard, which displays key details like the probable cause, topology, and associated automations to resolve the incident.

Some alerts in the console are presented in alert groups, based on analytics. By using the alert list, you can explore why these alerts were grouped together, and this understanding can further help with alert resolution.

Alerts are displayed in the Alert Viewer. From a ChatOps integration (Slack or Microsoft Teams), you can launch into an Alert Viewer that shows the alerts that are associated to a specific incident.

For more information, see Managing incidents and alerts.

ChatOps

IBM Cloud Pak for AIOps ChatOps support includes integrations with existing social collaboration tools, Slack, and Microsoft Teams. These integrations help to deliver AI-powered automation for ITOps and management within their preferred collaboration tool of choice.

From a ChatOps integration you can launch into the Cloud Pak for AIOps console Incidents and alerts tool to view more information about an alert that is being viewed in ChatOps.

IBM Cloud Pak for AIOps uses ChatOps to provide a message-based interface for reporting incidents with the ITOps that you are monitoring. This ChatOps interface displays the following types of operations information:

Table 1. Operations information available from the ChatOps interface

Operation	Description
Anomaly Detection	Provides details about detected anomalies in your real-time or offline data, such as the time when the incident was detected, the incident description, status, and more.
Relevant events	Groups-related events to aid incident diagnosis. Events can include alerts, PagerDuty alerts, or log anomalies.

Operation	Description
Probable cause	Derives root fault component, and derives the full scope of components that are affected by an incident.
Ticket Similarity	For a particular ticket, finds the highest "n"-ranked similar tickets from the past.
Next Best Action	For a particular incident, suggests the highest "n" actions from similar tickets from the past.

Automations

with IBM Cloud Pak for AIOps an engine and UI for automation is included that you can use to create automation policies, runbooks, and actions to help you resolve incidents quickly by setting up and enabling an automatic response as situations arise. Starting up additional capacity or restarting a failed service can happen before your team knows there is an issue.

For setting up automated procedures, you can proactively set up policies, runbooks, and actions to detect and remediate events, both current and future.

- Policies are rules that contain multiple condition and action sets. They can be triggered to automatically promote events to alerts, reduce noise by grouping alerts into an incident, and assign runbooks to remediate alerts. For more information, see Policies.
- Runbooks automate procedures that do not require human interaction, thereby increasing the efficiency of IT operations processes. With runbooks, your operators can spend more time innovating and are freed from performing time-consuming manual tasks. For more information, see Runbook Automation.
- Actions in runbooks are the collection of manual steps grouped into a single automated entity. An action improves runbook efficiency by automatically performing procedures and operations. For more information, see Actions

Change risk

IBM Cloud Pak for AIOps includes an AI algorithm for providing change risk assessment and includes the capability of surfacing change risk assessments through ChatOps.

The change risk AI algorithm provides an assessment of the risk of implementing a proposed change; for example, a code change or a new software version. Hundreds of changes can affect an application during its lifecycle. Change risk correlates historical change data with historical incident data and determines how likely it is that a specific code change would cause a problem. Using the assessment percentage provided, you can determine how safe it is to proceed with the change. Training this AI model can help you ensure that risky changes for an application are assessed before deployment. For more information, see Change risk training definition.

The change risk assessment in ChatOps provides you with a model confidence percentage that represents the confidence in the AI's prediction of the impact of a change. This percentage is relative to the risk associated with a new or updated ServiceNow change request ticket. The model confidence percentage is calculated based on the change risk AI model that is based on the change risk AI algorithm and the data that you feed into it. Change risk can be evaluated based on any type of change, whether it is changes to code, or to configurations, or to data. If you have historical change request tickets for that type of change, IBM Cloud Pak for AIOps can provide a model confidence percentage for it. For more information, see Change risk in ChatOps.

Note: This feature is only available for extended deployments. For more information about deployment types, see Incremental adoption.

Infrastructure automation

Infrastructure automation is included as an optional feature with IBM Cloud Pak for AIOps. This Infrastructure automation component consists of the following components, which were available with IBM Cloud Pak for Multicloud Management:

- Infrastructure management, previously called IBM Red Hat CloudForms.
- Managed services, previously called Terraform & Service Automation or IBM Cloud Automation Manager.

Infrastructure management delivers the insight, control, and automation enterprises need to address the challenges of managing virtual environments, which are far more complex than physical ones. This technology enables enterprises with existing virtual infrastructures to improve visibility and control, and those just starting virtualization deployments to build and operate a well-managed virtual infrastructure.

Managed services provide you with the capability to automate provisioning of infrastructure and virtual machine applications across multiple cloud environments with optional workflow orchestration.

Infrastructure automation enables IT Operations and Site Reliability Engineer (SRE) teams to use infrastructure as code practices to drive IT velocity and shift to the left of operations.

Infrastructure automation is installed within a separate cluster from IBM Cloud Pak for AIOps. For more information, see Infrastructure automation.

Bringing the features together

By combining AI-driven insights with a modern user experience, you can achieve reduced Mean Time to Resolution (MTTR) when an incident occurs and avoid incidents in the first place through a change risk advisor. The user experience brings in different types of data together and builds out an application perspective, delivering unique insights from the AI models.

Application impact avoidance

The trained AI models establish a baseline and automatically detect anomalies across structured and unstructured data. The models reduce noise and bring related entities across different data sets together into a complete “incident” ahead of a potential incident or application impact. A much smaller set of high-impact “incidents” versus a flood of events and incidents, provides deep insights with explainability.

These incidents help you to visually understand the probable cause to help prevent future downstream impact. The Next Best Action recommendation from past incidents, and the built-in Runbook Automation capabilities accelerate the time to remediation of an incident. The probable cause capability allows users to isolate and find the root cause of the problem faster.

Incident diagnosis

- Derive root fault component and probable cause of affected components to isolate problems faster.
- Detect anomalies by using automatic parsing to detect anomalies in logs and alerts earlier than rule-based alerts.

Insight delivery

- Integrate with pre-trained AI models to gain new insights from existing tools. Keep teams focused, surfacing insights and recommendations into existing workflows.
- Avoid application impact through correlating data across all relevant data sources. Detect hidden anomalies, anticipate issues, and resolve faster. Correlate a vast amount of unstructured and structured data in real time with AIOps tools.
- Proactively avoid risks and automate runbooks for more efficient workflows.
- Proactively avoid incidents with explainable AI that advises based on inputs across ITOps.
- Create a visual representation of the data that is flowing through your business and pinpoint bottlenecks in your operations.
- Draw on advanced AI and analytics to create a baseline that you'll use to track key performance indicators (KPIs) over time.

Architecture

IBM Cloud Pak® for AIOps combines a set of capabilities to provide a single solution that facilitates predicting, communicating, and resolving events before they become serious problems.

IBM Cloud Pak for AIOps integrates infrastructure and operations management into a single coherent structure across business applications, virtualized servers, network devices and protocols, and security and storage devices. IBM Cloud Pak for AIOps enables CIOs, site reliability engineers (SREs) and IT Operators by uncovering hidden insights from multiple sources of data (like logs, metrics, and events). It delivers those insights directly in to the tools that teams work with (like Slack, Microsoft Teams, and ServiceNow) in near real-time, and also presents them in a web-based user interface.

The core functions of IBM Cloud Pak for AIOps can be categorized into five capabilities: event management, incident detection and resolution, entity extraction, and insight delivery. These capabilities are supported by a network of integrations that gather data. It also offers all facets of the AIOps lifecycle from model training to execution. These capabilities are bolstered by IBM Cloud Pak foundational services.

A deployment of IBM Cloud Pak for AIOps can be a base deployment or an extended deployment that has log anomaly detection and ticket analysis (similar incidents and change risk) capabilities enabled. For more information about base deployments and extended deployments, see Incremental deployment.

For more information about the possible deployment configurations, see the following topics:

- Deploying IBM Cloud Pak for AIOps on OpenShift
- Deploying IBM Cloud Pak for AIOps on Linux

For more information about how these capabilities come together in IBM Cloud Pak for AIOps, see:

- IBM Cloud Pak for AIOps
- Infrastructure Automation
- IBM Cloud Pak for AIOps with Infrastructure Automation

IBM Cloud Pak for AIOps architecture

IBM Cloud Pak® for AIOps combines a set of capabilities to provide a single solution that facilitates predicting, communicating, and resolving events before they become serious problems.

The following sections illustrate how these capabilities come together in IBM Cloud Pak for AIOps:

- Features and Capabilities
- Deployment architecture
- Data flow architecture
- Extensibility architecture

Features and Capabilities

IBM Cloud Pak for AIOps is installed on Red Hat® OpenShift® Container Platform or Linux®. For more information about the supported platform versions, see the following topics:

- Supported Red Hat OpenShift Container Platform versions
- Planning an installation of IBM Cloud Pak for AIOps on Linux®

When installed, IBM Cloud Pak for AIOps includes access to the IBM Cloud Pak for AIOps console. This UI is a web-based application for accessing features and capabilities for IBM Cloud Paks, such as IBM Cloud Pak for AIOps. If you have multiple IBM Cloud Paks that are installed in the same environment, you can switch between working with each IBM Cloud Pak from this single UI.

Installing IBM Cloud Pak for AIOps gives you access to the following features and capabilities:

- AI modeling
- Applications
- Change risk
- Data ingestion
- Event grouping
- Log anomaly detection

- Metric anomaly detection
- Resolution automation (Runbooks, Policies, Actions)
- Similar incident comparison
- Incident management
- Topology

For more information about these features and capabilities, see features and capabilities.

Deployment architecture

The IBM Cloud Pak for AIOps deployment model requires a deployment on a Red Hat OpenShift Container Platform or Linux cluster. Data sources are configured with it for events, topology, metrics, and logs.

An IBM Cloud Pak for AIOps deployment can resemble the following diagrams. This diagram shows optional IBM Tivoli Network Manager and IBM Tivoli Netcool/Impact integrations.

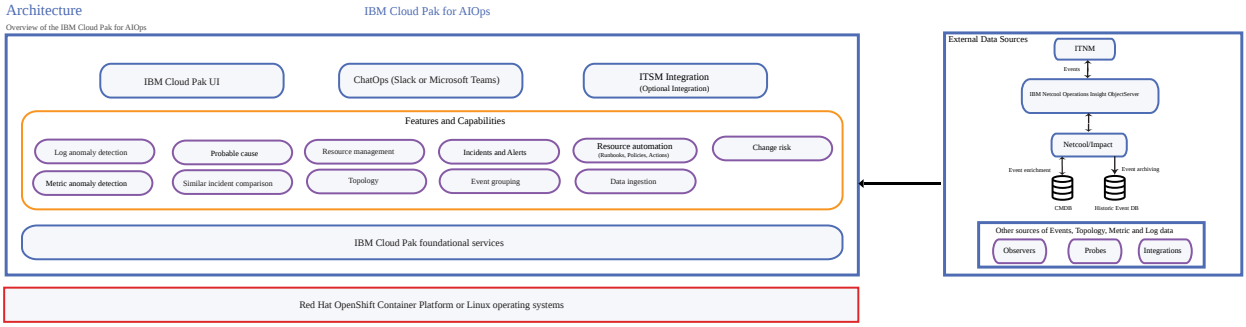


Figure. Architecture with Netcool components

This diagram shows the architecture without Netcool components.

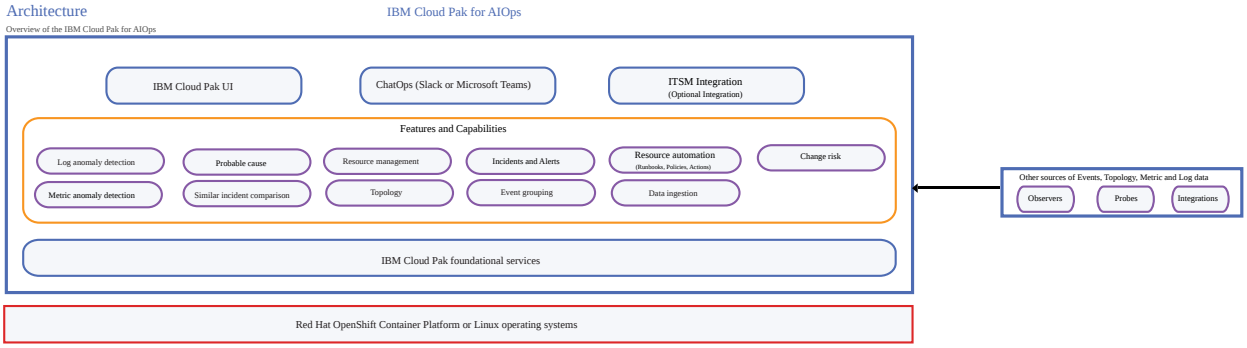


Figure. Architecture with no Netcool components

From these base deployments, there are typically two deployment patterns:

- You have Netcool components and want to reuse them with IBM Cloud Pak for AIOps.
- You have no Netcool components to reuse.

In the case where existing Netcool components exist, you can connect to them similar to how you can connect to other data sources through an integration. In this case, you would use the Netcool integration. This integration allows IBM Cloud Pak for AIOps to receive all the events in the IBM Tivoli Netcool/OMNIbus components (typically deployed on VMs).

Notes:

- A separate Netcool integration must be set up for each Aggregation ObjectServer pair that is present. The IBM Cloud Pak for AIOps schema must be normalized by using the mapping in the Netcool integration.
- Events flowing through integrations to IBM Cloud Pak for AIOps, aside from events that are sent through the Netcool integration, do not flow back to IBM Tivoli Netcool/OMNIbus. To handle custom enrichments or logic, use a IBM Tivoli Netcool/Impact integration with IBM Cloud Pak for AIOps.
- Event archiving to the REPORTER database can be done either directly from IBM Tivoli Netcool/OMNIbus, or by using IBM Cloud Pak for AIOps to feed events to IBM Tivoli Netcool/Impact. This process inserts the events into the event archive directly.
- Where IBM Tivoli Network Manager is present, it must be deployed in a resilient manner and connected to a IBM Tivoli Netcool/OMNIbus ObjectServer. This requirement is because IBM Tivoli Network Manager connects to IBM Tivoli Netcool/OMNIbus in two ways:
  - The IBM Tivoli Network Manager Gateway reads events from IBM Tivoli Netcool/OMNIbus and does root cause analysis and enriches them with topology metadata.
  - The IBM Tivoli Network Manager Probe sends events to the ObjectServer.

  It is more efficient to continue to connect IBM Tivoli Network Manager through a IBM Tivoli Netcool/OMNIbus integration as any events it generates flows into IBM Cloud Pak for AIOps.
- Any custom integrations, automations, or workflows that are done by using IBM Tivoli Netcool/Impact can either be automatically initiated from the IBM Tivoli Netcool/OMNIbus Aggregation ObjectServer pair by using an Event Reader, or from IBM Cloud Pak for AIOps directly by using the IBM Tivoli

Netcool/Impact integration.

Data flow architecture

IBM Cloud Pak for AIOps consists of many interconnected components. The following diagram shows an example of these interconnected components:

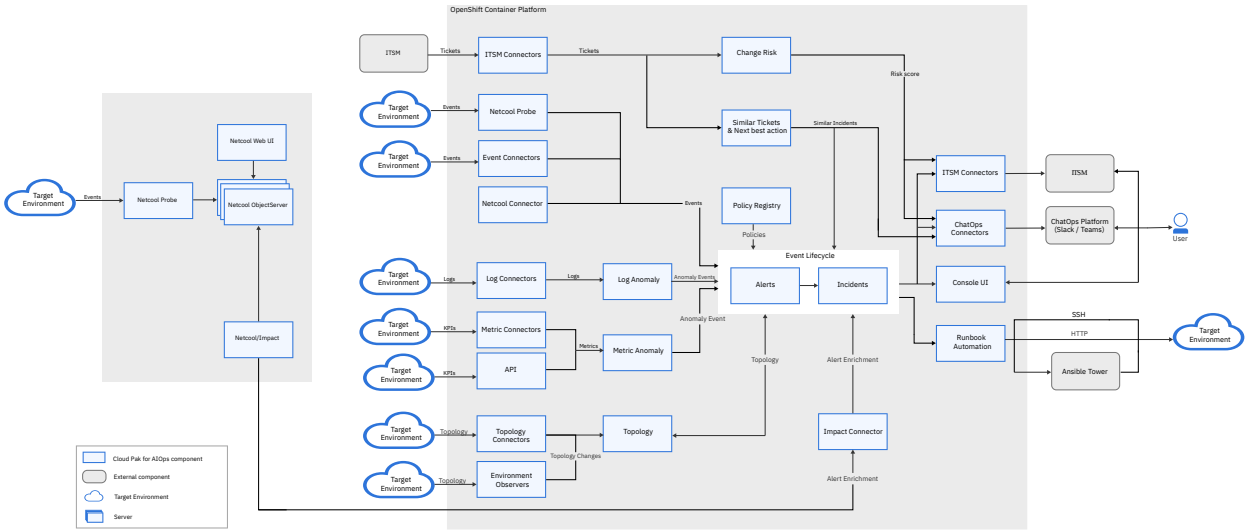


Figure. Data flow

The main workflow is the ingestion of events. Events are ingested from various sources, such as through integrations and Netcool probes. These events proceed through the event lifecycle, where de-duplication, suppression, correlation, incident creation, and many more insights and contexts are added. The resulting alerts and incidents are available to IT Operators through the IBM Cloud Pak for AIOps console, ChatOps, ITSM (ServiceNow) and APIs. In addition, the lifecycle incorporates automation through Runbooks to augment the alert or incident with additional diagnosis or attempts to automatically remediate the incident.

The next workflow brings in topology from the different observers of the estate. Topology merges these topology resources into a holistic view. It augments the topology with the observed events, as well as contributes to the context of alerts and incident and to insights. For example, underpinning the topological correlation and probable cause algorithms.

Both logs and metrics are brought in to provide more insights. They are analyzed for anomalies, which are then correlated into the incidents hence contributing to the full picture.

Finally, tickets are optionally created for the incidents to serve as a record of the incident. Resolved tickets are then ingested to provide context of similar occurrences in the past through similar tickets, and to convey the risk of a change request.

Extensibility architecture

IBM Cloud Pak for AIOps is highly extensible and customizable. IBM Cloud Pak for AIOps can include integrations for many of the popular data sources, events, metrics, logs, topology, and tickets. In addition to these default integrations, support is included for integrating with custom data sources, topology observers, and Netcool probes.

IBM Cloud Pak for AIOps supports a rich set of APIs allowing widespread integration. These APIs include retrieving and updating alerts, incidents, metrics, runbooks. For more information about API section, see APIs.

- For custom integrations without coding, IBM Cloud Pak for AIOps supports the use of webhook APIs for logs and events, the use of Kafka for logs and events, and the use of file-based integrations for topology. Custom metrics can be provided by using the metrics REST API.
- For building custom integrations with coding, IBM Cloud Pak for AIOps provides an SDK. This SDK supports custom user experience so that your custom integrations appear within the Cloud Pak for AIOps console integrations catalog. This SDK supports creating integrations for custom events, topology, logs, and tickets. In addition to ingesting tickets from custom ITSMs, the IBM Cloud Pak for AIOps SDK also supports ticket creation.

IBM Cloud Pak for AIOps also supports customizing the lifecycle of an event by using policies, which govern de-duplication, suppression, correlation, and incident creation. The policies can also be used to integrate with custom logic and enrichments by using IBM Tivoli Netcool/Impact. IBM Tivoli Netcool/Impact allows custom logic to be applied to a set of alerts or incidents. This logic can update the alert or incident itself, or run any logic based on the alert or incident. Runbooks can be associated with a defined set of alerts or incidents. These runbooks support custom logic run over SSH, REST or Ansible, and can be either manually or automatically run.

The detailed extensibility architecture resembles the following diagram:

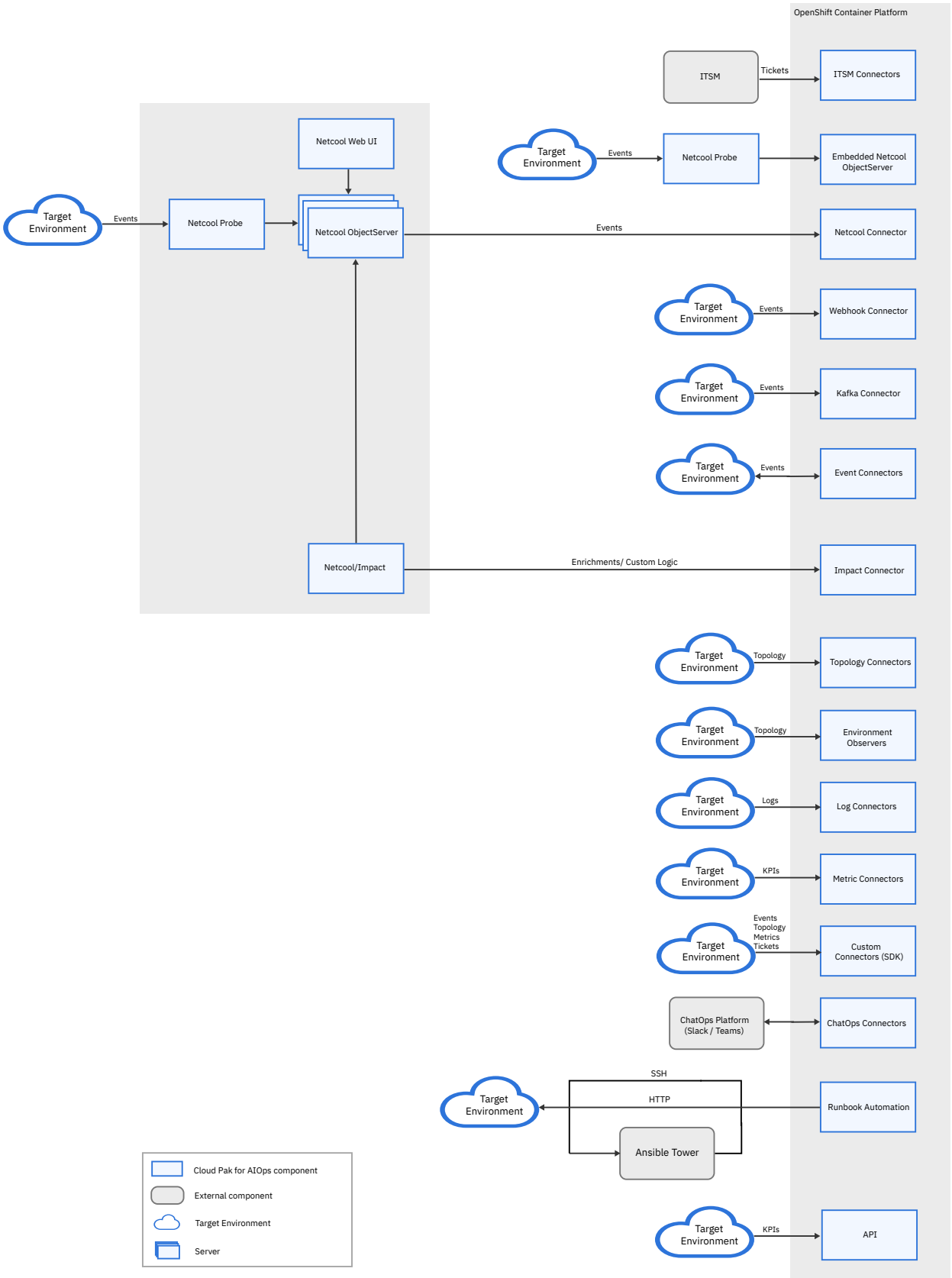


Figure. Extensibility

Infrastructure Automation architecture

Infrastructure Automation is a stand-alone module included within IBM Cloud Pak for AIOps. Infrastructure Automation brings together Infrastructure management and Managed services for an integrated, Infrastructure Automation capability for IT Operators.

The following sections illustrate how these capabilities come together in IBM Cloud Pak for AIOps:

- Deployment Architecture for Infrastructure Automation
- Deployment Architecture for IBM Cloud Pak for AIOps with Infrastructure Automation

Stand-alone Infrastructure Automation deployment architecture

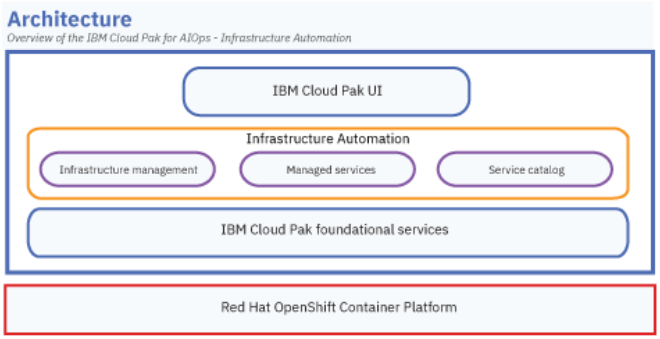


Figure. Infrastructure Automation architecture

Infrastructure automation is a stand-alone module within IBM Cloud Pak for AIOps. When you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure Automation.

Infrastructure Automation consists of these components (formerly part of IBM Cloud Pak for Multicloud Management):

- Managed services and service catalog, previously called Terraform & Service Automation or IBM Cloud Automation Manager
- Infrastructure management, previously called IBM Red Hat CloudForms.

Managed services provide the capability to automate provisioning of infrastructure and virtual machine applications by using Terraform and Ansible. It includes supported versions of Terraform that works across multiple cloud environments with comprehensive IT workflow orchestration. Using out-of-box integration with Ansible, workflows can interconnect Terraform provisioning templates with Ansible playbooks for configuration management, fully supporting infrastructure-as-code (IaC) and GitOps best practices.The service catalog allows authors administrators to publish services and users to self-service provision infrastructure and applications.

Infrastructure Management delivers the insight, control, compliance, and automation enterprises need to address the challenges of managing hybrid cloud environments, which are far more complex than physical ones. This technology enables enterprises with existing virtual, baremetal, and cloud infrastructures to improve visibility, control and compliance, and those just starting hybrid cloud deployments to build and operate a well-managed hybrid cloud infrastructure.

Infrastructure Automation enables IT Operations and Site Reliability Engineer (SRE) teams to use infrastructure as code practices to drive IT velocity and shift to the left of operations. With this modernized infrastructure management, you can perform the following tasks:

- Shorten the time to provision infrastructure (VMs, Cloud Services, Kubernetes) across on-premises, private, and public clouds
- Create repeatable templates for common infrastructure requests
- Enable self-service catalogs to avoid manual responses to tickets
- Scale infrastructure automatically based on demand
- Support day-2 operations, which include automated compliance
- You can provision, run, and manage workloads on the infrastructure of your choice.

IBM Cloud Pak for AIOps Infrastructure Automation is installed on Red Hat® OpenShift® Container Platform. For more information about the supported versions of Red Hat OpenShift Container Platform, see Supported Red Hat OpenShift Container Platform versions.

Infrastructure Automation with IBM Cloud Pak for AIOps deployment architecture

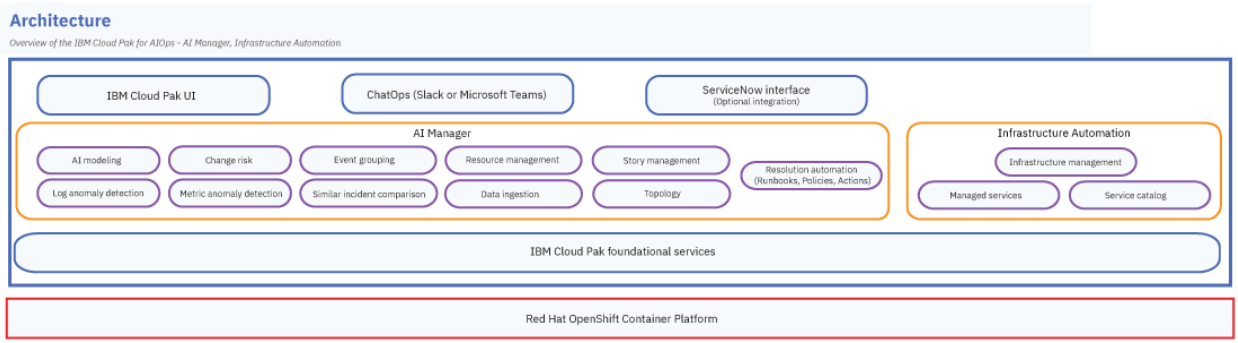


Figure. IBM Cloud Pak for AIOps with Infrastructure Automation architecture

Infrastructure automation can be used together with IBM Cloud Pak for AIOps to support use cases that require Terraform automation and service orchestration as part of the incident resolution process. Day 2 operations like changing the size of VMs, power management, or applying a patch are supported by Infrastructure Automation and can easily be accessed through a common console. They can also be automated through runbook actions by invoking Infrastructure Automation APIs. Both Infrastructure Automation and IBM Cloud Pak for AIOps integrate with ServiceNow and can be used together to calculate the risk score of infrastructure change requests.

Compliance

IBM Cloud Pak® for AIOps complies with the following security and privacy assessments, which can apply to all add-on services on IBM Cloud Pak for AIOps.

Customers are responsible for ensuring their own readiness for the laws and regulations that apply to them.

Customers are responsible for identifying and interpreting any relevant laws and regulations that might affect their users. Customers are also responsible for any actions their users might need to take to comply with these laws and regulations.

Customers must track whether personal information is present in their data. IBM Cloud Pak for AIOps is not aware of the presence of personal information in data imported from the client, such as log files. For more information about types of personal data that is processed by IBM Cloud Pak for AIOps, see Data Collection.

FIPS compliance

Federal Information Processing Standards (FIPS) are standards and guidelines that are issued by the National Institute of Standards and Technology (NIST) for federal government computer systems.

For more information, see Federal Information Processing Standards (FIPS).

FISMA compliance

For Federal Information Security Management Act (FISMA) level deployments, you need to follow the **FIPS compliance** and **Audit logging** sections.

For more information about FIPS compliance, see FIPS compliance.

For more information about Audit logging, see Audit logging.

For more questions around FISMA ready, contact your IBM Sales representative (or Business Partner).

GDPR

Customers are responsible for ensuring their own readiness for the laws and regulations, including the European Union General Data Protection Regulation.

For more information, see the following topics:

- Security and Privacy by Design (SPbD)
- Platform considerations for GDPR readiness

Federal Information Processing Standards (FIPS)

Federal Information Processing Standards (FIPS) are standards and guidelines that are issued by the National Institute of Standards and Technology (NIST) for federal government computer systems.

The standards are developed when compelling federal government requirements for standards, such as for security and interoperability, exist, but acceptable industry standards or solutions do not exist. Government agencies and financial institutions use these standards to ensure that products conform to specified security requirements.

Important: FIPs mode is not supported for deployments of IBM Cloud Pak for AIOps on Linux®.

Encryption with FIPS support enabled

When FIPS support is enabled, IBM Cloud Pak for AIOps uses cryptographic modules that are compliant with Level 1 of the Federal Information Processing Standard FIPS-140-2. Certificates that are used internally are encrypted by using FIPS-approved cryptography algorithms. FIPS-approved modules can optionally be used for the transmission of data. Traffic inside the IBM Cloud Pak for AIOps boundary is still secure, as traffic between nodes is automatically encrypted at the Red Hat® OpenShift® Container Platform level when TLS protection is enabled, while traffic inside a given node happens in-memory and does not leave the node.

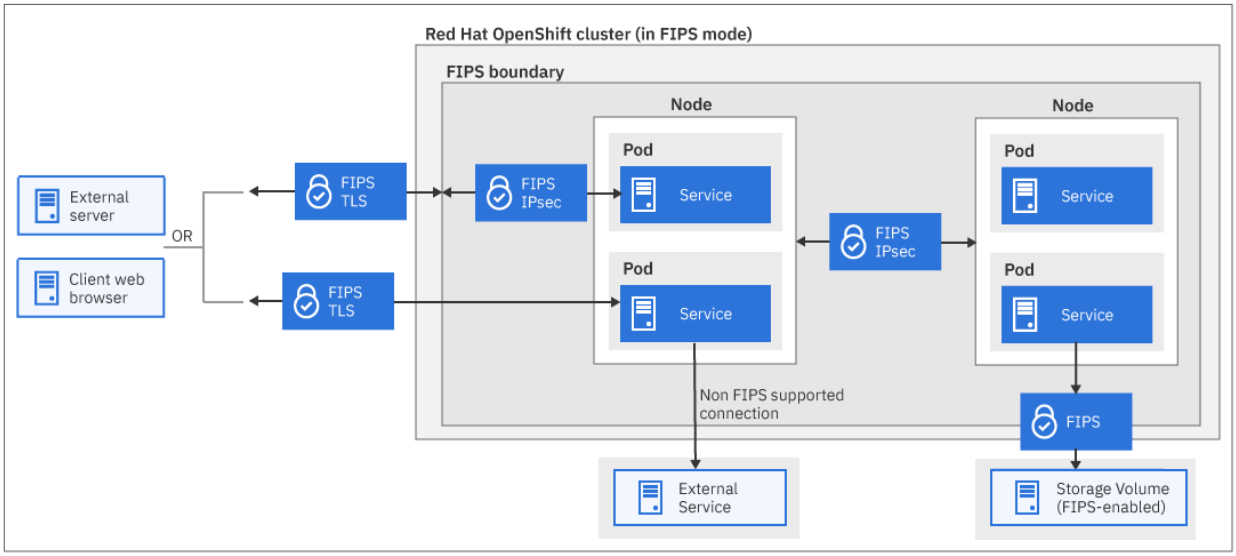


Figure. FIPs overview

FIPS (Federal Information Processing Standards) compliant encryption is validated for IBM Cloud Pak for AIOps services and components, including the IBM Cloud Pak foundational services that are used by IBM Cloud Pak for AIOps.

With FIPS enabled Data is FIPS encrypted at rest and inbound communications are FIPS encrypted. Outbound communications can support both FIPS enabled and non-enabled integrations. For FIPS enabled integrations, outbound integrations rely on the server to ensure FIPS ciphers are chosen. To ensure that integrations, including Observers, are FIPS enabled, an external service to mandate the use of FIPS compliant ciphers when negotiating encryption is required. Note that some restrictions to SSH actions accessing external endpoints apply when FIPS is enabled. For more information, see SSH algorithms.

Enabling FIPS support

To enable FIPS support, you must enable this support when you are installing Red Hat OpenShift Container Platform and IBM Cloud Pak for AIOps. For more information, see:

- IBM Cloud Pak for AIOps: Enabling Federal Information Processing Standards (FIPS) support
- Infrastructure Automation: Enabling Federal Information Processing Standards (FIPS) support

Enabling Federal Information Processing Standards (FIPS) support for IBM Cloud Pak for AIOps

Learn how to enable FIPS and run IBM Cloud Pak for AIOps on a FIPS-compliant system.

To enable support you need to install IBM Cloud Pak for AIOps on a Red Hat OpenShift Container Platform cluster that is running in a FIPS enabled mode. You also need to complete some tasks before you begin installing IBM Cloud Pak for AIOps on your cluster and complete other tasks while you are installing IBM Cloud Pak for AIOps. Depending on whether you plan to enable or set up some optional integrations, you might need to further complete tasks after your finish installing to ensure your integrations support FIPS.

Important:

- FIPs mode is not supported for deployments of IBM Cloud Pak for AIOps on Linux®.
- You can only enable FIPS support when you are installing. You cannot upgrade a non-FIPS enabled environment to a FIPS enabled environment. You also cannot backup an IBM Cloud Pak for AIOps from a non-FIPS enabled environment and restore it into a FIPS enabled environment. The remote kafka integration feature is not available on FIPS-compliant environments.

To enable FIPS support, complete the following tasks during your overall cluster and IBM Cloud Pak for AIOps installation:

1. Installing Red Hat OpenShift Container Platform and storage tasks required for compliance.

   These tasks must be completed when you are installing your Red Hat OpenShift Container Platform cluster and configuring your storage before you begin to install IBM Cloud Pak for AIOps on your OpenShift cluster.

2. Installing IBM Cloud Pak for AIOps tasks required for compliance

   While installing IBM Cloud Pak for AIOps you do need to complete some other required tasks to ensure you enable your environment to support FIPS.

3. Optional. Conditional tasks required for compliance

   Depending on the integrations that you plan to create for your IBM Cloud Pak for AIOps environment, you need to complete other tasks to enable FIPS support. For instance if you are integrating IBM Cloud Pak for AIOps with IBM Tivoli Netcool/OMNIbus, you need to configure the integration to support FIPS.

Installing Red Hat OpenShift Container Platform and storage tasks required for compliance

1. Enable FIPS support for your Red Hat OpenShift Container Platform cluster
2. Enable FIPS support for your storage

1. Enable FIPS support for your Red Hat OpenShift Container Platform cluster

You must enable FIPS support on your Red Hat OpenShift Container Platform cluster before you proceed with installing IBM Cloud Pak for AIOps. When you are installing IBM Cloud Pak for AIOps, the installation process automatically detects whether FIPS support is enabled on your Red Hat OpenShift Container Platform cluster. If FIPS is enabled on Red Hat OpenShift Container Platform, FIPS support is automatically enabled for IBM Cloud Pak for AIOps.

To enable FIPS for IBM Cloud Pak for AIOps, you must first enable FIPS support on your Red Hat OpenShift Container Platform cluster by completing the following tasks as part of installing Red Hat OpenShift Container Platform:

1. Enable FIPS mode on all of your nodes. For more information, see the Red Hat OpenShift Container Platform documentation about Support for FIPS cryptography
2. Install Red Hat OpenShift Container Platform in FIPS mode. For more information, see the Red Hat OpenShift Container Platform documentation about Installing a cluster in FIPS mode.

   Restriction: FIPS is supported only on x86_64 hardware.
3. Configure TLS protection for your node-to-node communication. Node-to-node communication must be TLS protected at all times.

   Recommended: Configure IPSec tunnels for communication between nodes in your cluster. IPSec is the validated and recommended method for implementing security across nodes for IBM Cloud Pak for AIOps. If you require a different method to secure your nodes, you can use your preferred method.

   With IPsec enabled, all network traffic between nodes on the OVN-Kubernetes Container Network Interface (CNI) cluster network travels through an encrypted tunnel. IPsec is disabled by default when you install OpenShift 4.x clusters. IPsec encryption can be enabled only during cluster installation and cannot be disabled after it is enabled.

   For more information about configuring IPsec encryption on Red Hat OpenShift Container Platform, see the Red Hat OpenShift Container Platform documentation about Configuring IPsec encryption.

2. Enable FIPS support for your storage

If your deployment's storage must be FIPS compliant, enable any FIPS settings and support for your chosen storage. Refer to your storage provider's documentation to ensure that your storage meets this requirement.

Red Hat® OpenShift® Data Foundation and FIPS

Red Hat® OpenShift® Data Foundation uses FIPS 140-2 certified cryptographic modules. You must use cluster wide encryption, and not encrypt persistent volumes individually. This is because Red Hat OpenShift Data Foundation persistent volume encryption is only available for block storage, and one of IBM Cloud Pak for AIOps's components, Zen, requires file storage. For more information, see the topic *Cluster-wide encryption* in the Red Hat OpenShift Data Foundation documentation.

Portworx and FIPS

Portworx uses FIPS 140-2 certified cryptographic modules. Portworx can encrypt each of the persistent volumes individually, or can encrypt the whole storage cluster. For more information, see the topic *Create encrypted PVCs* in the Portworx documentation.

Installing IBM Cloud Pak for AIOps tasks required for compliance

With your storage configured, continue your installation of IBM Cloud Pak for AIOps on your cluster.

Ensure that you complete the required installation tasks for compliance while you are installing IBM Cloud Pak for AIOps. These tasks must be completed while you are initially installing IBM Cloud Pak for AIOps.

For more information about, and instructions, for installing IBM Cloud Pak for AIOps, see:

- For more information about installing on specific cloud platforms, see Supported cloud environments for installing IBM Cloud Pak for AIOps.
- For more information about installing IBM Cloud Pak for AIOps overall, see Production installation. If you are installing a starter installation, see Starter installation.

Important: You must ensure that you have run the *Verify the installation* step for your chosen installation method, and that your deployment is healthy before you proceed.

Enable two-factor authentication (2FA)

To secure your environment with multi-factor authentication, you can configure single sign-on (SSO) between IBM Cloud Pak for AIOps and an identity provider that supports 2FA, such as IBM Security Verify. The identity provider's 2FA mechanism can then be used to authenticate IBM Cloud Pak for AIOps user login. Consult your chosen identity provider's documentation to setup 2FA.

For more information about IBM Security Verify, see IBM Security Verify.

Blog: A blog that provides an example of setting up single sign-on between a IBM Cloud Pak and IBM Security Verify is here: Tutorial: IBM Cloud Pak single sign-on (SSO) integration with IBM Security Verify.

Conditional tasks required for compliance

The following tasks are required depending on the integrations you plan to create for your IBM Cloud Pak for AIOps environment:

- IBM Cloud Pak for AIOps and Netcool integration compliance requirement: Enabling FIPS support

IBM Cloud Pak for AIOps and Netcool integration compliance requirement: Enabling FIPS support

If you plan to integrate IBM Cloud Pak for AIOps with IBM Tivoli Netcool/OMNIbus, you need to configure your integration to enable FIPS support. For more information, see Enabling Federal Information Processing Standards (FIPS) support for a Netcool integration.

Enabling Federal Information Processing Standards (FIPS) support for Infrastructure Automation

Learn how to enable FIPS and run Infrastructure Automation on a FIPS-compliant system.

To enable support you need to install Infrastructure Automation on a Red Hat OpenShift Container Platform cluster that is running in a FIPS enabled mode. You also need to complete some tasks before you begin installing Infrastructure Automation on your cluster and complete other tasks while you are installing Infrastructure Automation. Depending on whether you plan to enable or set up some optional integrations, you might need to further complete tasks after your finish installing to ensure your integrations support FIPS.

Important: You can only enable FIPS support when you are installing. You cannot upgrade a non-FIPS enabled environment to a FIPS enabled environment. You also cannot backup an Infrastructure Automation from a non-FIPS enabled environment and restore it into a FIPS enabled environment.

To enable FIPS support, complete the following tasks during your overall cluster and Infrastructure Automation installation:

1. Installing Red Hat OpenShift Container Platform and storage tasks required for compliance.

   These tasks must be completed when you are installing your Red Hat OpenShift Container Platform cluster and configuring your storage before you begin to install Infrastructure Automation on your OpenShift cluster.

2. Installing Infrastructure Automation tasks required for compliance

   While installing Infrastructure Automation you do need to complete some other required tasks to ensure you enable your environment to support FIPS:

3. Optional. Conditional tasks required for compliance

   Depending on the components that you plan to install in, or integrate with, your Infrastructure Automation environment, you need to complete other tasks to enable FIPS support. For instance if you plan to enable Managed services, you need to configure the deployment to support FIPS.

Installing Red Hat OpenShift Container Platform and storage tasks required for compliance

1. Enable FIPS support for your Red Hat OpenShift Container Platform cluster
2. Enable FIPS support for your storage

1. Enable FIPS support for your Red Hat OpenShift Container Platform cluster

You must enable FIPS support on your Red Hat OpenShift Container Platform cluster before you proceed with installing Infrastructure Automation. When you are installing Infrastructure Automation, the installation process automatically detects whether FIPS support is enabled on your Red Hat OpenShift Container Platform cluster. If FIPS is enabled on Red Hat OpenShift Container Platform, FIPS support is automatically enabled for Infrastructure Automation.

To enable FIPS for Infrastructure Automation, you must first enable FIPS support on your Red Hat OpenShift Container Platform cluster by completing the following tasks as part of installing Red Hat OpenShift Container Platform:

1. Enable FIPS mode on all of your nodes. For more information, see the Red Hat OpenShift Container Platform documentation about Support for FIPS cryptography

2. Install Red Hat OpenShift Container Platform in FIPS mode. For more information, see the Red Hat OpenShift Container Platform documentation about Installing a cluster in FIPS mode.

   Restriction: FIPS is supported only on x86_64 hardware.

3. Configure TLS protection for your node-to-node communication. Node-to-node communication must be TLS protected at all times.

   Recommended: Configure IPSec tunnels for communication between nodes in your cluster. IPSec is the validated and recommended method for implementing security across nodes for IBM Cloud Pak for AIOps. If you require a different method to secure your nodes, you can use your preferred method.

   With IPsec enabled, all network traffic between nodes on the OVN-Kubernetes Container Network Interface (CNI) cluster network travels through an encrypted tunnel. IPsec is disabled by default when you install OpenShift 4.x clusters. IPsec encryption can be enabled only during cluster installation and cannot be disabled after it is enabled.

   For more information about configuring IPsec encryption on Red Hat OpenShift Container Platform, see the Red Hat OpenShift Container Platform documentation about Configuring IPsec encryption.

2. Enable FIPS support for your storage

If your deployment's storage must be FIPS compliant, enable any FIPS settings and support for your chosen storage. Refer to your storage provider's documentation to ensure that your storage meets this requirement.

Red Hat® OpenShift® Data Foundation and FIPS

Red Hat® OpenShift® Data Foundation uses FIPS 140-2 certified cryptographic modules. You must use cluster wide encryption, and not encrypt persistent volumes individually. This is because Red Hat OpenShift Data Foundation persistent volume encryption is only available for block storage, and one of IBM Cloud Pak for AIOps's components, Zen, requires file storage. For more information, see the topic *Cluster-wide encryption* in the Red Hat OpenShift Data Foundation documentation.

Portworx and FIPS

Portworx uses FIPS 140-2 certified cryptographic modules. Portworx can encrypt each of the persistent volumes individually, or can encrypt the whole storage cluster. For more information, see the topic *Create encrypted PVCs* in the Portworx documentation.

Installing Infrastructure Automation tasks required for compliance

With your storage configured, continue your installation of Infrastructure Automation on your cluster.

Ensure that you complete the required installation tasks for compliance while you are installing Infrastructure Automation. These tasks must be completed while you are initially installing Infrastructure Automation.

For more information about, and instructions, for installing Infrastructure Automation overall, see Installating Infrastructure Automation.

Enable two-factor authentication (2FA)

To secure your environment with multi-factor authentication, you can configure single sign-on (SSO) between Infrastructure Automation and an identity provider that supports 2FA, such as IBM Security Verify. The identity provider's 2FA mechanism can then be used to authenticate Infrastructure Automation user login. Consult your chosen identity provider's documentation to setup 2FA.

For more information about IBM Security Verify, see IBM Security Verify.

Blog: A blog that provides an example of setting up single sign-on between a IBM Cloud Pak and IBM Security Verify is here: Tutorial: IBM Cloud Pak single sign-on (SSO) integration with IBM Security Verify.

Conditional tasks required for compliance

The following tasks are required depending on the components that you plan to enable for your Infrastructure Automation environment:

- Deploy Infrastructure Automation - Managed services in FIPS mode

Deploy Infrastructure Automation - Managed services in FIPS mode

If you need to deploy Managed services in your FIPS enabled environment, you can deploy the component after you complete your install by editing the **`IAConfig`** settings. This task can be completed after you complete your initial installation, but needs to be completed when you are initially deploying Managed services.

1. Use the OpenShift (**`oc`**) CLI to log in to your OpenShift cluster and switch to the project (namespace) where Infrastructure Automation is installed.
2. Run the following command to open the **`IAConfig`** configuration for editing:

   ```
   oc -n <namespace> edit IAConfig
   ```

   Where **`<namespace`** is the project (namespace) where Infrastructure Automation is installed.
3. Search for the **`spec`** section and edit the configuration to include the settings to deploy Managed services (**`ibm-management-cam-install`**) in FIPS mode (**`enableFIPS: true`**). Your updated configuration for deploying managed services can resemble the following example spec:

   ```
   spec:
       imagePullSecret: ibm-entitlement-key
       infraAutoComposableComponents:
       - enabled: true
         name: ibm-management-im-install
         spec: {}
       - enabled: true
         name: ibm-management-cam-install
         spec:
           manageservice:
             global:
               enableFIPS: true
             license:
               accept: true
       license:
         accept: true
   ```

Enabling Federal Information Processing Standards (FIPS) support for an IBM Tivoli Netcool/OMNIbus integration

If you plan to integrate IBM Cloud Pak for AIOps with IBM Tivoli Netcool/OMNIbus, you can configure your integration to enable support for Federal Information Processing Standards (FIPS).

Prerequisites

- You must have an existing FIPS-compliant installation of IBM Cloud Pak for AIOps. You cannot enable FIPS support for the integration on a noncompliant system.

  For more information about installing IBM Cloud Pak for AIOps to be FIPS-compliant, see Enabling support for Federal Information Processing Standards (FIPS).

- You must have an existing FIPS-compliant installation of IBM Tivoli Netcool/OMNIbus.

Procedure

To enable FIPS mode in IBM Tivoli Netcool/OMNIbus, complete the following steps on the host where IBM Tivoli Netcool/OMNIbus (on premises ObjectServer) is running:

1. Stop the ObjectServer pair and the IBM Tivoli Netcool/OMNIbus bidirectional gateway.

   - To stop the ObjectServer, complete the following steps:

     - Open the ObjectServer SQL prompt:

       ```
       $OMNIHOME/bin/nco_sql -server <OS_NAME> -user root -password <plain-text-pwd>
       ```

     - On the SQL prompt, run the following command:

       ```
       > alter system shutdown;
       > go
       ```

     Note: After you enter **go**, the SQL prompt disconnects as the ObjectServer is exiting.

   - To stop the bidirectional gateway, run the following command:

     ```
     kill -15 <pid>
     ```

     where **pid** is Process ID.

     Note: Use the **ps -ef | grep nco_g_objserv_bi** command to get the gateway's pid.

2. Enter the following information in the properties file of both the ObjectServers:

   ```
   PasswordEncryption   : 'AES'
   ConfigKeyFile        : '<key_file>'
   ```

   Note: To generate the key file, run the following command on the host:

   ```
   $OMNIHOME/bin/nco_keygen -o <key_file> -l 256
   ```

3. Start the ObjectServer pair and update the same FIPS-compliant password for the root user in each ObjectServer.

   1. To start the ObjectServer, run the following command:

      ```
      $OMNIHOME/bin/nco_objserv -name <OS_NAME> -propsfile <objectserver-properties-file> &
      ```

   2. To open the ObjectServer SQL prompt, run the following command:

      ```
      $OMNIHOME/bin/nco_sql -server <OS_NAME> -user root -password <plain-text-pwd>
      ```

   3. To alter root password, run the following SQL command:

      ```
      > alter user 'root' set password '<plain_text_pwd>';
      > go
      ```

   4. To verify that the Passwd column contains AES-encrypted passwords with 24 characters, run the following SQL command:

      ```
      > select UserName,Passwd from security.users;
      > go
      ```

4. Generate the encrypted IBM Tivoli Netcool/OMNIbus root password by using the ConfigKeyFile.

   - To create the encrypted password, run the following command:

     ```
     $OMNIHOME/bin/nco_aes_crypt -c AES_FIPS -k string -o <encrypted_file> -k <key_file> <root_plain_text_pwd>
     ```

     Note: For more information about generating a key in a key file, see Generating a key in a key file.

5. Apply the configurations in the properties file of the bidirectional gateway:

   ```
   Gate.ObjectServerA.Password  : '<encrypted_root_pwd>'
   Gate.ObjectServerB.Password  : '<encrypted_root_pwd>'
   ConfigKeyFile                : '<key_file>'
   ```

6. Place the following files in the **$NCHOME/etc/security** directory to enable FIPS mode in IBM Tivoli Netcool/OMNIbus:

   - **fips.conf**:

```
SP800_131MODE=TRUE
TLS12_ONLY=TRUE
```

- `sslciphers.conf`:

```
sslv3_remove=_
tls10_remove=_
tls11_remove=_
tls12_remove="TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WI
TH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_
SHA256"
```

7. Start the ObjectServer pair and the bidirectional gateway.

   - To start the ObjectServer, run the following command on the host:

     ```
     $OMNIHOME/bin/nco_objserv -name <OS_NAME> -propsfile <objectserver-properties-file> &
     ```

     For example,

     ```
     $OMNIHOME/bin/nco_objserv -name AGG_P -propsfile $OMNIHOME/etc/AGG_P.props &
     ```

   - To start the bidirectional gateway, run the following command on the host:

     ```
     $OMNIHOME/bin/nco_g_objserv_bi -propsfile <gateway-properties-file> &
     ```

     For example,

     ```
     $OMNIHOME/bin/nco_g_objserv_bi -propsfile $OMNIHOME/etc/AGG_GATE.props &
     ```

Audit logging

The audit logging capabilities within IBM Cloud Pak for AIOps provide you with the capability to track changes that were made to your system, when the changes were made, and by whom. The recorded audit logs include the record of actions that were completed or attempted by users or services of your IBM Cloud Pak for AIOps system. This auditing helps you with ensuring accountability, trackability, and regulatory compliance for data access, modification, and security within your IBM Cloud Pak for AIOps environment.

If you require that your environment supports the Federal Regulatory Compliance Standard (FISMA), you need to ensure that your environment has audit logging.

What is audited in IBM Cloud Pak for AIOps?

IBM Cloud Pak for AIOps capabilities audit the following types of actions, which can return, manage, manipulate, or access sensitive data:

- Successful and unsuccessful login attempts
- Authentication checks
- Authorization checks
- Account management events
- Object access
- Policy changes
- Privileged functions
- Data access
- Data changes
- Process tracking
- System events
- Administrator activities
- Permission changes

For more information and details about what is logged for IBM Cloud Pak for AIOps, see Logged actions and data for auditing.

How do you configure audit logging?

Application logging

For IBM Cloud Pak for AIOps capabilities, audit logging is configured and enabled by default.

For Infrastructure Automation, you need to enable audit logging during installation. For the list of audit-related installation parameters, see Installation parameters.

IBM Cloud Pak for AIOps uses the IBM Cloud Pak foundational services Identity Management (IM) and Integrated UI (Platform UI **Zen**) services. For more information about auditing logging for these services, see Auditing IM service.

Platform logging

If you have a deployment of IBM Cloud Pak for AIOps or Infrastructure Automation on Red Hat OpenShift, then you must configure audit logging for platform level calls. For Red Hat OpenShift Container Platform audit logging, see the Red Hat OpenShift Container Platform documentation Viewing audit logs.

If you have a deployment of IBM Cloud Pak for AIOps on Linux®, then platform logging is enabled by default. Logs are stored in the **aiopsctl** directory on each control plane node. Logs are retained for up to 30 days, with a maximum total log size of 1GB.

How do you view a record of audit logs?

- For IBM Cloud Pak for AIOps capabilities, the audit logs for all recorded actions are written to standard output **stdout** in JSON format.

  Viewing audit logs for a deployment on Red Hat OpenShift

  You can view an audit log for a pod by using the **oc logs** command.

  If you want to, or need to, view multiple audit logs, you can ingest and view the logs with a log aggregator, such as the logging subsystem for Red Hat OpenShift. With this logging subsystem, you can aggregate all the logs from your OpenShift Container Platform cluster, including IBM Cloud Pak for AIOps audit logs. You can then view the logs with the Kibana web console. For more information about the logging subsystem for Red Hat OpenShift, see the Red Hat OpenShift documentation Installing the logging subsystem for Red Hat OpenShift.

  You can also use other services for log aggregation, such as Falcon Logscale or an Elasticsearch, Logstash, and Kibana (ELK) stack.

  Run the following command to view the API audit logs, which are stored in the **ibm-nginx** pod logs:

  ```
  oc logs -l component=ibm-nginx -c ibm-nginx-container | grep eventTime
  ```

  Viewing audit logs for a deployment on Linux

  To review the audit logs, run the following command on a control plane node:

  ```
  ls ~/.aiopsctl/logs/
  ```

  Example output:

  ```
  aiopsctl.log  audit-2024-09-18T15-17-26.791.log  audit-2024-09-18T16-11-15.367.log  audit.log
  ```

- For Infrastructure Automation, Infrastructure management appliances use the systemd logger, journalctl for audit logging. Appliances that run UI workers log the UI requests. Appliances that run the web services role log the API requests.

  Logging for podified deployments use the standard output **stdout** logger. You can view these logs through an **oc logs** command. To view a log, you need to specify the correct UI or web service pod.

What is included in an audit log

The audit log messages for a recorded action or event in IBM Cloud Pak for AIOps include information to identify the action or event that occurred, when the action or event occurred, the action or event source and outcome, and the identity of the associated user or service.

For the more information about what is included in audit log messages, including example messages, see Audit log messages.

Logged actions and data for auditing

For audit logging, IBM Cloud Pak for AIOps capabilities log information about numerous user interactions, system processes, and events, which can each return, manage, manipulate, or access sensitive data. The following lists, detail the actions and data that is recorded in audit logs.

- IBM Cloud Pak for AIOps
- Infrastructure Automation

IBM Cloud Pak for AIOps

IBM Cloud Pak for AIOps capabilities audit the following types of actions, which can return, manage, manipulate, or access sensitive data:

- Successful and unsuccessful login attempts
- Authentication checks These checks run when a user attempts to log in to the UI. The record for these checks includes whether the attempt was successful. These checks do not include checks for internal system service-to-service authentication actions, such as when a secret is used to authenticate to a database.
- Authorization checks These checks run to determine whether a user has the required role and permission for completing an action.
- Account management events These events include actions for adding, changing, removing users, and managing credentials and role-based access control.
- Object access This access includes the access of data objects through the IBM Cloud Pak for AIOps UI or APIs.
- Policy changes The auditing of these changes includes policy changes for the management of user access controls, such as access policies. This auditing does not include event suppression policies.
- Privileged functions This auditing records changes for administrator access controls, such as access policies and their ability to run administrator-level tasks.
- Data access This auditing records the read-access any of data, such as viewing data on a UI page. The viewing of the following data through the UI or API is audited:
  - Integrations
    - Viewing the set of configured integrations
    - Viewing the status of an integration
    - Viewing the configuration of an integration

    - AIOps Insights
        - Viewing aggregate usage information, such as information about the number of resources, events, incidents, runbooks
    - AI Models
        - Viewing the set of configured AI model definitions
        - Viewing a AI model definition's status
        - Viewing a AI model definition's configuration
        - Viewing the status and details of an AI model version
    - Resources
        - Viewing the list of resources, resources groups, or applications
        - Searching resources, resource groups, or applications
        - Viewing the configuration of a resource, resource group, or application
        - Viewing the topology of an application
    - Incidents and alerts
        - Viewing the list of incidents or alerts
        - Viewing incident or alert details
    - Automations
        - Viewing the list of policies
        - Viewing a specific policy
        - Viewing the list of runbooks
        - Viewing a specific runbook
        - Viewing the list of automation actions
        - Viewing a specific automation action
- Data changes This auditing records actions for creating or changing data. Changes to the following types of data are recorded:
    - Integrations
        - Creating or editing an integrations
    - AI Models
        - Creating or editing an AI model definition
        - Creating an AI model version through AI training
        - Editing the state of an AI model version
    - Resources
        - Creating or editing a resource, resource group, or application
    - Alerts & Incident
        - Changing the state of an alert, including by policies
        - Editing an alert journal
        - Changing the incident details, such as the incident owner or state
    - Automations
        - Creating or editing a policy, runbook, or action
- Data deletions This auditing records the deletion of the following types of data:
    - Integrations
        - Deleting an integrations
    - AI models
        - Deleting an AI model definition or AI model version
    - Resources
        - Deleting a resource, resource group, or application
    - Automations
        - Deleting a policy, runbook, or action
- Process tracking This auditing records the action that caused a policy to run. A policy action is not caused by a user interaction. Instead, the action is caused by the policy service, with the data object for which the policy is applied being the target. Audit logs record an entry (policy ID, ID of the entity the policy ran on) whenever a policy runs due to one of the following actions occurring:
    - aiops.ibm.com/trigger/event-received
    - aiops.ibm.com/trigger/alert-pre-create
    - aiops.ibm.com/trigger/alert-created
    - aiops.ibm.com/trigger/alert-pre-update
    - aiops.ibm.com/trigger/alert-pre-delete
    - aiops.ibm.com/trigger/incident-pre-create
    - aiops.ibm.com/trigger/incident-created
    - aiops.ibm.com/trigger/incident-pre-update
    - aiops.ibm.com/trigger/incident-pre-delete

  One or more of these actions can occur when one of the following processes run:
    - Alert suppression through a policy
    - Alert creation through a policy
    - Correlating an alert to another alert through a policy
    - Creating an incident through a policy
    - Assigning or running an action through the UI, API, or policy
    - Assigning or running a runbook through the UI, API, or policy
    - Running an Ansible, HTTP or SSH automation action on an external system
    - Creating or updating an incident ticket through a policy
    - Creating or updating a ChatOps message through a policy
    - Determining a change risk score through a policy
    - Updating a change ticket with a change risk score through a policy
    - Providing model feedback
- System events This auditing records system and maintenance tasks that are not included within the preceding categories, such as the following actions:
    - Back up and restore

- Administrator activities This auditing records activities that require a user with an administrator role, including the following activities:
    - Integrations
        - Viewing the set of configured integrations
        - Viewing the status of an integrations
        - Viewing the configuration of an integrations
        - Creating, editing, or deleting an integrations
    - AI models
        - Viewing the set of configured AI model definitions
        - Viewing a AI model definition's status
        - Viewing a AI model definition's configuration
        - Viewing the status and details of an AI model version
        - Creating, editing, or deleting an AI model definition
        - Creating, editing, or deleting an AI model version
    - Resources
        - Viewing the configuration of a resource, resource group, or application
        - Creating, editing, or deleting a resource, resources group, or application
    - Automations
        - Viewing a specific runbook
        - Viewing the list of automation actions
        - Viewing a specific automation action
        - Creating, editing, or deleting a policy, runbook, or action
- Permission changes Records of the changes and management of user permissions.

Infrastructure Automation

Infrastructure Automation capabilities audit the following types of actions and data:

- Infrastructure management
- Managed services

Infrastructure management

Infrastructure management audits UI and API interactions. This logging includes UI privilege checks, invalid session requests through a session timeout or reloading a page after a user logs out.

Managed services

Managed services logging includes records for the following actions:

- Logging in or out of the UI
- Creating, editing, or deleting a cloud connection
- Creating, editing or deleting a DataTypes in shared parameters
- Creating, editing or deleting a DataObjects in shared parameters
- Adding or editing namespace for a DataObjects in shared parameters
- Creating, editing or deleting dynamic parameters
- Deploying a stack
- Modifying an existing stack (plan and apply)
- Modifying an existing stack record attributes
- Destroying or deleting a stack
- Creating, deleting an IAAS resource task
- Creating, editing, or deleting an IAAS resource settings
- Creating, editing, or deleting a template
- Creating, editing, or deleting a template version
- Creating, update or deleting provider engine
- Creating, update or deleting providers plugins
- Creating, editing, duplicating, retiring or deleting a service
- Adding or editing namespaces for a service
- Creating or deleting a service plan
- Creating, editing, or deleting a service category
- Adding or editing a service version
- Publishing, retiring, unretiring, duplicating, or deleting a service version
- Pushing to, or Pulling from, a Git service version
- Deploying a service-instance
- Resuming or Retrying a service-instance
- Terminating or deleting a service-instance
- Invoking or terminating a bind action for a service-instance
- Invoking a custom action for a service-instance
- Generating update plan for a service-instance
- Updating a service-instance
- Getting Upgrade or Downgrade service versions for a service-instance
- Generating service version change plan for a service-instance
- Running service version change for a service-instance
- Creating, editing, or deleting an independent custom action category
- Creating, editing, deleting, or invoking independent custom action
- Set or Unset namespace for independent custom action
- Lock or Unlock an independent custom action

- Creating, editing, or deleting an SMTP connection
- Adding or Deleting subscription store configuration

Audit log messages

For IBM Cloud Pak for AIOps capabilities, the log message entries for a recorded action or event include information to identify the action or event that occurred, when the action or event occurred, the action or event source and outcome, and the identity of the associated user or service.

Note: Some audit-specific fields, such as the username, are part of a generic message field and are not broken out into a dedicated field.

For Infrastructure Automation Infratructure management, the audit log messages include the HTTP method, username, role, request ID, session ID, requested URL path, and details for any specific privilege checks.

For example, a request to load the dashboard at `/dashboard/show` logs the `show` action and the privilege check (`dashboard_view`) for that page. Some pages can have multiple privilege checks as the UI determines what data a user can view or act upon.

Log messages

Log messages can include the following details:

Table. Log message details

Field Name	Required (IBM Cloud Pak for AIOps)	Required (UI)	Required (API)	Description	Example	Jsonpath in logging (API)
logType	Yes	Yes	Yes	An indicator of the fact that this is an audit message. This is used for allowing easy identifying audit versus other log messages. This is always set to `audit`	`audit`	
id	No	No	No	The unique ID for the request	`271bfd04-a625-48ec-b33c-c7e9958f646c`	
initiator.id	Yes	Yes	Yes	ID of the identity or user that initiated the action.	User IAM ID API key	`.request.headers.audit-initiator-id`
initiator.name	Yes	Yes	Yes	Name of the identity or user that initiated the action.	username for users `system` `policy-service` for policy actions.	`.request.headers.audit-initiator-name`
initiator.typeURI	Yes	Yes	Yes	CADF URI description of the initiator type. For UI audit logs, this URI is most likely the user.	`security/account/user` for cookie or user initiated actions `security/account/service` or `security/account/admin` for apikey or service or system initiated actions.	`.request.headers.audit-initiator-typeuri`

initiator.credential.type	Yes	Yes	Yes	Type of authentication provided by the initiator. For UI audit logs, the value is often `cookie`.	`token`
					`user`
					`apikey`
					`cookie`
					`unknown`
						`.request.headers.audit-initiator-credential-type`

initiator.credential.identity_status	No	No	No	Status of the identity authentication for the initiator or user.	`Confirmed`
					`Denied`
					`Unknown`
						`request.headers.audit-initiator-credential-identity_status`

initiator.host.address	No	No	No	Source IP from where the request originated.	IP address of the Red Hat OpenShift Load Balancer.	`.client_ip`
initiator.host.agent	N/A	N/A	Yes	The client type of API request	`curl/7.61.1` `HTTPie/2.4.0`	`.request.headers.user-agent`
target.name	Yes	Yes	Yes	The name of the service that is accepting the request. For UI audit logs, this is populated as an environment variable in the deployed UI server (`AUDIT_SERVICE_NAME`)	`service; resource` `api.acme.com`	`.service.host`
target.id	Yes	No	Yes	CRN of the service accepting this request. For UI audit logs, this value is populated as an environment variable in the deployed UI server (`AUDIT_SERVICE_CRN`)	`b7dd7db9-45f2-4c05-abb5-01466c2abcdf`	`.service.host`
target.typeURI	Yes	No	Yes	This is a constant value that is stored in `AUDIT_RESOURCE_CADF_URI`.	`service` for HTTP/Ansible `system` for Script	`service`

Field Name	Required (IBM Cloud Pak for AIOps)	Required (UI)	Required (API)	Description	Example	Jsonpath in logging (API)
requestData.path	Yes	Yes	Yes	The request path, such as the API, which indicates the specific request being made of the service.	`api/v2/inventories/d220b490-7c75-11ed-b6f3-23388646a3b9`	`.request.uri`

requestData.type	No	Yes	No	The request method, which indicates whether this is a view, create, update, or delete.	`GET`
					`PUT`
					`DELETE`
					`POST`
						`.request.method`

requestData.actionId	No	No	No	Automation/Action ID.	`c6ccc725-3e54-493d-a114-de5130e7c64b`	

action	Yes	No	Yes	The action that triggers an event. This is superceded by `requestData.method`.	`create`
					`update`
					`delete`
					`deploy`
					`authenticate`
						`view`
						`create`

outcome	Yes	Yes	Yes	Was the action succesful?	`success`
					`pending`
					`failure`
					`unknown`
						`success`

reason.reasonCode	Yes	Yes	Yes	The HTTP response code of the result.	`200` for success.	`.response.status`
reason.reasonSize	Yes	Yes	Yes	The HTTP response size of the result.	`116`	`.response.size`
reason.message	Yes	No	Yes	The HTTP response message explaining the result. For API, Follow Format 'action [custom data] [outcome]'	`OK` for success.	`view success`
eventTime	Yes	Yes	Yes	The time, date, and time zone of the event in Coordinated Universal Time (UTC)	`2022-11-14T11:28:54.753Z`	`.started_at`
attachments.content.correlation_id	No	No	No	An optional ID that can be used to correlate multi-part audit logs.	`cccccc-eeee-dddd-aaaa-bbbbbbbbbbbb`	

Examples

Example: Policy action audit log

The following message is an example audit log message for policy actions:

```
{
  "eventTime": "2024-05-21T15:22:23+00:00",
  "initiator": {
    "credential.type": "token",
    "id": "1000331001",
    "name": "cpadmin",
    "typeURI": "security/account/user"
  },
  "logType": "audit",
  "reason": {
    "reasonCode": "200"
  },
  "requestData": {
    "path": "/aiops/api/issue-resolution/v1/alerts",
    "type": "GET"
  },
  "target": {
    "name": "cpd-cp4waiops.example.com",
    "typeURI": "service"
  }
}
```

Example: UI audit log

The following JSON is an example of audit log message for UI pod:

```
{
  "logType": "audit",
  "initiator": {
    "id": "1000331001",
    "name": "cpadmin",
    "typeURI": "security/account/user",
    "credential": {
      "type": "cookie",
      "identity_status": "Confirmed"
    },
  },
  "requestData": {
    "path": "/aiops/homepage/api/application",
    "type": "GET"
  },
  "target": {
```

```
    "name": "--"
  },
  "outcome": "success",
  "reason": {
    "reasonCode": 200,
    "message": "OK"
  },
  "eventTime": "2024-11-01T03:13:06.747Z"
}
```

Example: Automation audit log

The following JSON is an example of automation audit log message:

```
{
"logType": "audit",
"attachments": {
  "content": {
  "correlation_id": "c8084070-9aca-11ef-a826-21984ee4e499",
  "context": "runbook:302457ed-cd0d-4634-b5df-ae332562fcd4,runbookinstance:c8cb3d062ab8d7b0783cbf47e45761d5"
},
"type": "application/json"
},
"initiator": {
  "id": "ld",
  "name": "user123@mymail.com",
  "typeURI": "security/account/user"
},
"level": 30,
"target": {
  "name": "mytarget@example.com",
  "typeURI": "service"
},
"requestData": {
  "path": "/json",
  "type": "GET",
  "actionId": "95735585-61fa-4544-a865-57652188dd00",
  "action": "http.execute"
},
"eventTime": "2024-11-04T16:35:20.326Z",
"reason": {
  "reasonCode": 200,
  "message": "OK"
},
"outcome": "success",
"msg": "Security Audit Log Record",
"time": "2024-11-04T16:35:20.883Z",
"v": 0
}
```

Example: API audit log

The following JSON is an example of API audit log message:

```
{
    "outcome": "success",
    "eventTime": "2023-02-03T06:13:17Z",
    "reason": {
        "reasonSize": 438,
        "message": "view success",
        "reasonCode": 200
    },
    "initiator": {
        "name": "admin",
        "host": {
            "agent": "curl/7.61.1",
            "address": "10.9.5.41"
        },
        "id": "1000330999",
        "typeURI": "security/account/user",
        "credential": {
            "idenity_status": "Confirmed",
            "type": "token"
        }
    },
    "requestData": {
        "type": "GET",
        "path": "/aiops/api/application-manager/topology-rest-observer/v1/healthcheck"
    },
    "target": {
        "id": "aiops-topology-rest-observer.katamari.9104.svc",
        "typeURI": "service",
        "name": "aiops-topology-rest-observer.katamari.9104.svc"
    },
    "action": "view"
}
```

Example: Infrastructure Management audit log

The following example log shows an audit log that includes log entries for successful requests, logon failures, requests that failed due to missing product features, and an invalid (stale) session.

```
[----] W, [2023-01-27T10:02:29.500256 #17089:5a398]  WARN -- audit: <AuditFailure> Username [blah], from:
[Base.audit_failure], Authentication failed for userid blah
[----] I, [2023-01-27T10:02:36.633339 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], from:
[Base.audit_success], User admin successfully validated by EVM
[----] I, [2023-01-27T10:02:36.636510 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], from:
[Base.audit_success], Authentication successful for user admin
[----] I, [2023-01-27T10:02:37.045266 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [e35c5068-9cee-41c9-89c7-a12024b61e82], Method [GET], Path [/dashboard/show] Action: show
[----] I, [2023-01-27T10:02:37.045501 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [e35c5068-9cee-41c9-89c7-a12024b61e82], Method [GET], Path [/dashboard/show] Features
checked: dashboard_view
[----] I, [2023-01-27T10:02:39.294107 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [387576de-5e70-4517-867e-9d10259b8b73], Method [GET], Path
[/dashboard/widget_chart_data/48] Action: widget_chart_data
[----] I, [2023-01-27T10:02:39.294856 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [387576de-5e70-4517-867e-9d10259b8b73], Method [GET], Path
[/dashboard/widget_chart_data/48] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:39.440698 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [4f302068-b92a-497f-8c20-4c3e99c33057], Method [GET], Path
[/dashboard/widget_report_data/19] Action: widget_report_data
[----] I, [2023-01-27T10:02:39.443928 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [4f302068-b92a-497f-8c20-4c3e99c33057], Method [GET], Path
[/dashboard/widget_report_data/19] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:39.647895 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [efaa6a1e-ad5c-4f9f-9daf-e95be6cd1af5], Method [GET], Path
[/dashboard/widget_report_data/17] Action: widget_report_data
[----] I, [2023-01-27T10:02:39.649264 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [efaa6a1e-ad5c-4f9f-9daf-e95be6cd1af5], Method [GET], Path
[/dashboard/widget_report_data/17] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:39.934904 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [5912f12d-3a7c-497e-9766-9c62e18ea247], Method [GET], Path [/dashboard/widget_menu_data/34]
Action: widget_menu_data
[----] I, [2023-01-27T10:02:39.935245 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [5912f12d-3a7c-497e-9766-9c62e18ea247], Method [GET], Path [/dashboard/widget_menu_data/34]
Features checked: dashboard_view
[----] I, [2023-01-27T10:02:39.990627 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [1a7b7ec1-8d37-44ee-bd1a-6b857e4d5de0], Method [GET], Path
[/dashboard/widget_chart_data/100] Action: widget_chart_data
[----] I, [2023-01-27T10:02:39.993080 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [1a7b7ec1-8d37-44ee-bd1a-6b857e4d5de0], Method [GET], Path
[/dashboard/widget_chart_data/100] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:40.183948 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [33b791e5-2ba9-4493-af46-4a66e91978eb], Method [GET], Path
[/dashboard/widget_report_data/20] Action: widget_report_data
[----] I, [2023-01-27T10:02:40.184786 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [33b791e5-2ba9-4493-af46-4a66e91978eb], Method [GET], Path
[/dashboard/widget_report_data/20] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:40.395302 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [37b40523-d8e0-4db2-8450-e1d20947f5c9], Method [GET], Path [/dashboard/widget_chart_data/6]
Action: widget_chart_data
[----] I, [2023-01-27T10:02:40.395566 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [37b40523-d8e0-4db2-8450-e1d20947f5c9], Method [GET], Path [/dashboard/widget_chart_data/6]
Features checked: dashboard_view
[----] I, [2023-01-27T10:02:40.441604 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [7615b645-1426-49c2-bede-cc6ab2ff0043], Method [GET], Path
[/dashboard/widget_chart_data/74] Action: widget_chart_data
[----] I, [2023-01-27T10:02:40.442043 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [7615b645-1426-49c2-bede-cc6ab2ff0043], Method [GET], Path
[/dashboard/widget_chart_data/74] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:40.665204 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [dd1cbac8-db78-4026-9582-3c8eb227d01f], Method [GET], Path
[/dashboard/widget_report_data/16] Action: widget_report_data
[----] I, [2023-01-27T10:02:40.666085 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [dd1cbac8-db78-4026-9582-3c8eb227d01f], Method [GET], Path
[/dashboard/widget_report_data/16] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:41.652177 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [8cecb5c5-13da-4404-b6c5-1a0bec224cfb], Method [GET], Path
[/dashboard/widget_report_data/18] Action: widget_report_data
[----] I, [2023-01-27T10:02:41.652878 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [8cecb5c5-13da-4404-b6c5-1a0bec224cfb], Method [GET], Path
[/dashboard/widget_report_data/18] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:21.677049 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [16bfe549-1d36-49bc-9ce5-6ec260c91021], Method [GET], Path [/ems_configuration/show_list]
Action: show_list
[----] I, [2023-01-27T10:07:25.466777 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [admin], Role [EvmRole-
super_administrator], Request [44f4f937-d391-4891-846e-95cf52939b47], Method [POST], Path
[/ems_configuration/report_data] Action: report_data
[----] I, [2023-01-27T10:07:25.888963 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [admin], from: [User.logoff],
User admin has logged off
[----] I, [2023-01-27T10:07:31.609288 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [joe], from:
[Base.audit_success], User joe successfully validated by EVM
[----] I, [2023-01-27T10:07:31.612001 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [joe], from:
[Base.audit_success], Authentication successful for user joe
[----] I, [2023-01-27T10:07:31.765188 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [39a967af-c46c-47ba-9e5b-1b22489593fc], Method [GET], Path [/dashboard/show] Action: show
[----] I, [2023-01-27T10:07:31.765429 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [39a967af-c46c-47ba-9e5b-1b22489593fc], Method [GET], Path [/dashboard/show] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:33.210210 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [59e3e495-6d25-4400-aff3-edbe48259f4a], Method [GET], Path [/dashboard/widget_report_data/19] Action:
widget_report_data
[----] I, [2023-01-27T10:07:33.210718 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [59e3e495-6d25-4400-aff3-edbe48259f4a], Method [GET], Path [/dashboard/widget_report_data/19] Features checked:
```

```
dashboard_view
[----] I, [2023-01-27T10:07:33.272283 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [c25464a9-76f0-495e-a063-af52e727fddf], Method [GET], Path [/dashboard/widget_report_data/17] Action:
widget_report_data
[----] I, [2023-01-27T10:07:33.272675 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [c25464a9-76f0-495e-a063-af52e727fddf], Method [GET], Path [/dashboard/widget_report_data/17] Features checked:
dashboard_view
[----] I, [2023-01-27T10:07:33.368293 #17089:5a398]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [2009ddce-eaa3-46fb-bc92-ea5fd67b3160], Method [GET], Path [/dashboard/widget_chart_data/6] Action:
widget_chart_data
[----] I, [2023-01-27T10:07:33.368664 #17089:5a398]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [2009ddce-eaa3-46fb-bc92-ea5fd67b3160], Method [GET], Path [/dashboard/widget_chart_data/6] Features checked:
dashboard_view
[----] I, [2023-01-27T10:07:33.520539 #17089:13e0c]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [4f942385-cdae-448f-98a4-5ad34f2f7e2d], Method [GET], Path [/dashboard/widget_chart_data/5] Action:
widget_chart_data
[----] I, [2023-01-27T10:07:33.521192 #17089:13e0c]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [4f942385-cdae-448f-98a4-5ad34f2f7e2d], Method [GET], Path [/dashboard/widget_chart_data/5] Features checked:
dashboard_view
[----] I, [2023-01-27T10:07:33.617704 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [6d4f162d-015a-4dc1-ad3b-9359b240e48d], Method [GET], Path [/dashboard/widget_chart_data/1] Action:
widget_chart_data
[----] I, [2023-01-27T10:07:33.618119 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [6d4f162d-015a-4dc1-ad3b-9359b240e48d], Method [GET], Path [/dashboard/widget_chart_data/1] Features checked:
dashboard_view
[----] I, [2023-01-27T10:07:33.728720 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [b9f522e3-d590-4c27-8792-fabfb6b9ff11], Method [GET], Path [/dashboard/widget_report_data/20] Action:
widget_report_data
[----] I, [2023-01-27T10:07:33.729039 #17089:66d00]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [b9f522e3-d590-4c27-8792-fabfb6b9ff11], Method [GET], Path [/dashboard/widget_report_data/20] Features checked:
dashboard_view
[----] I, [2023-01-27T10:07:33.817352 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [f12ff6f5-3c67-485a-8751-5526e33d40ca], Method [GET], Path [/dashboard/widget_report_data/18] Action:
widget_report_data
[----] I, [2023-01-27T10:07:33.818016 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [f12ff6f5-3c67-485a-8751-5526e33d40ca], Method [GET], Path [/dashboard/widget_report_data/18] Features checked:
dashboard_view
[----] I, [2023-01-27T10:07:33.924635 #17089:13e0c]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [5ee0792a-4262-4184-96d0-49c4dd7a2c13], Method [GET], Path [/dashboard/widget_report_data/16] Action:
widget_report_data
[----] I, [2023-01-27T10:07:33.924891 #17089:13e0c]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [5ee0792a-4262-4184-96d0-49c4dd7a2c13], Method [GET], Path [/dashboard/widget_report_data/16] Features checked:
dashboard_view
[----] W, [2023-01-27T10:07:41.826427 #17089:5a5dc]  WARN -- audit: <AuditFailure> Username [joe], Role [EvmRole-user],
Request [fbb15be3-4137-4f12-b10c-6ba4ab451652], Method [GET], Path [/ops/explorer] Action: explorer
[----] W, [2023-01-27T10:07:41.964517 #17089:5a5dc]  WARN -- audit: <AuditFailure> Username [joe], Role [EvmRole-user],
Request [75430ae8-c62b-40f3-a0f9-fb8b1e43b611], Method [GET], Path [/dashboard/auth_error] Authentication Error Redirect
[----] I, [2023-01-27T10:07:55.728336 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [8a36b777-e757-41d1-b90a-507fa97ab767], Method [GET], Path [/report/explorer] Action: explorer
[----] I, [2023-01-27T10:07:57.033934 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [919c4db0-ed64-4075-bc72-e22e85db77f5], Method [POST], Path [/report/tree_select?
id=root&text=All%2520Saved%2520Reports] Action: tree_select
[----] I, [2023-01-27T10:07:57.035044 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [919c4db0-ed64-4075-bc72-e22e85db77f5], Method [POST], Path [/report/tree_select?
id=root&text=All%2520Saved%2520Reports] Features checked: miq_report_saved_reports
[----] I, [2023-01-27T10:07:57.476854 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username [joe], Role [EvmRole-user],
Request [833b8d0a-d8e5-4788-98d6-d7361a1c277a], Method [POST], Path [/report/report_data] Action: report_data
[----] I, [2023-01-27T10:07:59.567570 #17089:5a398]  INFO -- audit: <AuditSuccess> Username [joe], from: [User.logoff],
User joe has logged off
[----] W, [2023-01-27T10:10:34.973016 #17089:66cb0]  WARN -- audit: <AuditFailure> Username [], Role [], Request
[710978e7-20ec-4709-8678-5ea03718eb43], Method [GET], Path [/ops/explorer] Invalid Session
```

Enable Auditing for Infrastructure Automation Managed services

You can enable auditing for Infrastructure Automation Managed services.

1. Enable auditing during installation or post-installation by updating the **iaconfig** configuration. You can update the configuration by using the following command.

   ```
   oc -n cp4aiops edit iaconfig
   ```

   - Find the **spec** and add **manageservice** to its **spec** section. For example:

     ```
     spec:
       infraAutoComposableComponents:
        name: ibm-management-cam-install
        spec:
          manageservice:
          global:
           audit: true
           license:
             accept: true
     license:
       accept: true
     ```

2. To configure the IBM Cloud Pak foundational services for receiving audit logs, see Architecture for Audit logging version 4.6.

Security and Privacy by Design (SPbD)

Security and Privacy by Design (SPbD) at IBM® is an agile set of focused security and privacy practices, including threat models, privacy assessments, security testing, and vulnerability management.

This document is intended to help you in your preparations for GDPR readiness. It provides information about features of IBM Cloud Pak® for AIOps that you can configure, and aspects of the product's use, to consider for GDPR readiness. This information is not an exhaustive list, due to the many ways that customers can choose and configure features, and the product can be used in itself and with third-party applications and systems.

IBM developed a set of SPbD processes and tools that are used by all of its business units. For more information about the IBM Secure Engineering Framework (SEF) and SPbD, see the IBM Redbooks Security in Development - The IBM Secure Engineering Framework (available in PDF format).

IBM also provides information about the features of IBM® Netcool® Operations Insight® that you can configure, how to use the product securely, and what to consider to help your organization with GDPR readiness. For more information, see Platform considerations for GDPR readiness.

As part of being secure by default, all intra-product communication is encrypted with TLS on deployments of IBM Cloud Pak for AIOps on Linux® and Red Hat® OpenShift® Container Platform. In addition, both deployment options adhere to best practices such as running containers as non-root. Deployments of IBM Cloud Pak for AIOps on Red Hat OpenShift Container Platform also take advantage of additional security that is provided by Red Hat OpenShift Container Platform. For more information about container security, see the Red Hat® documentation: Security and compliance.

As an additional layer of security, all communications between Red Hat OpenShift Container Platform nodes can be transparently encrypted (in additional to the TLS encryption at the application layer). To ensure that encryption in motion is enabled between each service or node within your Red Hat OpenShift Container Platform cluster, you must enable IPsec on Red Hat OpenShift Container Platform. When you install Red Hat OpenShift Container Platform, specify an empty object for the **ipsecConfig** parameter to enable IPsec encryption, as in the following example:

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    type: OVNKubernetes
    ovnKubernetesConfig:
      ipsecConfig: {}
```

Note: IPsec enablement status cannot be changed after cluster installation.

For more information, see IPsec encryption configuration and Specifying advanced network configuration in the Red Hat OpenShift Container Platform documentation.

Platform considerations for GDPR readiness

Notice

This document is intended to help you in your preparations for GDPR readiness. It provides information about features of IBM Cloud Pak® for AIOps that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness. This information is not an exhaustive list, due to the many ways that clients can choose and configure features, and the large variety of ways that the product can be used in itself and with third-party applications and systems.

Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.

The products, services, and other capabilities described are not suitable for all client situations and may have restricted availability. IBM® does not provide legal, accounting, or auditing advice or represent or warrant that its services or products will ensure that clients are in compliance with any law or regulation.

Table of Contents

- GDPR
- Product Configuration for GDPR
- Data Life Cycle
- Data Collection
- Data Storage
- Data Access
- Data Processing
- Data Deletion
- Data Monitoring
- Capability for Restricting Use of Personal Data

GDPR

General Data Protection Regulation (GDPR) has been adopted by the European Union ("EU") and applies from May 25, 2018.

Why is GDPR important?

GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

Read more about GDPR

- ibm.com/GDPR website

Product Configuration - considerations for GDPR Readiness

The following sections describe aspects of data management within IBM Cloud Pak for AIOps and provide information on capabilities to help customers with GDPR requirements.

Data Life Cycle

IBM Cloud Pak for AIOps is an AIOps platform that deploys advanced, explainable AI across the ITOps toolchain so you can confidently assess, diagnose, and resolve incidents across mission-critical workloads.

IBM Cloud Pak for AIOps deals primarily with technical data, some of which might be subject to GDPR. IBM Cloud Pak for AIOps also deals with information about users who manage the deployment. This data is described throughout this document for the awareness of customers responsible for meeting GDPR requirements. This data is persisted on local or remote file systems as configuration files or in databases. Applications that integrate with IBM Cloud Pak for AIOps might deal with other forms of personal data subject to GDPR. The mechanisms that are used to protect and manage data are also available to applications that integrate with IBM Cloud Pak for AIOps. Extra mechanisms might be required to manage and protect personal data that is collected by these applications.

To best understand IBM Cloud Pak for AIOps and its data flows, you must understand how Kubernetes and Docker work. These open source components are fundamental to IBM Cloud Pak for AIOps.

IBM Cloud Pak for AIOps includes a catalog of containerized software and services from IBM® in the default IBM Cloud Pak for AIOps repository list. To view a list of all the IBM Cloud Pak for AIOps charts, see IBM/charts. For considerations about GDPR for the products in the catalog, consult the documentation for those products. Some of the applications available in the catalog are open source software. It is the customer's responsibility to determine and implement any appropriate GDPR controls for open source software. Information on these packages is included in the catalog entry.

What types of data flow through IBM Cloud Pak for AIOps platform

IBM Cloud Pak for AIOps deals with several categories of technical data that might be considered as personal data. Categories include administrator user IDs and passwords, service user IDs and passwords, IP addresses, and Kubernetes node names. IBM Cloud Pak for AIOps also deals with information about users who manage the deployment. Integrated applications might introduce other categories of personal data unknown to IBM Cloud Pak for AIOps.

Information on how this technical data is collected or created, stored, accessed, secured, logged, and deleted is described in later sections of this document.

Personal data used for online contact with IBM

Customers can submit online comments/feedback/requests to contact IBM about IBM Cloud Pak for AIOps subjects in various ways, primarily:

- The public IBM Cloud Pak for AIOps Slack Community
- Public comments area on pages of IBM Cloud Pak for AIOps product documentation in the IBM Documentation

Typically, only the customer name and email address are used to enable personal replies for the subject of the contact. The use of personal data conforms to the IBM Online Privacy Statement.

Data Collection

IBM Cloud Pak for AIOps does not collect any special categories of personal data. It does create and manage technical data, such as an administrator user ID and password, service user IDs and passwords, IP addresses, and Kubernetes node names, which might be considered personal data. IBM Cloud Pak for AIOps also deals with information about users who manage the offering. All such information is only accessible by the administrator.

Applications that run on IBM Cloud Pak for AIOps might collect personal data.

To assess the use of IBM Cloud Pak for AIOps running containerized applications and your need to meet the requirements of GDPR, you must consider the types of personal data that the application collects and aspects of how that data is managed, such as:

- How is the data protected as it flows to and from the application? Is the data encrypted in transit?
- How is the data stored by the application? Is the data encrypted at rest?
- How are credentials, which are used to access the application, collected and stored?
- How are credentials, which are used by the application to access data sources, collected and stored?
- How is data collected by the application removed as needed?

This list is not a definitive list of the types of data collected by IBM Cloud Pak for AIOps. It is provided as an example for consideration. If you have any questions about the types of data, contact IBM.

Types of personal data

- Basic Personal Information (such as name, address, phone number, email)

- Technically Identifiable Personal Information (such as device IDs, usage-based identifiers, static IP addresses - when linked to an individual).

Special categories of personal data

- IBM Cloud Pak for AIOps was not designed to process any special categories of personal data.

Data storage

IBM Cloud Pak for AIOps persists technical data in stateful stores on local or remote file systems as configuration files or in databases. Consideration must be given to securing all data at rest. IBM Cloud Pak for AIOps supports encryption of data at rest in stateful stores. For more information, see Security considerations

Data access

IBM Cloud Pak for AIOps provides a number of groups and roles for controlling data access. The groups and roles enable differentiation between normal users and those with extra privileges. For more information, see Roles and permissions.

Data Processing

In general, data that is used for authentication must be in a directory service or LDAP. Databases are provisioned during installation. Make sure to maintain them throughout product lifecycles.

Basic recommendations

- Regularly back up data, according to your business needs and to the risk level.
- Encrypt data backups.
- When data is no longer used, delete the databases or archive them for future use.
- As a data controller, provide means to satisfy data access requests for personal information or other compliance requests.

Further considerations

- Make sure that control of access to databases is in place and effective.
- Use strong credentials.
- Protect the REST administration APIs with proper credentials.
- Use HTTPS or equivalent secure communication protocols for all integrations.
- Remove or change all default passwords.

Data Deletion

Article 17 of the GDPR states that data subjects have the right to request that their personal data is removed from the systems of controllers and processors, without undue delay. Implement appropriate controls and tools to satisfy this right.

IBM Cloud Pak for AIOps does not require any special method for data deletion.

Data that reflects personally identifiable information (PII) can be in all stages of the data processing pipeline. Data deletion must include all these stages. Administrators can use IBM Cloud Pak for AIOps features to remove user data.

Data monitoring

Regularly test, assess, and evaluate the effectiveness of your technical and organizational measures to comply with GDPR. These measures should include ongoing privacy assessments, threat modeling, centralized security logging, and monitoring, among others.

Capability for Restricting Use of Personal Data

Using the facilities summarized in this document, IBM Cloud Pak for AIOps enables a user to restrict usage of any technical data that is considered personal data.

Under GDPR, users have rights to access, modify, and restrict processing. Refer to other sections of this document to manage the following controls:

- Right to access
  - Administrators can use IBM Cloud Pak for AIOps features to provide individuals access to their data.
  - Administrators can use IBM Cloud Pak for AIOps features to provide individuals information about what data IBM Cloud Pak for AIOps platform holds about the individual.
- Right to modify
  - Administrators can use IBM Cloud Pak for AIOps features to allow an individual to modify or correct their data.
  - Administrators can use IBM Cloud Pak for AIOps features to correct an individual's data for them.
- Right to restrict processing
  - Administrators can use IBM Cloud Pak for AIOps features to stop processing an individual's data.

When configuring logs from their environments to be stored or processed in IBM Cloud Pak for AIOps, users are responsible for filtering out personal information or other sensitive information. For more information about filtering logs, see the following topics:

- Creating ELK integrations
- Creating Falcon LogScale integrations
- Creating Mezmo integrations

- Creating Splunk integrations

Data retention

IBM Cloud Pak for AIOps does retain data for processing, analysis, and historical purposes. Review the following default retention periods to learn the duration that data is retained.

Default data retention periods

Default retention periods for data

Data type	Default retention period
Logs	- Retention period: 3 days - Normalized logs are retained for 3 days.
Metrics	- Retention period: 15 days - Metric time series data is retained for 15 days.
Events	- Retention period: N/A - Events are not retained after processing.
Alerts	- Retention period: 90 days - Open alerts are kept indefinitely. - Closed alerts are archived and retained for 90 days.
Incidents	- Retention period: N/A - Open incidents are kept indefinitely. - Closed incidents are purged when closed.
Tickets	- Retention period: Indefinite - Closed tickets from a historical time frame are retrieved from the ticketing system based on a user defined date range. Each subsequent retrieval adds to the total data set. Retention controls are not provided.
Topology	- Retention period: 30 days - Valid resources are kept indefinitely. - Deleted or changed resources are deleted after 30 days. - Event correlation and associated analytics use the current topology.

Configuring the retention period for topology resources

You can configure the retention period for topology resources if required. For instance, you can increase the retention period to a maximum of 90 days.

The retention period for historical resource data is configured through the **`HISTORY_TTL`** time to live (TTL) configuration property. You can increase the retention period by increasing this property value. When a resource is deleted, the resource no longer appears in the UI. Historic representations of the resource, however, are retained and can be accessed in the UI history timeline until the history TTL limit is reached, after which the data is deleted.

Caution: Performance and scalability are affected by the number of resources that are managed, the amount of history present for each resource, and the ingestion rate. If you increase the retention period for historical resource data from the default of 30 days (up to a maximum of 90 days), your system performance (when rendering views) might deteriorate, if as a result of this increase resources have in excess of 25,000 historical entries.

To configure the **`HISTORY_TTL`** time to live (TTL) property, complete the following steps on your cluster:

1. Log in to your cluster.
2. Open the **`$ASM_HOME/.env`** file for editing in your preferred editor.
3. Edit the **`HISTORY_TTL`** setting to set your preferred retention period (in hours). You can change the default setting of **`720`** hours (30 days) to a maximum of **`2160`** hours (90 days).
4. Restart the topology service:

   ```
   $ASM_HOME/bin/asm_start.sh
   ```

Configuring the retention period for log resources

You can configure the retention period for log resources if required.

1. Log in to your OpenShift Container Platform cluster.
2. Go to **Deployments**.
3. In Deployments, search for **golden-signal** under the Name field.
4. Click **aimanager-aio-log-anomaly-golden-signals** resource.
5. Click **Environment**.
6. Edit the value for **LAGS_RETENTION_DAYS**. The default value is set to 3 days.

Getting started

Now that you want to get started with IBM Cloud Pak® for AIOps, it's time to learn about all of the capabilities afforded to you by IBM Cloud Pak, and how to take advantage of these capabilities.

Before installing IBM Cloud Pak for AIOps, take the time to understand your needs. IBM Cloud Pak for AIOps includes many features by default. If you have a clear understanding of what you're looking to do with IBM Cloud Pak for AIOps, your installation might be more straightforward.

Choosing the deployment platform

IBM Cloud Pak for AIOps can be installed on Red Hat® OpenShift® Container Platform or on Linux®.

Red Hat OpenShift is supported on most major cloud providers, and is a robust platform for large-scale Kubernetes deployments like IBM Cloud Pak for AIOps. It also provides a console interface for administering and managing your deployment. Linux is offered as an alternative platform to provide an easier getting started experience for users who are not familiar with Red Hat OpenShift.

Deployments on both platforms offer similar capabilities and scaling, but you must consider the following key differences when you are deciding which platform to install IBM Cloud Pak for AIOps on:

- **Hardware requirements**: IBM Cloud Pak for AIOps has the same hardware requirements on both platforms, but deployments on Linux use local storage and benefit from reduced resource requirements by not requiring a storage cluster or cloud storage.
- **Ease of installation**: The deployment of IBM Cloud Pak for AIOps on Red Hat OpenShift requires you to provision a cluster that is running Red Hat CoreOS, deploy Red Hat OpenShift, deploy and configure storage, and install IBM Cloud Pak for AIOps. You must have knowledge of the installation, configuration and administration of Red Hat OpenShift, and of the storage provider that you select. A deployment on Linux is simpler - it requires you to provision a cluster of virtual machines (VMs) and then use an IBM Cloud Pak for AIOps command line tool to configure local storage and install IBM Cloud Pak for AIOps.
- **Backup and restore**: The backup and restore process for deployments of IBM Cloud Pak for AIOps on Linux uses offline VM backups which requires outage time. The backup and restore process for deployments of IBM Cloud Pak for AIOps on Red Hat OpenShift is an online backup that uses IBM Storage Fusion (license restrictions apply), and does not require outage time.
- **Multi-zone high availability**: This is a technology preview feature for deployments of IBM Cloud Pak for AIOps on Red Hat OpenShift. This feature is not available for deployments of IBM Cloud Pak for AIOps on Linux (though deployments are still highly available).
- **FIPs mode**: This feature is not supported for deployments of IBM Cloud Pak for AIOps on Linux.
- **Secure Tunnel**: This feature is not supported for deployments of IBM Cloud Pak for AIOps on Linux.
- **Cognos integration**: This feature is not supported for deployments of IBM Cloud Pak for AIOps on Linux.

Workflows

To help first-time users get started, guided tours are included within the UI console. These tours introduce users to the various UI options and components, and demonstrate how to complete key tasks across the platform. Users can access the tours at any time by clicking the Tours icon in the console toolbar, and selecting their preferred tour.

For more information about working with the IBM Cloud Pak, see the following topics:

- Workflows
- Guided tours
- Roles and archetypes

Archetypes and user roles

For IBM Cloud Pak® for AIOps, different persona archetypes are defined to help you understand the different available and supported tasks so that you can get your environment up and running or begin your day to day operations quickly and efficiently.

Archetypes are a set of basic characteristics that a user role can fulfill. You can think of an archetype as a templated job description for a particular user role. For example, in IBM Cloud Pak for AIOps, the Site reliability expert is responsible for many tasks assessing and remediating problems. The Site reliability expert falls under the Infrastructure engineer archetype, that is, someone who is focused on keeping the entire IT infrastructure of an organization up and running 24 x 7.

The following defined archetypes and user roles are referenced within the product documentation as the user for completing different use cases and tasks:

- Application developer
- Infrastructure engineer

Archetype: Application developer

Application developers in IBM Cloud Pak for AIOps are responsible for building things outside of the scope of enterprise IT security. Application developers want to solve problems early so that they can focus on new feature development and not dwell on debugging IT issues. Application developers work hand in hand with the Security operations manager and Infrastructure engineer to identify and fix issues. The following user roles are often associated with the Application developer archetype:

- Software engineer
- Software developer
- DevOps engineer
- Full stack developer

The following table lists the high-level key tasks that Application developers can complete with IBM Cloud Pak for AIOps:

Table 1. Archetype: Application developer tasks and references

Tasks	References
Monitor outgoing integrations for issues	ChatOps insight management
Debug issues that are associated with applications and related services	Managing incidents

Archetype: Infrastructure engineer

Infrastructure engineers play a vital role in IBM Cloud Pak for AIOps as they ensure that systems are always up and running. They serve as the go-between from the strategy and high level thinking of the Security operations manager to the implementation by the Application developers. Infrastructure engineers are always aware of their platforms, ensuring that problems are detected and remediated as quickly as possible. If all goes well, the Infrastructure engineer is responsible for a platform that never goes down. The following user roles are often associated with the Infrastructure engineer archetype:

- Site reliability engineer
- Systems engineer
- Operations engineer
- Platform engineer

The following table lists the high-level key tasks that Infrastructure engineers can complete with IBM Cloud Pak for AIOps:

Table 2. Archetypes: Infrastructure engineer tasks and references

Tasks	References
Install IBM Cloud Pak for AIOps	Deploying IBM Cloud Pak for AIOps on OpenShift or Deploying IBM Cloud Pak for AIOps on Linux
Administer IBM Cloud Pak for AIOps	Cloud Pak administration
Setting up integrations	Configuring Integrations
Training AI models	Configuring AI training
Managing topologies	Application management
Monitor outgoing integrations for issues	ChatOps insight management
Debug issues that are associated with applications and related services	Managing incidents

IBM Cloud Pak for AIOps workflows

You decided to install IBM Cloud Pak® for AIOps, now it's time to explore the key workflows of your users and their interaction with IBM Cloud Pak for AIOps.

Understanding how you can accomplish your tasks, who's responsible, and how each workflow fits together gives you a holistic picture of your entire AIOps-infused IT infrastructure management experience. You can bring IBM Cloud Pak for AIOps to life by deploying more operators, customizing your configurations, and optimizing IBM Cloud Pak for AIOps performance. For more information about how to get the most out of your team, see Archetypes.

Installing IBM Cloud Pak for AIOps

Deployments on Red Hat® OpenShift® Container Platform

Platform engineers prepare all required hardware and software, and then install the Red Hat® OpenShift® Container Platform. They prepare OpenShift clusters and install IBM Cloud Pak for AIOps into those clusters.

For more information about installing IBM Cloud Pak for AIOps, see the following sections:

- Planning installation, configuration, and overall cluster requirements
- Preparing for and installing IBM Cloud Pak for AIOps

Deployments on Linux®

Platform engineers prepare all required hardware and software. They prepare the Linux cluster and install IBM Cloud Pak for AIOps onto the cluster.

For more information about installing IBM Cloud Pak for AIOps, see the following sections:

- Planning installation, configuration, and overall cluster requirements
- Preparing for and installing IBM Cloud Pak for AIOps

Guided tours

To help first-time users get started with some workflows, guided tours are included within the UI console across the different dashboards. These tours introduce users to the various UI options and components, and demonstrate how to complete key tasks across the platform. Users can access the tours at any time by clicking the Tours icon in the console toolbar, and selecting their preferred tour.

For more information about the available guided tours, see Guided tours.

Setting up integrations

You can create integrations for logs, to establish a baseline of normal behavior and identify anomalies. Other integrations provide event, applications, and infrastructure data to IBM Cloud Pak for AIOps.

Systems engineers create integrations to ticketing and to ChatOps tools. The integrations to ticketing tools have a dual purpose: they enable ticket data to be collected for change request and similar ticket training. The same integration also enables synchronization between IBM Cloud Pak for AIOps incidents and corresponding ServiceNow tickets. The integrations to Microsoft Teams and Slack ChatOps tools enable incident notifications to be pushed to these ChatOps tools. These notifications are for the attention of Site Reliability Engineers who are tasked with resolving those incidents.

IBM Cloud Pak for AIOps provides many connectors, probes, and observers by default for setting up integrations with common IBM and third-party systems and services. In addition, if a default integration is not available for connecting to a system or service that you want to use with IBM Cloud Pak for AIOps, you can set up a custom integration to connect to that system or service.

For more information about setting up integrations, see the following sections:

- Defining outgoing integrations
- Defining incoming integrations

Loading data and training AI

As a prerequisite to training your AI to provide insights on your monitored environment, systems engineers load event, log, metric, ticket, and other data from various data sources. They then run different types of AI training to generate AI models such as log anomaly models, which can easily identify and report on anomalies in your log data. Insights that are generated by these AI models are used to enrich incidents with related alert, similar ticket, and event grouping information.

For more information about loading data and training AI, see the following sections:

- Planning data loading and training
- Defining incoming integrations
- Configuring AI training

Building applications

Systems engineers are responsible for building the applications that are monitored by Site Reliability Engineers. Systems engineers build applications from the ground up, by first defining observer integrations to import resource topology data, and then defining templates to automatically generate resource groups from the incoming observer streams. When they have an incoming dynamic stream of resource groups, they build business applications that are made up of multiple resource groups.

For more information about building applications, see the following sections:

- Defining observer jobs
- Defining templates with template builder
- Creating and editing applications
- Viewing applications

Creating automations

System administrators set up policies, runbooks, and actions to detect and remediate incoming alerts. Policies are rules that contain a condition and a set of outcomes that can be manual or automated. Runbooks help to solve common operational problems by automating procedures that do not require human interaction. Procedure-based documentation can be transferred into manual runbooks. Actions in runbooks are the collection of several manual steps into a single automated entity. Actions can be added to create semi-automated or fully automated runbooks.

For more information about creating automations, see the following sections:

- Creating policies
- Creating a runbook
- Creating an action

Resolving incidents

Site reliability engineers are responsible for investigating and resolving incidents and incidents that are associated with applications and infrastructure. They use ChatOps tools such as Microsoft Teams and Slack to monitor incident channels, and investigate incidents by using application, topology, ticketing, alert, and similar ticket information provided in the incident notification. They act to resolve the incidents, where necessary involving Applications developers.

For more information about resolving incidents, see the following sections:

- ChatOps insight management
- Application incidents

Administering the cluster

Platform engineers administer the cluster on which IBM Cloud Pak for AIOps is installed on an ongoing basis. They ensure that all nodes, pods, and other elements within the cluster are performing as expected. They also support the systems engineer with complex cluster administration tasks such as multitenancy management, project (namespace) management, and authorization and authentication of users within the cluster.

Administering IBM Cloud Pak for AIOps

Systems engineers, or in some organizations, Operations engineers, are responsible for authenticating and managing users. They can also customize the UI experience by setting console branding and customizing what content users see on the home page. For more complex cluster-based tasks such as multitenancy management, project (namespace) management, and authorization and authentication of users within the cluster, they collaborate with the Platform engineer.

For more information about administering IBM Cloud Pak for AIOps, see the following sections:

- Managing user access control
- Customizing the platform UI

Deploying IBM Cloud Pak for AIOps on Linux

You can install a production deployment of IBM Cloud Pak® for AIOps directly on a Linux® cluster, without the need to install Red Hat® OpenShift® Container Platform. Online and offline deployments are supported.

If you need help deciding whether to install IBM Cloud Pak for AIOps on Red Hat OpenShift or Linux, see Choosing the deployment platform.

Read about planning an installation of IBM Cloud Pak for AIOps on Linux, and then select one of the installation options:

- Planning - learn about the system requirements for installing IBM Cloud Pak for AIOps on a Linux cluster.
- Online installation - follow these instructions if your cluster is connected to the internet.
- Offline installation - follow these instructions if your cluster is not connected to the internet.

Learn about upgrading, administering, and uninstalling IBM Cloud Pak for AIOps on Linux:

- Online upgrade
- Offline upgrade
- Administering
- Uninstalling

Planning an installation of IBM Cloud Pak for AIOps on Linux

Learn about the system requirements for an installation of IBM Cloud Pak for AIOps on Linux.

Before you begin

- There is not a small starter size option for the installation of IBM Cloud Pak® for AIOps on Linux®. You can only deploy a production size deployment.
- vCPU is defined as when one x86 CPU splits each of its physical cores into virtual cores (vCPU). It is assumed that one x86 CPU's physical core can be split into two logical vCPUs.
- If Linux is installed on VMware virtual machines, set the value of the `sched.cpu.latencySensitivity` parameter to high.
- You can deploy a base deployment of IBM Cloud Pak for AIOps that does not have log anomaly or ticket analysis capabilities, or an extended deployment of IBM Cloud Pak for AIOps with these capabilities. Learn more

Review the information in the following sections:

- Supported platforms
- Hardware requirements
- Processing abilities
- Custom sizing
- Storage requirements
- Configuring local volumes
- Network requirements
- High availability considerations
- Load balancing
- DNS requirements
- Security considerations
- Licensing

Supported platforms

IBM Cloud Pak for AIOps can be installed on the following platform versions:

- Red Hat® Enterprise Linux® 8.10 only
- Red Hat® Enterprise Linux® 9.4 only
- Ubuntu 24.04 LTS

The hardware architecture for installing IBM Cloud Pak for AIOps must be x86_64 (amd64).

If you want to install IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform or on a cloud platform, see Deploying IBM Cloud Pak for AIOps on OpenShift.

Hardware requirements

Multiple Linux nodes are needed, and IBM Cloud Pak for AIOps is installed on this cluster of nodes. The cluster must be reserved for the sole use of IBM Cloud Pak for AIOps.

Three of these nodes must be control plane nodes. The control plane nodes coordinate the running of IBM Cloud Pak for AIOps across the other nodes, and are the entry point into the product. Worker or agent nodes provide more compute resources to run IBM Cloud Pak for AIOps services. All the nodes can be assigned as agent nodes, but three of the nodes must also be assigned as control plane nodes.

The size of each of your nodes can vary, but each worker node must meet the minimum requirements in Table 1 to accommodate the placement of larger IBM Cloud Pak for AIOps pods, and the combined resources of your nodes must comply with the resource totals in Table 2.

You must ensure that the clocks on your Linux cluster are synchronized. Each node in the cluster must have access to an NTP server to synchronize their clocks. Discrepancies between the clocks on the nodes can cause IBM Cloud Pak for AIOps to experience operational issues.

Warning: Insufficient hardware will result in product instability and loss of function. Verify that your hardware is sufficiently sized for your expected workloads. For more information, see Processing abilities. In addition to the default production size deployment, you can choose to deploy a custom-sized deployment of IBM Cloud Pak for AIOps. For more information, see Custom sizing.

Minimum node requirements

Each of the worker and control plane nodes in your cluster must meet the minimum size requirements in the following table:

Table 1. Minimum node requirements

Resource	Requirement
vCPU per node	16
Memory per node (GB)	20
Disk per node (GB)	120

Overall cluster requirements

Your cluster must meet the overall size requirements in the following table.

Table 2. Overall cluster requirements

Resource	Base deployment requirement	Extended deployment requirement
Node count	9	9
Total vCPU	148	174
Total memory (GB)	358	416

Table 2 shows the hardware requirements for a deployment of IBM Cloud Pak for AIOps on a Linux cluster, for a base deployment and an extended deployment. For more information about the differences between a base deployment and an extended deployment, see Incremental adoption.

Important:

- Three of these nodes must be control plane nodes.
- An extra 1 vCPU and 3 GB memory is required for each integration that you configure. For example, if you configure two Netcool integrations then you will require an additional 2 vCPUs and 6 GB memory.
- You will also require an additional server for a load balancer. For more information, see Load balancing.

Integrations

When you are configuring an integration in IBM Cloud Pak for AIOps, it is important to consider the performance and footprint. If you have large amounts of data on the selected target system, you can expect relatively higher resource usage from collecting data from that system when compared to a system with less data. Additionally, the sizing of the IBM Cloud Pak for AIOps installation affects the number of resources that are available for running integrations. Review the following considerations for each integration category and the **integration installation** section before you configure a new integration.

Integration installation

IBM Cloud Pak for AIOps integrations for metrics, events, and logs are configured in the Integrations UI page, which can be used to create, edit, delete, and track the status of integrations. Integrations run as pods and have minimum and maximum resource allocations in terms of CPU, memory and storage. Some integrations offer flexibility in terms of how to install the integration in the UI, but some integrations only offer one option.

You can use the following two options to install integrations:

- **Local**: Install the integration in the same cluster and namespace where IBM Cloud Pak for AIOps is installed. The integration's status is displayed in the Integrations UI page and is automatically managed by IBM Cloud Pak for AIOps.
- **Remote**: Install the integration anywhere you choose, for example, a different network region, on SaaS, or remote on-premises (VM, cluster or container). After adding the integration, you can use the script to run the integration pod using podman.

Regardless of whether an integration is a local or remote installation, the data collected by IBM Cloud Pak for AIOps is stored for some time in Kafka.

Hardware requirements - Integrations

Before you create an integration for both local and remote installation, make sure that IBM Cloud Pak for AIOps or remote environment contains the hardware resources that are required to run the integrations.

The minimum resource requirements for the integrations are listed in the following table:

Table 3. Minimum requirements for integrations

Integration name	Memory limit (Mb)	CPU limit	Ephemeral storage limit (Mb)
AppDynamics New Relic Splunk	2500	1	500

Integration name	Memory limit (Mb)	CPU limit	Ephemeral storage limit (Mb)
AWS CloudWatch, Dynatrace (metrics-only), Infrastructure Management, Generic Webhook, Zabbix	4096	1	500
Dynatrace (metrics, events, topology)	10000	4	8000
DB2, GitHub, ServiceNow	800	1	1000
Email Notifications, IMPACT, Jira	512	1	1000
Instana	4096	4	1000
Netcool ObjectServer	4096	2	2000
Custom, Elk, Falcon LogScale, Mezmo	1536	1	1000

Notes:

- Dynatrace (metrics, events, topology) supports up to 5 integrations.
- The hardware sizing in the preceding table does not include resource slots for logs integrations or observer jobs. For more information about resource slots for logs integrations, see Performance considerations for logs data collection. For more information about defining observer jobs, see Observer jobs.

For performance considerations for integrations, see the following sections:

- Performance considerations for metrics data collection
- Performance considerations for logs data collection
- Performance considerations for events data collection
- Performance considerations for tickets data collection
- Performance considerations for observers

Processing abilities

Expand the following sections to find out about the processing abilities of a production deployment of IBM Cloud Pak for AIOps. Higher rates are supported by custom sizes. For more information about customizing the size of your IBM Cloud Pak for AIOps deployment according to your processing and footprint requirements, see Custom sizing.

▶ Supported resource number and throughput rates for production deployments

▶ Event, alert, and incident rates

Custom sizing

The default production deployment size enables the full capabilities of IBM Cloud Pak for AIOps to be used with the workload volumes that are stated in the Processing abilities section. If different workload volumes are required or resource constraints are an issue, then specific capabilities such as **`Metric Anomaly Detection`**, **`Log Anomaly Detection`**, and **`Runbook Automation`** can be sized accordingly. IBM Sales representatives and Business Partners have access to a custom sizing tool that can assess your runtime requirements and provide a custom profile that scales IBM Cloud Pak for AIOps components. The custom profile is applied when you install IBM Cloud Pak for AIOps. This custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. If you require custom sizing, contact IBM Sales representatives and Business Partners with details of your intended workloads.

The following table shows the processing abilities of some custom-sized deployments of IBM Cloud Pak for AIOps.

Example 1 (Minimum scale) represents a minimally sized deployment of IBM Cloud Pak for AIOps for the evaluation of event management. It demonstrates event analytics, noise reduction, and the automation of issue resolution on a small topology. Metric and log anomaly detection and change risk assessment are de-emphasized. Example 1 requires 49.36 vCPU and 155GB of memory to run on a 9 node Linux cluster, this includes an overhead of 12GB of memory and 0.36 of a vCPU. If there are more than 9 nodes in the cluster, then extra resources are required as follows: 0.06 of a core and 3GB for each control plane node, and 0.03 of a core and 0.5GB for each worker node.

Example 2 (Event management focused) represents a production deployment of IBM Cloud Pak for AIOps which is focused on event management capabilities. It supports event analytics, noise reduction, and the automation of issue resolution on a large topology. Metric and log anomaly detection and change risk assessment are de-emphasized. Example 2 requires 176.36 vCPU and 378 GB memory to run on a 9 node Linux cluster, this includes an overhead of 12GB of memory and 0.36 of a vCPU. If there are more than 9 nodes in the cluster, then extra resources are required as follows: 0.06 of a core and 3GB for each control plane node, and 0.03 of a core and 0.5GB for each worker node.

Maximum scale shows the upper limits that IBM Cloud Pak for AIOps can be scaled to, across all of its capabilities.

Table 5. Processing abilities of example custom-sized deployments

Component	Resource	Example 1	Example 2	Maximum scale
Change risk	Incidents and change request records per second	0	0	N/A
Metric anomaly detection	Maximum throughput - (KPIs) for all metric integration	0	0	5,000,000
Log anomaly detection	Maximum throughput (log messages per second) for all log integrations	0	0	25,000
Events (through Netcool integration)	Steady state event rate throughput per second Burst rate event throughput per second	10	600	700 1000
Automation runbooks	Fully automated runbooks run per second	1	2	4
Topology management	Maximum number of topology resources	5,000	5,000,000	15,000,000
UI users	Active users supported	5	40	40

Storage requirements

IBM Cloud Pak for AIOps deployments on Linux require 3000 GB of persistent local storage across all of the nodes in your Linux cluster. Each node must have a minimum of 250GB. To meet the local storage requirements, you must configure distributed storage across the nodes in your Linux cluster, and dedicate specific disks and logical volumes for application and platform storage. Follow the instructions in Configuring local volumes.

Important: The file system used by MinIO must be **`XFS`**, not **`ext4`**. The **`ext4`** file system has a limit on the number of inodes that can be created for each file system. If inode usage reaches 100%, the file system becomes read-only even if enough PVC space is available and MinIO is prevented from creating new files or directories. Refer to your storage provider's documentation for information about setting **`XFS`** as the file system.

Additional requirements for offline deployments

If you are installing in an air-gapped environment (offline), you must also ensure that you have adequate space to download the IBM Cloud Pak for AIOps images to the target registry in your offline environment. The IBM Cloud Pak for AIOps images total 143 GB.

IOPS requirements

The following table identifies the input/output operations per second (IOPS) per node that must be supported. Ensure that your hardware can support the expected IOPS per node. A minimum configuration of three nodes for the storage cluster is needed. Each node of the storage solution requires a minimum of one disk (SSD or high-performance storage array). The results for your storage can vary depending on your exact usage, data sets, hardware, storage solution, and more.

Table 6. IOPS requirements

Deployment size	Minimum	Recommended
Production deployment	10,000	20,000

Network requirements

The control plane nodes require the following access:

Table 7. Control plane node access

Port number	Direction	Protocol	Description
80	Inbound from load balancer	TCP	Application HTTP port
443	Inbound from load balancer	TCP	Application HTTPS port
6443	Inbound from cluster worker nodes Between control plane nodes	HTTPS	Control plane server API
5001	Between cluster nodes	TCP	Distributed registry
8472	Betwen cluster nodes	UDP	Virtual network

Worker nodes require the following access:

Table 8. Worker node access

Port number	Direction	Protocol	Description
6443	Outgoing to control plane nodes		IBM Cloud Pak for AIOps components
5001	Between cluster nodes	TCP	Distributed registry
8472	Between cluster nodes	UDP	Virtual network

Firewall rules

Some Linux distributions require extra firewall rules to avoid potential conflicts or restrictions.

1. Check whether the firewall is enabled.

   ```
   systemctl status firewalld
   ```

   If the firewall is enabled, then the command output provides the status of the firewall, including any active rules. If the firewall is disabled, then the command outputs the message *No such file or directory.*

2. If the firewall is enabled, run the following commands:

   ```
   firewall-cmd --permanent --add-port=6443/tcp #apiserver
   firewall-cmd --permanent --zone=trusted --add-source=10.42.0.0/16 #pods
   firewall-cmd --permanent --zone=trusted --add-source=10.43.0.0/16 #services
   firewall-cmd --reload
   ```

3. If you are permitting the collection of usage data, then ensure that outbound traffic to https://api.segment.io is allowed. For more information, see Updating usage data collection preferences.

nm-cloud-setup service

The **`nm-cloud-setup`** service must not be enabled. Run the following steps on each of your nodes.

1. Check if **`nm-cloud-setup.service`** is enabled.

   ```
   systemctl is-enabled nm-cloud-setup.service
   ```

   If the service is enabled, the command returns **`enabled`**.

2. If the service is enabled, then disable it and reboot the node.

```
systemctl disable nm-cloud-setup.service nm-cloud-setup.timer
reboot
```

High availability considerations

IBM Cloud Pak for AIOps uses a minimum of three control plane nodes and multiple worker nodes to improve resiliency if a node failure occurs.

A procedure is also provided to delete a failed and inaccessible node, for more information, see Recovering from node failure.

Load balancing

IBM Cloud Pak for AIOps requires a load balancer to distribute incoming traffic among the three control plane nodes. The load balancer is a mandatory component, and if you do not already have a load balancer then you must configure one of your choice.

The load balancer requires the following access:

Table 9. Load balancer access

Port number	Description
443	Inbound IBM Cloud Pak for AIOps requests
6443	Inter-cluster communications: inbound from worker nodes, outbound to control plane nodes
(optional) 80	For inbound IBM Cloud Pak for AIOps requests, redirects to secure port 443. If you disable this port, users must type `https://` as part of the URL when they first navigate to the IBM Cloud Pak for AIOps console.

The load balancer is the entry point for accessing IBM Cloud Pak for AIOps. It is important to try to ensure the load balancer's high availability, as any downtime or outage will render IBM Cloud Pak for AIOps inaccessible. Consider adding the load balancer to the startup configuration (such as `init.d`) on the server where it is installed. This configuration will enable the load balancer to launch automatically during the boot process, minimizing downtime if an outage occurs.

For more information about load balancers, see the IBM topic What is load balancing?

Expand the following section to view an example configuration for a HAProxy load balancer. This example is provided as a reference only, consider your own internal policies and practices when you create the configuration for your chosen load balancer.

▶ Example load balancer configuration

DNS requirements

IBM Cloud Pak for AIOps requires a Domain Name System (DNS) server in the environment, with resolution for the following two hosts:

- `<load_balancer_ip> aiops-cpd.<your fully-qualified server domain name>`
- `<load_balancer_ip> cp-console-aiops.<your fully-qualified server domain name>`

Where `<load_balancer_ip>` is the IP address of your load balancer.

Important: The DNS server must provide name resolution for all of the nodes in the cluster.

Security considerations

Ensure that you keep up to date on any security-related news or vulnerabilities with IBM Cloud Pak for AIOps by subscribing to security bulletins.

The `aiopsctl` tool must run with root privileges.

For more information about the permissions that are required for Cloud Pak for AIOps, see Permissions.

You can use a custom certificate for IBM Cloud Pak for AIOps instead of the default cluster certificate. For more information, see Using a custom certificate.

Licensing

License usage tracking is required. The `aiopsctl` tool deploys the IBM Cloud Pak foundational services License Service on your Linux cluster. This background service collects and stores the license usage information for tracking license consumption and for audit purposes. For more information, see Licensing.

Incremental adoption (IBM Cloud Pak for AIOps on Linux)

Learn about incremental adoption and the deployment types that are available for IBM Cloud Pak® for AIOps.

Overview

Incremental adoption offers the flexibility of two deployment types that provide a different combination of footprint and capabilities.

- **Base deployment**: has a smaller footprint than the extended deployment, with vCPU and memory savings of around 15% for a production deployment. The log anomaly detection and ticket analysis (similar incidents and change risk) capabilities are not enabled.
- **Extended deployment**: has a larger footprint and has the full capabilities of IBM Cloud Pak for AIOps enabled.

The IBM Cloud Pak for AIOps license entitles you to use either of the deployment types.

Capabilities and hardware requirements

Log anomaly detection collects and analyzes log data to identify deviations from expected patterns of behavior. For more information, see Log anomaly detection.

Ticket analysis collects ticket data and compares it with similar tickets previously associated with incidents to identify possible remediating changes, and the risk of implementing those changes. For more information, see Similar ticket comparison and Change risk.

For more information about the footprint of each deployment type, see Hardware requirements.

Selecting your deployment type (install and upgrade)

You can choose to install a base deployment or an extended deployment of IBM Cloud Pak for AIOps. The deployment type is configured by an environment variable at installation time, and defaults to a base deployment. For more information about setting the deployment type at installation time, see the topic for your installation method:

- Online installation
- Offline installation

Deployments that upgrade to IBM Cloud Pak for AIOps 4.8.0 have their deployment type set during upgrade. For more information about setting the deployment type during upgrade, see the topic for your installation method:

- Online upgrade
- Offline upgrade

Changing your deployment type (postinstall)

Subject to further hardware requirements, you can update the deployment type of a new or upgraded deployment at any time after installation from a base deployment to an extended deployment. For more information, see Updating the deployment type.

Configuring local volumes for an installation of IBM Cloud Pak for AIOps on Linux

Learn about configuring logical storage volumes for an installation of IBM Cloud Pak for AIOps on Linux.

Overview

Each control plane node in your Linux® cluster must have two separate disks in addition to the boot disk. One disk is dedicated to persistent storage for IBM Cloud Pak® for AIOps, while the other is reserved for platform storage. Similarly, each worker node in the cluster must have a dedicated disk for IBM Cloud Pak for AIOps's persistent storage. This protects and isolates application and platform data from the rest of the node, and provides capacity limiting that protects the root volume from being filled by application data.

IBM Cloud Pak for AIOps deployments on Linux require 3000 GB of persistent local storage across all of the nodes in your Linux cluster. Each node must have a minimum of 250GB.

The following procedure is suggested to create separate logical volumes for IBM Cloud Pak for AIOps's application storage, and for platform storage. The procedure is for logical volume management, and some steps may be different for your chosen storage management solution.

Warning: Although it is possible to configure a separate partition and volume on an existing disk, this is not recommended. For more information, see Multiple partitions on a disk in the Red Hat® Enterprise Linux® documentation.

Procedure

1. Configure logical volumes on the control plane nodes.

   Run the following steps on each of your control plane nodes.

   1. Identify two blank disks that can be used for IBM Cloud Pak for AIOps's application storage and for platform storage.

      ```
      fdisk -l
      ```

      Example output where **/dev/vdb** and **/dev/vdc** can be used:

      ```
      $ fdisk -l

      Device     Boot   Start       End   Sectors  Size Id Type
      /dev/vda1  *       2048   2099199   2097152    1G 83 Linux
      /dev/vda2       2099200 524287999 522188800  249G 8e Linux LVM

      Disk /dev/vdb: 200 GiB, 214748364800 bytes, 419430400 sectors
      Units: sectors of 1 * 512 = 512 bytes
      Sector size (logical/physical): 512 bytes / 512 bytes
      I/O size (minimum/optimal): 512 bytes / 512 bytes


      Disk /dev/vdc: 200 GiB, 214748364800 bytes, 419430400 sectors
      Units: sectors of 1 * 512 = 512 bytes
      Sector size (logical/physical): 512 bytes / 512 bytes
      I/O size (minimum/optimal): 512 bytes / 512 bytes
      ```

2. Export environment variables.

```
export APP_DRIVE_NAMES=(
  <app_drive_name>
)
export PLATFORM_DRIVE_NAMES=(
  <plat_drive_name>
)

export APP_STORAGE_PATH="/var/lib/aiops/storage"
export PLATFORM_STORAGE_PATH="/var/lib/aiops/platform"
```

Where

- **<app_drive_name>** is a blank disk that you identified in step 1. For example: "/dev/vdb"
- **<plat_drive_name>** is a different blank disk that you identified in step 1. For example: "/dev/vdc"

Note: You can set different values for $APP_STORAGE_PATH and $PLATFORM_STORAGE_PATH if you want. If you do, then note the values that you use and ensure that you specify these values when you install IBM Cloud Pak for AIOps. You must use the same value for $APP_STORAGE_PATH for all control plane and worker nodes, and you must use the same value for $PLATFORM_STORAGE_PATH for all control plane nodes.

3. Run the following commands to configure the drives and volumes:

```
# Create physical volumes
pvcreate "${APP_DRIVE_NAMES[@]}"
pvcreate "${PLATFORM_DRIVE_NAMES[@]}"

# Create volume groups
vgcreate aiops "${APP_DRIVE_NAMES[@]}"
vgcreate aiopspl "${PLATFORM_DRIVE_NAMES[@]}"

# Create logical volumes on the volume groups
lvcreate aiops -n aiops -l 100%FREE
lvcreate aiopspl -n aiopspl -l 100%FREE

# Format the new logical volumes
export APP_DMPATH=$(lvs --noheadings -o dmpath aiops | sed -e 's/^[[:space:]]*//')
export PLATFORM_DMPATH=$(lvs --noheading -o dmpath aiopspl | sed -e 's/^[[:space:]]*//')
mkfs.xfs -L aiops "${APP_DMPATH}"
mkfs.xfs -L aiopspl "${PLATFORM_DMPATH}"

# Create filesystem directories
mkdir -p "${APP_STORAGE_PATH}"
mkdir -p "${PLATFORM_STORAGE_PATH}"

# Add the new filesystem directories to `/etc/fstab`
echo "${APP_DMPATH}  ${APP_STORAGE_PATH}     xfs    defaults        0 0" >> /etc/fstab
echo "${PLATFORM_DMPATH}  ${PLATFORM_STORAGE_PATH}     xfs    defaults        0 0" >> /etc/fstab

# Mount the filesystems into the directories
systemctl daemon-reload
mount "${APP_STORAGE_PATH}"
mount "${PLATFORM_STORAGE_PATH}"
```

4. Verify that the mount points are correctly configured.

Run the following command to verify that you have two new mount points called **aiops** and **aiopspl** for the storage locations that you specified, on the disks that you identified.

```
lsblk --output NAME,KNAME,TYPE,MAJ:MIN,FSTYPE,SIZE,RA,MOUNTPOINT,LABEL
```

Example output where **vdb** and **vdc** are now configured as the mount points for **aiops** and **aiopspl**:

```
$ lsblk --output NAME,KNAME,TYPE,MAJ:MIN,FSTYPE,SIZE,RA,MOUNTPOINT,LABEL
NAME               KNAME TYPE MAJ:MIN FSTYPE       SIZE   RA MOUNTPOINT             LABEL
vda                vda   disk 252:0                 250G 4096
├─vda1             vda1  part 252:1   xfs             1G 4096 /boot
└─vda2             vda2  part 252:2   LVM2_member   249G 4096
  ├─rhel-root      dm-0  lvm  253:0   xfs           233G 4096 /
  └─rhel-swap      dm-1  lvm  253:1   swap           16G 4096 [SWAP]
vdb                vdb   disk 252:16  LVM2_member   200G 4096
└─aiops-aiops      dm-2  lvm  253:2   xfs           200G 4096 /var/lib/aiops/storage aiops
vdc                vdc   disk 252:16  LVM2_member   200G 4096
└─aiopspl-aiopspl dm-2  lvm  253:2   xfs           200G 4096 /var/lib/aiops/platform aiopspl
```

2. Configure logical volumes on the worker nodes.

Run the following steps on each of your worker nodes.

1. Identify a blank disk that can be used for IBM Cloud Pak for AIOps's application storage.

```
fdisk -l
```

Example output where **/dev/vdb** can be used:

```
$ fdisk -l

Device     Boot   Start       End   Sectors  Size Id Type
/dev/vda1  *       2048   2099199   2097152    1G 83 Linux
/dev/vda2       2099200 524287999 522188800  249G 8e Linux LVM
```

```
Disk /dev/vdb: 200 GiB, 214748364800 bytes, 419430400 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

2. Export environment variables.

```
export APP_DRIVE_NAMES=(
  <app_drive_name>
)

export APP_STORAGE_PATH="/var/lib/aiops/storage"
```

Where **<app_drive_name>** is a blank disk that you identified in step 1. For example: "/dev/vdb"

Note: You can set a different value for $APP_STORAGE_PATH if you want. If you do, then note the value that you use and ensure that you specify this when you install IBM Cloud Pak for AIOps. You must use the same value for $APP_STORAGE_PATH for all control plane and worker nodes.

3. Run the following commands to configure the drives and volumes:

```
# Create physical volumes
pvcreate "${APP_DRIVE_NAMES[@]}"

# Create volume groups
vgcreate aiops "${APP_DRIVE_NAMES[@]}"

# Create a logical volumes
lvcreate aiops -n aiops -l 100%FREE

# Format the new logical volume
export APP_DMPATH=$(lvs --noheadings -o dmpath aiops | sed -e 's/^[[:space:]]*//')
mkfs.xfs -L aiops "${APP_DMPATH}"

# Create filesystem directory
mkdir -p "${APP_STORAGE_PATH}"

# Add the new filesystem directory to `/etc/fstab`
echo "${APP_DMPATH}  ${APP_STORAGE_PATH}     xfs    defaults        0 0" >> /etc/fstab

# Mount the filesystem into the directory
systemctl daemon-reload
mount "${APP_STORAGE_PATH}"
```

4. Verify that the mount point is correctly configured.

Run the following command to verify that you have a new mount point called **aiops** for the storage location that you specified on the disk that you identified.

```
lsblk --output NAME,KNAME,TYPE,MAJ:MIN,FSTYPE,SIZE,RA,MOUNTPOINT,LABEL
```

Example output where **vdb** is now configured as the mount points for **aiops**:

```
$ lsblk --output NAME,KNAME,TYPE,MAJ:MIN,FSTYPE,SIZE,RA,MOUNTPOINT,LABEL
NAME               KNAME TYPE MAJ:MIN FSTYPE       SIZE   RA MOUNTPOINT             LABEL
vda                vda   disk 252:0                 250G 4096
├─vda1             vda1  part 252:1   xfs            1G 4096 /boot
└─vda2             vda2  part 252:2   LVM2_member  249G 4096
  ├─rhel-root      dm-0  lvm  253:0   xfs          233G 4096 /
  └─rhel-swap      dm-1  lvm  253:1   swap          16G 4096 [SWAP]
vdb                vdb   disk 252:16  LVM2_member  200G 4096
└─aiops-aiops      dm-2  lvm  253:2   xfs          200G 4096 /var/lib/aiops/storage aiops
```

Note: If the logical volume sizes need to be increased after installation, follow the instructions in Resizing storage (IBM Cloud Pak for AIOps on Linux).

Online installation of IBM Cloud Pak for AIOps on Linux

Learn about installing IBM Cloud Pak for AIOps on Linux.

Overview

You can install a production deployment of IBM Cloud Pak® for AIOps on a Linux® platform, without installing Red Hat® OpenShift® Container Platform.

The deployment is configured to collect usage data. If you want to disable usage data collection, then you can do this post-install. For more information, see Updating usage data collection preferences.

Limitations

- Starter-size deployments are not supported.

- The following features and capabilities are only available for deployments of IBM Cloud Pak for AIOps on Red Hat OpenShift, and are not available for deployments of IBM Cloud Pak for AIOps on Linux:
  - FIPS support
  - Secure tunnel
  - Multi-zone HADR
  - IBM Cognos® Analytics integration

Before you begin

Ensure that you meet the following prerequisites:

- Your cluster meets all of the requirements that are detailed in Planning an installation of IBM Cloud Pak for AIOps on Linux. The Linux cluster must be reserved for the sole use of IBM Cloud Pak for AIOps, and you must have a minimum of three control planes nodes.
- Local storage is configured in accordance with the instructions in Configuring local volumes.
- You have the credentials for the root user. Root user must be used to install IBM Cloud Pak for AIOps.
- The worker nodes and the client machine that you are running the installation from have network connectivity to the control plane nodes.
- You have a load balancer configured in accordance with the details in Load balancing.

Important: If IBM Sales representatives and Business Partners supplied you with a custom profile ConfigMap to customize your deployment, then you must follow their instructions to apply it during installation. The custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. For more information about custom sizing, see Custom sizing.

Installation Procedure

The installation procedure requires you to download the **aiopsctl** tool, which installs IBM Cloud Pak for AIOps and its prerequisites on your cluster.

1. Retrieve your entitlement key
2. Optionally configure a custom certificate
3. Create environment variables
4. Install the **aiopsctl** tool and register cluster nodes
5. Install IBM Cloud Pak for AIOps
6. Verify your installation
7. Access the Cloud Pak for AIOps console

1. Retrieve your entitlement key

Obtain the IBM entitlement key that is assigned to your **IBMid**. The key is needed to pull the IBM Cloud Pak for AIOps images from the IBM® Entitled Registry.

1. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Entitlement key* section, select **Copy** to copy the entitlement key to the clipboard.

2. Optionally configure a custom certificate

If you want to use your own custom certificate for IBM Cloud Pak for AIOps instead of the default cluster certificate, then use the following steps to create a certificate and key that you can supply as parameters at installation time.

1. Ensure that you have the following three PEM-encoded X.509 certificate files:
   - **caintermediate.pem**: The intermediate certificate that issued your server certificate.
   - **aiops.pem**: An IBM Cloud Pak for AIOps certificate, which includes the two fully qualified domain names (FQDNs) for **aiops-cpd** and **cp-console-aiops** in the Subject Alternative Name (SAN) list.
   - **aiops.key.pem**: A key file for the signed certificate in aiops.pem

   Tip: You can create the FQDN strings to use for **aiops-cpd** and **cp-console-aiops** by prepending **cp-console-aiops** and **aiops-cpd** to your load balancer's host name. For example, if your load balancer host name is **loadbalancerhost.acme.com**, then the FQDN strings are **cp-console-aiops.loadbalancerhost.acme.com** and **aiops-cpd.loadbalancerhost.acme.com**.
2. Concatenate the server and intermediate certificates into one file called **aiops-certificate-chain.pem**.

   ```
   cat aiops.pem caintermediate.pem > aiops-certificate-chain.pem
   ```

Note: If you do not install IBM Cloud Pak for AIOps with a custom certificate, you can switch to using a custom certificate after installation. For more information, see Using a custom certificate (IBM Cloud Pak for AIOps on Linux).

3. Create environment variables

Create and then source a shell script that is named *aiops_var.sh*, which defines the environment variables that are used to provide installation parameters for your deployment. Use the following codeblock as a template, replacing the brackets **< ... >** with values for your environment. It is important that you keep this file.

For more information about choosing a deployment type, see Incremental adoption. Subject to further hardware requirements, you can update the deployment type post-install. For more information, see Updating the deployment type.

Important: If you used an alternative path for APP_STORAGE_PATH or PLATFORM_STORAGE_PATH when you set up your local storage in Configuring local volumes, then you must change the values of these environment variables accordingly.

```
#================================================================================================================
```

```
# IBM Cloud Pak for AIOps installation variables (Linux)
#================================================================================================================================
========
export TARGET_USER="root"
export ACCEPT_LICENSE=false # Set to true to agree to the license terms.

# ------------------------------------------------------------------------------------------------------------------------------
---------
# IBM Entitled Registry
# ------------------------------------------------------------------------------------------------------------------------------
---------
export IBM_ENTITLEMENT_KEY=<ibm-entitlement-key> # Set to the entitlement key retrieved in previous step.

# ------------------------------------------------------------------------------------------------------------------------------
---------
# Hostnames
# `<load_balancer_hostname>` - the hostname of your load balancer
# `<control_plane_node_n>` - the FQDN or IP address of each control plane node. For example,
"control_plane_node_1.example.com"
# `<worker_n>` - the FQDN or IP address of each worker node. For example, "worker_1.example.com"
# ------------------------------------------------------------------------------------------------------------------------------
---------
export LOAD_BALANCER_HOST="<load_balancer_hostname>"
export CONTROL_PLANE_NODE="<control_plane_node_1>"
export ADDITIONAL_CONTROL_PLANE_NODES=(
  "<control_plane_node_2>"
  "<control_plane_node_3>"
)
export WORKER_NODES=(
  "<worker_1>"
  "<worker_2>"
  "<worker_3>"
  "<worker_4>"
  "<worker_5>"
  "<worker_6>"
  "<worker_7>"
)

# ------------------------------------------------------------------------------------------------------------------------
# Incremental adoption - set your deployment type.
# Set to `extended` to install an extended deployment with log anomaly detection and ticket analysis capabilities
# Set to `base` to install a base deployment without log anomaly detection and ticket analysis capabilities
# ------------------------------------------------------------------------------------------------------------------------
export DEPLOY_TYPE="base"

# ------------------------------------------------------------------------------------------------------------------------------
--------
# Storage
# ------------------------------------------------------------------------------------------------------------------------------
--------
export APP_STORAGE_PATH="/var/lib/aiops/storage"
export PLATFORM_STORAGE_PATH="/var/lib/aiops/platform"
```

Run the following command to source your script and set the environment variables:

```
. ./aiops_var.sh
```

4. Install the `aiopsctl` tool and register cluster nodes

1. Run the following commands to install **aiopsctl** on your cluster, and then register the control plane nodes and worker nodes.

```
AIOPSCTL_TAR="aiopsctl-linux_amd64.tar.gz"
AIOPSCTL_INSTALL_URL="https://github.com/IBM/aiopsctl/releases/download/v4.8.0/${AIOPSCTL_TAR}"
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} curl -LO "${AIOPSCTL_INSTALL_URL}"
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} tar xf "${AIOPSCTL_TAR}"
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} mv aiopsctl /usr/local/bin/aiopsctl

echo "Installing main control plane node ${CONTROL_PLANE_NODE}"
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node up --accept-license=${ACCEPT_LICENSE} --role=control-
plane --registry-token="${IBM_ENTITLEMENT_KEY}" --app-storage "${APP_STORAGE_PATH}" --platform-storage
"${PLATFORM_STORAGE_PATH}" --load-balancer-host="${LOAD_BALANCER_HOST}"

K3S_TOKEN=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --token-only)
K3S_HOST=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --server-url-only)

echo "Installing additional control plane nodes"
for CP_NODE in "${ADDITIONAL_CONTROL_PLANE_NODES[@]}"; do
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${CP_NODE} curl -LO "${AIOPSCTL_INSTALL_URL}"
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${CP_NODE} tar xvf "${AIOPSCTL_TAR}"
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${CP_NODE} mv aiopsctl /usr/local/bin/aiopsctl
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${CP_NODE} aiopsctl cluster node up --accept-
license=${ACCEPT_LICENSE} --role=control-plane --server-url="${K3S_HOST}" --token="${K3S_TOKEN}" --registry-
token="${IBM_ENTITLEMENT_KEY}" --app-storage "${APP_STORAGE_PATH}" --platform-storage "${PLATFORM_STORAGE_PATH}" --
load-balancer-host="${LOAD_BALANCER_HOST}"
done

echo "Installing worker nodes"
K3S_LB_HOST="https://${LOAD_BALANCER_HOST}:6443"
for WORKER_NODE in "${WORKER_NODES[@]}"; do
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${WORKER_NODE} curl -LO "${AIOPSCTL_INSTALL_URL}"
```

```
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${WORKER_NODE} tar xvf "${AIOPSCTL_TAR}"
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${WORKER_NODE} mv aiopsctl /usr/local/bin/aiopsctl
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${WORKER_NODE} aiopsctl cluster node up --accept-
license=${ACCEPT_LICENSE} --role=worker --server-url="${K3S_LB_HOST}" --token="${K3S_TOKEN}" --registry-
token="${IBM_ENTITLEMENT_KEY}" --app-storage "${APP_STORAGE_PATH}"
done
```

Note: The preceding commands install the Red Hat OpenShift CLI (oc) if it is not already installed. If it is already installed, then you must ensure that it is at version 4.14 or higher.

2. Run the following command to see information about your cluster:

   ```
   aiopsctl cluster node info
   ```

   Example output:

   ```
   #  aiopsctl cluster node info
   Node Details
   Role: Control plane
   Server URL: https://test-server.acme.com:6443
   Token: ***** (Use --show-secrets to reveal token)
   ```

3. Check for network connectivity issues.

   VMWare vSphere virtual machines (VMs) with Red Hat® Enterprise Linux® can have connectivity issues. If you are not using vSphere VMs for your IBM Cloud Pak for AIOps deployment, then skip this step. Otherwise, run the following lookup command:

   ```
   nslookup kubernetes.default.svc.cluster.local 10.43.0.10
   ```

   If the lookup is successful, then proceed to the next step. If the lookup fails, then run the following command on each of your VMs to rectify connectivity problems:

   ```
   ethtool -K flannel.1 tx-checksum-ip-generic off
   ```

   For more information, see Installation on Linux fails with network connectivity errors when using `aiopsctl` to install on vSphere VMs.

5. Install IBM Cloud Pak for AIOps

Run the `aiopsctl` tool to install IBM Cloud Pak for AIOps.

If you did not configure a custom certificate in step 2 then run the following command:

```
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl server up --load-balancer-host="${LOAD_BALANCER_HOST}" --mode
"${DEPLOY_TYPE}"
```

If you configured a custom certificate in step 2 then run the following command:

```
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl server up --load-balancer-host="${LOAD_BALANCER_HOST}" --mode
"${DEPLOY_TYPE}" --certificate-file aiops-certificate-chain.pem --key-file aiops.key.pem
```

6. Verify your installation

The installation takes one to two hours to complete. If the installation is unsuccessful, an error message is displayed and a nonzero exit code is returned.

Run the following command to check the status of the components of your IBM Cloud Pak for AIOps installation:

```
aiopsctl status
```

Example output for a healthy installation:

```
$ aiopsctl status
o- [12 Aug 24 08:40 PDT] Getting cluster status
Control Plane Node(s):
    test-server-1.acme.com Ready
    test-server-2.acme.com Ready
    test-server-3.acme.com Ready

Worker Node(s):
    test-agent-1.acme.com Ready
    test-agent-2.acme.com Ready
    test-agent-3.acme.com Ready
    test-agent-4.acme.com Ready
    test-agent-5.acme.com Ready
    test-agent-6.acme.com Ready
    test-agent-7.acme.com Ready

o- [12 Aug 24 08:40 PDT] Checking AIOps installation status

  15 Ready Components
    cluster
    aimanager
    lifecycletrigger
    aiopsanalyticsorchestrator
    baseui
    elasticsearchcluster
    aiopsedge
    rediscp
```

```
    asm
    zenservice
    aiopsui
    commonservice
    kafka
    issueresolutioncore
    lifecycleservice

  AIOps installation healthy
```

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

7. Access the Cloud Pak for AIOps console

Run the following command on a control plane node to see the URL, username, and password for your IBM Cloud Pak for AIOps deployment.

```
aiopsctl server info --show-secrets
```

Example output:

```
aiopsctl server info --show-secrets

Cluster Access Details
URL:      aiops-cpd.test-server.acme.com
Username: cpadmin
Password: abcdefghijklmno
```

Important: Ensure that your environment's Domain Name System (DNS) is correctly configured to resolve the hosts for accessing the Cloud Pak for AIOps console. For more information, see DNS requirements.

What to do next

Important: Any commands that use `oc` must be run from a control plane node.

- Define integrations and applications with Defining.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/OMNIbus integrations.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/Impact integrations.

To uninstall your IBM Cloud Pak for AIOps on Linux deployment, follow the instructions in Uninstalling a deployment of IBM Cloud Pak for AIOps on Linux.

Offline installation of IBM Cloud Pak for AIOps on Linux (bastion host)

Learn about installing IBM Cloud Pak for AIOps on Linux in an offline environment with a bastion host.

Overview

If your Linux® cluster does not have internet access, you can still install IBM Cloud Pak® for AIOps by using an offline or air-gapped deployment. In this scenario, your air-gapped (offline) environment has a target registry, and a Linux cluster on which IBM Cloud Pak for AIOps is to be installed. The bastion host has access to the internet and to the air-gapped environment. IBM Cloud Pak for AIOps images are mirrored from the IBM Entitled Registry through the bastion server to the target registry in the offline environment. IBM Cloud Pak for AIOps can then be installed in the offline environment by using the target registry.

You can install a production deployment of IBM Cloud Pak for AIOps on a Linux platform, without installing Red Hat® OpenShift® Container Platform.

The deployment is configured to collect usage data. If you want to disable usage data collection, then you can do this post-install. For more information, see Updating usage data collection preferences.

Limitations

- Starter-size deployments are not supported.
- The following features and capabilities are only available for deployments of IBM Cloud Pak for AIOps on Red Hat OpenShift, and are not available for deployments of IBM Cloud Pak for AIOps on Linux:
  - FIPS support
  - Secure tunnel
  - Multi-zone HADR
  - IBM Cognos® Analytics integration

Before you begin

Ensure that you meet the following prerequisites:

- Your cluster meets all of the requirements that are detailed in Planning an installation of IBM Cloud Pak for AIOps on Linux. The Linux cluster must be reserved for the sole use of IBM Cloud Pak for AIOps, and you must have a minimum of three control planes nodes.
- Local storage is configured in accordance with the instructions in Configuring local volumes.
- You have the credentials for the root user. The root user must be used to install IBM Cloud Pak for AIOps.
- The bastion host and worker nodes have network connectivity to the control plane nodes.
- You have a load balancer configured in accordance with the details in Load balancing.

Important: If IBM Sales representatives and Business Partners supplied you with a custom profile ConfigMap to customize your deployment, then you must follow their instructions to apply it during installation. The custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. For more information about custom sizing, see Custom sizing.

Installation Procedure

The installation procedure requires you to download the **`aiopsctl`** tool, which installs IBM Cloud Pak for AIOps and its prerequisites on your cluster.

1. Setup the mirroring environment
2. Retrieve your entitlement key
3. Optionally configure a custom certificate
4. Create environment variables
5. Mirror images
6. Install the **`aiopsctl`** tool and register cluster nodes
7. Install IBM Cloud Pak for AIOps
8. Verify your installation
9. Access the Cloud Pak for AIOps console

1. Setup the mirroring environment

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description
• `icr.io` • `cp.icr.io` • `dd0.icr.io` • `dd2.icr.io` • `dd4.icr.io` • `dd6.icr.io`	Allow access to these hosts on port 443 to enable access to the **`IBM Cloud Container Registry`**, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.
• `dd1-icr.ibm-zh.com` • `dd3-icr.ibm-zh.com` • `dd5-icr.ibm-zh.com` • `dd7-icr.ibm-zh.com`	If you are located in China, also allow access to these hosts on port 443.
`github.com`	GitHub houses CASE files, IBM Cloud Pak tools, and scripts.
`mirror.openshift.com`	for the `oc` CLI

1.1 Download documentation and scripts for offline access

Download the Cloud Pak for AIOps 4.8.0 PDF (this documentation) so that you can access it offline.

1.2 Set up a target registry

You must have a local Docker type production-grade registry available to store the IBM Cloud Pak for AIOps images in. The registry must meet the following requirements:

- The registry must support OCI Manifests, in addition to Docker Manifest V2, schema 2.
- supports multi-architecture images.
- is accessible from the cluster nodes.
- must serve images over HTTPS.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.
- must have 143 GB of storage to hold all the software that is to be transferred to the target registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one.

1.3 Prepare a bastion host

The bastion host must meet the following requirements:

- is able to connect to the internet
- is able to connect to the air-gapped environment with access to the cluster and the target registry.

2. Retrieve your entitlement key

Obtain the IBM entitlement key that is assigned to your **IBMid**. The key is needed to pull the IBM Cloud Pak for AIOps images from the IBM® Entitled Registry.

1. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Entitlement key* section, select **Copy** to copy the entitlement key to the clipboard.

3. Optionally configure a custom certificate

If you want to use your own custom certificate for IBM Cloud Pak for AIOps instead of the default cluster certificate, then use the following steps to create a certificate and key that you can supply as parameters at installation time.

1. Ensure that you have the following three PEM-encoded X.509 certificate files:
   - **cacintermediate.pem**: The intermediate certificate that issued your server certificate.
   - **aiops.pem**: An IBM Cloud Pak for AIOps certificate, which includes the two fully qualified domain names (FQDNs) for **aiops-cpd** and **cp-console-aiops** in the Subject Alternative Name (SAN) list.
   - **aiops.key.pem**: A key file for the signed certificate in aiops.pem

   Tip: You can create the FQDN strings to use for **aiops-cpd** and **cp-console-aiops** by prepending **cp-console-aiops** and **aiops-cpd** to your load balancer's host name. For example, if your load balancer host name is **loadbalancerhost.acme.com**, then the FQDN strings are **cp-console-aiops.loadbalancerhost.acme.com** and **aiops-cpd.loadbalancerhost.acme.com**.
2. Concatenate the server and intermediate certificates into one file called **aiops-certificate-chain.pem**.

   ```
   cat aiops.pem cacintermediate.pem > aiops-certificate-chain.pem
   ```

Note: If you do not install IBM Cloud Pak for AIOps with a custom certificate, you can switch to using a custom certificate after installation. For more information, see Using a custom certificate (IBM Cloud Pak for AIOps on Linux).

4. Create environment variables

Create and then source a shell script that is named *aiops_var.sh*, which defines the environment variables that are used to provide installation parameters for your deployment. Use the following codeblock as a template, replacing the brackets **< ... >** with values for your environment. It is important that you keep this file.

For more information about choosing a deployment type, see Incremental adoption. Subject to further hardware requirements, you can update the deployment type post-install. For more information, see Updating the deployment type.

Important: If you used an alternative path for APP_STORAGE_PATH or PLATFORM_STORAGE_PATH when you set up your local storage in Configuring local volumes, then you must change the values of these environment variables accordingly.

```
#============================================================================================================================================
# IBM Cloud Pak for AIOps installation variables (Linux)
#============================================================================================================================================
export TARGET_USER="root"
export ACCEPT_LICENSE=false # Set to true to agree to the license terms.

# ------------------------------------------------------------------------------------------------------------------------------------
# IBM Entitled Registry
# ------------------------------------------------------------------------------------------------------------------------------------
export IBM_ENTITLEMENT_KEY=<ibm-entitlement-key> # Set to the entitlement key retrieved in previous step.

# ------------------------------------------------------------------------------------------------------------------------------------
# Hostnames
# `<load_balancer_hostname>` - the hostname of your load balancer
# `<control_plane_node_n>` - the FQDN or IP address of each control plane node. For example, "control_plane_node_1.example.com"
# `<worker-n>` - the FQDN or IP address of each worker node. For example, "worker_1.example.com"
# ------------------------------------------------------------------------------------------------------------------------------------
export LOAD_BALANCER_HOST="<load_balancer_hostname>"
export CONTROL_PLANE_NODE="<control_plane_node_1>"
export ADDITIONAL_CONTROL_PLANE_NODES=(
  "<control_plane_node_2>"
  "<control_plane_node_3>"
)
export WORKER_NODES=(
  "<worker_1>"
  "<worker_2>"
  "<worker_3>"
  "<worker_4>"
  "<worker_5>"
  "<worker_6>"
  "<worker_7>"
)

# ------------------------------------------------------------------------------------------------------------------------------------
# Incremental adoption - set your deployment type.
# Set to `extended` to install an extended deployment with log anomaly detection and ticket analysis capabilities
# Set to `base` to install a base deployment without log anomaly detection and ticket analysis capabilities
```

```
# ------------------------------------------------------------------------------------------------------------
export DEPLOY_TYPE="base"

# ------------------------------------------------------------------------------------------------------------
--------
# Storage
# ------------------------------------------------------------------------------------------------------------
--------
export APP_STORAGE_PATH="/var/lib/aiops/storage"
export PLATFORM_STORAGE_PATH="/var/lib/aiops/platform"

# ------------------------------------------------------------------------------------------------------------
--------
# Variables for offline installation
# If you need to specify a custom path for your target registry, then export TARGET_REGISTRY with a custom path. For
example:
# export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT/my/custom/path
# ------------------------------------------------------------------------------------------------------------
--------
export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
export TARGET_REGISTRY_USER=<your-registry-username>
export TARGET_REGISTRY_PASSWORD=<your-registry-password>
export SKIP_TLS=false # Set to true if your target registry does not provide a signed TLS certificate
```

Run the following command to source your script and set the environment variables:

```
. ./aiops_var.sh
```

5. Mirror images

1. Connect your bastion host to the internet and disconnect it from the air-gapped environment.
2. Run the following commands to download the **aiopsctl** tool and its prerequisites.

   ```
   AIOPSCTL_TAR="aiopsctl-linux_amd64.tar.gz"
   AIOPSCTL_INSTALL_URL="https://github.com/IBM/aiopsctl/releases/download/v4.8.0/${AIOPSCTL_TAR}"
   curl -LO "${AIOPSCTL_INSTALL_URL}"
   tar xf "${AIOPSCTL_TAR}"
   mv aiopsctl /usr/local/bin/aiopsctl

   aiopsctl bastion login cp.icr.io -u cp -p ${IBM_ENTITLEMENT_KEY}
   aiopsctl bastion login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD} --insecure-skip-
   tls-verify="${SKIP_TLS}"
   unset IBM_ENTITLEMENT_KEY

   aiopsctl bastion mirror-images --registry ${TARGET_REGISTRY}
   ```

   You can use the following command to monitor the mirroring process:

   ```
   tail -f /root/.aiopsctl/logs/aiopsctl.log
   ```

3. Disconnect the bastion host from the internet, and connect it to the air-gapped environment.

6. Install the `aiopsctl` tool and register cluster nodes

1. Run the following commands to install **aiopsctl** on your offline cluster, and then register the control plane nodes and worker nodes.

   ```
   scp /usr/local/bin/aiopsctl ${TARGET_USER}@${CONTROL_PLANE_NODE}:/usr/local/bin/aiopsctl

   if [ "$(command -v "oc")"  ]; then
     scp "$(command -v "oc")" ${TARGET_USER}@${CONTROL_PLANE_NODE}:/usr/local/bin/oc
   else
     scp ~/.aiopsctl/mirror/oc ${TARGET_USER}@${CONTROL_PLANE_NODE}:/usr/local/bin/oc
   fi

   echo "Installing main control plane node ${CONTROL_PLANE_NODE}"

   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node up --accept-license=${ACCEPT_LICENSE} --role=control-
   plane --registry="${TARGET_REGISTRY}" --registry-user="${TARGET_REGISTRY_USER}" --registry-
   token="${TARGET_REGISTRY_PASSWORD}" --offline --insecure-skip-tls-verify="${SKIP_TLS}" --app-storage
   "${APP_STORAGE_PATH}" --platform-storage "${PLATFORM_STORAGE_PATH}" --load-balancer-host="${LOAD_BALANCER_HOST}"

   # Configure kubeconfig for oc CLI
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} "mkdir -p ~/.kube && ln -s /etc/rancher/k3s/k3s.yaml ~/.kube/config"

   K3S_TOKEN=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --token-only)
   K3S_HOST=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --server-url-only)

   echo "Installing additional control plane nodes"
   for CP_NODE in "${ADDITIONAL_CONTROL_PLANE_NODES[@]}"; do
     ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} scp /usr/local/bin/aiopsctl
   ${TARGET_USER}@${CP_NODE}:/usr/local/bin/aiopsctl
     ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${CP_NODE} aiopsctl cluster node up --accept-
   license=${ACCEPT_LICENSE} --role=control-plane --server-url="${K3S_HOST}" --token="${K3S_TOKEN}" --
   registry="${TARGET_REGISTRY}" --registry-user="${TARGET_REGISTRY_USER}" --registry-
   token="${TARGET_REGISTRY_PASSWORD}" --offline --app-storage "${APP_STORAGE_PATH}" --platform-storage
   "${PLATFORM_STORAGE_PATH}" --load-balancer-host="${LOAD_BALANCER_HOST}"
   ```

```
done

echo "Installing worker nodes"
K3S_LB_HOST="https://${LOAD_BALANCER_HOST}:6443"
for WORKER_NODE in "${WORKER_NODES[@]}"
do
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} scp /usr/local/bin/aiopsctl
${TARGET_USER}@${WORKER_NODE}:/usr/local/bin/aiopsctl
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${WORKER_NODE} aiopsctl cluster node up --accept-
license=${ACCEPT_LICENSE} --role=worker --server-url="${K3S_LB_HOST}" --token="${K3S_TOKEN}" --
registry="${TARGET_REGISTRY}" --registry-user="${TARGET_REGISTRY_USER}" --registry-
token="${TARGET_REGISTRY_PASSWORD}" --app-storage "${APP_STORAGE_PATH}" --offline --insecure-skip-tls-
verify="${SKIP_TLS}"
done
```

Note: The preceding commands install the Red Hat OpenShift CLI (oc) if it is not already installed. If it is already installed, then you must ensure that it is at version 4.14 or higher.

2. Run the following command to see information about your cluster:

   ```
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info
   ```

   Example output:

   ```
   # ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info
   Node Details
   Role: Control plane
   Server URL: https://test-server.acme.com:6443
   Token: ***** (Use --show-secrets to reveal token)
   ```

3. Check for network connectivity issues.

   VMWare vSphere virtual machines (VMs) with Red Hat® Enterprise Linux® can have connectivity issues. If you are not using vSphere VMs for your IBM Cloud Pak for AIOps deployment, then skip this step. Otherwise, run the following lookup command:

   ```
   nslookup kubernetes.default.svc.cluster.local 10.43.0.10
   ```

   If the lookup is successful, then proceed to the next step. If the lookup fails, then run the following command on each of your VMs to rectify connectivity problems:

   ```
   ethtool -K flannel.1 tx-checksum-ip-generic off
   ```

   For more information, see Installation on Linux fails with network connectivity errors when using `aiopsctl` to install on vSphere VMs.

7. Install IBM Cloud Pak for AIOps

Run the `aiopsctl` tool to install IBM Cloud Pak for AIOps.

If you did not configure a custom certificate in step 3 then run the following command:

```
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl server up --load-balancer-host="${LOAD_BALANCER_HOST}" --mode
"${DEPLOY_TYPE}"
```

If you configured a custom certificate in step 3 then run the following command:

```
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl server up --load-balancer-host="${LOAD_BALANCER_HOST}" --mode
"${DEPLOY_TYPE}" --certificate-file aiops-certificate-chain.pem --key-file aiops.key.pem
```

8. Verify your installation

The installation takes one to two hours to complete. If the installation is unsuccessful, an error message is displayed and a nonzero exit code is returned.

Run the following command to check the status of the components of your IBM Cloud Pak for AIOps installation:

```
aiopsctl status
```

Example output for a healthy installation:

```
$ aiopsctl status
o- [12 Aug 24 08:40 PDT] Getting cluster status
Control Plane Node(s):
    test-server-1.acme.com Ready
    test-server-2.acme.com Ready
    test-server-3.acme.com Ready

Worker Node(s):
    test-agent-1.acme.com Ready
    test-agent-2.acme.com Ready
    test-agent-3.acme.com Ready
    test-agent-4.acme.com Ready
    test-agent-5.acme.com Ready
    test-agent-6.acme.com Ready
    test-agent-7.acme.com Ready

o- [12 Aug 24 08:40 PDT] Checking AIOps installation status

  15 Ready Components
```

```
    cluster
    aimanager
    lifecycletrigger
    aiopsanalyticsorchestrator
    baseui
    elasticsearchcluster
    aiopsedge
    rediscp
    asm
    zenservice
    aiopsui
    commonservice
    kafka
    issueresolutioncore
    lifecycleservice

  AIOps installation healthy
```

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

9. Access the Cloud Pak for AIOps console

Run the following command on a control plane node to see the URL, username, and password for your IBM Cloud Pak for AIOps deployment.

```
aiopsctl server info --show-secrets
```

Example output:

```
aiopsctl server info --show-secrets

Cluster Access Details
URL:      aiops-cpd.test-server.acme.com
Username: cpadmin
Password: abcdefghijklmno
```

Important: Ensure that your environment's Domain Name System (DNS) is correctly configured to resolve the hosts for accessing the Cloud Pak for AIOps console. For more information, see DNS requirements.

What to do next

Important: Any commands that use `oc` must be run from a control plane node.

- Define integrations and applications with Defining.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/OMNIbus integrations.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/Impact integrations.

To uninstall your IBM Cloud Pak for AIOps on Linux deployment, follow the instructions in Uninstalling a deployment of IBM Cloud Pak for AIOps on Linux.

Online upgrade of IBM Cloud Pak for AIOps on Linux

Use these instructions to upgrade an online deployment of IBM Cloud Pak® for AIOps 4.7.0 or later to 4.8.0.

Overview

This procedure is for deployments of IBM Cloud Pak for AIOps on Linux®. This procedure can be used on an online deployment of IBM Cloud Pak for AIOps 4.7.0 or later, and can still be used if the deployment has hotfixes applied.

You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 4.6.1 or earlier. Upgrade of deployments of IBM Cloud Pak® for AIOps on Linux older than 4.7.0 is not supported. If you have a deployment of IBM Cloud Pak for AIOps older than 4.7.0, then you must uninstall it and then install 4.8.0.

If you have an offline deployment of IBM Cloud Pak for AIOps on Linux, follow the instructions in Offline upgrade of IBM Cloud Pak for AIOps on Linux.

If you have a deployment of IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform, follow the instructions in Upgrading IBM Cloud Pak for AIOps on OpenShift.

Before you begin

Ensure that you meet the following prerequisites:

- The worker nodes and the client machine that you are running the upgrade from have network connectivity to the control plane nodes.
- You have the credentials for the root user. Root user must be used to upgrade IBM Cloud Pak for AIOps.

Warnings:

- Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources are lost when IBM Cloud Pak for AIOps is upgraded, and must be manually reapplied after upgrade. For more information, see Manual adjustments are not persisted.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your online IBM Cloud Pak for AIOps deployment.

1. Ensure cluster readiness
2. Update the aiopsctl tool on the cluster nodes
3. Upgrade IBM Cloud Pak for AIOps
4. Verify your deployment
5. Post upgrade actions

1. Ensure cluster readiness

Recommended: Take a backup of IBM Cloud Pak for AIOps before you upgrade to v4.8.0. For more information, see Back up and restore (IBM Cloud Pak for AIOps on Linux).

1. Ensure that your cluster still meets all the prerequisites for deployment. For more information, see Planning an installation of IBM Cloud Pak for AIOps on Linux.
2. Set environment variables.
    1. Set an environment variable for your deployment mode.

        From IBM Cloud Pak for AIOps 4.8.0, deployment mode is configured with a command line argument instead of with a configuration file.

        Edit the **aiops_var.sh** file that you created when you installed IBM Cloud Pak for AIOps, and update the **DEPLOY_TYPE** environment variable to the required value. Set to **extended** if you have an extended deployment with log anomaly detection and ticket analysis capabilities, or set to **base** if you have a base deployment without log anomaly detection and ticket analysis capabilities.

    2. Run the following command from the directory that the script is in, to set the environment variables that are used later.

        ```
        . ./aiops_var.sh
        ```

    Note: If you do not still have **aiops_var.sh** from when you installed IBM Cloud Pak for AIOps, then follow the instructions in Create environment variables.
3. Delete any evicted **connector-orchestrator** pods.
    1. Run the following command to check whether there are any evicted **connector-orchestrator** pods.

        ```
        ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} oc get pods -n aiops | grep connector-orchestrator
        ```

    2. Cleanup any evicted connector-orchestrator pods.

        If the previous command returned any pods with a *STATUS* of **Evicted**, then run the following command to delete each of them.

        ```
        ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} oc delete pod -n aiops <connector_orchestrator>
        ```

        Where **<connector_orchestrator>** is a pod returned by the previous step.
4. Run the following command to ensure the safety of the Postgres database during upgrade.

    Important: Failure to run this step might lead to a complete loss of data.

    ```
    ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} oc label clusters.postgresql.k8s.enterprisedb.io -n aiops aiops-installation-edb-postgres operator.ibm.com/opreq-do-not-uninstall='true'
    ```

2. Update the aiopsctl tool on the cluster nodes

Run the following commands to update the **aiopsctl** tool on all the cluster nodes.

Note: When the command to update the first control plane node is run, you are prompted whether you want to accept the upgrade. Subsequent nodes use the **-y** flag to bypass this prompt.

```
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl update --version v4.8.0

echo "Upgrading main control plane node ${CONTROL_PLANE_NODE}"
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node up --accept-license=${ACCEPT_LICENSE} --role=control-plane

echo "Upgrading additional control plane nodes"
for CP_NODE in "${ADDITIONAL_CONTROL_PLANE_NODES[@]}"; do
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${CP_NODE} aiopsctl update -y --version v4.8.0
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${CP_NODE} aiopsctl cluster node up -y --accept-license=${ACCEPT_LICENSE} --role=control-plane
done

echo "Upgrading worker nodes"
for WORKER_NODE in "${WORKER_NODES[@]}"; do
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${WORKER_NODE} aiopsctl update -y --version v4.8.0
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${WORKER_NODE} aiopsctl cluster node up -y --accept-license=${ACCEPT_LICENSE} --role=worker
done
```

3. Upgrade IBM Cloud Pak for AIOps

Run the following command to upgrade IBM Cloud Pak for AIOps to v4.8.0.

```
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl server up --load-balancer-host="${LOAD_BALANCER_HOST}" --mode
"${DEPLOY_TYPE}"
```

Note: You are prompted whether you want to accept the upgrade.

4. Verify your deployment

The upgrade takes around an hour to complete. If the upgrade is unsuccessful, an error message is displayed and a nonzero exit code is returned.

1. Check the status of your deployment.

   Run the following command to check the status of the components of your IBM Cloud Pak for AIOps installation:

   ```
   aiopsctl status
   ```

   Example output for a healthy deployment:

   ```
   $ aiopsctl status
   o- [12 Aug 24 08:40 PDT] Getting cluster status
   Control Plane Node(s):
       test-server-1.acme.com Ready
       test-server-2.acme.com Ready
       test-server-3.acme.com Ready

   Worker Node(s):
       test-agent-1.acme.com Ready
       test-agent-2.acme.com Ready
       test-agent-3.acme.com Ready
       test-agent-4.acme.com Ready
       test-agent-5.acme.com Ready
       test-agent-6.acme.com Ready
       test-agent-7.acme.com Ready

   o- [12 Aug 24 08:40 PDT] Checking AIOps installation status

     15 Ready Components
       cluster
       aimanager
       lifecycletrigger
       aiopsanalyticsorchestrator
       baseui
       elasticsearchcluster
       aiopsedge
       rediscp
       asm
       zenservice
       aiopsui
       commonservice
       kafka
       issueresolutioncore
       lifecycleservice

     AIOps installation healthy
   ```

2. Run the following command and check that the VERSION that is returned is 4.8.0.

   ```
   aiopsctl server version
   ```

If the upgrade fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

5. Post upgrade actions

1. If you previously took a backup of your deployment, it is recommended that you take a new back up. For more information, see Back up and restore (IBM Cloud Pak for AIOps on Linux).
2. If the *EXPIRY_SECONDS* environment variable was set for configuring log anomaly alerts, the environment variable was not retained in the upgrade. After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly alerts.
3. If you have a metric integration configured that stops working after upgrade, then you must follow the instructions in After upgrade, a metric integration goes into a failed state.

Offline upgrade of IBM Cloud Pak for AIOps on Linux

Use these instructions to upgrade an offline deployment of IBM Cloud Pak® for AIOps 4.7.0 or later to 4.8.0.

Overview

This procedure is for deployments of IBM Cloud Pak for AIOps on Linux®. This procedure can be used on an online deployment of IBM Cloud Pak for AIOps 4.7.0 or later, and can still be used if the deployment has hotfixes applied.

You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 4.6.1 or earlier. Upgrade of deployments of IBM Cloud Pak® for AIOps on Linux older than 4.7.0 is not supported. If you have a deployment of IBM Cloud Pak for AIOps older than 4.7.0, then you must uninstall it and then install 4.8.0.

If you have an online deployment of IBM Cloud Pak for AIOps on Linux, follow the instructions in Online upgrade of IBM Cloud Pak for AIOps on Linux.

If you have a deployment of IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform, follow the instructions in Upgrading IBM Cloud Pak for AIOps on OpenShift.

Before you begin

Ensure that you meet the following prerequisites:

- The worker nodes and the client machine that you are running the upgrade from have network connectivity to the control plane nodes.
- You have the credentials for the root user. Root user must be used to upgrade IBM Cloud Pak for AIOps.

Warnings:

- Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources are lost when IBM Cloud Pak for AIOps is upgraded, and must be manually reapplied after upgrade. For more information, see Manual adjustments are not persisted.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your offline IBM Cloud Pak for AIOps deployment.

1. Ensure cluster readiness
2. Mirror the images
3. Update the aiopsctl tool on the cluster nodes
4. Upgrade IBM Cloud Pak for AIOps
5. Verify your deployment
6. Post upgrade actions

1. Ensure cluster readiness

Recommended: Take a backup of IBM Cloud Pak for AIOps before you upgrade to v4.8.0. For more information, see Back up and restore (IBM Cloud Pak for AIOps on Linux).

1. Ensure that your cluster still meets all of the prerequisites for an air-gapped deployment. For more information, see Planning an installation of IBM Cloud Pak for AIOps on Linux and Set up the mirroring environment.
2. Set environment variables.
    1. Set an environment variable for your deployment mode.

        From IBM Cloud Pak for AIOps 4.8.0, deployment mode is configured with a command line argument instead of with a configuration file.

        Edit the **`aiops_var.sh`** file that you created when you installed IBM Cloud Pak for AIOps, and update the **`DEPLOY_TYPE`** environment variable to the required value. Set to **`extended`** if you have an extended deployment with log anomaly detection and ticket analysis capabilities, or set to **`base`** if you have a base deployment without log anomaly detection and ticket analysis capabilities.
    2. Run the following command from the directory that the script is in, to set the environment variables that are used later.

        ```
        . ./aiops_var.sh
        ```

    Note: If you do not still have **`aiops_var.sh`** from when you installed IBM Cloud Pak for AIOps, then follow the instructions in Create environment variables.
3. Delete any evicted **`connector-orchestrator`** pods.
    1. Run the following command to check whether there are any evicted **`connector-orchestrator`** pods.

        ```
        ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} oc get pods -n aiops | grep connector-orchestrator
        ```
    2. Cleanup any evicted connector-orchestrator pods.

        If the previous command returned any pods with a *STATUS* of **`Evicted`**, then run the following command to delete each of them.

        ```
        ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} oc delete pod -n aiops <connector_orchestrator>
        ```

        Where **`<connector_orchestrator>`** is a pod returned by the previous step.
4. Run the following command to ensure the safety of the Postgres database during upgrade.

    Important: Failure to run this step might lead to a complete loss of data.

    ```
    ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} oc label clusters.postgresql.k8s.enterprisedb.io -n aiops aiops-installation-edb-postgres operator.ibm.com/opreq-do-not-uninstall='true'
    ```

2. Mirror the images

1. Connect your bastion host to the internet and disconnect it from the air-gapped environment.
2. Run the following commands to download the new version of the **aiopsctl** tool and mirror the new images.

```
AIOPSCTL_TAR="aiopsctl-linux_amd64.tar.gz"
AIOPSCTL_INSTALL_URL="https://github.com/IBM/aiopsctl/releases/download/v4.8.0/${AIOPSCTL_TAR}"
curl -LO "${AIOPSCTL_INSTALL_URL}"
tar xf "${AIOPSCTL_TAR}"
mv aiopsctl /usr/local/bin/aiopsctl

aiopsctl bastion login cp.icr.io -u cp -p ${IBM_ENTITLEMENT_KEY}
aiopsctl bastion login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD} --insecure-skip-
tls-verify="${SKIP_TLS}"
unset IBM_ENTITLEMENT_KEY

aiopsctl bastion mirror-images --registry ${TARGET_REGISTRY}
```

   You can use the following command to monitor the mirroring process:

```
tail -f /root/.aiopsctl/logs/aiopsctl.log
```

3. Disconnect the bastion host from the internet, and connect it to the air-gapped environment.

3. Update the aiopsctl tool on the cluster nodes

Run the following commands to update the **aiopsctl** tool on all the cluster nodes.

Note: When the command to update the first control plane node is run, you are prompted whether you want to accept the upgrade. Subsequent nodes use the **-y** flag to bypass this prompt.

```
scp /usr/local/bin/aiopsctl ${TARGET_USER}@${CONTROL_PLANE_NODE}:/usr/local/bin/aiopsctl

echo "Upgrading main control plane node ${CONTROL_PLANE_NODE}"

ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node up --accept-license=${ACCEPT_LICENSE} --role=control-plane
--offline

echo "Upgrading additional control plane nodes"
for CP_NODE in "${ADDITIONAL_CONTROL_PLANE_NODES[@]}"; do
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} scp /usr/local/bin/aiopsctl ${TARGET_USER}@${CP_NODE}:/usr/local/bin/aiopsctl
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${CP_NODE} aiopsctl cluster node up --accept-
license=${ACCEPT_LICENSE} -y --role=control-plane --offline
done

echo "Upgrading worker nodes"
for WORKER_NODE in "${WORKER_NODES[@]}"
do
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} scp /usr/local/bin/aiopsctl
${TARGET_USER}@${WORKER_NODE}:/usr/local/bin/aiopsctl
  ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${WORKER_NODE} aiopsctl cluster node up --accept-
license=${ACCEPT_LICENSE} -y --role=worker --offline
done
```

4. Upgrade IBM Cloud Pak for AIOps

Run the following command to upgrade IBM Cloud Pak for AIOps to v4.8.0.

```
ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl server up --load-balancer-host="${LOAD_BALANCER_HOST}" --mode
"${DEPLOY_TYPE}"
```

Note: You are prompted whether you want to accept the upgrade.

5. Verify your deployment

The upgrade takes around an hour to complete. If the upgrade is unsuccessful, an error message is displayed and a nonzero exit code is returned.

1. Check the status of your deployment.

   Run the following command to check the status of the components of your IBM Cloud Pak for AIOps installation:

```
aiopsctl status
```

   Example output for a healthy deployment:

```
$ aiopsctl status
o- [12 Aug 24 08:40 PDT] Getting cluster status
Control Plane Node(s):
    test-server-1.acme.com Ready
    test-server-2.acme.com Ready
    test-server-3.acme.com Ready

Worker Node(s):
    test-agent-1.acme.com Ready
    test-agent-2.acme.com Ready
```

```
    test-agent-3.acme.com Ready
    test-agent-4.acme.com Ready
    test-agent-5.acme.com Ready
    test-agent-6.acme.com Ready
    test-agent-7.acme.com Ready

o- [12 Aug 24 08:40 PDT] Checking AIOps installation status

  15 Ready Components
    cluster
    aimanager
    lifecycletrigger
    aiopsanalyticsorchestrator
    baseui
    elasticsearchcluster
    aiopsedge
    rediscp
    asm
    zenservice
    aiopsui
    commonservice
    kafka
    issueresolutioncore
    lifecycleservice

  AIOps installation healthy
```

2. Run the following command and check that the VERSION that is returned is 4.8.0.

```
aiopsctl server version
```

If the upgrade fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

6. Post upgrade actions

1. If you previously took a backup of your deployment, it is recommended that you take a new back up. For more information, see Back up and restore (IBM Cloud Pak for AIOps on Linux).
2. If the *EXPIRY_SECONDS* environment variable was set for configuring log anomaly alerts, the environment variable was not retained in the upgrade. After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly alerts.
3. If you have a metric integration configured that stops working after upgrade, then you must follow the instructions in After upgrade, a metric integration goes into a failed state.

Administering IBM Cloud Pak for AIOps on Linux

After you install IBM Cloud Pak® for AIOps, you can access several options for administering and customizing your deployment.

- Adding a control plane node
- Adding a worker node
- Backup and restore
- Configuring topology resources for production
- Recovering from node failure
- Removing a node
- Resizing storage
- Restarting the environment
- Updating the deployment type
- Using a custom certificate

See also Cloud Pak administration

To gather data about your deployment for support or troubleshooting purposes, you can install and run the **`MustGather`** script:

- MustGather tool

Adding a control plane node (IBM Cloud Pak for AIOps on Linux)

Learn how to add a control plane node to a Linux® cluster that has a deployment of IBM Cloud Pak® for AIOps running on it.

Procedure

1. Set environment variables.

1. Run the following command to source the shell script **aiops_var.sh** that you created when you installed IBM Cloud Pak for AIOps on your Linux cluster.

   ```
   . ./aiops_var.sh
   ```

   For more information about **aiops_var.sh**, see *Create environment variables* in Online installation of IBM Cloud Pak for AIOps on Linux or Offline installation of IBM Cloud Pak for AIOps on Linux.

2. Set an environment variable for the new control plane node that you want to add.

   ```
   export NEW_CP_NODE="<new_cp_node>"
   ```

   Where **<new_cp_node>** is the FQDN or IP address of the control plane node to be added, for example **test-server4.acme.com**.

3. If the main control plane node is down or offline, then run the following command:

   ```
   export CONTROL_PLANE_NODE="<other_cp_node>"
   ```

   Where **<other_cp_node>** is the FQDN or IP address of a working control plane node that is an entry in the **ADDITIONAL_CONTROL_PLANE_NODES** array in **aiops_var.sh**.

2. Configure a new control plane node.

   Run the commands in one of the following codeblocks, depending on whether your deployment is online or offline.

   Online deployments

   ```
   K3S_TOKEN=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --token-only)
   K3S_HOST=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --server-url-only)

   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_CP_NODE} curl -LO "${AIOPSCTL_INSTALL_URL}"
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_CP_NODE} tar xvf "${AIOPSCTL_TAR}"
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_CP_NODE} mv aiopsctl /usr/local/bin/aiopsctl
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_CP_NODE} aiopsctl cluster node up --accept-
   license=${ACCEPT_LICENSE} --role=control-plane --server-url="${K3S_HOST}" --token="${K3S_TOKEN}" --registry-
   token="${IBM_ENTITLEMENT_KEY}" --app-storage "${APP_STORAGE_PATH}" --platform-storage "${PLATFORM_STORAGE_PATH}" --
   load-balancer-host="${LOAD_BALANCER_HOST}"
   ```

   Offline deployments

   Run the following commands from your bastion host:

   ```
   K3S_TOKEN=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --token-only)
   K3S_HOST=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --server-url-only)

   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} scp /usr/local/bin/aiopsctl
   ${TARGET_USER}@${NEW_CP_NODE}:/usr/local/bin/aiopsctl
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_CP_NODE} aiopsctl cluster node up --accept-
   license=${ACCEPT_LICENSE} --role=control-plane --server-url="${K3S_HOST}" --token="${K3S_TOKEN}"
   registry="${TARGET_REGISTRY}" --registry-user="${TARGET_REGISTRY_USER}" --registry-
   token="${TARGET_REGISTRY_PASSWORD}"  --app-storage "${APP_STORAGE_PATH}" --platform-storage
   "${PLATFORM_STORAGE_PATH}" --load-balancer-host="${LOAD_BALANCER_HOST}"
   ```

3. Update the environment variables script **aiops_var.sh**. This is to maintain the accuracy of the environment variables file.

   When your new control plane node is succesfully added and configured, update the **aiops_var.sh** file.

   - If you added the new control plane node as an additional control plane node, then add an entry for it to the **ADDITIONAL_CONTROL_PLANES** array.
   - If you added the new control plane node as the main control plane node, then set the value of **CONTROL_PLANE_NODE** to be your new control plane node.

4. Update your load balancer's configuration for the control plane node that you just added.

Adding a worker node (IBM Cloud Pak for AIOps on Linux)

Learn how to add a worker node to a Linux® cluster that has a deployment of IBM Cloud Pak® for AIOps running on it.

Procedure

1. Set environment variables.

   1. Run the following command to source the shell script **aiops_var.sh** that you created when you installed IBM Cloud Pak for AIOps on your Linux cluster.

      ```
      . ./aiops_var.sh
      ```

      For more information about **aiops_var.sh**, see *Create environment variables* in Online installation of IBM Cloud Pak for AIOps on Linux or Offline installation of IBM Cloud Pak for AIOps on Linux.

   2. Set an environment variable for the new worker node that you want to add.

      ```
      export NEW_WORKER_NODE="<new_worker_node>"
      ```

      Where **<new_worker_node>** is the FQDN or IP address of the worker node to be added, for example **agent8.acme.com**.

2. Run the following commands to configure your new worker node.

   For online deployments:

   ```
   K3S_TOKEN=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --token-only)
   K3S_HOST="https://${LOAD_BALANCER_HOST}:6443"

   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_WORKER_NODE} curl -LO "${AIOPSCTL_INSTALL_URL}"
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_WORKER_NODE} tar xvf "${AIOPSCTL_TAR}"
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_WORKER_NODE} mv aiopsctl /usr/local/bin/aiopsctl
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_WORKER_NODE} aiopsctl cluster node up --accept-
   license=${ACCEPT_LICENSE} --role=worker --server-url="${K3S_HOST}" --token="${K3S_TOKEN}" --registry-
   token="${IBM_ENTITLEMENT_KEY}" --app-storage "${APP_STORAGE_PATH}"
   ```

   For offline deployments:

   ```
   K3S_TOKEN=$(ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster node info --token-only)
   K3S_HOST="https://${LOAD_BALANCER_HOST}:6443"

   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} scp -r ~/.aiopsctl ${TARGET_USER}@${NEW_WORKER_NODE}:~/.aiopsctl/
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} scp /usr/local/bin/aiopsctl
   ${TARGET_USER}@${NEW_WORKER_NODE}:/usr/local/bin/aiopsctl
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NEW_WORKER_NODE} aiopsctl cluster node up --accept-
   license=${ACCEPT_LICENSE} --role=worker --server-url="${K3S_HOST}" --token="${K3S_TOKEN}" --
   registry="${TARGET_REGISTRY}" --registry-user="${TARGET_REGISTRY_USER}" --registry-
   token="${TARGET_REGISTRY_PASSWORD}" --app-storage "${APP_STORAGE_PATH}" --offline --insecure-skip-tls-
   verify="${SKIP_TLS}"
   ```

3. Update the environment variables script **aiops_var.sh**.

   When your worker node is succesfully added and configured, add an entry for it to the **WORKER_NODES** array for the new control plane node. This is to maintain the accuracy of the environment variables file.

Back up and restore (IBM Cloud Pak for AIOps on Linux)

Learn how to back up and restore a deployment of IBM Cloud Pak® for AIOps on a Linux® cluster.

Overview

The backup and restore procedure for deployments of IBM Cloud Pak for AIOps on Linux requires you to backup the virtual machines (VMs) in the cluster, and to use the VM backups for restoration as necessary. This procedure can be used for online and offline deployments of IBM Cloud Pak for AIOps.

The restore process can be used for the following scenarios:

- restore to a new cluster
- restore to the same cluster, which has missing worker or control nodes
- restore to the same cluster, which has some missing integrations, resource groups, or topology data.

The backup procedure is disruptive and requires the shutdown of the Linux cluster.

Backup

1. Shutdown the Linux cluster. Run steps 1-5 from Restart the environment. This step is mandatory - a backup can be used for restore only if it was taken after the Linux cluster was shut down.

2. Use a backup and restore tool of your choice to backup all of your VMs. If your chosen tool allows, it is recommended to group VMs together for backup (such as entire vApp backup on VMware®). This simplifies the backup process and helps ensure that all the cluster VMs are backed up consistently.

Restore

Note: The cluster that you are restoring to must have at least the same size and resources as the cluster that the backup was created on.

1. Use the backup and restore tool to restore the VMs.

2. Restart IBM Cloud Pak for AIOps. Run steps 6-8 from Restart the environment.

Configuring topology resources for production (IBM Cloud Pak for AIOps on Linux)

Learn how to configure the File observer for more than 600,000 resources.

Configuring the File observer

Additional configuration is required if you are using the File observer for more than 600,000 resources.

If IBM Sales representatives and Business Partners did not use the custom sizing tool to supply you with a custom profile ConfigMap to customize your deployment, then run the following command from a control plane node:

```
cat <<EOF | oc apply -f -
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ibm-aiops-orchestrator
    app.kubernetes.io/name: custom-sized-profiles
    app.kubernetes.io/part-of: ibm-aiops
    aiops-custom-size-profile-version: 4.2.0-2023
  name: aiops-custom-size-profile
data:
  # WARNING: Modifications to this ConfigMap may cause your AIOPs installation to become unstable.
  profiles: |
    generatedfor: HA
    configmaps:
    - name: aiops-topology-sizing
      data:
        asm: |
          file-observer:
            specs:
              replicas: 1
            containers:
              file-observer:
                resources:
                  requests:
                    cpu: 100m
                    memory: 750Mi
                  limits:
                    cpu: 1000m
                    memory: 1750Mi
                env:
                  - name: JVM_ARGS
                    value: -Xms300M -Xmx1400M
EOF
```

If IBM Sales representatives and Business Partners used the custom sizing tool to supply you with a custom profile ConfigMap to customize your deployment, then use the following steps to adjust the value of *JVM_ARGS* for the File observer to *-Xms300M -Xmx1400M*:

1. Log in to a control plane node and then run the following command:

   ```
   oc project aiops
   ```

2. Run the following commands to edit your custom profile configmap:

   ```
   CUSTOM_SIZE_PROFILE=$(oc get installation -n aiops -o jsonpath='{.items[0].status.customProfileConfigmap}')
   oc edit configmap "${CUSTOM_SIZE_PROFILE}" -n aiops
   ```

3. Adjust the value of *JVM_ARGS* for the File observer to *-Xms300M -Xmx1400M* and save your changes.

   Example excerpt:

   ```
   - name: aiops-topology-sizing
     data:
       asm: |
         file-observer:
           specs:
             replicas: 1
           containers:
             file-observer:
               resources:
                 requests:
                   cpu: 100m
                   memory: 750Mi
                 limits:
                   cpu: 1000m
                   memory: 1750Mi
               env:
                 - name: JVM_ARGS
                   value: -Xms300M -Xmx1400M
   ```

Recovering from node failure (IBM Cloud Pak for AIOps on Linux)

Learn about recovering from a failed worker or control plane node on a Linux® cluster that has a deployment of IBM Cloud Pak® for AIOps running on it.

Overview

IBM Cloud Pak for AIOps is highly available and able to withstand a single node failure. IBM Cloud Pak for AIOps replicates stateful and nonstateful pods and storage, and can also move nonstateful pods to alternative nodes if the node that they are scheduled on fails.

If a node in your Linux cluster fails and is inaccessible, use the procedure in this topic to delete it. Stateful pods that were on the failed node are rescheduled to alternative nodes, and data is replicated from other healthy replicas.

Note: The following procedures are also available:

- if you no longer want a node in your Linux cluster to be used by your IBM Cloud Pak for AIOps deployment, follow the instructions in Removing a node.
- if you want to uninstall IBM Cloud Pak for AIOps completely, follow the instructions in Uninstalling IBM Cloud Pak for AIOps on Linux.

Procedure

1. Run the following command to identify a failed node.

   ```
   oc get node
   ```

   A failed worker node has a *STATUS* of **NotReady** and does not have a ROLE of **control-plane.etcd.master**.

   A failed control plane node has a *STATUS* of **NotReady** and has a ROLE of **control-plane.etcd.master**.

   Example output for a Linux cluster with a failed worker node called **agent1.acme.com**.

   ```
   NAME               STATUS    ROLES                       AGE   VERSION
   agent1.acme.com    NotReady  <none>                      7d2h  v1.30.2+k3s2
   agent2.acme.com    Ready     <none>                      7d2h  v1.30.2+k3s2
   agent3.acme.com    Ready     <none>                      7d2h  v1.30.2+k3s2
   agent4.acme.com    Ready     <none>                      7d2h  v1.30.2+k3s2
   agent5.acme.com    Ready     <none>                      7d2h  v1.30.2+k3s2
   agent6.acme.com    Ready     <none>                      7d2h  v1.30.2+k3s2
   agent7.acme.com    Ready     <none>                      7d2h  v1.30.2+k3s2
   server1.acme.com   Ready     control-plane.etcd.master   7d2h  v1.30.2+k3s2
   server2.acme.com   Ready     control-plane.etcd.master   7d2h  v1.30.2+k3s2
   server3.acme.com   Ready     control-plane.etcd.master   7d2h  v1.30.2+k3s2
   ```

2. Add another node to your Linux cluster so that replicas can be distributed across the additional node.

   - If you have a failed worker node, follow the instructions in Adding a worker node.
   - If you have a failed control plane node, follow the instructions in Adding a control plane node.

3. Delete the failed node.

   Run the following command from a control plane node to delete the failed node and relocate its workloads and storage to other nodes in the cluster.

   ```
   aiopsctl cluster delete-node <node_name>
   ```

   Where **<node_name>** is the name of the failed worker or control plane node.

   You are prompted with a warning that data loss might occur, and a query whether to continue. To avoid data loss, examine the **Data stored on node** table to check that the replication of your data stores is sufficient such that data loss does not occur if you continue to delete the node. Check that the last column **Replicas remaining after deletion** does not have *0* for any row, and that an extra warning message to contact IBM Support is not displayed. Type **y** at the prompt if you want to continue.

   Example output for a cluster where it is safe to continue and remove the node:

   ```
   $ aiopsctl cluster delete-node agent1.acme.com
   [WARN] Removing node agent1.acme.com. This may cause data loss, read the information below carefully before
   continuing.

   Data stored on node:

   Name                          Total replicas  Replicas on node  Replicas remaining after deletion
   Kafka                         3               0                 3
   Identity Management Postgres  2               0                 2
   Zen MinIO                     3               1                 2
   CouchDB                       3               0                 3
   Elasticsearch                 3               0                 3
   Zen Postgres                  2               0                 2
   AIOps Postgres                3               1                 2
   AIOps MinIO                   5               0                 5
   Redis                         3               1                 2
   Cassandra                     3               0                 3

   Are you sure you wish to continue? (y/n): y

   o- [24 Jul 24 08:44 CDT] Deleting node agent1.acme.com...
   o- [24 Jul 24 08:44 CDT] Recovering datastores...

   # Node is removed
   ```

   Note: If you have only one control plane node, then you cannot delete that node, and **aiopsctl** prevents you from doing so.

4. Update the environment variables file **aiops_var.sh**.

   Edit the shell script **aiops_var.sh** that you created when you installed IBM Cloud Pak for AIOps on your Linux cluster. This is to maintain the accuracy of the environment variables file.

   For more information about **aiops_var.sh**, see *Create environment variables* in Online installation of IBM Cloud Pak for AIOps on Linux or Offline installation of IBM Cloud Pak for AIOps on Linux.

If you deleted a worker node:

Remove the entry for the deleted worker node from **aiops_var.sh**.

If you deleted a control plane node:

- If the deleted control plane node is in the ADDITIONAL_CONTROL_PLANES array, then remove the entry for the deleted control plane node.
- If the deleted control plane node is the CONTROL_PLANE_NODE, then replace the value of CONTROL_PLANE_NODE with the value of the new main control plane node that you created in step 2, if you have not already done so.

5. If you added and deleted a control plane node, then update your load balancer's configuration.

Removing a node (IBM Cloud Pak for AIOps on Linux)

Learn how to remove a worker node or a control plane node from a deployment of IBM Cloud Pak® for AIOps on Linux®.

Overview

If you no longer want a worker node or a control plane node in your Linux cluster to be used by your IBM Cloud Pak for AIOps deployment, then use the following procedure to remove it.

Note: The following procedures are also available:

- if you want to uninstall IBM Cloud Pak for AIOps completely, follow the instructions in Uninstalling IBM Cloud Pak for AIOps on Linux.
- if you want to delete and recover from a single unresponsive worker node in your IBM Cloud Pak for AIOps deployment, follow the instructions in Recovering from node failure.

Procedure

Important:

- You must check that the removal of a node will not cause there to be insufficient resources for IBM Cloud Pak for AIOps. For more information, see Planning an installation. You must always have three control plane nodes.
- If you have only one control plane node, do not remove it as this will cause IBM Cloud Pak for AIOps to stop functioning.

1. Before you remove a node, ensure that you first add a new node. For more information, see Adding a control plane node or Adding a worker node.
2. Set environment variables.
    1. Run the following command to source the shell script **aiops_var.sh** that you created when you installed IBM Cloud Pak for AIOps on your Linux cluster.

        ```
        . ./aiops_var.sh
        ```

        For more information about **aiops_var.sh**, see *Create environment variables* in Online installation of IBM Cloud Pak for AIOps on Linux or Offline installation of IBM Cloud Pak for AIOps on Linux.

    2. Set an additional temporary environment variable for the node that you are removing.

        ```
        export NODE_TO_REMOVE=<node_to_remove>
        ```

        Where **<node_to_remove>** is the FQDN or IP address of the node that you want to remove. For example, **node1.example.com**.

3. Run the following command to uninstall **k3s** on the node that you want to remove.

    ```
    ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl cluster delete-node ${NODE_TO_REMOVE}
    ```

    You are prompted with a warning that data loss might occur, and a query whether to continue. To avoid data loss, examine the **Data stored on node** table to check that the replication of your data stores is sufficient such that data loss does not occur if you continue to delete the node. Check that the last column **Replicas remaining after deletion** does not have *0* for any row, and that an extra warning message to contact IBM Support is not displayed. Type **y** at the prompt if you want to continue.

    Example output for a cluster where it is safe to continue and remove the node:

    ```
    $ aiopsctl cluster delete-node agent1.acme.com
    [WARN] Removing node agent1.acme.com. This may cause data loss, read the information below carefully before continuing.

    Data stored on node:

    Name                          Total replicas  Replicas on node  Replicas remaining after deletion
    Kafka                         3               0                 3
    Identity Management Postgres  2               0                 2
    Zen MinIO                     3               1                 2
    CouchDB                       3               0                 3
    Elasticsearch                 3               0                 3
    Zen Postgres                  2               0                 2
    AIOps Postgres                3               1                 2
    AIOps MinIO                   5               0                 5
    Redis                         3               1                 2
    Cassandra                     3               0                 3
    ```

```
Are you sure you wish to continue? (y/n): y

o- [24 Jul 24 08:44 CDT] Deleting node agent1.acme.com...
o- [24 Jul 24 08:44 CDT] Recovering datastores...

# Node is removed
```

Note: If you have only one control plane node, then you cannot delete that node, and `aiopsctl` prevents you from doing so.

4. Run one of the following commands to remove the node from your IBM Cloud Pak for AIOps deployment, depending on whether it is a worker node or a control plane node.

   Worker node:

   ```
   ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} ssh ${TARGET_USER}@${NODE_TO_REMOVE} aiopsctl cluster node down --role
   worker --app-storage ${APP_STORAGE_PATH}
   ```

   Control plane node:

   ```
   ssh ${TARGET_USER}@${NODE_TO_REMOVE} aiopsctl cluster node down --role control-plane
   ```

5. Update the environment variables file `aiops_var.sh`

   When your node is successfully removed, remove its entry from `aiops_var.sh`. This is to maintain the accuracy of the environment variables file.

   For more information about `aiops_var.sh`, see *Create environment variables* in Online installation of IBM Cloud Pak for AIOps on Linux or Offline installation of IBM Cloud Pak for AIOps on Linux.

   If you are removing a worker node:

   Remove the entry for the worker node that you removed from the WORKER_NODES array.

   If you are removing a control plane node:

   - If the control plane node that you removed is in the ADDITIONAL_CONTROL_PLANES array, then remove the entry for the control plane node from the ADDITIONAL_CONTROL_PLANES array.
   - If the control plane node that you removed is the CONTROL_PLANE_NODE, then replace the value of CONTROL_PLANE_NODE with the value of the new main control plane node that you created in step 1, if you have not already done so.

6. If you removed a control plane node, then update your load balancer's configuration.

Resizing storage (IBM Cloud Pak for AIOps on Linux)

Learn about increasing the logical storage volumes on an existing installation of IBM Cloud Pak® for AIOps on Linux®.

Note: Do not attempt to manually increase the size of your persistent volume claims (PVCs). PVC capacity limits are not relevant for deployments of IBM Cloud Pak for AIOps on Linux. Capacity is only limited by the size of your logical storage volumes.

Use your preferred method to monitor your disk usage. If you become close to running out of disk space, use the following steps to increase the volume sizes on your Linux cluster nodes as necessary:

1. Export environment variables.

   ```
   export DRIVE_NAME=<drive_name>
   export NEW_SIZE=<new_size>
   ```

   Where

   - `<drive_name>` is the name of the disk drive that was specified for IBM Cloud Pak for AIOps application storage when you set up local storage in Configuring local volumes.
   - `<new_size>` is the new increased size for the volume, in Gigabytes (G).

   For example:

   ```
   export DRIVE_NAME="/dev/vdb"
   export NEW_SIZE="199G"
   ```

2. Extend the logical volume.

   ```
   export APP_DMPATH=$(lvs --noheadings -o dmpath aiops | sed -e 's/^[[:space:]]*//')
   lvextend -L "${NEW_SIZE}" "{APP_DMPATH}"
   ```

3. Grow the file system.

   ```
   xfs_growfs "${STORAGE_PATH}"
   ```

Restarting the environment (IBM Cloud Pak for AIOps on Linux)

Learn how to shutdown and restart the Linux cluster where IBM Cloud Pak for AIOps is deployed.

Overview

Use this procedure before a known maintenance window or outage to shut down the Linux cluster where IBM Cloud Pak for AIOps is installed, and to restart the cluster and workloads afterward.

Warning: If you need to shut down the cluster where IBM Cloud Pak for AIOps is installed, then you must use the following procedure. Failure to do so can result in data loss or corruption.

Procedure

1. Validate the installation
2. Check the certificates
3. Prepare to scale down
4. Scale down the workloads and drain the nodes
5. Shut down the cluster
6. Restart the cluster
7. Scale up the workloads
8. Validate the installation

1. Validate the installation

Run the describe command:

```
kubectl describe installations.orchestrator.aiops.ibm.com -n aiops
```

Review the `ComponentStatus` fields to confirm that all components are marked as `Ready` and the phase is `Running`.

Example output:

```
Name:         ibm-cp-aiops
Namespace:    aiops
API Version:  orchestrator.aiops.ibm.com/v1alpha1
Kind:         Installation
Spec:
...
Status:
Componentstatus:
   Aimanager:                        Ready
   Aiopsanalyticsorchestrator:       Ready
   Aiopsedge:                        Ready
   Aiopsui:                          Ready
   Asm:                              Ready
   Baseui:                           Ready
   Cluster:                          Ready
   Commonservice:                    Ready
   Elasticsearch:                    Ready
   Flinkdeployment:                  Ready
   Issueresolutioncore:              Ready
   Kafka:                            Ready
   Lifecycleservice:                 Ready
   Lifecycletrigger:                 Ready
   Rediscp:                          Ready
   Tunnel:                           Ready
   Zenservice:                       Ready
Phase:                    Running
```

2. Check the certificates

Ensure that none of the certificates have problems or are expired.

Run the following command:

```
while read l; do echo "$l" | grep '^NAME' || (n=$(echo $l | sed 's/ .*//'); s=$(echo $l | sed 's/^[^ ]* *\([^
]*\).*/\1/'); x=$(kubectl get secret -n $n $s -o jsonpath='{.data.tls\.crt}' | base64 -d | openssl x509 -noout -enddate
2>/dev/null | sed 's!notAfter=!!'); echo "$l" | sed 's![^ ][^ ]*$!'"$x"'!'); done< <(kubectl get secret -A --field-
selector=type==kubernetes.io/tls -o custom-
columns=NAMESPACE:.metadata.namespace,NAME:.metadata.name,EXPIRY:.metadata.name)
```

Example output excerpt:

```
ibm-licensing   ibm-license-service-cert                                  Jan  8  13:32:07  2025  GMT
ibm-licensing   ibm-license-service-cert-internal                         Jan  7  13:31:12  2026  GMT
ibm-licensing   ibm-licensing-service-prometheus-cert                     Jan  7  13:31:25  2026  GMT
aiops           aimanager-aio-log-anomaly-feedback-learning-cert          Jan  7  14:01:43  2026  GMT
aiops           aimanager-aio-log-anomaly-golden-signals-cert             Jan  7  14:01:43  2026  GMT
aiops           aimanager-aio-oob-recommended-actions-cert                Jan  7  14:01:43  2026  GMT
<...>
```

Renew or re-create any certificates that have problems, are expired, or will expire before the cluster is restarted.

3. Prepare to scale down

1. Cordon all of the worker and control plane nodes.

   From a control plane node, run the following command for each of the worker and control plane nodes:

   ```
   kubectl cordon <node>
   ```

   Where **<node>** is the name of the node to cordon.

2. Make a note of the number of replicas.

   1. Make a note of the number of replicas for each StatefulSet.

      ```
      kubectl get statefulsets -n aiops
      ```

      Example output:

      ```
      NAME                                    READY   AGE
      aimanager-ibm-minio                     1/1     42m
      aiops-ibm-elasticsearch-es-server-all   1/1     86m
      aiops-installation-redis-server         3/3     84m
      aiops-ir-analytics-spark-worker         2/2     63m
      aiops-ir-core-ncobackup                 0/0     75m
      aiops-ir-core-ncoprimary                0/0     76m
      aiops-topology-cassandra                1/1     83m
      c-example-couchdbcluster-m              1/1     77m
      zen-minio                               3/3     76m
      ```

      Note:

      - If you do not have a IBM® Netcool® Operations Insight® probe integration, then `aiops-ir-core-ncobackup` and `aiops-ir-core-ncoprimary` has zero replicas.
      - If you upgraded from an earlier version of IBM Cloud Pak for AIOps, you also have an `icp-mongodb` StatefulSet.

   2. Make a note of the number of replicas for each StrimziPodSet.

      ```
      kubectl get strimzipodset -n aiops
      ```

      Example output:

      ```
      NAME                          PODS   READY PODS   CURRENT PODS   AGE
      iaf-system-kafka              3      3            3              13d
      iaf-system-zookeeper          3      3            3              13d
      ```

   3. Make a note of the number of replicas for each Flink deployment by using the following command:

      ```
      kubectl get deployment | grep flink | grep -v "operator"
      ```

      Example output:

      ```
      NAME                                    PODS   READY PODS   CURRENT PODS   AGE
      aiops-ir-lifecycle-flink                1/1     1            1             137m
      aiops-ir-lifecycle-flink-taskmanager    1/1     1            1             137m
      aiops-lad-flink                         1/1     1            1             139m
      aiops-lad-flink-taskmanager             2/2     2            2             139m
      ```

      Notes:

      - If you have a base deployment, then `aiops-lad-flink` and `aiops-lad-flink-taskmanager` do not show in the preceding output.
      - If you upgraded from an earlier version of IBM Cloud Pak for AIOps, you also have an `icp-mongodb` StatefulSet.

4. Scale down the workloads and drain the nodes

1. Scale down the operator deployments in the IBM Cloud Pak for AIOps namespace.

   ```
   kubectl scale deployment -l olm.owner.kind=ClusterServiceVersion -n aiops --replicas=0
   ```

   Run the following command to check that the number of replicas for each of the operator deployments is now 0.

   ```
   kubectl get deployment -n aiops -l olm.owner.kind=ClusterServiceVersion
   ```

   Example output:

   ```
   NAME                                          READY   UP-TO-DATE   AVAILABLE   AGE
   aimanager-operator-controller-manager         0/0     0            0           47m
   aiopsedge-operator-controller-manager         0/0     0            0           47m
   asm-operator                                  0/0     0            0           47m
   iaf-flink-operator-controller-manager         0/0     0            0           54m
   ibm-aiops-orchestrator-controller-manager     0/0     0            0           58m
   ibm-common-service-operator                   0/0     0            0           56m
   ibm-commonui-operator                         0/0     0            0           53m
   ibm-elastic-operator-controller-manager       0/0     0            0           54m
   ibm-events-operator-v5.0.1                    0/0     0            0           54m
   ibm-iam-operator                              0/0     0            0           54m
   ibm-ir-ai-operator-controller-manager         0/0     0            0           47m
   ibm-redis-cp-operator                         0/0     0            0           49m
   ibm-secure-tunnel-operator                    0/0     0            0           48m
   ibm-watson-aiops-ui-operator-controller-manager   0/0     0            0           48m
   ibm-zen-operator                              0/0     0            0           54m
   ```

```
ir-core-operator-controller-manager                0/0     0            0           47m
ir-lifecycle-operator-controller-manager           0/0     0            0           47m
operand-deployment-lifecycle-manager               0/0     0            0           55m
postgresql-operator-controller-manager-1-18-12     0/0     0            0           54m
```

Note: If you upgraded from an earlier version of IBM Cloud Pak for AIOps, you also have an `icp-mongodb-operator` deployment.

2. Scale down the StatefulSets that you noted in step 3.2.

   You can use the Cloud Pak for AIOps console, or create a shell script to do this.

   If you have a base deployment, then remove the following lines from the example shell script:

```
kubectl scale deployment aiops-lad-flink --replicas=0 -n aiops
kubectl scale deployment aiops-lad-flink-taskmanager --replicas=0 -n aiops
```

   Note:

   - If you have a base deployment, then `aiops-lad-flink` and `aiops-lad-flink-taskmanager` do not show in the preceding output.
   - If you upgraded from an earlier version of IBM Cloud Pak for AIOps, you also have an `icp-mongodb` StatefulSet.

   If you upgraded from an earlier version of IBM Cloud Pak for AIOps, then add the following line to the example shell script:

```
kubectl scale statefulsets icp-mongodb --replicas=0 -n aiops
```

   Example shell script:

```
#!/bin/bash

kubectl scale statefulsets aimanager-ibm-minio --replicas=0 -n aiops
sleep 2
kubectl scale statefulsets aiops-installation-redis-server --replicas=0 -n aiops
sleep 2
kubectl scale statefulsets aiops-ir-analytics-spark-worker --replicas=0 -n aiops
sleep 2
kubectl scale statefulsets aiops-ir-core-ncobackup --replicas=0 -n aiops
sleep 2
kubectl scale statefulsets aiops-ir-core-ncoprimary --replicas=0 -n aiops
sleep 2
kubectl scale deployment aiops-ir-lifecycle-flink --replicas=0 -n aiops
sleep 2
kubectl scale deployment aiops-ir-lifecycle-flink-taskmanager --replicas=0 -n aiops
sleep 2
kubectl scale statefulsets aiops-topology-cassandra --replicas=0 -n aiops
sleep 2
kubectl scale statefulsets c-example-couchdbcluster-m --replicas=0 -n aiops
sleep 2
kubectl scale deployment aiops-lad-flink --replicas=0 -n aiops
sleep 2
kubectl scale deployment aiops-lad-flink-taskmanager --replicas=0 -n aiops
sleep 2
kubectl scale statefulsets aiops-ibm-elasticsearch-es-server-all --replicas=0 -n aiops
sleep 2
kubectl scale statefulsets zen-minio --replicas=0 -n aiops
sleep 2
```

   Run the following command to check that the number of replicas for each of the StatefulSets is now 0.

```
kubectl get statefulsets -n aiops
```

   Example output:

```
NAME                                    READY   AGE
aimanager-ibm-minio                     0/0     42m
aiops-ibm-elasticsearch-es-server-all   0/0     86m
aiops-installation-redis-server         0/0     84m
aiops-ir-analytics-spark-worker         0/0     63m
aiops-ir-core-ncobackup                 0/0     75m
aiops-ir-core-ncoprimary                0/0     76m
aiops-topology-cassandra                0/0     83m
c-example-couchdbcluster-m              0/0     77m
zen-minio                               0/0     76m
```

3. Run the following command to check that the number of replicas for each of the Flink Deployments is now 0.

```
kubectl get deployments -n aiops | grep flink | grep -v "operator"
```

   Example output:

```
NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
aiops-ir-lifecycle-flink              0/0     0            0           137m
aiops-ir-lifecycle-flink-taskmanager  0/0     0            0           137m
aiops-lad-flink                       0/0     0            0           139m
aiops-lad-flink-taskmanager           0/0     0            0           139m
```

4. Shutdown the `Kafka` and `ZooKeeper` pods.

```
kubectl delete pod -l ibmevents.ibm.com/name=iaf-system-kafka -n aiops
kubectl delete pod -l ibmevents.ibm.com/name=iaf-system-zookeeper -n aiops
```

Run the following command to check that the Kafka and ZooKeeper pods have successfully shutdown. If the shutdown is complete, no pods are returned.

```
kubectl get pod -l ibmevents.ibm.com/controller=strimzipodset -n aiops
```

5. Scale down the PostgreSQL pods.

   When shutting down a PostgreSQL cluster, it is best to remove the primary replica last. The following script removes each database replica in the cluster with the primary removed last.

```
#!/bin/bash

AIOPS_NAMESPACE=aiops

# Get array of Postgres clusters
CLUSTERS=($(kubectl get clusters.postgresql.k8s.enterprisedb.io -n "${AIOPS_NAMESPACE}" -o go-template='{{range
.items}}{{.metadata.name}}{{" "}}{{end}}'))

# For each Postgres cluster, shutdown primary last
for cluster_name in "${CLUSTERS[@]}"; do
    primary=$(kubectl get clusters.postgresql.k8s.enterprisedb.io -n "${AIOPS_NAMESPACE}" "${cluster_name}" -o go-
template='{{.status.currentPrimary}}')
    instances=($(kubectl get clusters.postgresql.k8s.enterprisedb.io -n "${AIOPS_NAMESPACE}" "${cluster_name}" -o
go-template='{{range .status.instanceNames}}{{print . " "}}{{end}}'))
    for instance_name in "${instances[@]}"; do
        # Shutdown non-primary replicas
        if [ "${instance_name}" != "${primary}" ]; then
            kubectl delete pod -n "${AIOPS_NAMESPACE}" "${instance_name}" --ignore-not-found
        fi
    done

    # Shutdown the primary once all other replicas are down
    kubectl delete pod -n "${AIOPS_NAMESPACE}" "${primary}" --ignore-not-found
done
```

   Wait for all the Postgres pods to be deleted. All pods are deleted when the following command returns no pods:

```
kubectl get pod -l k8s.enterprisedb.io/podRole=instance -n aiops
```

6. (Optional) After the StatefulSets and StrimziPodSets are scaled down, drain all of the worker and control plane nodes. This step is not necessary if you are running a backup.

   From a control plane node, run the following command for each of the worker and control plane nodes:

```
kubectl drain <node>
```

   Where **<node>** is the name of the node to drain.

   Note: Some pods, such as storage pods, do not stop because this would violate the disruption budget. If this problem occurs, run the commands in each node until only the storage pods are left, and then stop the command and drain the next node.

5. Shut down the cluster

1. Shut down all the worker nodes on the cluster.
2. Shut down all the control plane nodes on the cluster.

6. Restart the cluster

1. Re-export the environment variables that you saved in step 1.1.
2. Restart the cluster nodes in the following order:
   1. Restart the control plane nodes. Check whether all the control plane nodes are in **ready** status by running the following command:

      ```
      kubectl get nodes
      ```

   2. Restart the worker nodes. Check whether all worker nodes are in **ready** status by running the following command:

      ```
      kubectl get nodes
      ```

3. After all the nodes are up, uncordon the control plane and worker nodes.

   From a control plane node, run the following command for each of the worker and control plane nodes:

```
kubectl uncordon <node>
```

   Where **<node>** is the name of the node to uncordon.

7. Scale up the workloads

Scaling up the workloads in the following order helps to minimize startup time and resource contention issues.

1. Scale the events operator back up.

```
kubectl scale deployment --replicas=1 $(kubectl get deployment -o custom-columns=NAME:.metadata.name --no-headers -n
aiops  | grep '^ibm-events-operator-') -n aiops
```

2. Check whether the Kafka and Zookeeper pods are running again. This can take a few minutes.

   ```
   kubectl get pod -l ibmevents.ibm.com/controller=strimzipodset -n aiops
   ```

   Example output when the Kafka and Zookeeper pods are running:

   ```
   NAME                                  READY   STATUS    RESTARTS   AGE
   iaf-system-kafka-0                    1/1     Running   0          13d
   iaf-system-kafka-1                    1/1     Running   0          13d
   iaf-system-kafka-2                    1/1     Running   0          13d
   iaf-system-zookeeper-0                1/1     Running   0          13d
   iaf-system-zookeeper-1                1/1     Running   0          13d
   iaf-system-zookeeper-2                1/1     Running   0          13d
   ```

3. You need to scale up each of the StatefulSets to the number of replicas as noted in step 3.2.

   1. Scale up Cassandra, Elasticsearch, and Spark StatefulSets in the following order:

      - aiops-topology-cassandra
      - aiops-ibm-elasticsearch-es-server-all
      - aiops-ir-analytics-spark-worker

      Run the following command to scale up each StatefulSet:

      ```
      kubectl scale statefulsets <statefulset> --replicas=<number of replicas> -n aiops
      ```

      Where:

      - **<statefulset>** is the StatefulSet to be scaled up
      - **<number_of_replicas>** is the number of replicas the StatefulSet it to be scaled up to

      For example,

      ```
      kubectl scale statefulsets aiops-topology-cassandra --replicas=1 -n aiops
      ```

   2. Scale the Flink deployments in the following order:

      - aiops-ir-lifecycle-flink
      - aiops-ir-lifecycle-flink-taskmanager
      - aiops-lad-flink
      - aiops-lad-flink-taskmanager

      Note: If you have a base deployment, then do not scale up **aiops-lad-flink** and **aiops-lad-flink-taskmanager**.

      Run the following command to scale up the Flink deployments:

      ```
      kubectl scale deployment <flink_deployment> --replicas=<number of replicas> -n aiops
      ```

      Where **<flink_deployment>** is the name of the Flink deployment.

   3. Scale the following StatefulSets in the specified order:

      - aiops-ir-core-ncoprimary
      - aiops-ir-core-ncobackup
      - c-example-couchdbcluster-m
      - aiops-installation-redis-server
      - aimanager-ibm-minio
      - zen-minio

      Run the following command to scale up each StatefulSet:

      ```
      kubectl scale statefulsets <statefulset> --replicas=<number of replicas> -n aiops
      ```

      Where:

      - **<statefulset>** is the StatefulSet to be scaled up
      - **<number_of_replicas>** is the number of replicas the StatefulSet it to be scaled up to

4. Scale up the operator deployments.

   ```
   kubectl scale deployment -l olm.owner.kind=ClusterServiceVersion -n aiops --replicas=1
   ```

8. Validate the installation

Note: After a complete cluster restart, it might take approximately an hour for the installation to start running again.

Run the describe command:

```
kubectl describe installations.orchestrator.aiops.ibm.com -n aiops
```

Review the **ComponentStatus** fields to confirm that all components are marked as **Ready** and the phase is **Running**.

Updating the deployment type (IBM Cloud Pak for AIOps on Linux)

Learn how to switch between the base deployment and extended deployment types.

Overview

After IBM Cloud Pak® for AIOps is installed, you can change your deployment type from a base deployment to an extended deployment, or from an extended deployment to a base deployment. For more information about the base deployment and extended deployment types, see Incremental adoption.

If you have log or ticket integrations configured that require an extended deployment, you are prevented from changing to a base deployment until these integrations are removed.

If you are changing from a base deployment to an extended deployment, the **aiopsctl** tool will check that your cluster meets the additional hardware requirements. For more information, see Hardware requirements.

Procedure

1. Set environment variables.
    1. Edit the **aiops_var.sh** file that you created when you installed IBM Cloud Pak for AIOps, and update the **DEPLOY_TYPE** environment variable to the required value.
    2. Run the following command from the directory that the script is in, to set the environment variables that are used later.

        ```
        . ./aiops_var.sh
        ```

    Note: If you do not still have **aiops_var.sh** from when you installed IBM Cloud Pak for AIOps, then follow the instructions for your deployment:

    Online deployment: Create environment variables or Offline deployment: Create environment variables.
2. Run the following command to update your deployment type.

    ```
    ssh ${TARGET_USER}@${CONTROL_PLANE_NODE} aiopsctl server up --load-balancer-host="${LOAD_BALANCER_HOST}" --mode "${DEPLOY_TYPE}"
    ```

Using a custom certificate (IBM Cloud Pak for AIOps on Linux)

Learn how to add a custom certificate to a deployment of IBM Cloud Pak® for AIOps on Linux®.

Procedure

If you want to use your own custom certificate for IBM Cloud Pak for AIOps instead of the default cluster certificate, then use the following steps after you have installed IBM Cloud Pak for AIOps on Linux.

1. Ensure that you have the following three PEM-encoded X.509 certificate files:
    - **caintermediate.pem**: The intermediate certificate that issued your server certificate.
    - **aiops.pem**: An IBM Cloud Pak for AIOps certificate, which includes the two fully qualified domain names (FQDNs) for **aiops-cpd** and **cp-console-aiops** in the Subject Alternative Name (SAN) list.
    - **aiops.key.pem**: A key file for the signed certificate in aiops.pem

    You can find the FQDNs for **aiops-cpd** and **cp-console-aiops** with the following command:

    ```
    kubectl get ingress -n aiops
    ```
2. Concatenate the server and intermediate certificates into one file called **aiops-certificate-chain.pem**.

    ```
    cat aiops.pem caintermediate.pem > aiops-certificate-chain.pem
    ```
3. Update the system configuration with the new certificate.

    ```
    aiopsctl server custom-certificate --certificate-file aiops-certificate-chain.pem --key-file aiops.key.pem
    ```

Installing and running the MustGather tool (IBM Cloud Pak for AIOps on Linux)

Learn about installing and running the MustGather tool for IBM Cloud Pak for AIOps on Linux.

Installing and running the MustGather tool

The MustGather tool is used for troubleshooting and support purposes. It collects data and logs about a deployment of IBM Cloud Pak for AIOps. To install and run the latest version of the MustGather tool, run the following command on the control plane node:

```
aiopsctl mustgather
```

Advanced configuration

You can also supply parameters to the MustGather tool. For more information, see Running the MustGather tool. In the example code blocks in the MustGather topics, replace **waiops-mustgather.sh** with **aiopsctl mustgather** for a deployment of IBM Cloud Pak for AIOps on Linux.

There are additional options available for deployments of IBM Cloud Pak for AIOps on Linux. Run the following command to seee usage:

```
aiopsctl mustgather -h
```

Example output:

```
root@acme-server:~# aiopsctl mustgather -h
Run the AIOps MustGather script to collect server information for support

Usage:
  aiopsctl mustgather [flags]

Flags:
  -a, --alias string      MustGather configuration alias
  -f, --force-install     Force MustGather to be reinstalled
  -h, --help              help for mustgather
  -o, --options string    Additional options that will be supplied to the MustGather script
  -u, --update            Check for MustGather updates

Global Flags:
      --accept-license     Accept the license
  -c, --config string      Config file location (default "~/.aiopsctl.yaml")
  -l, --log-file string    Log filename, set to stdout to print to console
      --show-secrets       Allow the display of secrets in output
  -v, --verbose            Verbose logging
      --wait-timeout int   Wait timeout for services to come ready in minutes (default 180)
```

Uninstalling IBM Cloud Pak for AIOps on Linux

Learn how to uninstall an online or offline (airgap) deployment of IBM Cloud Pak for AIOps on Linux, and to clean up the resources that it created.

Procedure

Use the following steps to uninstall IBM Cloud Pak for AIOps from your Linux cluster.

1. Export environment variables.

   Important: If you used an alternative path for APP_STORAGE_PATH when you set up your local application storage in Configuring local volumes, then you must change the value of this environment variable accordingly.

   ```
   export TARGET_USER="root"
   export CONTROL_PLANE_NODE="<control_plane_node_1>"
   export ADDITIONAL_CONTROL_PLANE_NODES=(
     "<control_plane_node_2>"
     "<control_plane_node_3>"
   )
   export WORKER_NODES=(
     "<worker_1>"
     "<worker_2>"
     "<worker_3>"
     "<worker_4>"
     "<worker_5>"
     "<worker_6>"
     "<worker_7>"
   )
   export APP_STORAGE_PATH="/var/lib/aiops/storage"
   ```

   Where:

   - `<control_plane_node_n>` is the FQDN or IP address of your control plane node. For example, "control-plane.example.com"
   - `<worker_n>` - is the FQDN or IP address of each worker node, where n is the number of nodes. For example, "worker-1.example.com"

2. Run the following commands from the control plane node:

   ```
   for WORKER_NODE in "${WORKER_NODES[@]}"
   do
     ssh ${TARGET_USER}@${WORKER_NODE} aiopsctl cluster node down --role worker --app-storage ${APP_STORAGE_PATH}
   done

   for CP_NODE in "${ADDITIONAL_CONTROL_PLANE_NODES[@]}"
   do
     ssh ${TARGET_USER}@${CP_NODE} aiopsctl cluster node down --role control-plane
   done

   aiopsctl cluster node down --role control-plane
   ```

Deploying IBM Cloud Pak for AIOps on OpenShift

You can install IBM Cloud Pak® for AIOps, Infrastructure Automation, or IBM Cloud Pak for AIOps and Infrastructure Automation on Red Hat® OpenShift® Container Platform. Online and offline deployments are supported.

If you need help deciding whether to install IBM Cloud Pak for AIOps on Red Hat OpenShift or Linux, see Choosing the deployment platform.

Notes:

- IBM Cloud Pak for AIOps and Infrastructure Automation can co-exist on the same cluster, but you cannot have multiple instances of either of them on the same cluster.
- You can optionally verify the IBM Cloud Pak for AIOps images before you install IBM Cloud Pak for AIOps. For more information, see Verifying images.
- After IBM Cloud Pak for AIOps is installed, you can optionally integrate with IBM Cognos® Analytics. For more information, see Visualizing dashboards and reports.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, you can optionally connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/OMNIbus integrations.

Read about planning an installation of IBM Cloud Pak for AIOps on Red Hat OpenShift, and then select one of the installation topics.

- Planning
- Installing IBM Cloud Pak for AIOps
- Installing Infrastructure Automation

Learn about upgrading, administering, and uninstalling IBM Cloud Pak for AIOps on Red Hat OpenShift:

- Upgrading
- Administering
- Uninstalling

Planning an installation of IBM Cloud Pak for AIOps on OpenShift

Review the following information to help you plan your installation.

Incremental adoption

Learn about incremental adoption and the deployment types that are available. For more information, see Incremental adoption (IBM Cloud Pak for AIOps).

Supported platforms

Learn about the Red Hat® OpenShift® Container Platform versions and public clouds that you can deploy on. Supported platforms.

Hardware requirements

Review the memory and hardware requirements for your deployment. For more information, see Hardware requirements.

Storage requirements

Review the storage requirements for your deployment, and learn about supported storage providers and storage classes. For more infomation, see Storage requirements.

Network requirements

Review the list of ports that need to be exposed for use in inbound and outbound communication. For more information, see Network requirements

High availability

Review information about how to configure high availability (HA) for your deployment. For more information, see High availability (HA) considerations

Security considerations

IBM Cloud Pak for AIOps supports different mechanisms for securing your environment and your data. In addition, as you are planning or going through an installation of IBM Cloud Pak for AIOps, you might need to create or provide certificates. For more information, see Security considerations.

Licensing

As part of using IBM Cloud Pak for AIOps, tracking your license usage is required. Although IBM Cloud Pak for AIOps provides license options not tracked directly by the product, to track usage, you must deploy the IBM Cloud Pak foundational services License Service on all clusters where IBM Cloud Pak for AIOps is installed. This background service collects and stores the license usage information for tracking license consumption and for audit purposes. For more information, Licensing.

Incremental adoption (IBM Cloud Pak for AIOps on OpenShift)

Learn about incremental adoption and the deployment types that are available for IBM Cloud Pak® for AIOps.

Overview

incremental adoption offers the flexibility of two deployment types that provide a different combination of footprint and capabilities.

- **Base deployment**: has a smaller footprint than the extended deployment, with vCPU and memory savings of around 15% for a production deployment. The log anomaly detection and ticket analysis (similar incidents and change risk) capabilities are not enabled.
- **Extended deployment**: has a larger footprint and has the full capabilities of IBM Cloud Pak for AIOps enabled.

The IBM Cloud Pak for AIOps license entitles you to use either of the deployment types.

Capabilities and hardware requirements

Log anomaly detection collects and analyzes log data to identify deviations from expected patterns of behavior. For more information, see Log anomaly detection.

Ticket analysis collects ticket data and compares it with similar tickets previously associated with incidents to identify possible remediating changes, and the risk of implementing those changes. For more information, see Similar ticket comparison and Change risk.

For more information about the footprint of each deployment type, see Hardware requirements.

Selecting your deployment type (install)

You can choose to install a base deployment or an extended deployment of IBM Cloud Pak for AIOps. An editable value in the `Installation` custom resource, which is defaulted for a base deployment, determines the deployment type.

Deployments that upgrade to IBM Cloud Pak for AIOps 4.8.0 inherit their previous deployment type, which for deployments earlier than 4.8.0 is extended deployment.

For more information about setting the deployment type at installation time, see the topic for your installation method:

- Starter installation
- Online production installation (CLI method)
- Online production installation (console method)
- Installing in an air-gapped environment by using a bastion host
- Installing in an air-gapped environment by using a portable device

Changing your deployment type (postinstall)

Subject to further hardware requirements, you can update the deployment type of a new or upgraded deployment at any time after installation to be either a base deployment or an extended deployment. For more information, see Updating the deployment type.

Supported platforms

Learn about which Red Hat OpenShift Container Platform versions and public cloud platforms are supported for IBM Cloud Pak® for AIOps.

- Red Hat® OpenShift® Container Platform
  - IBM Fusion HCI Systems
- Linux®
- Cloud platforms
  - IBM Cloud®
  - AWS Marketplace
  - AWS GovCloud
  - Microsoft® Azure
  - Google Cloud Platform

Red Hat OpenShift Container Platform

IBM Cloud Pak for AIOps adheres to the IBM Cloud Pak and Red Hat OpenShift version compatibility objective, and support and recommend installing on a supported Red Hat OpenShift EUS minor version.

For more information about support timelines for Red Hat OpenShift, including when a specific version is going out of support, see the Red Hat OpenShift Container Platform Life-Cycle Policy in the Red Hat OpenShift documentation.

Table 1. Supported OpenShift versions

Deployment	Platform	Red Hat OpenShift Container Platform version
IBM Cloud Pak for AIOps	Linux x86_64	4.14 4.15 4.16 (minimum 4.16.4) 4.17

Important:

- If you are planning to upgrade Red Hat OpenShift as part of an upgrade to IBM Cloud Pak for AIOps 4.8.0, you must complete the IBM Cloud Pak for AIOps upgrade before you upgrade Red Hat OpenShift.

- The hardware architecture for installing IBM Cloud Pak for AIOps must be x86_64 (amd64).
- Red Hat OpenShift v4.14 and onwards support heterogeneous architectures, but IBM Cloud Pak for AIOps does not support heterogeneous architectures. All nodes in the Red Hat OpenShift cluster must have the same architecture.

Red Hat OpenShift installation mode

IBM Cloud Pak for AIOps supports the OpenShift **`OwnNamespace`** and **`AllNamespaces`** installation modes. The **`OwnNamespace`** installation mode is the recommended installation mode because it provides better resilience and performance.

In the **`OwnNamespace`** installation mode, you must create and select a project (namespace) for the IBM Cloud Pak for AIOps operator to be installed into. In this mode, the operator watches and is available for use only in this single namespace. The **`All namespaces on the cluster (default)`** installation mode in the UI installation of the IBM Cloud Pak for AIOps operator is equivalent to the **`AllNamespaces`** mode in the CLI installation. In this mode, the operator is installed in the **`openshift-operators`** project (namespace), and watches and is available to all namespaces in the cluster. The Operator Deployment Lifecycle Manager (ODLM) installs suboperators in the same project (namespace) that you created the **`Installation`** custom resource instance in. Each suboperator watches and is available for use in its project (namespace).

For more information on Red Hat OpenShift operator installation modes, see Operator groups in the Red Hat OpenShift Container Platform documentation.

Resource allocation for nodes

Red Hat OpenShift nodes must have adequate resources reserved. For more information, see Allocating resources for nodes in an OpenShift Container Platform cluster in the Red Hat OpenShift documentation, and the Red Hat Customer Portal.

IBM Fusion HCI Systems

IBM Fusion is a hyperconverged infrastructure configurated appliance to deploy bare metal Red Hat OpenShift on-premise along with IBM Fusion software. This is a simple turn-key enterprise-grade solution to deploy Red Hat OpenShift for consistent, rapid deployment and management.

You can install IBM Cloud Pak for AIOps on Red Hat OpenShift with IBM Fusion HCI Systems as your storage provider.

For more information, see IBM Fusion HCI Systems.

Linux

You can install a production deployment of IBM Cloud Pak for AIOps directly on Linux, without the need to install Red Hat OpenShift. For more information, see Deploying IBM Cloud Pak for AIOps on Linux.

Cloud platforms

IBM Cloud Pak for AIOps is supported on the following cloud platforms:

- IBM Cloud®
- AWS Marketplace
- AWS GovCloud
- Microsoft® Azure
- Google Cloud Platform

For more information about the storage providers that you can use on the following platforms, see Recommended storage providers.

IBM Cloud

You can install IBM Cloud Pak for AIOps on IBM Cloud® with:

- Red Hat OpenShift
- Red Hat® OpenShift® on IBM Cloud® service (ROKS). For more information about ROKS, see the Red Hat OpenShift documentation Red Hat OpenShift on IBM Cloud.

For more information about installing IBM Cloud Pak for AIOps on ROKS, see Installing IBM Cloud Pak for AIOps on ROKS.

AWS Marketplace

You can install IBM Cloud Pak for AIOps on AWS with:

- Red Hat® OpenShift®
- Red Hat OpenShift Service on AWS (ROSA). For more information about ROSA, see the Red Hat documentation Red Hat OpenShift Service on AWS.

For more information about installing IBM Cloud Pak for AIOps on ROSA, see Installing IBM Cloud Pak for AIOps on ROSA.

AWS GovCloud

You can install IBM Cloud Pak for AIOps on AWS GovCloud.

For more information about installing IBM Cloud Pak for AIOps on AWS into a government region, see Installing IBM Cloud Pak for AIOps on AWS GovCloud.

Microsoft® Azure

You can install IBM Cloud Pak for AIOps on Azure with:

- Red Hat OpenShift

- Azure Red Hat OpenShift (ARO). For more information about ARO, see the Microsoft documentation Azure Red Hat OpenShift.

  For more information about installing IBM Cloud Pak for AIOps on (ARO), see Installing IBM Cloud Pak for AIOps on ARO.

Google Cloud Platform (GCP)

You can install IBM Cloud Pak for AIOps on GCP with Red Hat OpenShift.

For more information about IBM Cloud Pak for AIOps on GCP, see Installing IBM Cloud Pak for AIOps on GCP.

Hardware requirements

Learn about the hardware requirements for a deployment of IBM Cloud Pak® for AIOps on Red Hat® OpenShift® Container Platform. Your hardware must be able to support Red Hat OpenShift, IBM Cloud Pak for AIOps, and your chosen storage solution.

Before you begin

- You cannot change your selected deployment size after installation.
- Multi-zone high availability disaster resolution (HADR) is available as a nonproduction technology preview.. For more information, see Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HADR).
- vCPU is defined as when one x86 CPU splits each of its physical cores into virtual cores (vCPU). It is assumed that one x86 CPU's physical core can be split into two logical vCPUs.
- If Red Hat OpenShift is installed on VMware virtual machines, set the value of the `sched.cpu.latencySensitivity` parameter to high.
- Persistent storage is also required, for more information, see Storage requirements.

You can deploy a starter, production, or custom sized deployment of IBM Cloud Pak for AIOps. The processing abilities and hardware requirements vary for each deployment size. Review the information in the following sections:

- Hardware requirements - Red Hat OpenShift
- Hardware requirements - IBM Cloud Pak for AIOps
- Integrations
- Requests and limits
- Processing abilities
- Custom sizing
- Infrastructure Automation

Hardware requirements - Red Hat OpenShift

A Red Hat OpenShift cluster has master nodes and worker nodes. Tables 1 and 2 show the minimum node requirements for Red Hat OpenShift to withstand the demands of starter and production deployments of IBM Cloud Pak for AIOps.

IBM Cloud Pak for AIOps requires your Red Hat OpenShift cluster to have at least three master nodes. IBM Cloud Pak for AIOps has many Kubernetes operators that interface with the `API server` and `etcd` storage, and so master nodes must be adequately sized.

The number of worker nodes that are required varies, and depends on the storage and processing requirements of your IBM Cloud Pak for AIOps deployment. The size of each of your worker nodes can vary, but the combined resources of your worker nodes must comply with the resource totals in Tables 3 and 4. Each worker node must meet the minimum requirements in Table 2 to accommodate the placement of larger IBM Cloud Pak for AIOps pods.

A higher recommended vCPU is also given in Table 2. When you are selecting the size of your worker nodes, you must balance cost considerations against the benefits of improved resiliency and the ease of scheduling workloads. Over-allocating resources can improve resiliency by enabling sufficient resources to be available if a worker node fails, and having well-sized worker nodes makes the placement of workloads easier. If your largest worker node becomes unavailable, resiliency is improved if your smallest worker node is able to handle the largest worker node's workloads. The degree of over-allocation of resources is correlated with the extent of failure scenarios that can be accommodated.

Table 1. Master node minimum requirements

Resource	Starter	Production
Master node count	3	3
vCPU per node	4	4
Memory per node (GB)	16	16
Disk per node (GB)	120	120

Table 2. Worker node minimum requirements

Resource	Starter	Production
Minimum vCPU per node	8	16
Recommended vCPU per node	16	16
Memory per node (GB)	12	20
Disk per node (GB)	120	120

Note: The numbers that are specified are minimums. For high production workloads, you might need a larger disk per node (GB) for your worker loads.

Hardware requirements - IBM Cloud Pak for AIOps

Table 3 and table 4 show the hardware requirements for starter and production deployments of IBM Cloud Pak for AIOps, for a base deployment and an extended deployment. For more information about the differences between a base deployment and an extended deployment, see Incremental adoption.

The resource requirements are given for the following scenarios:

- Table 3: IBM Cloud Pak for AIOps only - you are deploying IBM Cloud Pak for AIOps on an existing Red Hat OpenShift cluster.
- Table 4: IBM Cloud Pak for AIOps and Red Hat OpenShift - you are creating a new Red Hat OpenShift cluster and deploying IBM Cloud Pak for AIOps onto it.

Table 3. Minimum requirements for starter and production deployments, without Red Hat OpenShift

Resource	Base deployment IBM Cloud Pak for AIOps only (Starter)	Extended deployment IBM Cloud Pak for AIOps only (Starter)	Base deployment IBM Cloud Pak for AIOps only (Production)	Extended deployment IBM Cloud Pak for AIOps only (Production)
Master node count	•	•	•	•
Minimum worker node count	3	3	6	6
Total vCPU	47	55	136	162
Total memory (GB)	123	136	310	368

Table 4. Minimum requirements for starter and production deployments, with Red Hat OpenShift

Resource	Base deployment IBM Cloud Pak for AIOps and Red Hat OpenShift (Starter)	Extended deployment IBM Cloud Pak for AIOps and Red Hat OpenShift (Starter)	Base deployment IBM Cloud Pak for AIOps and Red Hat OpenShift (Production)	Extended deployment IBM Cloud Pak for AIOps and Red Hat OpenShift (Production)
Master node count	3	3	3	3
Minimum worker node count	3	3	6	6
Total vCPU	59	67	148	174
Total memory (GB)	171	184	358	416

The Red Hat OpenShift master and worker nodes must meet the minimum size requirements in Hardware requirements - Red Hat OpenShift only.

Note:

- These values do not include CPU and memory resources for hosting a storage provider, such as Red Hat® OpenShift® Data Foundation or Portworx. Storage providers can require more nodes, more resources, or both, to run. The extra resources that are needed can vary based on your selected storage provider. A general recommendation is for one extra worker node for a starter deployment, and for three extra worker nodes for a production deployment. Consult your storage provider's documentation for exact requirements.
- An additional 1 vCPU and 3GB memory is required for each integration that you configure. For example, if you configure two Netcool integrations then you will require an additional 2 vCPUs and 6GB memory.

Warning: Insufficient hardware will result in product instability and loss of function. Verify that your hardware is sufficiently sized for your expected workloads. For more information, see Processing abilities. In addition to the default starter and production size deployments, you can choose to deploy a custom-sized deployment of IBM Cloud Pak for AIOps. For more information, see Custom sizing.

Integrations

When you are configuring an integration in IBM Cloud Pak for AIOps, it is important to consider the performance and footprint. If you have large amounts of data on the selected target system, you can expect relatively higher resource usage from collecting data from that system when compared to a system with less data. Additionally, the sizing of the IBM Cloud Pak for AIOps installation affects the number of resources that are available for running integrations. Review the following considerations for each integration category and the **integration installation** section before you configure a new integration.

Integration installation

IBM Cloud Pak for AIOps integrations for metrics, events, and logs are configured in the Integrations UI page, which can be used to create, edit, delete, and track the status of integrations. Integrations run as pods and have minimum and maximum resource allocations in terms of CPU, memory and storage. Some integrations offer flexibility in terms of how to install the integration in the UI, but some integrations only offer one option.

You can use the following two options to install integrations:

- **Local**: Install the integration in the same cluster and namespace where IBM Cloud Pak for AIOps is installed. The integration's status is displayed in the Integrations UI page and is automatically managed by IBM Cloud Pak for AIOps.
- **Remote**: Install the integration anywhere you choose, for example, a different network region, on SaaS, or remote on-premises (VM, cluster or container). After adding the integration, you can use the script to run the integration pod using podman.

Regardless of whether an integration is a local or remote installation, the data collected by IBM Cloud Pak for AIOps is stored for some time in Kafka.

Note: Before you set up your first integration, make sure that Kafka storage is correctly configured and scaled for the production environment. If the Kafka persistent volume claims (PVCs) is near capacity, increase each of the PVCs by an extra 60 GB. For more information, see Increasing Kafka PVC.

Hardware requirements - Integrations

Before you create an integration for both local and remote installation, make sure that IBM Cloud Pak for AIOps or remote environment contains the hardware resources that are required to run the integrations.

The minimum resource requirements for the integrations are listed in the following table:

Table 1. Minimum requirements for integrations

Integration name	Memory limit (Mb)	CPU limit	Ephemeral storage limit (Mb)

Integration name	Memory limit (Mb)	CPU limit	Ephemeral storage limit (Mb)
AppDynamics New Relic Splunk	2500	1	500
AWS CloudWatch, Dynatrace (metrics-only), Infrastructure Management, Generic Webhook, Zabbix	4096	1	500
Dynatrace (metrics, events, topology)	10000	4	8000
DB2, GitHub, ServiceNow	800	1	1000
Email Notifications, IMPACT, Jira	512	1	1000
Instana	4096	4	1000
Netcool ObjectServer	4096	2	2000
Custom, Elk, Falcon LogScale, Mezmo	1536	1	1000

Notes:

- Dynatrace (metrics, events, topology) supports up to 5 integrations.
- The hardware sizing in the preceding table does not include resource slots for logs integrations or observer jobs. For more information about resource slots for logs integrations, see Performance considerations for logs data collection. For more information about defining observer jobs, see Observer jobs.

For performance considerations for integrations, see the following sections:

- Performance considerations for metrics data collection
- Performance considerations for logs data collection
- Performance considerations for events data collection
- Performance considerations for tickets data collection
- Performance considerations for observers

Requests and limits

In Kubernetes, requests define the resources that must be provided, and limits define the upper boundary of extra resources that can be allocated. Resource quotas can optionally be used to dictate the allowed totals for all requests and limits in a namespace. If a resource quota is set, then it must be at an adequate level to ensure that workload placement is not inhibited. If you do not deploy resource quotas, then skip this section.

The following table shows the requests and limits that the IBM Cloud Pak for AIOps installation namespace require for a typical starter or production deployment.

Table 5. Requests and limits requirements for the IBM Cloud Pak for AIOps installation namespace

Resource name	IBM Cloud Pak for AIOps installation namespace (Starter)	IBM Cloud Pak for AIOps installation namespace (Production)
CPU Request	75	227
Memory Request (Gi)	195	466
Ephemeral Storage Request (Gi)	105	200
CPU Limit	210	436
Memory Limit (Gi)	260	533
Ephemeral Storage Limit (Gi)	340	600

Important: Deployment of additional services impacts these values. Review these values regularly to ensure that they still meet your requirements. Extra resources are required for upgrade, for more information see Ensure cluster readiness.

Use the Red Hat OpenShift command line to create or edit a **`ResourceQuota`** object with these values. For more information, see Quotas in the Red Hat OpenShift documentation.

The following example YAML file show the YAML to create **`ResourceQuota`** objects for the IBM Cloud Pak for AIOps installation namespace in a production deployment of IBM Cloud Pak for AIOps:

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: aiops-large
  namespace: cp4aiops
spec:
  hard:
    requests.cpu: "227"
    requests.memory: 466Gi
    requests.ephemeral-storage: 200Gi
    limits.cpu: "436"
    limits.memory: 533Gi
    limits.ephemeral-storage: 600Gi
    pods: '275'
```

Processing abilities

Expand the following sections to find out about the processing abilities of starter and production sized deployments of IBM Cloud Pak for AIOps. Higher rates are supported by custom sizes. For more information about customizing the size of your IBM Cloud Pak for AIOps deployment according to your processing and footprint requirements, see Custom sizing.

Supported resource number and throughput rates for starter and production deployments

▶ Expand this section for details.

Event, alert, and incident rates

▶ Expand this section for details.

Custom sizing

The default starter and production deployment sizes enable the full capabilities of IBM Cloud Pak for AIOps to be used with the workload volumes that are stated in the Processing abilities section. If different workload volumes are required or resource constraints are an issue, then specific capabilities such as **`Metric Anomaly Detection`**, **`Log Anomaly Detection`**, and **`Runbook Automation`** can be sized accordingly. IBM Sales representatives and Business Partners have access to a custom sizing tool that can assess your runtime requirements and provide a custom profile that scales IBM Cloud Pak for AIOps components. The custom profile is applied when you install IBM Cloud Pak for AIOps. This custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. If you require custom sizing, contact IBM Sales representatives and Business Partners with details of your intended workloads.

Table 7 shows the processing abilities of some custom-sized deployments of IBM Cloud Pak for AIOps.

Example 1 (Minimum scale) represents a minimally sized deployment of IBM Cloud Pak for AIOps for the evaluation of event management. It demonstrates event analytics, noise reduction, and the automation of issue resolution on a small topology. Metric and log anomaly detection and change risk assessment are de-emphasized. Example 1 requires 49 vCPU and 143 GB memory.

Example 2 (Event management focused) represents a production deployment of IBM Cloud Pak for AIOps which is focused on event management capabilities. It supports event analytics, noise reduction, and the automation of issue resolution on a large topology. Metric and log anomaly detection and change risk assessment are de-emphasized. Example 2 requires 176 vCPU and 366 GB memory.

Maximum scale shows the upper limits that IBM Cloud Pak for AIOps can be scaled to, across all of its capabilities.

Table 7. Processing abilities of example custom-sized deployments

Component	Resource	Example 1	Example 2	Maximum scale
Change risk	Incidents and change request records per second	0	0	N/A
Metric anomaly detection	Maximum throughput - (KPIs) for all metric integration	0	0	5,000,000
Log anomaly detection	Maximum throughput (log messages per second) for all log integrations	0	0	25,000
Events (through Netcool integration)	Steady state event rate throughput per second Burst rate event throughput per second	10	600	700 1000
Automation runbooks	Fully automated runbooks run per second	1	2	4
Topology management	Maximum number of topology resources	5,000	5,000,000	15,000,000
UI users	Active users supported	5	40	40

Infrastructure Automation

For more information about the hardware requirements for your Infrastructure Automation deployment on Red Hat® OpenShift® Container Platform, see Hardware requirements.

Performance considerations for metrics data collection

When you configure an integration in IBM Cloud Pak® for AIOps that collects metrics data, such as AppDynamics, AWS CloudWatch, Dynatrace, Instana, New Relic, Splunk, or Zabbix, it is important to consider the amount of data on the target system when you select the metrics.

If you have large amounts of data on a selected target system, for example thousands of hosts, you can expect higher resource usage when you collect data from that system as compared to a system with less data. For target systems with many hosts, consider the following actions to help ensure that your integration works as expected:

- Do not select all the metrics. Instead, review the documentation for the target system that you are pulling data for, understand the metrics and select the metrics that are useful. All metrics are not suited for all the use cases. Select a small, meaningful set of metrics, and expand the selected metrics when you confirm the initial use case.
- Do not select all the hosts for an integration. Some metrics integrations offer the ability to filter by specific hosts or you can define a filter to narrow down the metric data pulled for specific hosts.
- Use filters when possible to reduce the risk of exceeding the limitations of the API or the integration. The filters that are available for each integration are documented in the topic for each integration.
- Each integration needs a unique API token. Each API token has its own requests limit. If integrations share API tokens, then they are at a risk of exceeding the request limits.
- When you configure some integrations in the UI, you can reduce the length of time for historical data collection or reduce the base parallelism to decrease the resource usage of the integration. If multiple metrics are selected, consider collecting data for a time that is less then two weeks.
- Do not create more than 10 metrics integrations on a small cluster or 20 integrations on a large cluster at one time. If multiple integrations are running simultaneously, then more KPIs are sent to Cloud Pak for AIOps. The number of KPIs collected by the metrics integrations must not exceed the limitations for metric anomaly detection. For more information, see Custom sizing.

The following table lists the filters that are available for metric integrations:

Table. Available filters for metric integrations

Metric integration name	Metrics filtering	Host filtering	Namespace filtering	Application filtering	Technology filtering	Index filtering	Management zone	Tags
AppDynamics	Yes	No	No	Yes	No	No	No	No
AWS CloudWatch	Yes	No	Yes	No	No	No	No	No
Dynatrace (metrics-only)	Yes	No	No	No	Yes	No	No	Yes
Dynatrace (metrics, events, topology)	Yes	No	No	No	Yes	No	Yes	Yes
Instana	Yes	No	No	No	Yes	No	No	No
New Relic	Yes	No	No	Yes	No	No	No	No
Splunk	Yes	Yes	No	No	No	Yes	No	No
Zabbix	Yes	Yes	No	No	Yes	No	No	No

Performance considerations for logs data collection

When you configure an integration in IBM Cloud Pak® for AIOps that collects logs data, such as Custom, Elk, Falcon LogScale, Kafka, Mezmo, and Splunk Enterprise, it is important to consider the resource requirements for creating a new integration.

Consider the following actions to help ensure that your integration works as expected:

- Use filters when possible to reduce the risk of exceeding the limitations of the API or integration. The filters available for each integration are documented in the topic for each integration.
- Each integration needs a unique API token. Each API token has its own requests limit. If integrations share API tokens, then they are at a risk of exceeding the request limits.
- Monitor the of the number of logs integrations that are created and the size of the Cloud Pak for AIOps installation. The amount of logs data collected per interval must not exceed the limitations of log anomaly detection. For more information, see Custom sizing.

Cloud Pak for AIOps applies various analytics and transformations to the stream of log data. For scaling to high-volume workloads, the data stream is parallelized using the concept of slots. Each slots consists of a given unit of CPU and memory. Each integration requires one or more slots dependencing on the base parallelism setting, data collection setting, and the data volume. Additional slots are required to support high availability (failover) scenarios, which are available only on large installations of Cloud Pak for AIOps.

When creating a new integration, make sure that there are enough slots on Cloud Pak for AIOps to support the logs integration.

Performance considerations for events data collection

Learn more about the performance considerations for Generic Webhook and Kafka integrations in IBM Cloud Pak® for AIOps.

Generic Webhook integration

When you configure the Generic Webhook integration, it is important to consider the maximum event rate (per second) when you create an integration.

The Generic Webhook integration provides the option to configure a maximum event rate, which can have a minimum of 10 and a maximum of 1000 events per second with a default value of 100. To avoid the rejection of events, scale the maximum event rate if you anticipate exceeding the maximum events rate limit for the integration.

Kafka integration

When you are configuring the Kafka integration, it is important to consider the following performance factors:

- The Kafka replication factor is set to one replica by default. If you are implementing a production deployment of Cloud Pak for AIOps, you might lose data if your Kafka pods fail or restart. If the data collection is enabled in your Kafka integration when the Kafka pods go down, you might experience a gap in the data that your integration generated during the downtime.
- The maximum Kafka message size limit in Cloud Pak for AIOps is 1048588 bytes. If you frequently send in messages that are close to the maximum size, it can adversely impact the performance of the integration. If the maximum message size limit is exceeded, the message is rejected.

Performance considerations for tickets data collection

Learn more about the performance considerations for Jira, GitHub and ServiceNow integrations in IBM Cloud Pak® for AIOps.

When you configure ticketing integrations in Cloud Pak for AIOps, it is important to consider the following performance factors:

- Use filters when possible to reduce the risk of exceeding the limitations of the API or integration. The filters available for each integration are documented in the topic for each integration.
- Each integration needs a unique API token. Each API token has its own requests limit. If integrations share API tokens, then they are at a risk of exceeding the request limits.
- If the GitHub or Jira integrations exceed API rate limits from the target system, there might be a delay in data collection. The GitHub or Jira integrations implement a pause and wait strategy until the API rate limit is refreshed before making additional requests.
- If available in the UI for the integration that you are configuring, you can adjust the data collection interval to reduce the resource usage of the integration.

Performance considerations for observers

When you configure observer integrations in IBM Cloud Pak® for AIOps, then use the following considerations:

- Observer filtering capabilities need to be used where possible to help fine-tune what the integration provides to Cloud Pak for AIOps. This approach ensures that Cloud Pak for AIOps focuses on managing only what is necessary to deliver value, while maximizing efficiency in both data source and data processing.
- Observer data is sent to a Kafka topic for processing by the topology-service microservice, but the processing rate is significantly lower than the production rate, causing data to accumulate in the Kafka topic. If large observer jobs are run in parallel, processing might exceed Kafka's 8-hour default message retention time, leading to data loss, discarded control plane messages, and stuck jobs. To mitigate these risks, it is recommended to run large observer jobs sequentially and moderate the number of jobs that run in parallel.
- Merge, group processing and populating the inventory database can continue after a given observer job is marked as finished, which indicates that the topology microservice has consumed and processed the data.
- If you are using the File observer, files should be constrained to 1m resources to keep the quantity of tracked metadata and processing times manageable.
- Observer load jobs automate the process of determining the differences between subsequent loads of the same data. The system simplifies the authoring of a data source by automatically handling changes between loads, such as resources A,B,C being provided at time 1 and resources A,C,D being provided at time 2. The observer downloads the set of identifiers for the previous run from the topology service REST APIs at job start-up and compares the IDs for time 1 and 2. The system automatically deletes any IDs that are not found in time 2, eliminating the need for a linger concept. The system resurrects resource B if it reappears at time 3.
- If you are using the File or REST observers (or writing a new one), then consider the resource properties that are required to meet business goals. It is not useful to provide large number of properties if a better experience and more up-to-date data can be obtained through federation to the source by using a right-click contextual tool launch. Otherwise, every property needs to be stored and managed, which can slow the processing down. In extreme cases, Kafka might discard the message from the observer as the maximum message size is 1.5 Mb.
- If you are using the File or REST observers (or writing a new one), consider the number of edges (relationships) that each resource contains to avoid the supernodes. A supernode in a topology graph is a node with disproportionately high number of relationships. It is recommended to exercise caution when you are exceeding 400 relationships incident to a given resource. The reason is that the user experience can suffer. In extreme cases, such as thousands of relationships, Kafka might discard the message from the observer as the maximum message size of Kafka is 1.5 Mb. Relationships in topology manager can accumulate across individual contributions to a resource by using merge processing.
- If you are using the File or REST observers (or writing a new one), consider the order in which data is provided to the Topology Manager. For example, the order of lines in a file or the sequence in which the **`applyVertex`** method is executed in a COTS observer. Topology Manager contains the logic to handle out-of-sequence data and reassemble it, but this can consume the memory and reduce performance. Consider the case where resource A is related to resources B, C, and D. If A is sent first with relationships to B, C and D as part of it, then placeholders need to be tracked for B, C, and D until the full records arrive. In extreme cases, tens or hundreds of thousands of records might be generated after A is first seen. To minimize placeholder tracking, it is recommended to send B, C, and D before A, if possible. Alternatively, relate B, C, and D to A by using a relationship from/to each of those for A. To increase the likelihood that the resources that are referenced without relationships are previously sent to the topology service, you can generate a sorted list of resources in Python by using the following code:

  ```
  for resource in sorted(topology, key=lambda k: (len(topology[k]['_references']), topology[k]['uniqueId'])):
  ```

  This code sorts the resources by the number of relationships and unique ID, ensuring that resources with fewer relationships are sent first.
- When providing data to the topology service, you choose the operation to perform such as, **`insert-replace`**, **`insert-update`**, or **`delete`**. The topology service performs the chosen operation. **`Insert-replace`** is the simplest and most reliable method, as it does not require advanced user knowledge to use correctly. Incorrect use of **`insert-update`** might result in a topology with decreasing accuracy over time.
- Observers can provide groups to Topology Manager automatically. Consider the meaning of group to the business, the number of groups created and the number of resources within each group. Groups over 2,000 resources can result in a poor user experience. Defining groups by using templates is recommended, as it provides greater administrative control over the creation of groups, including how, when, and why they are created. If a group must be created by using an observer directly, then provide an option to suppress or filter the groups that it creates.

Storage requirements

Learn about the storage requirements for IBM Cloud Pak® for AIOps.

- Persistent storage requirements
- Persistent storage sizing
- Recommended storage providers

Persistent storage requirements

IBM Cloud Pak for AIOps requires persistent storage that supports the RWO (read-write-once) access mode. Use of the Red Hat OpenShift APIs for Data Protection (OADP) backup and restore method requires persistent storage that supports the RWX (read-write-many) access mode. For more information about the backup and restore methods available for IBM Cloud Pak for AIOps, see Back up and restore (IBM Cloud Pak for AIOps on OpenShift).

Important: The file system used by MinIO must be `XFS`, not `ext4`. The `ext4` file system has a limit on the number of inodes that can be created for each file system. If inode usage reaches 100%, the file system becomes read-only even if enough PVC space is available and MinIO is prevented from creating new files or directories. Refer to your storage provider's documentation for information about setting `XFS` as the file system.

Storage class requirements

For production environments, storage classes must have `allowVolumeExpansion` enabled. This allows persistent volumes to be expanded if necessary, to avoid storage from filling up and causing unrecoverable failures. This is also highly recommended for starter deployments, since without it you are limited to the default capacity that might not be sufficient for your specific needs. To enable `allowVolumeExpansion`, edit the storage class to enable expansion. Follow the instructions in the Red Hat documentation Enabling volume expansion support.

IOPS requirements

Table 1 identifies the input/output operations per second (IOPS) per node that must be supported for a 3-node storage solution. Ensure that your hardware can support the expected IOPS per node. A minimum configuration of three nodes for the storage cluster is needed. Each node of the storage solution requires a minimum of one disk (SSD or high-performance storage array).

Table 1. IOPS requirements

Deployment size	Minimum	Recommended
Starter deployment	2,000	5,000
Production deployment	10,000	20,000

The results for your storage can vary depending on your exact usage, data sets, hardware, storage solution, and more. The preceding IOPS minimum and recommended values are based on a Red Hat® OpenShift® Data Foundation storage solution.

Persistent storage sizing

Red Hat OpenShift uses the Kubernetes persistent volume (PV) framework. PVs are storage resources in the cluster, and persistent volume claims (PVCs) are storage requests that are made on those PVs by IBM Cloud Pak for AIOps. For more information about persistent storage in Red Hat OpenShift clusters, see Persistent storage overview.

A starter deployment requires 463 Gi of persistent storage, and a production deployment requires 1928 Gi of persistent storage.

Table 2. Persistent storage sizing

Component	Starter deployment PVC size (Gi) Total (number of PVCs x PVC size)	Production deployment PVC size (Gi) Total (number of PVCs x PVC size)	Access mode	Storage mode
Cassandra (data and backup)	100 (2x50)	300 (6x50)	RW0	Block
Common service db	20 (2x10)	40 (4x10)	RWO	Block
Topology file observer	5 (1x5)	5 (1x5)	RWO	Filesystem
CouchDB	20 (1x20)	60 (3x20)	RWO	Filesystem
Elasticsearch	100 (1x100)	300 (3 x 100)	RWO	Block
Kafka	60 (1x60)	300 (3x100)	RWO	Block
Zookeeper	10 (1x10)	30 (3x10)	RWO	Block
Redis	3 (3x1)	3 (3x1)	RWO	Block
AIOps MinIO	50 (1x50)	500 (5X100)	RWO	Block
Zen MinIO	30 (3x10)	30 (3x10)	RWO	Block
EDB Postgres	15 (1x15)	300 (3x100)	RWO	Block
CS Postgres backup	10 (1x20)	20 (1x20)	RWO	Block
Zen Objectstore backup	20 (1x20)	20 (1x20)	RWO	Block
Zen Metastoredb EDB	20 (2x10)	20 (2x10)	RWO	Block
Total	463	1928		

The preceding table provides PVC sizings that provide adequate space to begin working with IBM Cloud Pak for AIOps, but you may need to increase these values later. For more information, see Resizing storage.

Notes:

- The access mode is ReadWriteOnce (RWO) for each PV. Usage of OADP backup and restore requires ReadWriteMany (RWX) access mode. For more information about access modes, see Access modes.
- Depending on the size of the training sets, usage, and data retention needs, your deployment can require additional storage.
- The storage type is shown for each of the PVs, and is either Filesystem or Block. For more information about persistent storage, see the Red Hat documentation about Persistent storage overview.

Additional storage requirements for offline (airgap) deployments

If you are installing in an air-gapped environment (offline), you must ensure that the target registry and any intermediary portable devices have adequate space to download the IBM Cloud Pak for AIOps images. The IBM Cloud Pak for AIOps images total 140 GB.

If you are also going to install IBM Cognos® Analytics, then you will require an additional 155 GB.

Additional storage requirements for backup and restore

The backup and restore of IBM Cloud Pak for AIOps is based on the storage and database backup of critical data stores, cluster resources, and component data and settings, such as issue resolution policy settings, Runbooks, automated actions, metric data, log anomaly models, related secrets and configmaps, and more. With the backup and restore feature, you can recover your services to a previous point in time backup to recover from data corruption, system failures, or user errors.

To support backing up your IBM Cloud Pak for AIOps environment, you need to provision storage specifically for storing and retrieving backups. This storage also needs to be accessible by any environment where a backup might need to be restored. The backup process for IBM Cloud Pak for AIOps requires S3 compliant object stores for storing and retrieving backups. For instance, you can use the following public or private cloud options for provisioning this storage:

- MinIO
- Red Hat OpenShift Data Foundation object store
- IBM Cloud Object Storage
- AWS Cloud

Tip: As your data grows, the size of your backup storage might need to grow. Ensure that sufficient storage is provisioned to accommodate any expected growth in data size. For more information about the backup and restore process, see Backing up and restoring IBM Cloud Pak for AIOps.

Recommended storage providers

The recommended storage providers are:

- IBM Cloud
- Red Hat OpenShift Data Foundation (ODF)
- IBM Fusion Data Foundation
- IBM Fusion Global Data Platform
- IBM Storage Scale Container Native
- Portworx
- AWS native storage

The following table shows the tested and supported storage providers for the platforms that IBM Cloud Pak for AIOps can be deployed on.

Table 2. Supported storage providers by platform

Platform	IBM Cloud Storage (Block and File)	Red Hat® OpenShift® Data Foundation	IBM Fusion Data Foundation	IBM Fusion Global Data Platform	IBM Storage Scale Container Native	Portworx	AWS native storage
Azure Red Hat OpenShift (ARO)		Yes				Yes	
Google Cloud Platform (GCP)		Yes				Yes	
Red Hat OpenShift Container Platform		Yes	Yes	Yes	Yes	Yes	Yes
Red Hat OpenShift on IBM Cloud (ROKS)	Yes	Yes	Yes	Yes	Yes		
AWS GovCloud						Yes	
Red Hat OpenShift Service on AWS (ROSA)						Yes	Yes

Notes:

- IBM Storage Scale Container Native and Red Hat® OpenShift® Data Foundation are part of IBM Storage Suite for IBM Cloud Paks.
- IBM Cloud Pak for AIOps requires persistent RWX (read-write-many) storage for OADP backup and restore. Red Hat® does not currently support Red Hat® OpenShift® Data Foundation (ODF) on ROSA. Portworx is the only recommended storage provider that provides a RWX storage solution for a deployment of IBM Cloud Pak for AIOps on ROSA. Portworx is available for a free 30 day trial, but will require a license for a longer timespan and production usage.

The preceding storage providers are the only providers that are tested and validated for a deployment of IBM Cloud Pak for AIOps. You can choose to use an alternate storage provider if they meet the requirements for deploying IBM Cloud Pak for AIOps. Your chosen storage provider must meet the same storage and hardware requirements as the recommended storage providers. For instance, for deploying IBM Cloud Pak for AIOps, your chosen provider must support the required access modes and storage modes. If you choose to use an alternate storage provider, your overall performance can differ from any listed sizings, throughput rates or other performance metrics that are listed in the IBM Cloud Pak for AIOps documentation. Work with your IBM Sales representative (or Business Partner) to ensure that your chosen storage provider is sufficient for your deployment plan.

IBM Cloud

The following storage classes are required by IBM Cloud Pak for AIOps, and are created when Red Hat OpenShift on IBM Cloud (ROKS) is installed:

Storage provider	Storage class name	Large block storage class name
IBM Cloud® (without OADP backup and restore)	ibmc-block-gold	ibmc-block-gold
IBM Cloud® (with OADP backup and restore)	ibmc-file-gold-gid	ibmc-block-gold

Installing IBM Cloud storage

For more information about IBM Cloud storage, see Storing data on classic IBM Cloud File Storage and Storing data on classic IBM Cloud Block Storage in the IBM Cloud Docs.

Red Hat OpenShift Data Foundation (ODF)

The following storage classes are required by IBM Cloud Pak for AIOps, and are created when Red Hat OpenShift Data Foundation is installed:

Storage provider	Storage class name	Large block storage class name
Red Hat® OpenShift® Data Foundation (without OADP backup and restore)	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
Red Hat® OpenShift® Data Foundation (with OADP backup and restore)	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd

Installing Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation is available for purchase through the IBM Storage Suite for IBM Cloud Paks. Red Hat OpenShift Data Foundation is an implementation of the open source Ceph storage software, which is engineered to provide data and storage services on Red Hat OpenShift. Use version 4.12 or higher.

For more information about deploying Red Hat OpenShift Data Foundation, see Deploying OpenShift Data Foundation in the Red Hat documentation. Choose the appropriate deployment instructions for your deployment platform.

Important: Red Hat OpenShift Data Foundation provide additional configuration steps, which can alleviate potential problems with delayed space reclamation. For more information, see Reclaiming space on target volumes in the Red Hat OpenShift Data Foundation documentation.

These additional configuration steps are not specific to IBM Cloud Pak for AIOps, and are required for any application that experiences a high volume of writes that are then deleted. For IBM Cloud Pak for AIOps, this might be an event storm where large numbers of events are ingested and resolved, or when a large quantity of short-term metric data is ingested and then deleted. Failure to properly configure storage can cause out of disk space errors that might not be readily apparent. In some scenarios, you might need to contact your storage provider to ensure proper storage configuration for your needs.

IBM Fusion Data Foundation

The following storage classes are required by IBM Cloud Pak for AIOps:

Storage provider	Storage class name	Large block storage class name
IBM Fusion Data Foundation (without OADP backup and restore)	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation (with OADP backup and restore)	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd

Installing IBM Fusion Data Foundation

IBM Fusion Software version 2.9.0 and IBM Fusion HCI Systems version 2.9.x are compatible with IBM Cloud Pak for AIOps.

To learn more about IBM Fusion and how to install IBM Fusion Data Foundation service, see the following two methods:

IBM Fusion Software

IBM Fusion Data Foundation is a highly integrated collection of cloud storage and data services for Red Hat OpenShift Container Platform. It is available as part of the Red Hat OpenShift Container Platform service catalog, packaged as an operator to facilitate simple deployment and management.

The IBM Fusion Software provides the following features:

- IBM Fusion software is a software-only solution that can be deployed on a variety of hardware platforms.
- IBM Fusion software is designed to scale up to meet the needs of larger environments.
- IBM Fusion software can be integrated with a variety of third-party solutions.

Before you install IBM Fusion, ensure that you meet all of the prerequisites. For more information, see Prerequisites.

For more information about Deploying IBM Fusion, see Deploying IBM Fusion.

For more information about how to install Data Foundation service using IBM Fusion, see Data Foundation.

For more information about IBM Fusion Software storage class, see Data Foundation.

IBM Fusion HCI Systems

The IBM Fusion Data Foundation service provides a foundational data layer for applications to function and interact with data in a simplified, consistent, and scalable manner.

The IBM Fusion HCI Systems provides the following features:

- IBM Fusion HCI system is a pre-integrated, pre-tested, and pre-configured appliance that combines hardware and software.
- IBM Fusion HCI system is designed to scale out to meet growing storage needs.
- IBM Fusion HCI system is tightly integrated with IBM's other hybrid cloud solutions, such as IBM Cloud and IBM Power Systems.

For more information about Prerequisites of IBM Fusion HCI system, see Planning and prerequisites.

For more information about Deploying IBM Fusion HCI System, see Deploying IBM Fusion HCI System.

For more information about how to install Data Foundation service using IBM Fusion HCI Systems, see Data Foundation.

For more information about IBM Fusion HCI Systems storage class, see Data Foundation.

IBM Fusion Global Data Platform

The following storage classes are required by IBM Cloud Pak for AIOps:

Storage provider	Storage class name	Large block storage class name
IBM Fusion Global Data Platform: IBM Fusion	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
IBM Fusion Global Data Platform: IBM Fusion HCI System	ibm-storage-fusion-cp-sc	ibm-storage-fusion-cp-sc

Installing IBM Fusion Global Data Platform

The Global Data Platform storage type provides the following features:

- File storage
- High availability via capacity-efficient erasure coding
- Metro and regional disaster recovery
- CSI snapshot support with built-in application consistency
- Encryption at rest
- Ability to mount file systems hosted by remote IBM Storage Scale clusters.

Important: IBM Fusion Global Data Platform provide additional configuration steps, which can alleviate potential problems with delayed space reclamation. For more information, see Enabling reclaim space operation using Annotating PersistentVolumeClaims in the IBM Fusion documentation.

These additional configuration steps are not specific to IBM Cloud Pak for AIOps, and are required for any application that experiences a high volume of writes that are then deleted. For IBM Cloud Pak for AIOps, this might be an event storm where large numbers of events are ingested and resolved, or when a large quantity of short-term metric data is ingested and then deleted. Failure to properly configure storage can cause out of disk space errors that might not be readily apparent. In some scenarios, you might need to contact your storage provider to ensure proper storage configuration for your needs.

You can deploy Global Data Platform service by using the following two methods:

IBM Fusion Software

Before you install IBM Fusion, ensure that you meet all of the prerequisites. For more information, see Prerequisites.

For more information about Deploying IBM Fusion, see Deploying IBM Fusion.

For more information about how to install Global Data Platform using IBM Fusion Software, see Global Data Platform.

For more information about IBM Fusion Software storage class, see IBM Storage Scale.

IBM Fusion HCI Systems

For more information about Prerequisites of IBM Fusion HCI system, see Planning and prerequisites.

For more information about Deploying IBM Fusion HCI System, see Deploying IBM Fusion HCI System.

For more information about how to deploy Global Data Platform service using IBM Fusion HCI Systems, see Global Data Platform.

For more information about IBM Fusion HCI Systems storage class, see IBM Storage Scale.

IBM Storage Scale Container Native

The following storage class is required by IBM Cloud Pak for AIOps, and is created when IBM Storage Scale Container Native is installed.

Storage provider	Storage class name	Large block storage class name
IBM Storage Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc

This class is used as both the ReadWriteMany storage class and large block storage class when you are installing IBM Cloud Pak for AIOps. This StorageClass includes a parameter to set permissions of data within the StorageClass to **`shared: true`**, which is required to support the Kubernetes SubPath feature. For more information about the permissions field for the class, see the IBM Spectrum Scale CSI Driver documentation.

Installing IBM Storage Scale Container Native

IBM Storage Scale Container Native is available for purchase through the IBM Storage Suite for IBM Cloud Paks. IBM Spectrum Scale is a cluster file system that provides concurrent access to a single file system or set of file systems from multiple nodes. The nodes can be SAN-attached, network attached, a mixture of SAN-attached and network attached, or in a shared nothing cluster configuration. This enables high-performance access to this common set of data to support a scale-out solution or to provide a high availability platform. IBM Storage Scale Container Native Storage Access must be at version 5.1.1.3 or higher, with IBM Spectrum Scale Container Storage Interface version 2.3.0 or higher.

For more information, see the IBM Storage Scale Container Native documentation.

To use IBM Storage Scale Container Native with IBM Cloud Pak for AIOps, you do not require a separate license. You can use up to 12 TB of IBM Storage Scale Container Native storage for up to 36 months, fully supported by IBM, within your production environments (Level 1 and Level 2). If you exceed these terms, a separate license is required.

To install and use IBM Storage Scale Container Native, your cluster must meet the requirements in the IBM Storage Scale Container Native documentation.

To install IBM Storage Scale Container Native and the IBM Spectrum Scale Container Storage Interface, follow the IBM Storage Scale Container Native installation documentation.

If you encounter any errors or issues with installing or using IBM Storage Scale Container Native or the IBM Spectrum Scale Container Storage Interface, refer to the following documentation:

- IBM Storage Scale Container Native
- IBM Spectrum Scale Container Storage Interface

Portworx

The following storage classes are required by IBM Cloud Pak for AIOps, and must be created using the instructions in the following section:

Storage provider	Storage class name	Large block storage class name
Portworx	px-csi-aiops	px-csi-aiops
Portworx (multi-zone HA)	px-csi-aiops-mz	px-csi-aiops-mz

Installing Portworx

Important: `sharedv4` volumes require NFS ports to be open. For more information, see the Portworx documentation Open NFS Ports.

1. Install the Portworx operator and configure a Portworx StorageCluster.

   Two editions of Portworx are available: Portworx Enterprise and Portworx Essentials. Portworx Enterprise is suitable for production deployments. Portworx Essentials is suitable only for demonstration deployments, as it can be used only on clusters of five nodes or less, and includes storage size limits. For more information, see the Portworx documentation Installing Portworx on Openshift.

   Note: You must be a cluster administrator to install Portworx on the cluster.

2. Define a custom Portworx storage class.

   The custom Portworx storage class will be used for file and block storage. The storage class is scoped to the cluster, so setting the project (namespace) is not required.

   If you are not using the Multi-zone HA technical preview, then use Option 1 to define **`px-csi-aiops`**. If you are using the Multi-zone HA technical preview, then use Option 2 to define **`px-csi-aiops-mz`**.

   Option 1: Define `px-csi-aiops`

   Log in to your OpenShift cluster's console. Click the plus icon on the upper right to open the **Import YAML** dialog box, paste in the following content, and then click **Create**.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: px-csi-aiops
provisioner: pxd.portworx.com
parameters:
  fs: xfs
  io_profile: db_remote
  repl: '2'
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

   Option 2: Define `px-csi-aiops-mz`

   Log in to your OpenShift cluster's console. Click the plus icon on the upper right to open the **Import YAML** dialog box, paste in the following content, and then click **Create**.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: px-csi-aiops-mz
provisioner: pxd.portworx.com
parameters:
  fs: xfs
  io_profile: db_remote
  repl: '3'
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

AWS native storage

The following storage classes are required by IBM Cloud Pak for AIOps. **`gp3-csi`** is created for you, but you must create **`efs-sc`** using the instructions in the following section.

Storage provider	Storage class name	Large block storage class name
AWS native storage (without OADP backup and restore)	gp3-csi	gp3-csi
AWS native storage (with OADP backup and restore)	efs-sc	gp3-csi

Amazon Elastic Block Store (EBS) provides block storage - the storage class is **`gp3-csi`**, and is created when ROSA is installed. Amazon Elastic File System (EFS) provides file storage - the storage class is **`efs-sc`**. You must create the **`efs-sc`** storage class using the instructions in the following section.

Installing AWS native storage

For more information about Amazon Web Services (AWS) native storage, see Cloud Storage on AWS in the AWS documentation.

For more information, see **`Setting up the AWS EFS CSI Driver Operator`** and **`Creating the AWS EFS storage class`** in the Red Hat OpenShift documentation. You can also review the information in the Red Hat article Enabling the AWS EFS CSI Driver Operator on ROSA.

Network requirements

IBM Cloud Pak for AIOps uses ports to establish communication within OpenShift Container Platform and to the external integrations. Review the list of ports that need to be exposed for use in inbound and outbound communication with IBM Cloud Pak for AIOps.

Inbound communication ports

Inbound communication ports are used to establish connection from external sources to IBM Cloud Pak for AIOps capabilities. IBM Cloud Pak for AIOps exposes the following ports to achieve this communication.

- *TCP 80*: Default HTTP port.
- *TCP 443*: Default HTTPS port.
- *TCP 4100*: Used for Tabular Data Stream to connect to the on-prem IBM Tivoli Netcool/OMNIbus.

Notes:

- TCP 80 and TCP 443 ports must be available on all nodes, including master, worker, and storage nodes. These are the standard ports for HTTP/HTTPS inbound connections as defined by the OpenShift Container Platform routes. OpenShift Container Platform routes provide an abstraction for an external access so you do not have to deal directly with load balancers, inbound controllers, and other networking components.
- Cloud Pak for AIOps pods use only the standard ports such as port 80 or port 443 as defined by OpenShift Container Platform routes for HTTP/HTTPS inbound connections. The inbound connections from external sources can include browsers for accessing the console, an external Kafka client that needs to place data on a Kafka topic, Slack inbound data, and more.
- TCP 4100, TCP 16311, and TCP 9081 ports must be available on the worker nodes.

Connect to externally hosted probes

If you plan to connect to externally hosted probes to ObjectServers in Cloud Pak for AIOps, NodePorts are required. You can map any available port from the range of permissible ports for NodePort services to the target 4100 port. Hence, the port that matters from an inbound network perspective is the port that is used in the NodePort definition.

For more information about external probes that connect to ObjectServers (primary and backup), see Configuring the Cloud system.

Outbound communication ports

Cloud Pak for AIOps needs access to outbound ports to establish connection for probes and integrations.

- **IBM Tivoli Netcool/OMNIbus integration**

  For Netcool/OMNIbus integration, the bulk of data transfer is usually done over Tabular Data Stream port 4100 on the Netcool side. Cloud Pak for AIOps is able to establish a connection to 4100 port from each of the worker nodes. The Netcool/OMNIbus integration uses the IDUC port. IDUC indicates which alerts are inserted, deleted, or updated since the last IDUC cycle, which allows the integration to know what to request. This integration initially connects to ObjectServer by using Tabular Data Stream and then ObjectServer announces the host and port to use for IDUC. Therefore, firewall rules allow connectivity to the IDUC port.

  Notes:

  - If a firewall prevents Cloud Pak for AIOps connecting to Netcool on your VMs, you need to update the firewall rules to correct this situation.
  - The Netcool admin can modify the default 4100 port. It is recommended to engage the Netcool admin so that the admin can confirm all the ports for the Netcool components, such as ObjectServers, and Netcool/Impact.

  For more information about Netcool/OMNIbus to Cloud Pak for AIOps integration, see Creating IBM Netcool Operations Insight ObjectServer integrations.

  For more information about IDUC port, see the following links:

  - Set up the Netcool OMNIbus ObjectServer IDUC port.
  - Specifying the IDUC port
- **IBM Tivoli Netcool/Impact integration**

  The default ports for Netcool/Impact integration are 16311 and 9081. The 16311 port is used to access the GUI and the 9081 port is used to access the Netcool/Impact API.

  Netcool/Impact admin can decide to use different ports. It is recommended to get all the ports directly from admin in case the default ports have changed.

  For more information about Netcool/Impact to Cloud Pak for AIOps integration, see Creating IBM Tivoli Netcool/Impact integrations.

- **Probes**

  For probes that send events directly to Cloud Pak for AIOps, the IDUC port that is specified for the on-prem ObjectServer configuration needs to allow connectivity for Cloud Pak for AIOps. This IDUC port needs to be accessible from each of the worker nodes.

  In environments where a firewall is enabled between the servers or where iptables are configured on the servers, the `ncp_g_event` process can connect to the main ObjectServer listening port but might fail to connect to the IDUC port. While the main ObjectServer listening port uses a fixed port, the IDUC listening port is randomly selected by default. A fixed IDUC listening port must be specified to define firewall rules or iptables settings that allow IBM Tivoli Network Manager to connect to the ObjectServer on the IDUC listening port.

  For more information about IDUC port, see the following links:

  - Set up the Netcool OMNIbus ObjectServer IDUC port.
  - Specifying the IDUC port
- **Other integrations**

  For integrations such as Netcool, ELK, Splunk, Humio, Mezmo and topology observers, Cloud Pak for AIOps uses a PULL model.

  The configurations of these integrations (endpoint exposed by the external source) dictate the port and connection parameters that Cloud Pak for AIOps uses to pull data, such as event and log from those external systems. The port number that is used by that external system depends on how the external integration was configured. Cloud Pak for AIOps does not dictate the value of port number. The outbound ports for each of these integrations need to be open on each of the worker nodes.

In some cases, these integrations can exist within the same intra-network where Cloud Pak for AIOps is installed. In other cases, you might be using a SaaS offering of the integrations (such as Instana or New Relic). For such cases, if you do not want to allow direct outbound access from the OpenShift Container Platform cluster to the SaaS service, you can use the Secure Tunnel approach. This approach requires a VM with internet access, which allows a VM to connect to the SaaS instance.

For more information about the Secure Tunnel approach, see Secure Tunnel.

For more information about other integrations, see the following topics.

  - Dynatrace Events integration
  - Dynatrace Topology integration
  - Splunk Events integration
  - SevOne Events integration
  - SevOne Topology integration
  - ITNM Topology integration

- **Internal communication within the OpenShift Container Platform cluster where Cloud Pak for AIOps is installed**

  Cloud Pak for AIOps internally uses other ports such as 8080, 8383 and 8686 for communication between its own pods, which run on the same OpenShift Container Platform cluster.

High Availability (HA) considerations

You can set up high availability (HA) for your IBM Cloud Pak® for AIOps deployment by selecting a production (large) sizing profile.

IBM Cloud Pak for AIOps

HA configures your deployment with extra replication of its services, and affinity rules to provide redundancy. A HA configuration allows your deployment to tolerate upgrades and service interruptions to a portion of the infrastructure. Failures in the following domains are minimized:

- individual pod failure: Red Hat OpenShift recovers individual failed pods.
- cluster node failure (master or worker): Red Hat OpenShift recovers non-stateful deployments.

For a high availability deployment, you must set a value of **`large`** in your IBM Cloud Pak for AIOps **`Installation`** custom resource when you create your IBM Cloud Pak for AIOps instance. Sizing is configured in your IBM Cloud Pak for AIOps installation custom resource with the parameter **Capacity** (Red Hat® OpenShift® Container Platform UI) or **spec.size** (Red Hat OpenShift CLI), and can be set to **`small`** for starter deployments or **`large`** for production deployments.

Multi-zone HA

A technology preview of installing IBM Cloud Pak for AIOps on a multi-zone architecture is also available. For more information, see Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HA).

Multi-region cold standby disaster recovery

A multi-region cold standby disaster recovery solution for IBM Cloud Pak for AIOps is also available. For more information, see Multi-region cold standby disaster recovery

Infrastructure Automation

For more information about high availability (HA) considerations for your Infrastructure Automation deployment, see High Availability (HA) considerations.

Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HA)

BETA The use of the multi-zone high availability (HA) deployment architecture for IBM Cloud Pak for AIOps is available only for technology preview, and is not for production usage.

Review the following sections:

- Overview
- Installing on a multi-zone architecture
- Limitations while workloads are rescheduled

Overview

IBM Cloud Pak® for AIOps is starting to implement multi-zone availability, as provided by the Red Hat® OpenShift® Container Platform. This feature is not available for deployments of IBM Cloud Pak for AIOps on Linux®.

Installing IBM Cloud Pak for AIOps on a Red Hat OpenShift cluster that spans multiple availability zones improves resiliency. The distribution of pod replicas in different zones means that if a zone becomes unavailable, then remaining replicas in other zones are still able to service requests. Red Hat OpenShift is also able to reschedule failed nodes' workloads from a zone that is subject to an outage to nodes in an alternative healthy zone.

For more information about the benefits of multi-zone architectures, see Running in multiple zones in the Kubernetes documentation.

Red Hat OpenShift tolerates nodes (and by extension zones) being down for five minutes, in addition to a node monitoring interval. For that reason, and accounting for the additional image pull and pod startup times, node and zone failures take approximately six minutes to start recovering. For more information, see Using remote worker node at the network edge: Kubernetes zones in the Red Hat OpenShift documentation.

Note: Multi-zone availability is only supported for production deployments of IBM Cloud Pak for AIOps. A starter deployment of IBM Cloud Pak for AIOps can be deployed on a multi-zone architecture, but it will not be highly available.

Installing on a multi-zone architecture

Use the following steps to install IBM Cloud Pak for AIOps on a multi-zone architecture.

1. Prepare your cluster

   Before you install IBM Cloud Pak for AIOps, you must ensure that your Red Hat OpenShift cluster is configured for multi-zone by labeling the cluster nodes with a zone. Cluster administrator permission is required to add labels to your nodes.

   If you are installing IBM Cloud Pak for AIOps on a cloud platform, then refer to your cloud provider's documentation to learn how to select a multi-zone type when you are setting up your cluster, and then skip to step 2, Configure storage.

   If you are installing directly on Red Hat OpenShift on-premises, then label your Red Hat OpenShift cluster nodes as follows.

   1. Label cluster nodes with zones.

      Distribute your cluster nodes across three availability zones, ensuring that you have a cluster master node in each availability zone. Allocate each node to an availability zone by labeling the node with the zone:

      ```
      oc label node <node-name> topology.kubernetes.io/zone=<zone-name>
      ```

      Where

      - `<node-name>` is the fully qualified domain name of your node, for example *worker0.mycluster.subdomain.example.com*.
      - `<zone-name>` is the name of the zone that you want the node to be in.

   2. Label cluster nodes with regions.

      IBM Cloud Pak for AIOps has a dependency that requires the Kubernetes region label to be set on each node. The region label can be set to the same value on all nodes.

      ```
      oc label node <node-name> topology.kubernetes.io/region=<region-name>
      ```

      Where

      - `<node-name>` is the fully qualified domain name of your node, for example *worker0.mycluster.subdomain.example.com*.
      - `<region-name>` is the region name.

   When the `topology.kubernetes.io/zone` label is specified for the nodes in a cluster, Kubernetes tries to schedule a ReplicaSet's pods on different worker nodes in different zones. For example, a deployment of three replicas should result in one replica being deployed in each zone of a three zone cluster.

2. Configure storage

   Refer to your storage provider's documentation for instructions on configuring storage classes for a multi-zone configuration that enables data to remain available if a cluster node becomes unavailable.

3. Install IBM Cloud Pak for AIOps

   Starter deployments are not configured to support multi-zone HA. Use the following instructions to install a production installation of IBM Cloud Pak for AIOps.

   Installing IBM Cloud Pak for AIOps (Production installation)

   IBM Cloud Pak for AIOps uses a preferred scheduling approach instead of a hard cluster topology requirement. This approach provides greater administrative flexibility and helps IBM Cloud Pak for AIOps to remain functional despite any cluster topology issues. Red Hat OpenShift distributes pods on the nodes according to node availability and other considerations such as available resources, node taints, or whether an image already exists on a node. These considerations might sometimes conflict.

   The installation procedure has an optional step to run a status checker script. This script reports whether pods are optimally distributed for resilience within your multi-zone cluster. Manual intervention might sometimes be necessary to move pods to the correct zone. The deletion of one or more incorrectly placed pods is usually sufficient, but it may be necessary to cordon nodes in a zone to help direct Red Hat OpenShift to distribute the pods to the proper zone. For more information about rebalancing, see Rebalancing workloads on a multi-zone architecture (multi-zone HA).

   Note: The Secure Tunnel function is enhanced to better handle zone outages. If you use Secure Tunnel, when you configure connector and tunnel worker replicas, ensure that the replica count is equal to or greater than the number of zones in your environment. The number of replicas is preferably a multiple of the number of zones. For example, if your environment has three zones then you might configure two replicas for each zone, to give a total of six replicas. For more information about configuring connector and tunnel replicas, see Creating Secure Tunnel connections.

Some resources might not failover as expected, and might require intervention. Contact IBM Support if you encounter any problems when you are using this technology preview.

Limitations while workloads are rescheduled

If the following functions are affected by a zone outage, then they are compromised until Red Hat OpenShift reschedules the affected workloads to nodes in an unaffected zone. This takes approximately six minutes.

- Integrations do not collect or forward data.

- Alerts, Events, and Incidents from IBM® Netcool® Operations Insight® probes are paused.
- IBM® Netcool® Agile Service Manager observers do not run.
- AI model management UI is not accessible.
- AI training may be paused and there is no new training of selected analytics.
- You cannot view, edit, delete, create, or deploy integrations.
- Insights Dashboard is not accessible.
- Resource enrichment functionality, which matches resources to resources, updates status, and adds queriable tags is paused.
- The matching of resources to events is paused.
- Timed jobs such as alert expiration and closure do not run.
- Netcool Operations Insight probe data archiving is paused.
- Secure tunnel controller pod - changes to the tunnel connections or application mappings fail. If one of the connection worker pods is not available for Secure Tunnel to use then performance might be impacted, depending on the current workload.
- Runbook Automation: SSH actions and HTTP actions that are running might lose connection to their target if the outage affects the pod that is running them. In this scenario, no further output is received from the target. For SSH actions, the execution of the remote script on the target machine stops. Ansible actions are run by the Ansible Automation Platform. For HADR considerations of Ansible actions, refer to the Ansible Automation Platform documentation.
- No new usage metrics

Multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps

Multi-region cold standby disaster recovery (DR) enables the restoration of your IBM Cloud Pak for AIOps deployment if an unrecoverable failure impacts the deployment or its hosting cluster, data center, or region.

Overview

The multi-region cold standby disaster recovery solution for IBM Cloud Pak for AIOps uses IBM Fusion to create and restore backups of your IBM Cloud Pak for AIOps deployment and its data. For more information about IBM Fusion, see the Fusion documentation.

IBM Fusion uses the concept of a hub and spoke. A Red Hat® OpenShift® Container Platform cluster that has an IBM Fusion backup & restore service installation on it is the hub cluster. The hub manages the backups, and backups can also be taken or restored on a hub cluster. A Red Hat OpenShift cluster that has an IBM Fusion backup & restore agent installation on it is a spoke cluster. Backups can be taken or restored on a spoke, but the spoke cannot manage the backups.

IBM Fusion takes a backup of your IBM Cloud Pak for AIOps deployment, and stores it in external object storage. If a problem occurs that renders your IBM Cloud Pak for AIOps deployment unusable, then the backup can be restored to a Red Hat OpenShift spoke cluster. The entire IBM Cloud Pak for AIOps deployment is restored, along with its data.

Entitlement to IBM Cloud Pak for AIOps provides you with entitlement to the IBM Fusion Data Foundation service in internal deployment mode only. To implement multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps, you must purchase a full IBM Fusion license for access to the backup and disaster recovery features. For more information, see the license terms.

Restrictions

- IBM Fusion and the IBM Cloud Pak for AIOps deployment to be backed up must be installed on the same Red Hat OpenShift cluster.
- The backup and restore procedures cannot be used on offline (air gapped) deployments of IBM Cloud Pak for AIOps.
- The backup and restore procedures cannot be used if IBM Cloud Pak for AIOps is installed with Infrastructure Automation.
- The backup and restore procedures cannot be used on `AllNamespace` deployments of IBM Cloud Pak for AIOps. For more information about installation modes, see Operator installation mode.
- If IBM Cognos® Analytics is installed, it is not backed up or recovered.
- User interface customizations are not backed up or restored.

Before you begin

Red Hat OpenShift

Ensure that you meet the following prerequisites for the hub cluster, and also for the spoke cluster if you are using one:

- The Red Hat OpenShift version is supported by IBM Cloud Pak for AIOps and IBM Fusion v2.8.x or v2.9.x.
  - IBM Cloud Pak for AIOps: For more information, see Supported Red Hat OpenShift versions.
  - IBM Fusion: For more information, see Prerequisites in the IBM Fusion documentation.
- The Red Hat OpenShift command-line interface (oc) is installed. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI in the Red Hat OpenShift documentation.
- You are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).

Storage

Ensure that you meet the following prerequisites for your persistent and external storage.

- In addition to your IBM Cloud Pak for AIOps deployment's persistent storage, you must have external object storage available for IBM Fusion to use as the backup storage location. This is where the IBM Cloud Pak for AIOps volume backups will be stored. This can be any S3 compatible storage, or Microsoft® Azure Blob Storage.
- The amount of external object storage that is required is affected by the amount of data to be stored, the change rate of the data, and the storage efficiencies that can be made from compression and deduplication. As a guide, for a 30-day retention period, you must have object storage capacity equivalent to a minimum of 125% of your IBM Cloud Pak for AIOps deployment's persistent storage. For example, if you have 100 GB of persistent storage for IBM Cloud Pak for AIOps, then you must have 125 GB of object storage available for backups.
- The persistent storage that IBM Cloud Pak for AIOps uses must be Container Storage Interface (CSI) compatible, and must support volume snapshot and restore. The CSI driver must be v1 or higher.

- If you are using Portworx as your storage provider, then IBM Fusion must be at v2.9.0 or higher, or backup and restore will fail.
- PersistentVolumes (PVs) must be created with `volumeMode` set to *Filesystem*.
- The storage class that is used by IBM Cloud Pak for AIOps must have `volumeBindingMode` set to *Immediate*, and `allowVolumeExpansion` set to *true*.

For more information, see the `Backup & Restore` section in System requirements in the IBM Fusion documentation.

Backup and restore procedures

Use the instructions in the following sections to backup or restore an IBM Cloud Pak for AIOps deployment.

- Multi-region cold standby disaster recovery: Backing up IBM Cloud Pak for AIOps
- Multi-region cold standby disaster recovery: Restoring IBM Cloud Pak for AIOps
- Troubleshooting multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps

Multi-region cold standby disaster recovery: Backing up IBM Cloud Pak for AIOps

Use IBM Fusion to create a backup of your IBM Cloud Pak for AIOps deployment.

Backup procedure

Ensure that you meet the prerequisites in Before you begin.

Use the following instructions to install IBM Fusion on the same Red Hat OpenShift cluster as IBM Cloud Pak for AIOps, and then create a backup of IBM Cloud Pak for AIOps.

1. Ensure that you have a VolumeSnapshotClass
2. Install IBM Fusion
3. Deploy the IBM Fusion Backup & Restore service
4. Configure the IBM Fusion Backup & Restore service for IBM Cloud Pak for AIOps

1. Ensure that you have a VolumeSnapshotClass

To enable IBM Fusion backups, you must have a `VolumeSnapshotClass` defined for one of the storage classes that host IBM Cloud Pak for AIOps. For more information, see Prerequisites in the IBM Fusion documentation.

Run the following command to check whether a `VolumeSnapshotClass` is defined already:

```
oc get volumesnapshotclass
```

If the preceding command does not return anything, then you do not have a `VolumeSnapshotClass` defined. Refer to your storage provider's documentation to create one.

2. Install IBM Fusion

You must install IBM Fusion v2.8.x or v2.9.x.

Follow the *Installing* instructions for your Red Hat OpenShift deployment type in Deploying IBM Fusion in the IBM Fusion documentation. Accept the default values.

Note: As you are installing IBM Fusion on the same Red Hat OpenShift cluster as IBM Cloud Pak for AIOps, you already have an entitlement pull secret configured.

3. Deploy the IBM Fusion Backup & Restore service

1. Install the IBM Fusion Backup & Restore service.

   Follow the instructions in Backup and restore hub in the IBM Fusion documentation.

   The `ibm-backup-restore` project (namespace) is created, and the backup and restore service is installed in that project.

2. Configure the backup storage location for IBM Fusion to store the IBM Cloud Pak for AIOps backups in.

   The backup storage location must be external object storage, and can be any S3 compatible storage or Microsoft® Azure Blob Storage.

   Follow the instructions in Adding a backup storage location in the IBM Fusion documentation.

3. Create a backup policy.

   The backup policy specifies the backup location, frequency, and retention period of the backup. Specify the backup storage location that you created in the previous step as the backup location.

   Follow the instructions in Creating backup policy in the IBM Fusion documentation.

4. Select the IBM Cloud Pak for AIOps application that you want to backup, and assign the backup policy to it.

Follow the instructions in Selecting applications for protection in the IBM Fusion documentation.

Important: Before you select **Assign** in the user interface, you must deselect **Run backup now** to prevent the backup from running immediately.

5. Find the name of your policy assignment.

   Your policy assignment name is the IBM Cloud Pak for AIOps installation namespace concatenated with the backup policy name. Run the following command to find it, and then make a note of it:

   ```
   oc get fpa -n ibm-spectrum-fusion-ns
   ```

4. Configure the IBM Fusion Backup & Restore service for IBM Cloud Pak for AIOps

1. Run the following commands to set the environment variables that you need.

   ```
   export PROJECT_CP4AIOPS=<namespace>
   export CUSTOM_PROFILE=<custom_profile>
   export POLICY_ASSIGNMENT_NAME=<policy_assignment_name>
   ```

   Where:

   - **<namespace>** is the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in, for example **cp4aiops**.
   - **<custom_profile>** is blank unless IBM Sales representatives and Business Partners used the custom sizing tool to supply you with a custom profile ConfigMap to customize your deployment. If you were supplied with a custom profile ConfigMap, then this is the name of your custom profile ConfigMap, usually **aiops-custom-size-profile**.
   - **<policy_assignment_name>** is the policy assignment name that you found in the previous step.

2. Find your installation name and set an environment variable for it.

   1. Find the name of your IBM Cloud Pak for AIOps installation.

      ```
      oc get installation -n ${PROJECT_CP4AIOPS}
      ```

      Example output, where **ibm-cp-aiops** is the name of the IBM Cloud Pak for AIOps installation:

      ```
      NAME          PHASE    LICENSE   STORAGECLASS STORAGECLASSLARGEBLOCK AGE
      ibm-cp-aiops  Running  Accepted  rook-cephfs  rook-ceph-rbd          34h
      ```

   2. Export an environment variable with the name of your installation.

      ```
      export INSTALLATION_NAME=<name>
      ```

      Where **<name>** is the name returned by the previous command, for example **ibm-cp-aiops**.

3. Run the following commands to add labels to IBM Cloud Pak for AIOps resources so that they are correctly identified.

   ```
   oc -n $PROJECT_CP4AIOPS label secret ${INSTALLATION_NAME}-edb-secret manual=${INSTALLATION_NAME}-edb-secret
   oc -n $PROJECT_CP4AIOPS label secret ${INSTALLATION_NAME}-redis-secret manual=${INSTALLATION_NAME}-redis-secret
   oc -n $PROJECT_CP4AIOPS label secret ibm-zen-secret-jwks manual=ibm-zen-secret-jwks
   oc -n $PROJECT_CP4AIOPS label secret ibm-zen-secret-jwt manual=ibm-zen-secret-jwt
   oc -n $PROJECT_CP4AIOPS label secret ibm-zen-secret-jwt-private manual=ibm-zen-secret-jwt-private
   oc -n $PROJECT_CP4AIOPS label secret aiopsedge-config-encryption-history manual=aiopsedge-config-encryption-history
   oc -n $PROJECT_CP4AIOPS label secret platform-auth-ldaps-ca-cert manual=platform-auth-ldaps-ca-cert
   oc -n $PROJECT_CP4AIOPS label secret ibm-entitlement-key manual=ibm-entitlement-key-secret
   oc -n $PROJECT_CP4AIOPS label subscriptions.operators.coreos.com ibm-aiops-orchestrator manual=ibm-aiops-
   orchestrator
   ```

4. Download the IBM Cloud Pak for AIOps backup and restore IBM Fusion recipe.

   The IBM Fusion recipe defines the sequence of actions and checks that IBM Fusion must use for the backup and restore of IBM Cloud Pak for AIOps resources and PVCs.

   Download the recipe **aiops-backup-restore.yaml**, from github.com/IBM.

5. Adjust the recipe if you have a non-default installation name.

   If your installation name is not **ibm-cp-aiops**, then edit the recipe that you downloaded, and alter all occurences of **ibm-cp-aiops** to be your installation name.

6. Adjust the recipe for custom sizing.

   IBM Sales representatives and Business Partners may have used a custom sizing tool to supply you with a custom profile to customize your IBM Cloud Pak for AIOps deployment at installation time.

   If IBM Sales representatives and Business Partners did not supply you with a custom profile, then skip this step.

   If IBM Sales representatives and Business Partners supplied you with a custom profile, then add a label to the custom profile ConfigMap in your IBM Cloud Pak for AIOps deployment. Run the following command:

   ```
   oc label cm ${CUSTOM_PROFILE} cm=aiops-custom-size-profile
   ```

7. Create the IBM Cloud Pak for AIOps recipe custom resource.

   Run the following command to create the recipe custom resource.

   ```
   oc apply -f aiops-backup-restore.yaml
   ```

8. Configure your policy assignment to use the IBM Cloud Pak for AIOps back and restore recipe.

   Patch the policy assignment with the details of the IBM Cloud Pak for AIOps back and restore recipe.

   ```
   oc -n ibm-spectrum-fusion-ns patch policyassignment ${POLICY_ASSIGNMENT_NAME} --type merge  -p '{"spec":{"recipe":
   {"name":"aiops-backup-restore-recipe", "namespace":"ibm-spectrum-fusion-ns"}}}'
   ```

What to do next

Backups occur according to the schedule that you set in your policy. If you want to run a backup immediately, follow the instructions in Backup now.

You can monitor the progress of the backup from **Backup & restore** > **Jobs** in the IBM Fusion user interface.

To view the IBM Fusion logs, or to see the resources that were backed up, see Troubleshooting multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps.

Multi-region cold standby disaster recovery: Restoring IBM Cloud Pak for AIOps

Use IBM Fusion to restore your IBM Cloud Pak for AIOps deployment from backup.

Restore procedure

IBM Fusion recovers the IBM Cloud Pak for AIOps instance to a working state on a target cluster, along with its data. Your target cluster can be the hub cluster, or a different Red Hat OpenShift cluster (a spoke).

Ensure that you meet the prerequisites in Before you begin. User interface customizations are not restored.

Follow the steps for Restoring to the hub cluster or Restoring to a spoke cluster.

Restoring to the hub cluster

If you are restoring IBM Cloud Pak for AIOps to the same Red Hat OpenShift cluster that you created your backup on, then you must completely uninstall your IBM Cloud Pak for AIOps deployment, and remove the deployment namespaces from it (usually **cp4aiops**). The Red Hat OpenShift cluster must not have any residual IBM Cloud Pak for AIOps artifacts. For more information, see Uninstalling IBM Cloud Pak for AIOps.

1. Get the entitlement key.

   Follow the instructions in Obtaining the entitlement key in the IBM Fusion documentation.

2. Create a global pull secret.

   Follow the instructions in Creating image pull secret in the IBM Fusion documentation.

3. Create the required catalogs.

   Follow the instructions in Create the catalog sources.

4. Install Cert Manager

   Skip this step if a certificate manager is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not have a certificate manager, then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

   For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

   1. Run the following command to create the resource definitions that you need:

      ```
      cat << EOF | oc apply -f -
      apiVersion: v1
      kind: Namespace
      metadata:
        name: ibm-cert-manager
      ---
      apiVersion: operators.coreos.com/v1
      kind: OperatorGroup
      metadata:
        name: ibm-cert-manager-operator-group
        namespace: ibm-cert-manager
      ---
      apiVersion: operators.coreos.com/v1alpha1
      kind: Subscription
      metadata:
        name: ibm-cert-manager-operator
        namespace: ibm-cert-manager
      spec:
        channel: v4.2
        installPlanApproval: Automatic
        name: ibm-cert-manager-operator
        source: ibm-cert-manager-catalog
      ```

```
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-cert-manager get pods
   ```

   Example output for a successful IBM Cloud Pak® foundational services Cert Manager installation:

   ```
   NAME                                       READY   STATUS    RESTARTS   AGE
   cert-manager-cainjector-674854c49d-vstq4   1/1     Running   0          8d
   cert-manager-controller-646d4bd6fd-zwmqm   1/1     Running   0          8d
   cert-manager-webhook-8598787c8-s41kt       1/1     Running   0          8d
   ibm-cert-manager-operator-c96957695-dkxnm  1/1     Running   0          8d
   ```

5. Install the License Service

   Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

   IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

   1. Run the following command to create the resource definitions that you need:

      ```
      cat << EOF | oc apply -f -
      apiVersion: v1
      kind: Namespace
      metadata:
        name: ibm-licensing
      ---
      apiVersion: operators.coreos.com/v1
      kind: OperatorGroup
      metadata:
        name: ibm-licensing-operator-group
        namespace: ibm-licensing
      spec:
        targetNamespaces:
        - ibm-licensing
      ---
      apiVersion: operators.coreos.com/v1alpha1
      kind: Subscription
      metadata:
        name: ibm-licensing-operator-app
        namespace: ibm-licensing
      spec:
        channel: v4.2
        installPlanApproval: Automatic
        name: ibm-licensing-operator-app
        source: ibm-licensing-catalog
        sourceNamespace: openshift-marketplace
      EOF
      ```

   2. Run the following command to ensure that the IBM Cloud Pak® foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

      ```
      oc -n ibm-licensing get pods
      ```

      Example output for a successful IBM Cloud Pak® foundational services License Service installation:

      ```
      NAME                                              READY   STATUS    RESTARTS   AGE
      ibm-licensing-operator-db4cd746c-xzmlf            1/1     Running   0          8d
      ibm-licensing-service-instance-596b99588f-76cc5   1/1     Running   0          8d
      ```

   For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

6. Update the IBM Fusion Backup & Restore transaction manager cluster roles.

   Run the following command:

   ```
   oc get clusterrole transaction-manager-ibm-backup-restore -o json | jq '.rules += [{"verbs":
   ["get","list"],"apiGroups":["orchestrator.aiops.ibm.com"],"resources":["installations"]},{"verbs":
   ["get","list"],"apiGroups":["ai-manager.watson-aiops.ibm.com"],"resources":["aimanagers"]},{"verbs":
   ["get","list"],"apiGroups":["lifecycle.ir.aiops.ibm.com"],"resources":["lifecycleservices"]}]' | oc apply -f -
   ```

7. Follow the instructions in Restoring the application in the IBM Fusion documentation.

   Select **Same cluster**, and then for **Project destination** select *Use the same project the application is already using*. Accept the default values for the other fields.

   Note: You might notice that the `aiops-configmaps` resource group shows as failed under the **Restore sequence** section of the IBM Cloud Pak for AIOps restore recipe but the overall restore is successful and works as intended.

8. Delete any duplicate observer jobs on the restored cluster.

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. For any duplicate jobs, click the options menu (three vertical dots) and click **Delete**.

9. From the IBM Cloud Pak for AIOps console, select and edit each `incident` policy that utilises a `ticket` integration, and manually reassign the relevant ticketing system (Github/ServiceNow).

Restoring to a spoke cluster

If you are restoring IBM Cloud Pak for AIOps to a different cluster, ensure that your target cluster meets the prerequisites for an IBM Cloud Pak for AIOps deployment, and then install IBM Fusion and the IBM Fusion backup agent on it.

1. Spoke cluster prerequisites

- You must have the same storage provider and storage classes available for your target spoke cluster as you did for your IBM Cloud Pak for AIOps deployment on the hub cluster.
- The target spoke cluster must have the same deployment platform and Red Hat OpenShift version as the source hub cluster. Review the Planning section and ensure that the environment that you are restoring to meets the system requirements.
- The Red Hat OpenShift cluster that you restore on must not have a previous IBM Cloud Pak for AIOps deployment on it, and must not have any residual IBM Cloud Pak for AIOps artifacts or deployment namespaces (usually `cp4aiops`). If you need to uninstall IBM Cloud Pak for AIOps, see the instructions in Uninstalling IBM Cloud Pak for AIOps.

2. Install IBM Fusion on the target spoke cluster

    1. Get the entitlement key.

       Follow the instructions in Obtaining the entitlement key in the IBM Fusion documentation.

    2. Create a global pull secret.

       Follow the instructions in Creating image pull secret in the IBM Fusion documentation.

    3. Create the required catalogs.

       Follow the instructions in Create the catalog sources.

    4. Install IBM Fusion.

       You must install IBM Fusion v2.8.x or v2.9.x.

       Follow the `Installing` instructions for your Red Hat OpenShift deployment type in Deploying IBM Fusion in the IBM Fusion documentation. Accept the default values.

3. Install Cert Manager

   Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

   For more information about IBM Cloud Pak® foundational services Cert Manager, see Certificate Management in the IBM Cloud Pak foundational services documentation.

    1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-cert-manager-operator-group
  namespace: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-cert-manager-operator
  namespace: ibm-cert-manager
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-cert-manager-operator
  source: ibm-cert-manager-catalog
  sourceNamespace: openshift-marketplace
EOF
```

    2. Run the following command to ensure that the IBM Cloud Pak® foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-cert-manager get pods
```

Example output for a successful IBM Cloud Pak® foundational services Cert Manager installation:

```
NAME                                       READY   STATUS    RESTARTS   AGE
cert-manager-cainjector-674854c49d-vstq4   1/1     Running   0          8d
cert-manager-controller-646d4bd6fd-zwmqm   1/1     Running   0          8d
cert-manager-webhook-8598787c8-s41kt       1/1     Running   0          8d
ibm-cert-manager-operator-c96957695-dkxnm  1/1     Running   0          8d
```

4. Install the License Service

   Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

   IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

   1. Run the following command to create the resource definitions that you need:

      ```
      cat << EOF | oc apply -f -
      apiVersion: v1
      kind: Namespace
      metadata:
        name: ibm-licensing
      ---
      apiVersion: operators.coreos.com/v1
      kind: OperatorGroup
      metadata:
        name: ibm-licensing-operator-group
        namespace: ibm-licensing
      spec:
        targetNamespaces:
        - ibm-licensing
      ---
      apiVersion: operators.coreos.com/v1alpha1
      kind: Subscription
      metadata:
        name: ibm-licensing-operator-app
        namespace: ibm-licensing
      spec:
        channel: v4.2
        installPlanApproval: Automatic
        name: ibm-licensing-operator-app
        source: ibm-licensing-catalog
        sourceNamespace: openshift-marketplace
      EOF
      ```

   2. Run the following command to ensure that the IBM Cloud Pak® foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

      ```
      oc -n ibm-licensing get pods
      ```

      Example output for a successful IBM Cloud Pak® foundational services License Service installation:

      ```
      NAME                                              READY   STATUS    RESTARTS   AGE
      ibm-licensing-operator-db4cd746c-xzmlf            1/1     Running   0          8d
      ibm-licensing-service-instance-596b99588f-76cc5   1/1     Running   0          8d
      ```

   For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

5. Get connection YAML from the hub cluster.

   On the hub cluster, follow the **`Use the Fusion UI`** instructions in Connect clusters in the IBM Fusion documentation.

   When you have copied the connection snippet, return to these steps. The connection snippet is used in the next step to connect the hub and spoke clusters.

6. Install the **`Backup & Restore Agent`** on the spoke cluster, and connect the spoke cluster to the hub cluster.

   On the spoke cluster, follow the procedure in Backup & Restore spoke in the IBM Fusion documentation.

   For the **Hub Connection Snippet** field, use the connection snippet that you copied in the previous step. The storage classes must be the same storage classes that the IBM Cloud Pak for AIOps deployment on the hub cluster used.

7. On the spoke cluster, update the IBM Fusion Backup & Restore transaction manager cluster roles.

   Run the following command:

   ```
   oc get clusterrole transaction-manager-ibm-backup-restore -o json | jq '.rules += [{"verbs":
   ["get","list"],"apiGroups":["orchestrator.aiops.ibm.com"],"resources":["installations"]},{"verbs":
   ["get","list"],"apiGroups":["ai-manager.watson-aiops.ibm.com"],"resources":["aimanagers"]},{"verbs":
   ["get","list"],"apiGroups":["lifecycle.ir.aiops.ibm.com"],"resources":["lifecycleservices"]}]' | oc apply -f -
   ```

8. Follow the instructions in Restoring the application in the IBM Fusion documentation, and wait for the restore to complete.

   Select **New cluster**, and then accept the default values for the other fields.

Note: You might notice that the `aiops-configmaps` resource group shows as failed under the **Restore sequence** section of the IBM Cloud Pak for AIOps restore recipe but the overall restore is successful and works as intended.

9. Redeploy remote integrations.

   Re-run the bootstrap procedure for any remote connections that are configured.

   1. Log in to IBM Cloud Pak for AIOps console.
   2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
   3. For each remote integration, click the integration type on the **Manage integrations** tab of the Integrations page, then:
      1. Click the **Download** link in the **Remote deployment script** column for the integration.
      2. Click **Download as a sh-file** to download the deployment script as a `bootstrap.sh` file.
      3. From the Red Hat OpenShift CLI, switch to the target project (namespace).

         ```
         oc project <namespace>
         ```

         Where `<namespace>` is the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in, for example *cp4aiops*.
      4. Run the `bootstrap.sh` deployment script.

10. If you have secure tunnel connections, then re-create them. For more information, see Create the secure tunnel connection on your restored cluster.
11. Delete any duplicate observer jobs on the restored cluster.
    1. Log in to IBM Cloud Pak for AIOps console.
    2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
    3. For any duplicate jobs, click the options menu (three vertical dots) and click **Delete**.
12. From the IBM Cloud Pak for AIOps console, select and edit each `incident` policy that utilises a `ticket` integration, and manually reassign the relevant ticketing system (Github/ServiceNow).

What to do next

The restore takes a few hours to run. You are notified on the IBM Fusion user interface (UI) when the restore is complete. You can also monitor the progress of the restore from **Backup & restore** > **Jobs** in the IBM Fusion UI.

Important: If you restored to a spoke cluster, the URL and password for accessing the IBM Cloud Pak for AIOps console are different from the URL and password that you used on the hub cluster. Follow the steps in Access the Cloud Pak for AIOps console to find the new URL and password.

(Optional) After the restore is complete, you can verify the status of your deployment by using the steps in Verify your installation (CLI) or Verify your installation (console).

To view the IBM Fusion logs, or to see the resources that were backed up or restored, see Troubleshooting multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps.

Troubleshooting multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps

Troubleshoot your IBM Fusion backup or restore of IBM Cloud Pak for AIOps.

General troubleshooting

- Retrieve backup or restore logs
- List resources that were backed up or restored

Restore issues

- Policy configuration is not retained on the restored deployment
- PostgresSQL instance is out of sync after fusion restore

General troubleshooting tips

Retrieve backup or restore logs

1. Download `getRecipeWorkflow.sh` from github.com/IBM.
2. Obtain the job UID for the backup or restore job. The job UID can be obtained from the IBM Fusion user interface, or from the command-line.

   From the IBM Fusion user interface:

1. Select **Applications** from the left navigation menu.
2. Select the Application from the table, for example `filebrowser`.
3. Select **Backups**.
4. For the backup or restore job of interest, select the right ... menu and select **Details**.
5. Click the job name, for example `filebrowser-demo-daily-apps.test.stglabs.ibm.com-202311220800`.
6. Note the **Job ID** in the **Details** window, for example: `63eb6ea5-dc24-41ef-9fb7-199c48df1508`.`

From the command-line:

1. Open the YAML file for the backup custom resource. For example:

   ```
   oc get fbackup filebrowser-demo-daily-apps.test.stglabs.ibm.com-202311220800 -n ibm-spectrum-fusion-ns -o yaml
   ```

2. Locate the `uid` in `spec: metatdata`, for example: `uid: 63eb6ea5-dc24-41ef-9fb7-199c48df1508`.

3. Run `getRecipeWorkflow.sh` with the job UID.

   ```
   getRecipeWorkflow.sh <job_type> <job_uid>
   ```

   Where

   - `<job_type>` is `backup` or `restore`.
   - `<job_uid>` is the job UID that you obtained.

   Example command:

   ```
   getRecipeWorkflow.sh backup 63eb6ea5-dc24-41ef-9fb7-199c48df1508
   ```

List resources that were backed up or restored

1. Download `getResources.sh` from github.com/IBM.
2. Use step 2 in Retrieve backup or restore logs to get the job UID.
3. Run `getResources.sh` with the job UID.

   ```
   getReesources.sh <job_type> <job_uid>
   ```

   Where

   - `<job_type>` is `backup` or `restore`.
   - `<job_uid>` is the job UID that you obtained.

   Example command:

   ```
   getResources.sh backup 63eb6ea5-dc24-41ef-9fb7-199c48df1508
   ```

Troubleshooting restore issues

Policy configuration is not retained on the restored deployment

Incident policies that previously had a ticket integration are missing the ticket integration on the restored deployment. If GitHub or ServiceNow are configured, then external tickets are not created in them. The policy page has a **Warning** similar to the following example:

```
The previously selected ticket connection Github -mygit is no longer available. Please check the status of your connector.
```

Solution: From the IBM Cloud Pak for AIOps console, edit the policy and add the connection that is shown in the warning.

PostgresSQL instance is out of sync after fusion restore

After restore, an EDB Postgres instance has a different value for the Log Sequence Number (LSN), which means that it is not synchronized with the other instances.

1. If it is not already installed, install the `kubectl` plug-in.

   Run the following command:

   ```
   curl -sSfL \
   https://github.com/EnterpriseDB/kubectl-cnp/raw/main/install.sh | \
   sudo sh -s -- -b /usr/local/bin
   ```

2. Run the following command to check whether all the EDB Postgres instances are synchronized.

   ```
   oc cnp status ibm-cp-aiops-edb-postgres
   ```

   If a Postgres instance is not synchronized, then the value for `Current LSN` is not the same as the `Current LSN` for the other instances.

   Example output where a Postgres instance is not synchronized: `worker9.ir-backend-backup-acme.com` has a different Log Sequence Number (LSN):

```
 Name                          Database  Current     Replication            Status  QoS         Manager  Node
                               Size      LSN         role                                       Version
 ibm-cp-aiops-edb-postgres-1   83 MB     6/C214B440  Primary                OK      Burstable   1.18.13  worker7.ir-
backend-backup-acme.com
 ibm-cp-aiops-edb-postgres-2   82 MB     6/C214B440  Standby (async)        OK      Burstable   1.18.13  worker2.ir-
backend-backup-acme.com
 ibm-cp-aiops-edb-postgres-3   82 MB     6/A124B330  Standby (file based)   OK      Burstable   1.18.13  worker9.ir-
backend-backup-acme.com
```

Solution: If one or more of the Postgres instances are not synchronized, follow the instructions in Postgres EDB database crashes.

Security considerations

IBM Cloud Pak® for AIOps supports different mechanisms for securing your environment and your data.

IBM Cloud Pak for AIOps builds on the security features provided by Red Hat OpenShift Container Platform by creating Security Context Constraints (SCC), service accounts, and roles. These features ensure that IBM Cloud Pak for AIOps pods and users have the lowest level of privileges to the Red Hat OpenShift Container Platform that is needed for them. These features protect sensitive customer data with strong encryption controls and improve the oversight of access control across applications and the platform itself.

In addition, ensure that you keep up to date on any security-related news or vulnerabilities with IBM Cloud Pak for AIOps by subscribing to security bulletins.

Security context constraints (SCC)

Administrators can use SCCs to control permissions for pods on their Red Hat OpenShift Container Platform cluster. These permissions include actions that a pod can perform and what resources it can access. Red Hat OpenShift Container Platform clusters contain default security context constraints (SCCs). IBM Cloud Pak for AIOps runs with the **`restricted`** or **`restricted-v2`** SCC that is included within this default set of constraints. For more information, see Red Hat® OpenShift® Container Platform SCCs and Red Hat - Managing Security Context Constraints.

Note: SecurityContextConstraints do not apply to the **`default`** or **`kube-system`** namespaces.

Permissions

For more information about the permissions that are required for Cloud Pak for AIOps, see Permissions.

Volume level encryption

You can configure Portworx encryption or host-level encryption, depending on your requirements. Decide which level of encryption is needed for each stateful data store in your environment.

- For more information about using encrypted volumes on Red Hat OpenShift Container Storage, see Creating a storage class for persistent volume encryption.
- For more information about using the encryption feature for Portworx volumes, see Create encrypted PVCs.

Security hardening

For production environments, ensure that the Red Hat OpenShift Container Platform cluster where you are installing IBM Cloud Pak for AIOps is configured securely. To help harden the security for your cluster, consider completing the following tasks:

- Configure the Red Hat OpenShift Container Platform to enable etcd data encryption. For more information, see the Red Hat documentation about Encrypting etcd data.
- Implement a network egress policy that strictly controls what outbound calls can be made from applications in the cluster and from the IBM Cloud Pak for AIOps project (namespace). For more information, see the Red Hat documentation about Creating a network policy.
- If you use your own certificate authority (CA), replace the default certificate manager CA with your own CA.
- Ensure that your Red Hat OpenShift Container Platform and Kubernetes environment is configured according to the Center for Internet Security (CIS) Benchmark for Red Hat Enterprise Linux based OpenShift V4 and Kubernetes V1.6 security hardening. For more information, see the Center for Internet Security (CIS) Benchmarks about Securing Kubernetes.
- Configure lockout mechanisms on your authentication server to handle excessive login attempts.
- Shorten session timeouts to the shortest tolerable duration.
- Use your own certificate authority and monitor certificate expiration dates.
- Configure rate limiting connections for basic protection against distributed denial-of-service (DDoS) attacks. For more information about how to configure this limiting, see the Red Hat documentation about Route configuration.
- Limit users from viewing information about other users. By default, user IDs, user groups, roles, and permissions for other users are visible to users. To block non-administrative users from viewing user groups, roles, and permissions,

    1. Edit the config map **`product-configmap`** to add the configuration **`usermgmt_limit_user_info: "true"`**.

2. Restart the user management pods, by running the following command:

```
oc delete pod -l component=usermgmt
```

Protecting against HTTP Host header attacks

The mandatory request header specifies the domain that you want to access can be altered as part of an exploitative attack. If your installation does not use a load balancer or proxy that depends on http **HOST** headers, then use the following steps to enable HTTP Host header injection protection:

1. Ensure that the HTTP Host header matches the value of **URL_PREFIX** from the ConfigMap **product-configmap**. If necessary, specify the project (namespace) of your Cloud Pak for AIOps deployment.
2. In configmap **product-configmap**, add or edit the **HOST_INJECTION_CHECK_ENABLED** entry and set its value to **true**. To set the entry by using the Red Hat OpenShift, select the **product-configmap** in your Cloud Pak for AIOps project (namespace). From the menu, select **Edit configmap**. On the YAML page, find or add the **HOST_INJECTION_CHECK_ENABLED** entry, and set the value to **true**.
3. Restart the **ibm-nginx-*** pods.
   - If you are using the Red Hat OpenShift console, select the pods and use the delete pod action.
   - If you are using the CLI, use the **oc delete <pod-name> -n <namespace>** command.

     Where **<namespace>** is the project (namespace) of your Cloud Pak for AIOps deployment.

If you need to disable this protection for any reason, remove the **HOST_INJECTION_CHECK_ENABLED** entry, or set its value to **false**, and restart the Nginx pods.

Certificates

If you want to use a custom certificate for IBM Cloud Pak for AIOps, then you can do this with either of the following methods:

- Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
- If you would like to use a custom certificate for IBM Cloud Pak for AIOps, then after installation is complete follow the instructions in Using a custom certificate.

Image verification

You can optionally verify the IBM Cloud Pak for AIOps images before you install IBM Cloud Pak for AIOps. For more information, see Verifying images.

Verifying images

Use this information to verify the signatures on the IBM Cloud Pak® for AIOps images.

Digital signatures provide a way for consumers of content to ensure that what they download is authentic (it originated from the expected source) and has integrity (it is what it is expected to be). All images for IBM Cloud Pak for AIOps are signed.

Prerequisites

- Ensure that you these command line tools are installed (they can usually be installed on Linux® with the package manager):
  - GNU Privacy Guard v2 (package name **gpg2**)
  - OpenSSL (package name **openssl**)
  - skopeo
- The IBM Cloud Pak for AIOps public key must exist on the same machine as the command line tools. Copy the following text block exactly as shown into a text editor, and save it in a file named **cp4waiops-public.pub.asc**:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGc7ZOEBEADTRz2VX7ukVRhPvwpq+LUhbc9pwCIXCA2Mj9M5/OIY7UEZ57+9
bs0pKbumxa4H6JOKl4iQd+jdlmeZU5W6jeJoDqQ7YPIniKemnqtbGqIDNN7bvkAe
t7hcX4QdgQTezO2r7AsEkjRnTioXBhP1Ie+RHbnZlWRexwSa2GPaZmewpH6ysFBI
6mysLpXiBRs0vRtNwPvliBjZIwtLXZYPF83mt2YvB2uzrRrDPD9OCfNW8vUpdvuu
EsGHps/LzV4iob6TWWcbQSvagnq3KG8/Z6CAEOWDCJFR78kgqtm0bN/SKXFxFXMi
n+OLwNhLm4E+l0aJ3lGUCfjAhY5q4WftUPoytnNLrolyKoHSCM08h1YGm0EB/xVQ
z6+sl+YmXtV0QEx+yyaV51IONXKHBsAWiyoiOr7KmHF6SPIbxXPK13QQqgkeqRAU
MY1t9lUxSuVugdImBeVtNBf5iSJp7FYMAzF4q8Tpr2KzB3gPyM+f9jf89+ZpiUZg
LsX7Xp3Kl6TlwvIU+HU2LHhBJZ6KnQgUou7Q7y4gIeypnUDmb3lZwg5eRQBvA4sQ
xS2Ct3+nBM9g991FnFzK0N7YrdjgyiODx0QLY7RG+T2MHwaQj0apjy23qY5Otnix
3YtDK4qwazp14ciaQY02eFtQxtk0Vwfom83n/QTO8QCs6aBGA8RcpaBFuwARAQAB
tDFJQk0gQ2xvdWQgUGFrIGZvciBXYXRzb24gQUlPcHMgPHBzaXJ0QHVzLmlibS5j
b20+iQI6BBMBCAAkBQJnO2ThAhsPBQsJCAcCBhUKCQgLAgQWAgMBAh4BBQkAAAAA
AAoJEHjfy5NvBiOtm/wP/1g1Rt/0WjdcQqmhnhqscd83jbtBQ2n8f1Ehx1Wlm07w
5KGu8ARswSTTVr1KCEU1DIqmeCel+WS834xzJVkUC6ZHOENGRLN6F8SafarJWHpe
AqtkgPj2vddKznSRqSIwugOFKEu6VuENAHDC3yPUIFGYntH+M1e2DLvr/ls4DyhS
jVfQ0HG6Of7+nn+APxsMXo5s393r8w8SyXi1j/JjlEraQQIStxsIlA5bzL/kDWZ0
1TOykJ9ZECeJkEAD9F07o1ammPrT/mMZO0XJUUX5TQAy9l1o3kczUKbSZRT8PO8p
cWdd/O4zC5K/156rviAYg9NwjMwWG359d5i8VMBGJhSe0o9IUmUAMVEfxeVdc+U1
```

```
3Y4UrHp6GBzw7+jb4OS1wmTVercgI0FjL99+FHQcRQd2Jje95fEV/l7Flww1OybS
+t1B57FX9+YdzjcGFZV/SqSg4CokyBJQsZx7UNRBTQtqyC8oJhLU3A9sW/3r10Ry
FkLDb/l/zfnU7EoUNqQwvlmkYyxBNgmFD1WCdOoTGmVfL/7hhUNEVp455TPCnhIZ
jGaRiGmehH+qbPFSr1+kOPZw5qh+RurtWJpCXya/8jaypW3naqY8PMxYybfgSa9Q
MlmVerxDTlspub/nGDTnW0LQNzYt/UlGecVpdyAn0oU2y83Y2jvOXbAd2lH/WG73
=wCG8
-----END PGP PUBLIC KEY BLOCK-----
```

- You must have a list of images to verify. To get a list of container images used in Cloud Pak for AIOps, refer to the procedure in Downloading or listing container images. In the following procedure, the example image *cp.icr.io/cp/cp4waiops/node-server:v3.7.0-00000000.0000-000000000* is used.

  Note: This tag is an example for demonstration purposes; it is not a real tag. Obtain the most up-to-date image names and tags with the procedure in Downloading or listing container images.

Procedure

1. Log in to the entitled registry `cp.icr.io` so that you can pull images from that registry.

   ```
   podman login cp.icr.io --username cp --password <entitlement_key>
   ```

   If you need to obtain the entitlement key that is assigned to your ID, complete the following steps:

   1. Log in to MyIBM Container Software Library with the IBMid and password that are associated with the entitled software.
   2. In the Entitlement keys section, select **Copy key** to copy the entitlement key to the clipboard.
2. Import the IBM Cloud Pak for AIOps public key on the machine that you prepared according to the Prerequisites section:

   ```
   sudo gpg2 --import cp4waiops-public.pub.asc
   ```

   Note: This step needs to be done only once on each machine you use for signature verification.

3. Calculate the fingerprint:

   ```
   fingerprint=$(sudo gpg2 --fingerprint --with-colons IBM Cloud Pak for AIOps | grep fpr | tr -d 'fpr:')
   ```

   This command stores the key's fingerprint in an environment variable called *fingerprint*, which is need for the command to verify the signature. When you exit your shell session, the variable is deleted. The next time that you log in to your machine, you can set it again by rerunning the command.

4. Create a directory for the image and use skopeo to pull it into local storage:

   ```
   mkdir images
   skopeo copy docker://cp.icr.io/cpopen/ibm-aiops-orchestrator:3.2.0-2021-00-00-0000-00000000 dir:$HOME/images
   ```

   This command downloads the image as a set of files and places them in the images directory (or another directory that you choose).

   Note: There is a manifest file named *images/manifest.json*, and a signature file named *images/signature-1*. You reference both these files in the next step (in the command to verify the signature).

5. Verify the signature:

   ```
   for SIGNATURE in $(ls $HOME/images | grep signature); do
       skopeo standalone-verify $HOME/images/manifest.json cp.icr.io/cp/cp4waiops/node-server:v3.7.0-00000000.0000-
   000000000 ${fingerprint} $HOME/images/$SIGNATURE 2> /dev/null
       if [ $? -eq 0 ]; then
          break
       fi
   done
   ```

   If the signature is verified, you get a confirmation similar to the following output. Otherwise, no output is provided:

   ```
   Signature verified, digest sha256:0000000000000000000000000000000000000000000000000000000000000000
   ```

Downloading or listing IBM Cloud Pak for AIOps images

You might need to download all of the container images for IBM Cloud Pak for AIOps, for example, to copy them to an internal registry, scan them for vulnerabilities, or verify their signatures. This procedure describes how to download container images, although you may need to use different tools, depending on your particular environment.

1. Follow the procedure in Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a bastion host, up to and including the "Section 2 Download the CASE".

   These steps:

   - download the product CASE packages, which contain metadata for the container images required to deploy each capability. Each CASE package also contains the required scripts to mirror images to a private registry, and to configure the target cluster to use the private registry as a mirror.
   - generate comma-separated value (CSV) files listing the images for all capabilities (one CSV file is created for each capability and required dependencies). The CSV files, combined with your IBM Entitled Registry entitlement key, are used to download or mirror the images manually in the scripts below.

2. Log in to your container tool to access the entitled registry, using the username `cp` and your entitlement key.

3. Use a shell script to process all images from all capabilities supported on the target architecture.

   For example, for x86/amd64 architecture:

   ```
   tail -q -n +2 $HOME/offline/*-images.csv | while IFS="," read registry image_name tag digest mtype os arch variant
   insecure digest_source image_type groups; do
   if [[ "$arch" == "amd64" ]]; then
     podman pull "$registry/$image_name@$digest"
     podman tag "$registry/$image_name@$digest" "$registry/$image_name:$tag"
   fi
   done
   ```

   Instead of podman, you can use alternative container tools such as skopeo. Use one skopeocommand to replace the podman pull podman tag commands. For example:

   ```
   skopeo copy --dest-tls-verify=false docker://$registry/$image_name@$digest docker-daemon:$registry/$image_name:$tag
   ```

   You can also use skopeo to save images to disk, or to copy them directly to another container registry.

   Note: If you need your clusters to access these images from a private registry, the image digests must be preserved. If this is your intention, IBM recommends using the procedure in Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline).

Licensing

Review the following to learn about the licensing and entitlements for IBM Cloud Pak® for AIOps 4.8.0.

Notice: This License Guide is intended to provide only supplementary information to assist you in deploying the Programs you have licensed from IBM within your purchased entitlement. Your license agreement (such as the IBM International Program License Agreement (IPLA) or equivalent and its transaction documents, including the License Information for IBM Cloud Pak® for AIOps 4.8.0), is the sole and complete agreement between you and IBM regarding use of the Program.

- Viewing the license
- What's included in your purchaseof IBM Cloud Pak® for AIOps 4.8.0
- Red Hat Products
- Entitlement details
- License reporting

Viewing the license

To review the license agreement for IBM Cloud Pak® for AIOps 4.8.0 to find more information about licensing and entitlements for this offering, see L-BBAE-L53CM5.

What's included in your purchase of IBM Cloud Pak® for AIOps 4.8.0

Your purchase of IBM Cloud Pak for AIOps includes the following entitlements:

- Bundled Programs
- Supporting Programs
- Non-Production features
- Prohibited Components

Evaluation Period

The evaluation period starts upon agreement and lasts for 60 days.

Bundled Programs

As a licensee, you are authorized to install and use the followinig Bundled Programs:

- IBM Netcool Operations Insight Operations Management
- IBM Netcool Operations Insight Connection
- IBM Operations Analytics Predictive Insights
- IBM Robotic Process Automation Platform
- IBM Process Mining
- Monitoring

Supporting Programs

As a licensee, you are authorized to install and use the following supporting programs:

- IBM Netcool Operations Insight Agile Service Manager
- IBM Cloud Pak Foundational Services
- IBM Fusion
- IBM Cognos Analytics User Cartridge for IBM Cloud Pak for Data
- IBM Cognos Analytics Administrator Cartridge for IBM Cloud Pak for Data

Non-Production

As a licensee, the following components are limited for internal development and test environments for internal non-production activities ("Non-Production"):

- Story prioritization from persistence analysis and golden signals extraction
- Deploy to single ROSA cluster configured across multiple AWS Zones
- Multi-zone high availability and disaster recovery (HADR)
- IBM Cognos Analytics Integration for AIOps Custom Dashboards and Reporting

Prohibited Components

As a licensee, you are not authorized to use any of the following components or functions of the Program:

- Application Discovery component (of IBM Netcool Operations Insight Agile Service Manager)
- Network Discovery component (of IBM Netcool Operations Insight Agile Service Manager)
- Attended Bot Agent add-on (of IBM Robotic Process Automation)
- ABBYY FineReader OCR Provider (of IBM Robotic Process Automation)
- Business Process Model and Notation Export/Import Capability (of IBM Process Mining)
- Decision Model and Notation Export/Import Capability (of IBM Process Mining)

Red Hat Products

As a licensee, your purchase of IBM Cloud Pak® for AIOps 4.8.0 includes restricted entitlements for the following Red Hat products in the quantities and/or ratios set out in your license agreement:

- Red Hat OpenShift Container Platform

Entitlement details

Various metrics and conversion ratios apply to your entitlement of IBM Cloud Pak® for AIOps 4.8.0. These metrics and conversion ratios include, but not limited to, the following:

- Managed Client Device
- Managed Device
- Process
- Resource Unit
- Virtual Processor Core
- Virtual Server

The following table lists some of the entitlements per resource unit (perpetual) for IBM Cloud Pak for AIOps:

Table. Entitlement per resource units

Event Processing & Analytics	Metric Processing & Analytics	Log Processing & Analytics
- 1 Managed Device = 1 resource - 20 Client Devices = 1 resource - 10,000 Monthly Events = 7 resource	- 1 Managed Device = 1 resource - 20 Client Devices = 1 resource	• 50GB Monthly logs = 7 resource

License reporting

As part of using IBM Cloud Pak for AIOps, tracking your license usage is required. Although IBM Cloud Pak for AIOps provides license options not tracked directly by the product, to track usage, you must install and deploy the IBM Cloud Pak foundational services License Service on all clusters where IBM Cloud Pak for AIOps is installed.

IBM Cloud Pak foundational services License Service

This background License Service collects and stores the license usage information for tracking license consumption and for audit purposes. You can retrieve this data upon request for monitoring and compliance. You can also retrieve an audit snapshot of the data that is audit evidence.

Note: For IBM Cloud Pak for AIOps, the license service collects usage data for only IBM Cloud Pak for AIOps. Although IBM Cloud Pak for AIOps provides license options not tracked directly by the product, to track usage, you must deploy the IBM Cloud Pak for AIOps License Service on all clusters where IBM Cloud Pak for AIOps is installed.

Refer to the IBM Cloud Pak foundational services documentation to learn about how License Service collects and measures the license usage of IBM Cloud Pak for AIOps, and how to retrieve the license usage data.

- License Service
- License Service Reporter

Passport Advantage part numbers

You might need to download a package from IBM® Passport Advantage® to install some products and extensions for IBM Cloud Pak® for AIOps. This requirement can be due to your entitlement or based on how you choose to install the product or extension.

Use these tables to find the files for IBM Cloud Pak® for AIOps that are included in Passport Advantage and the corresponding part numbers.

- Downloading packages
- Quick start guide
- IBM Cloud Pak for AIOps Core packages
- IBM Cloud Pak for AIOps Supporting product packages

- Infrastructure Automation packages
- IBM Cloud Pak for AIOps Topology Manager product packages
- IBM Cloud Pak for AIOps Metrics Manager product packages
- IBM Cloud Pak for AIOps Extension eAssemblies

Downloading packages

Complete the following procedure to download your packages:

1. Go to the Passport Advantage Online tab at Passport Advantage and log in with your IBM ID.
2. Find your part number.
3. Search for your files by using the part numbers that are listed in the tables.
4. Download the files to a directory on your computer.

Core component eAssemblies

The following table lists the available multiplatform eAssembly number for IBM Cloud Pak for AIOps.

Table 1. Software eAssembly part numbers for base components

eAssembly #	Description
G0DB3EN	IBM Cloud Pak for AIOps 4.8.0 Installation Packages eAssembly

Quick start guide

The following table lists the quick start guide eImage number for IBM Cloud Pak for AIOps.

Table 1. Software eImage part number for quick start guide

eImage #	Description
M0NXBEN	IBM Cloud Pak for AIOps 4.8.0 Quick Start Guide

Supporting product eAssemblies

The following tables list the available eAssembly numbers for IBM Cloud Pak® for AIOps supporting products.

Table 2. Software eAssembly part numbers for supporting products

eAssembly #	Description
G0DCBML	IBM Robotic Process Automation for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DCCML	IBM Process Mining for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DCDML	IBM Netcool Operations Insight V1.6.13 Quick Start Guide for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DCTML	IBM Netcool Operations Insight v1.6.13 Cloud Paks for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DCVML	IBM Tivoli Netcool/OMNIbus v8.1 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DCWML	IBM Tivoli Netcool Impact v7.1 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DCXML	IBM Operations Analytics Log Analysis v1.3.8 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DCYEN	IBM Operations Analytics Log Analysis Service Desk Extension v1.1 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DCZEN	IBM Operations Analytics Advanced Insights v1.3.6 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DD0ML	Jazz for Service Management 1.1.3.19 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DDGML	IBM Db2 Server Edition v11.5.x from Netcool Operations Insight V1.6.13 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DDHML	IBM Tivoli Monitoring for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DDJEN	IBM Tivoli Monitoring Agents for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DDLEN	IBM Tivoli Netcool/OMNIbus Probes for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DDMEN	IBM Tivoli Netcool/OMNIbus Probe Dependencies for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DDNML	IBM Tivoli Netcool/OMNIbus Event Forwarding V8.1 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DFVEN	IBM Tivoli Netcool/OMNIbus Data Exchange V8.1 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DFWEN	Network Health Dashboard v4.2 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DFXEN	Device Dashboard v1.1.0.2 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DGRML	IBM Cognos Analytic Server 11.1 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DGSML	IBM Cognos Analytic Server 11.2 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DGWML	IBM Tivoli Monitoring V6.3.0.7 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DGYML	IBM Tivoli Monitoring V6.3.0.2 QSG for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DH1ML	IBM Tivoli Monitoring V6.3.0.7 Agent for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DH2ML	IBM DB2 Advanced Workgroup Server Edition 11.1 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DH3ML	IBM DB2 Standard Edition 11.5.4 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DH4ML	IBM Tivoli System Automation For Multiplatform Base V3.2.0 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DH5EN	Netcool System Service Monitor Component V4.0.1 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DH7ML	Jazz for Service Management V1.1.2.0 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DH8ML	ITCAM Transactions V7.4.0.1 Response Time and Internet SerVice Monitoring for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DH9ML	ITCAM Transactions V7.4.0.1 Transaction Tracking for IBM Cloud Pak for AIOps 4.8.0 eAssembly

eAssembly #	Description
G0DHBML	IBM Rational 8.6 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DJ4ML	ITCAM for Microsoft Applications Version 6.3.1 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DJ5ML	ITCAM for Applications V7.2.2.2 for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DJ6ML	IBM Tivoli Monitoring for Virtual Environments V7.2.0.3 QSG for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DJ8ML	ITCAM for Application Diagnostics V7.1.0.4: Managing Server Component for IBM Cloud Pak for AIOps 4.8.0 eAssembly

Infrastructure Automation packages

The following table lists the eAssembly number for Infrastructure Automation - Infrastructure Management.

Table. Software eAssembly part numbers for Infrastructure Automation

eAssembly #	Description
G0DB2EN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure Automation eAssembly

The following table lists the eImage numbers for Infrastructure Automation - Infrastructure Management.

Table. Software eImage part number for Infrastructure Automation

eImage #	Description
M0NX4EN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for VMware vSphere with PVSCSI controller
M0NX5EN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for Red Hat OpenStack Platform
M0NX6EN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for Amazon EC2
M0NX7EN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for Red Hat Virtualization
M0NX8EN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation RPM updates
M0NX9EN	IBM Cloud Pak for AIOps 4.8.0 Infrastructure automation for Google Compute Engine

Topology Manager eAssemblies

Table 4. Software eAssembly part numbers for Topology Manager supporting products

eAssembly #	Description
G0DFYEN	IBM Netcool Agile Service Manager v1.1 Base for IBM Cloud Pak for AIOps 4.8.0 eAssembly
G0DFZEN	IBM Netcool Agile Service Manager v1.1 Observers for IBM Cloud Pak for AIOps 4.8.0 eAssembly

Metrics Manager eAssemblies

Table 5. Software eAssembly part numbers for Metrics Manager supporting products

eAssembly #	Description
G0DGTEN	IBM Operations Analytics Predictive Insights v1.3.6 for IBM Cloud Pak for AIOps 4.8.0 eAssembly

Supporting products and extensions

After you purchase an extension, follow these instructions to install your IBM Cloud Pak® for AIOps extension.

The part numbers for supporting products and extensions can be found in Passport Advantage part numbers.

Installing the Network Manager extension

You can integrate an on-premises IBM Netcool Network Management or IBM® Netcool® Operations Insight® - Network Management installation with IBM Cloud Pak for AIOps, while continuing to discover and manage physical and virtual network infrastructure.

To install the Network Manager extension, see Installing Network Management. To configure the IBM Cloud Pak for AIOps Network Manager observer to load data from the Network Manager extension into IBM Cloud Pak for AIOps, see Configuring Network Manager Observer jobs.

Installing IBM Cloud Pak for AIOps on OpenShift

You can choose to complete a standard production environment installation of IBM Cloud Pak® for AIOps, or a streamlined small starter installation for a non-production proof-of-concept or trial deployment.

Starter installation

A starter installation provides a concise installation of IBM Cloud Pak for AIOps for deployments that do not require a sustained workload. Use a starter installation only for proof-of-concept or experimentation environments.

- Online starter installation of IBM Cloud Pak for AIOps

Note: If you require an air-gapped (offline) deployment, you must complete a production installation.

Production installation

Production installations are intended for real world use cases and typically require more system and storage resources.

- Installing IBM Cloud Pak for AIOps (Production installation)

Online starter installation of IBM Cloud Pak for AIOps (CLI)

If you are a trial user of IBM Cloud Pak® for AIOps, or you want a proof-of-concept deployment that does not require a sustained workload, consider a starter installation to get a smaller, nonproduction deployment up and running quickly.

If you are installing on Azure Red Hat OpenShift (ARO), Google Cloud Platform (GCP), Red Hat® OpenShift® on IBM Cloud® service (ROKS), or Red Hat OpenShift Service on AWS (ROSA), then review the information in Installing on cloud platforms.

If you require a production-grade deployment, or an offline air-gapped deployment, see Installing IBM Cloud Pak for AIOps.

Before you begin

- You must know whether you are deploying a base deployment or a extended deployment of IBM Cloud Pak for AIOps. For more information, see Incremental adoption.
- Review the Planning section, and ensure that your cluster meets the system requirements for a starter deployment.
- Online installations of IBM Cloud Pak for AIOps can be run entirely as a non-root user, and do not require that user to have sudo access.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that the IBM Cloud Pak for AIOps operators need, see Permissions (IBM Cloud Pak for AIOps).
- A user with `cluster-admin` privileges is needed for the following operations:
  - Create a custom project
  - Install Cert Manager
  - Install the License Service
  - Create the catalog sources
  - Create an OperatorGroup

Limitations

- After installation, you cannot increase the size of your IBM Cloud Pak for AIOps deployment from a starter deployment to a larger production deployment.
- High availability is not supported for starter deployments.
- This starter installation uses the `OwnNamespace` installation mode, which is the recommended installation mode for IBM Cloud Pak for AIOps, to provide better resilience and performance. For more information, see Operator installation mode.
- If IBM Sales representatives and Business Partners supplied you with a custom profile ConfigMap to customize your deployment, then you must follow their instructions to apply it during installation. The custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. For more information about custom sizing, see Custom sizing.

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the `IBM Cloud Container Registry` and IBM Cloud Pak® foundational services catalog source.

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

For more information, see Configuring your firewall for OpenShift Container Platform.

1. Install and configure Red Hat OpenShift Container Platform

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.

2. Install the Red Hat OpenShift (**oc**) command-line interface (CLI) on your cluster's boot node and run **oc login**. For more information, see Getting started with the Red Hat OpenShift CLI.

3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a **MachineConfig** custom resource to configure chrony to connect to your NTP servers.

4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:

   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

2. Configure storage

Configure your own storage for use with IBM Cloud Pak for AIOps. For more information, see Storage.

Note: Storage classes and storage providers cannot be changed after you install IBM Cloud Pak for AIOps. OADP backup and restore requires that a ReadWriteMany (RWX) storage class must be provided. If OADP backup and restore is not needed, a ReadWriteOnce (RWO) storage class can be provided as the **RWX-storage-class-name** in the installation instance CR YAML file. This configuration cannot be changed after IBM Cloud Pak for AIOps is installed.

3. Create a custom project (namespace)

Create a project (namespace) called **cp4aiops** for your IBM Cloud Pak for AIOps deployment, by running the following command:

```
oc create namespace cp4aiops
```

4. Create an OperatorGroup in your custom project (namespace)

Create the Operator group by running the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cp4aiops-operator-group
  namespace: cp4aiops
spec:
  targetNamespaces:
    - cp4aiops
EOF
```

5. Create the entitlement key pull secret

1. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.

2. In the *Entitlement keys* section, select **Copy** to copy your entitlement key to the clipboard.

3. From the Red Hat OpenShift CLI, run the following command:

   ```
   oc create secret docker-registry ibm-entitlement-key \
       --docker-username=cp \
       --docker-password=<entitlement-key> \
       --docker-server=cp.icr.io \
       --namespace=cp4aiops
   ```

   Where **<entitlement-key>** is the entitlement key that you copied in the previous step.

6. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The usage data is collected by the **cp4waiops-metricsprocessor** pod, and is sent to and stored in IBM controlled GDPR-compliant systems. The collection of usage data is enabled by default, but can be disabled. For transparency, the **cp4waiops-metricsprocessor** pod's logs contain all the information that is collected. The usage data that is collected is numeric, or is about the deployment type and platform. It does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of integrations of each type (For example ServiceNow, Instana, Falcon Logscale)
- Secure tunnel enablement: whether *connection* (which controls whether you can create a secure tunnel) is enabled in the *Installation* custom resource
- Deployment type: base deployment or extended deployment
- Deployment platform: Red Hat® OpenShift® Container Platform or Linux®

Configuring the collection of usage data

If you do not want to disable the collection of usage data, run the following steps.

1. Set environment variables.

```
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial` or `poc`>
export PROJECT_CP4AIOPS=cp4aiops
```

2. Configure the usage data with your customer details.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME}
--from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT}
```

3. If you have a firewall enabled, ensure that outbound traffic to https://api.segment.io is allowed.

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must run the command given in *Disabling the collection of usage data*.

Disabling the collection of usage data

If you want to disable the collection of usage data, run the following commands.

1. Set environment variables.

```
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial` or `poc`>
export PROJECT_CP4AIOPS=cp4aiops
```

2. Disable usage data collection

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME}
--from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT} --from-
literal=enableCollection=false
```

Note: You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

7. Create the catalog sources

1. Run the following command to create the catalog sources for IBM Cloud Pak for AIOps and IBM Cloud Pak® foundational services Cert Manager and License Service in the *openshift-marketplace* namespace.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-aiops-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-aiops-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-aiops-catalog@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cert-manager-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-cert-manager
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-cert-manager-operator-catalog
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-licensing-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM License Service Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-licensing-catalog
EOF
```

2. Verify that the *ibm-aiops-catalog*, *ibm-cert-manager-catalog* and *ibm-licensing-catalog* `CatalogSource` objects are in the output that is returned by the following command:

```
oc get CatalogSources -n openshift-marketplace
```

Example output:

```
oc get CatalogSources -n openshift-marketplace
NAME                    DISPLAY                     TYPE   PUBLISHER   AGE
```

```
ibm-aiops-catalog        ibm-aiops-catalog            grpc   IBM          2m
ibm-cert-manager-catalog ibm-cert-manager             grpc   IBM          2m
ibm-licensing-catalog    IBM License Service Catalog  grpc   IBM          2m
```

8. Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-cert-manager-operator-group
  namespace: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-cert-manager-operator
  namespace: ibm-cert-manager
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-cert-manager-operator
  source: ibm-cert-manager-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-cert-manager get pods
```

Example output for a successful IBM Cloud Pak® foundational services Cert Manager installation:

```
NAME                                       READY   STATUS    RESTARTS   AGE
cert-manager-cainjector-674854c49d-vstq4   1/1     Running   0          8d
cert-manager-controller-646d4bd6fd-zwmqm   1/1     Running   0          8d
cert-manager-webhook-8598787c8-s41kt       1/1     Running   0          8d
ibm-cert-manager-operator-c96957695-dkxnm  1/1     Running   0          8d
```

9. Install the License Service

Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-licensing
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-licensing-operator-group
  namespace: ibm-licensing
spec:
  targetNamespaces:
  - ibm-licensing
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-licensing-operator-app
  namespace: ibm-licensing
```

```
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-licensing-operator-app
  source: ibm-licensing-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-licensing get pods
   ```

   Example output for a successful IBM Cloud Pak® foundational services License Service installation:

   ```
   NAME                                                READY   STATUS    RESTARTS   AGE
   ibm-licensing-operator-db4cd746c-xzmlf              1/1     Running   0          8d
   ibm-licensing-service-instance-596b99588f-76cc5     1/1     Running   0          8d
   ```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

10. Verify cluster readiness

Run the prerequisite checker script to verify whether your Red Hat OpenShift cluster is correctly set up for an IBM Cloud Pak for AIOps installation.

Download the prerequisite checker script from github.com/IBM, and run it with the following command:

```
./prereq.sh -n cp4aiops
```

Example output:

```
# ./prereq.sh -n cp4aiops

[INFO] Starting IBM Cloud Pak for AIOps prerequisite checker v4.8...

CLI: oc

[INFO] ================================Platform Version Check================================
[INFO] Checking Platform Type....
[INFO] You are using Openshift Container Platform
[INFO] OCP version 4.16.7 is compatible but only nodes with AMD64 architectures are supported at this time.
[INFO] ================================Platform Version Check================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Cert Manager Check================================
[INFO] Checking for Cert Manager operator

[INFO] Successfully functioning cert-manager found.

CLUSTERSERVICEVERSION             NAMESPACE
ibm-cert-manager-operator.v4.2.8  ibm-cert-manager

[INFO] ================================Cert Manager Check================================

[INFO] ================================Licensing Service Operator Check================================
[INFO] Checking for Licensing Service operator

[INFO] Successfully functioning licensing service operator found.

CLUSTERSERVICEVERSION          NAMESPACE
ibm-licensing-operator.v4.2.8  ibm-licensing

[INFO] ================================Licensing Service Operator Check================================

[INFO] ================================Starter or Production Install Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource Summary================================================
[INFO]                                                Nodes     |      vCPU       |  Memory(GB)
[INFO] Starter (Non-HA) Base (available/required)       [  9 / 3 ]    [  144 / 47 ]    [  289 / 123 ]
[INFO]      (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 3 ]    [  144 / 55 ]    [  289 / 136 ]
```

```
[INFO] Production (HA) Base (available/required)        [  9 / 6 ]   [  144 / 136 ]   [  289 / 310 ]
[INFO]      (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 6 ]   [  144 / 162 ]   [  289 / 368 ]
[INFO] ================================Resource Summary=============================================================
[INFO] Cluster currently has resources available to create a Starter (Non-HA) install of Cloud Pak for AIOps

[INFO] ================================Prerequisite Checker Tool Summary================================
      [  PASS  ] Platform Version Check
      [  PASS  ] Storage Provider
      [  PASS  ] Starter (Non-HA) Base Install Resources
      [  FAIL  ] Production (HA) Base Install Resources
      [  PASS  ] Cert Manager Operator Installed
      [  PASS  ] Licensing Service Operator Installed
[INFO] ================================Prerequisite Checker Tool Summary================================
```

11. Install the IBM Cloud Pak for AIOps operator

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

Run the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-aiops-orchestrator
  namespace: cp4aiops
spec:
  channel: v4.8
  installPlanApproval: Automatic
  name: ibm-aiops-orchestrator
  source: ibm-aiops-catalog
  sourceNamespace: openshift-marketplace
EOF
```

Warning: installPlanApproval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

After a few minutes, the IBM Cloud Pak for AIOps operator is installed. Verify that the all of the components have a state of **Succeeded** by running the following command:

```
oc get csv -n cp4aiops | egrep "ibm-aiops-orchestrator"
```

Example output:

```
$ oc get csv -n cp4aiops | egrep "ibm-aiops-orchestrator"
ibm-aiops-orchestrator.v4.8.0         IBM Cloud Pak for AIOps               4.8.0   Succeeded
```

12. Install IBM Cloud Pak for AIOps

Note: A maximum of one IBM Cloud Pak for AIOps custom resource is allowed per cluster.

Run the following command to create an instance of the IBM Cloud Pak for AIOps custom resource called **ibm-cp-aiops**.

Warning: The pakModules **aiopsFoundation**, **applicationManager**, and **aiManager** must be enabled as in the following YAML. Do not change these values to false.

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ibm-cp-aiops
  namespace: cp4aiops
spec:
  imagePullSecret: ibm-entitlement-key
  license:
    accept: <license_acceptance>
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: false
  - name: logAnomalyDetection
    enabled: <enable_log_anomaly_detection>
  size: small
  storageClass: <storage_class_name>
  storageClassLargeBlock: <large_block_storage_class_name>
  topologyModel: application
EOF
```

Where

- **<license_acceptance>** - is set to *true* to agree to the license terms.
- **<enable_log_anomaly_detection>** - set to *true* to install an extended deployment with log anomaly detection and ticket analysis capabilities enabled. Set to *false* to install a base deployment without log anomaly detection and ticket analysis capabilities enabled. For more information, see Incremental adoption.
- **<storage_class_name>** and **<large_block_storage_class_name>** are the storage classes that you want to use, as detailed in the following table. For more information about storage, see Storage.

Note: To confirm that you have the storage classes for your chosen storage provider as shown in the table, run **oc get sc**.

Table 1. Storage provider classes

Storage provider	storage_class_name	large_block_storage_class_name
IBM Cloud®	ibmc-block-gold	ibmc-block-gold
IBM Cloud® with OADP backup and restore	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
Red Hat® OpenShift® Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Global Data Platform	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc
IBM Storage Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	px-csi-aiops	px-csi-aiops
Portworx (multi-zone HA)	px-csi-aiops-mz	px-csi-aiops-mz
AWS native storage	gp3-csi	gp3-csi
AWS native storage with OADP backup and restore	efs-sc	gp3-csi

13. Verify your installation

Run the following command to check that the **PHASE** of your installation is **Updating**.

```
oc get installations.orchestrator.aiops.ibm.com -n cp4aiops
```

Example output:

```
NAME            PHASE      LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
ibm-cp-aiops    Updating   Accepted   rook-cephfs     rook-ceph-block           3m
```

It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to **Running**. If your installation phase does not change to **Running**, then use the following command to find out which components are not ready:

```
oc get installation.orchestrator.aiops.ibm.com -o yaml -n cp4aiops | grep 'Not Ready'
```

Example output:

```
lifecycleservice: Not Ready
zenservice: Not Ready
```

To see details about why a component is **Not Ready** run the following command, where **<component>** is the component that is not ready, for example **zenservice**.

```
oc get <component> -o yaml -n cp4aiops
```

(Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

14. Log in to the IBM Cloud Pak for AIOps console

1. Find the password for the **admin** username by running the following command:

   ```
   oc -n cp4aiops get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d
   ```

2. Find the URL to access the IBM Cloud Pak for AIOps console with the following command.

   ```
   oc get route -n cp4aiops cpd -o jsonpath='{.spec.host}'
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

Based on the sample output, your console URL would be **`https://cpd-cp4aiops.apps.mycluster.mydomain`**

3. Enter the URL in your browser to open the IBM Cloud Pak for AIOps console and log in with a username of **`cpadmin`** and the password that you found in the previous step.

What to do next

- Define integrations and applications with Defining.
- You can integrate with IBM Cognos® Analytics. For more information, see Integrating IBM Cognos Analytics with IBM Cloud Pak for AIOps.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/OMNIbus integrations.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/Impact integrations.

Installing IBM Cloud Pak for AIOps (Production installation)

Depending on whether your OpenShift cluster is connected to the internet, choose either an online or an offline installation.

If you are installing on Azure Red Hat OpenShift (ARO), Google Cloud Platform (GCP), Red Hat® OpenShift® on IBM Cloud® service (ROKS), Red Hat OpenShift Service on AWS (ROSA), or AWS GovCloud, then review the information in Installing on cloud platforms.

Online installation

If your cluster is connected to the internet, then you can install IBM Cloud Pak® for AIOps using the Red Hat® OpenShift® Container Platform console or the command line interface (CLI). Choose one of the following topics.

- Online production installation of IBM Cloud Pak for AIOps (console method)
- Online production installation of IBM Cloud Pak for AIOps (CLI method)

Offline installation

If your Red Hat OpenShift cluster is not connected to the internet, then you can perform an airgapped (offline) installation.

- Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline)

Online production installation of IBM Cloud Pak for AIOps (console method)

If your cluster is connected to the internet, you can complete a production installation of IBM Cloud Pak® for AIOps with the Red Hat® OpenShift® Container Platform console.

Before you begin

- You must know whether you are deploying a base deployment or a extended deployment of IBM Cloud Pak for AIOps. For more information, see Incremental adoption.
- Review the Planning section.
- Online installations of IBM Cloud Pak for AIOps can be run entirely as a non-root user, and do not require that user to have sudo access.
- Some steps must still be run with the command line. Ensure that you are logged in to your Red Hat OpenShift cluster with **`oc login`** for any steps that use the Red Hat OpenShift command-line interface (CLI).
- The display names of some Red Hat OpenShift console components, such as window titles and push buttons, vary between Red Hat OpenShift versions. The following instructions are based on Red Hat OpenShift version 4.14 console components.
- If you require details about the permissions that the IBM Cloud Pak for AIOps operators need, see Permissions (IBM Cloud Pak for AIOps).
- A user with **`cluster-admin`** privileges is needed for the following operations:
  - Create a custom project
  - Create the catalog sources
  - Install Cert Manager
  - Install the License Service
  - Install the operator

Important: If IBM Sales representatives and Business Partners supplied you with a custom profile ConfigMap to customize your deployment, then you must follow their instructions to apply it during installation. The custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. For more information about custom sizing, see Custom sizing.

Installation procedure

Follow these steps to install IBM Cloud Pak for AIOps.

1. Install and configure Red Hat OpenShift
2. Configure storage

3. Create a custom project (namespace)
4. Create the entitlement key secret
5. Configure usage data collection
6. Create the catalog sources
7. Install Cert Manager
8. Install the License Service
9. Verify cluster readiness
10. Install the operator
11. Install IBM Cloud Pak for AIOps
12. Verify your installation
13. Create an EgressFirewall to restrict outgoing traffic
14. Access the IBM Cloud Pak for AIOps console

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the **`IBM Cloud Container Registry`** and IBM Cloud Pak® foundational services catalog source.

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

For more information, see Configuring your firewall for OpenShift Container Platform.

1. Install and configure Red Hat OpenShift

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running. You must have administrative access to your Red Hat OpenShift cluster.

For more information about supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.
2. Install the Red Hat OpenShift command line interface (**`oc`**) on your cluster's boot node and run **`oc login`**. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.
3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a **`MachineConfig`** custom resource to configure chrony to connect to your NTP servers.
4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:
   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

2. Configure storage

The storage configuration must satisfy your sizing requirements. For more information about the storage classes that are needed for installing IBM Cloud Pak for AIOps, see Storage.

Note: Storage classes and storage providers cannot be changed after you install IBM Cloud Pak for AIOps. OADP backup and restore requires that a ReadWriteMany (RWX) storage class must be provided. If OADP backup and restore is not needed, a ReadWriteOnce (RWO) storage class can be provided as the **`RWX-storage-class-name`** in the installation instance CR YAML file. This configuration cannot be changed after IBM Cloud Pak for AIOps is installed.

3. Create a custom project (namespace)

Create a project (namespace) to deploy IBM Cloud Pak for AIOps into.

A project is a Kubernetes namespace. You must create a custom project (namespace) and not use the **`default`**, **`kube-system`**, **`kube-public`**, **`openshift-node`**, **`openshift-infra`**, or **`openshift`** projects (namespaces). This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects (namespaces).

1. From your Red Hat OpenShift console, click **Home** > **Projects**.
2. Select **Create Project**, specify the **Name** of the project that you want to create, for example *cp4aiops* and click **Create**.

4. Create the entitlement key secret

Complete the following steps to create a registry secret to enable your deployment to pull the IBM Cloud Pak for AIOps images from the IBM® Entitled Registry.

1. Obtain the entitlement key that is assigned to your IBMid. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Active entitlement keys* section, select **Copy** to copy the entitlement key to the clipboard.
3. From your Red Hat OpenShift console, click **Workloads** > **Secrets**.
4. From the **Project** menu, select the project that you created earlier in Create a custom project (namespace).
5. Click the **Create** button, and select **Image pull secret** from the menu. The **Create image pull secret** form is displayed. Enter the following values and then click **Create**.
   - **Secret name**: `ibm-entitlement-key`
   - **Authentication type**: `Image registry credentials`
   - **Registry server address**: `cp.icr.io`
   - **Username**: `cp`
   - **Password**: use the entitlement key that you copied in step 2.

5. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The usage data is collected by the `cp4waiops-metricsprocessor` pod, and is sent to and stored in IBM controlled GDPR-compliant systems. The collection of usage data is enabled by default, but can be disabled. For transparency, the `cp4waiops-metricsprocessor` pod's logs contain all the information that is collected. The usage data that is collected is numeric, or is about the deployment type and platform. It does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of integrations of each type (For example ServiceNow, Instana, Falcon Logscale)
- Secure tunnel enablement: whether *connection* (which controls whether you can create a secure tunnel) is enabled in the *Installation* custom resource
- Deployment type: base deployment or extended deployment
- Deployment platform: Red Hat® OpenShift® Container Platform or Linux®

Use the following steps to configure or disable usage data collection.

1. From your Red Hat OpenShift console, click **Workloads** > **Secrets**.
2. From the **Project** menu, select the project that you created earlier in Create a custom project (namespace).
3. Click the **Create** button, and select **Key/value secret** from the menu. The **Create key/value secret** form is displayed. Enter the following values and then click **Create**.
   - **Secret name**: `aiops-metrics-processor`
   - Add the following **Key**/**Value** pairs:
     - **customerName**: your company name
     - **customerICN**: your IBM Customer Number (ICN)
     - **environment**: `trial` for testing, `poc` for proof of concept, or `production` for production environments.
   - If you want to disable usage data collection, also add the following key/value pair: **enableCollection**: false
4. If you have a firewall enabled, ensure that outbound traffic to https://api.segment.io is allowed.

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must create this secret with **enableCollection** set to false.

You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

6. Create the catalog sources

1. Run the following command to create the catalog sources for IBM Cloud Pak for AIOps and IBM Cloud Pak® foundational services Cert Manager and License Service in the *openshift-marketplace* namespace.

   Log in to your Red Hat OpenShift cluster's console. Click the plus icon in the upper right to open the **Import YAML** dialog box, paste in the following YAML, and then click **Create**.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-aiops-catalog
  namespace: openshift-marketplace
spec:
```

```
  displayName: ibm-aiops-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-aiops-catalog@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cert-manager-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-cert-manager
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-cert-manager-operator-catalog
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-licensing-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM License Service Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-licensing-catalog
```

2. Go to **Administration** > **Cluster Settings**. Under **Configuration** > **OperatorHub** > **Sources**, verify that the *ibm-aiops-catalog*, *ibm-cert-manager-catalog* and *ibm-licensing-catalog* `CatalogSource` objects are present.

7. Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Log in to your Red Hat OpenShift cluster's console.
2. Click **Operators** > **OperatorHub**. The **OperatorHub** page is displayed.
3. In the **All Items** field, enter `IBM Cert Manager`. The **IBM Cert Manager** operator is displayed.
4. Click the **IBM Cert Manager** tile. The **IBM Cert Manager** window is displayed.
5. Click **Install**. You see the **Install Operator** page.
6. Set the **Update Channel** to the `v4.2` version. If the Channel `v4.2` version is not available, click other **IBM Cert Manager** tile from OperatorHub to install the correct version.
7. Set **Installation Mode** to `All namespaces on the cluster (default)`.
8. Set **Installed Namespace** to `ibm-cert-manager(Operator recommended)`.
9. Set **Update approval** to `Automatic`.
10. Click **Install**.

8. Install the License Service

Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not know whether the License Service is already installed, then see Verifying if License Service is already installed on the cluster in the IBM Cloud Pak foundational services documentation.

IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

Follow the instructions in Installing the License Service with OpenShift console in the IBM Cloud Pak foundational services documentation, from step 2 `Create the ibm-licensing namespace` onwards.

9. Verify cluster readiness

Run the prerequisite checker script to verify whether your Red Hat OpenShift cluster is correctly set up for a IBM Cloud Pak for AIOps installation.

Download the prerequisite checker script from github.com/IBM, and run it with the following command:

```
./prereq.sh -n <project>
```

Where `<project>` is the project that your IBM Cloud Pak for AIOps installation is deployed in.

Important: If you are installing on a multi-zone cluster, then also specify the `-m` flag to assess whether there are sufficient resources to withstand a zone outage.

Example output:

```
# ./prereq.sh -n cp4aiops

[INFO] Starting IBM Cloud Pak for AIOps prerequisite checker v4.8...

CLI: oc

[INFO] ================================Platform Version Check================================
[INFO] Checking Platform Type....
[INFO] You are using Openshift Container Platform
[INFO] OCP version 4.16.7 is compatible but only nodes with AMD64 architectures are supported at this time.
[INFO] ================================Platform Version Check================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Cert Manager Check================================
[INFO] Checking for Cert Manager operator

[INFO] Successfully functioning cert-manager found.

CLUSTERSERVICEVERSION             NAMESPACE
ibm-cert-manager-operator.v4.2.8  ibm-cert-manager

[INFO] ================================Cert Manager Check================================

[INFO] ================================Licensing Service Operator Check================================
[INFO] Checking for Licensing Service operator

[INFO] Successfully functioning licensing service operator found.

CLUSTERSERVICEVERSION          NAMESPACE
ibm-licensing-operator.v4.2.8  ibm-licensing

[INFO] ================================Licensing Service Operator Check================================

[INFO] ================================Starter or Production Install Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource Summary================================================
[INFO]                                                Nodes     |     vCPU       |  Memory(GB)
[INFO] Starter (Non-HA) Base (available/required)       [  9 / 3 ]   [  144 / 47 ]    [  289 / 123 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 3 ]   [  144 / 55 ]    [  289 / 136 ]

[INFO] Production (HA) Base (available/required)        [  9 / 6 ]   [  144 / 136 ]   [  289 / 310 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 6 ]   [  144 / 162 ]   [  289 / 368 ]
[INFO] ================================Resource Summary================================================
[INFO] Cluster currently has resources available to create a Starter (Non-HA) install of Cloud Pak for AIOps

[INFO] ================================Prerequisite Checker Tool Summary================================
      [  PASS  ] Platform Version Check
      [  PASS  ] Storage Provider
      [  PASS  ] Starter (Non-HA) Base Install Resources
      [  FAIL  ] Production (HA) Base Install Resources
      [  PASS  ] Cert Manager Operator Installed
      [  PASS  ] Licensing Service Operator Installed
[INFO] ================================Prerequisite Checker Tool Summary================================
```

10. Install the operator

For more information about installing operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

Note: During the initial installation of Cloud Pak for AIOps, Kubernetes jobs might fail and re-run. If a job succeeds on the second or third attempt, there can be one or two pods in `Error` state and one pod in the `Completed` state. If a job fails repeatedly, the attempt is abandoned, and the logs from failed pods can be used to determine the cause of the failure. When you determine the cause for the failure, you can delete the job, and the operator can recreate it to reattempt the operations.

1. Log in to your Red Hat OpenShift cluster's console.
2. Click **Operators** > **OperatorHub**. The **OperatorHub** page is displayed.

3. In the **All Items** field, enter `IBM Cloud Pak for AIOps`. The **IBM Cloud Pak for AIOps** operator is displayed.
4. Click the **IBM Cloud Pak for AIOps** tile with the CatalogSource tag *ibm-aiops-catalog*. The **IBM Cloud Pak for AIOps** window is displayed.
5. Click **Install**. The **Install Operator** page is displayed.
6. Enter the following values:
   - Set **Update channel** to *v4.8*.
   - **Installation mode** - For more information about installation modes, see Operator installation mode.
   - **Installed Namespace** - If you are using the `OwnNamespace` installation mode (a specific namespace), then set this field to be the project (namespace) in which to install the operator, such as `cp4aiops`. If you are using the `AllNamespaces` installation mode, then set this field to `openshift-operators`.
   - Set **Update approval** to *Automatic*. This must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.
7. Click **Install** and wait for the IBM Cloud Pak for AIOps operator to install.
8. Verify that the IBM Cloud Pak for AIOps operator is successfully installed.

   Navigate to **Operators** > **Installed Operators**, and select your project from the **Projects** dropdown. **IBM Cloud Pak for AIOps** and its dependant operators in the project are listed with a **Status** of *Succeeded*.

11. Install IBM Cloud Pak for AIOps

Create an instance of the IBM Cloud Pak for AIOps custom resource. A maximum of one IBM Cloud Pak for AIOps custom resource is allowed per cluster.

1. From your Red Hat OpenShift console, click **Operators** > **Installed Operators**.
2. From the **Project** dropdown menu, select the project that you want to create the IBM Cloud Pak for AIOps instance in. Use the project that you created earlier in Create a custom project (namespace).

   Note: You cannot use the `default`, `kube-system`, `kube-public`, `openshift-node`, `openshift-infra`, or `openshift` projects. This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects.
3. Select **IBM Cloud Pak for AIOps** operator, then click the **IBM Cloud Pak for AIOps** tab.
4. Click **Create Installation**. The default **Form View** is displayed.

   Warning: The pakModules `aiopsFoundation`, `applicationManager`, and `aiManager` must be enabled. Do not change these values to false.

   Enter the following values:
   - **Name**: Specify the name that you want your IBM Cloud Pak for AIOps instance to be called, for example `ibm-cp-aiops`.
   - **license**: Expand the **license** section and read the agreement. Toggle the **License Acceptance** switch to **true** to accept the license.
   - **File Storage Class** and **Large Block Storage Class** are the storage classes that you want to use, as detailed in the following table. For more information about storage, see Storage.
   - **Enable Secure Tunnel**: Set to true if you want to install Secure Tunnel. For more information about Secure Tunnel, see Secure Tunnel.
   - **Image Pull Secret**: Select the `ibm-entitlement-key` secret that you created in the step Create the entitlement key secret.
   - **Resource Overrides ConfigMap** Do not edit this field unless you have been supplied with a custom `ConfigMap` by an IBM Sales representative (or Business Partner).
   - **Size**: Select the size that you require for your IBM Cloud Pak for AIOps installation.
   - **Topology resource group terminology**: Specify `application` or `service` as the terminology to be used for collections of topology resource groups. The default is `application`.
   - **Zones**: If you are installing on a multi-zone cluster, then add the names of the zones that you are using. The zone names must exactly match the zone labels that you applied to each of your nodes in step 1a of Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HA).

     Note: To confirm that you have the storage classes for your chosen storage provider as shown in the table, run `oc get sc`.

Table 1. Storage provider classes

Storage provider	File Storage Class	Large Block Storage Class
IBM Cloud®	ibmc-block-gold	ibmc-block-gold
IBM Cloud® with OADP backup and restore	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
Red Hat® OpenShift® Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Global Data Platform	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc

Storage provider	File Storage Class	Large Block Storage Class
IBM Storage Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	px-csi-aiops	px-csi-aiops
Portworx (multi-zone HA)	px-csi-aiops-mz	px-csi-aiops-mz
AWS native storage	gp3-csi	gp3-csi
AWS native storage with OADP backup and restore	efs-sc	gp3-csi

5. Configure your deployment to be a base deployment or an extended deployment.

   For more information about these deployment types, see Incremental adoption.

   Your deployment defaults to a base deployment which does not have log anomaly detection and ticket analysis capabilites. If you want a base deployment, then proceed to the next step.

   If you want an extended deployment with log anomaly detection and ticket analysis capabilites, then switch to the **YAML** view and set the value of **enabled** to **`true`**.

   Example excerpt:

   ```
   spec:
     pakModules:
     - enabled: true
       name: logAnomalyDetection
   ```

   You can update your deployment type after installation. For more information, see Updating the deployment type.

6. Click **Create** to create a custom resource that is an instance of IBM Cloud Pak for AIOps.

12. Verify your installation

1. After a few minutes, use the following steps to check the status of your installation. Click **Operators** > **Installed Operators**.
2. From the **Project** list, select the project (namespace) that IBM Cloud Pak for AIOps is deployed in.
3. Select **IBM Cloud Pak for AIOps** and then click the **IBM Cloud Pak for AIOps** tab.
4. Under **Installations**, look for the entry with the name that you specified for your IBM Cloud Pak for AIOps instance, and verify that it has a **Status** of *Phase: Updating*. It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **Status** changes to *Phase: Running*.
5. (Optional) If you want to see more detail about the status of the installation's components, select the entry with the name that you specified for your IBM Cloud Pak for AIOps instance, and then switch to the **YAML** view. Scroll down to the **Status** section near the end of the YAML. A component's installation is complete and successful when the component has a value of **`Ready`**.

   Example YAML:

   ```
   status:
     size: small
     customProfileConfigmap: aiops-custom-size-profile
     customProfileValidationStatus: >-
       Custom profile configmap not found, continue installation process without customization
     storageclasslargeblock: rook-ceph-rbd
     componentstatus:
       issueresolutioncore: Ready
       kafka: Ready
       aiopsanalyticsorchestrator: Ready
       aiopsedge: Ready
       tunnel: Ready
       lifecycleservice: Ready
       zenservice: Ready
       flinkcluster: Ready
       cluster: Ready
       elasticsearchcluster: Ready
       aiopsui: Ready
       redissentinel: Ready
       <...>
   ```

(Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems

13. Create an EgressFirewall

There is no egress firewall policy defined when you install IBM Cloud Pak for AIOps, so outgoing traffic from workload pods to the internal and external network is unrestricted.

To create a more secure environment, use the following steps.

1. Create an **`EgressFirewall`** on your Red Hat OpenShift cluster to limit egress from the IBM Cloud Pak for AIOps project (namespace).

   For more information about creating an **`EgressFirewall`**, see Configuring an egress firewall for a project.

   Note: You can only have one **`EgressFirewall`** per project/namespace.

2. Configure exceptions to the **`EgressFirewall`**.

   Edit your **`EgressFirewall`** to add exceptions for the following IBM Cloud Pak for AIOps components that have egress dependencies, otherwise these IBM Cloud Pak for AIOps components fail when they attempt egress.

   1. Allow egress to any external services, such as the following integrations:
      - Kubernetes
      - GitHub
      - Microsoft® Teams
      - ServiceNow
      - Slack
      - VMware® vCenter
   2. Configure your **`EgressFirewall`** to allow traffic for your GitHub, Kubernetes, ServiceNow, and VMware vCenter integrations.

      Edit your **`EgressFirewall`** to allow or deny egress, as in the following example:

```
kind: EgressFirewall
metadata:
  name: default
spec:
  egress:
  - type: Allow
    to:
      cidrSelector: <1.2.3.0/24>
  - type: Allow
    to:
      dnsName: <www.github.com>
  - type: Allow
    to:
      dnsName: <www.developer.kubernetes.com>
  - type: Allow
    to:
      dnsName: <www.developer.servicenow.com>
  - type: Allow
    to:
      dnsName: <www.developer.vcenter.com>
  - type: Deny
    to:
      cidrSelector: <0.0.0.0/0>
```

      Where the values you enter for **`dnsName`** and **`cidrSelector`** are the DNS names and addresses of your GitHub, Kubernetes, ServiceNow, or VMware vCenter sources.

14. Access the IBM Cloud Pak for AIOps UI

After you successfully install IBM Cloud Pak for AIOps, you can use the IBM Cloud Pak Administration panel to manage the underlying deployment, or use the IBM Cloud Pak for AIOps console to use IBM Cloud Pak for AIOps.

IBM Cloud Pak Administration panel

You can use the **Launch Admin Hub** link to access the IBM Cloud Pak Administration panel:

1. Log in to the Red Hat OpenShift Container Platform web console as an administrator.
2. Click **Operators** > **Installed Operators**.
3. Click **IBM Cloud Pak for AIOps**.
4. On the *Operator Details* page, click the **IBM Cloud Pak for AIOps** tab, and then click the IBM Cloud Pak for AIOps installation name.
5. In the Details tab, right-click on the URL underneath **Launch Admin Hub**, and select **Open Link in New Tab**.
6. On the IBM Cloud Pak Administration panel login page, select one of the following login options:
   - **OpenShift authentication**: The **`kubeadmin`** user is automatically used to log in to the Administration panel. The **`kubeadmin`** user has the same privileges as the Administration panel admin user.
   - **IBM provided credentials (admin only)**: The default username to access the console is **`admin`**. To obtain the username and password, see Obtain IBM provided credentials (admin only).

Cloud Pak for AIOps console

You can use the **Launch Cloud Pak in IBM Automation** link to access the Cloud Pak for AIOps console:

1. Log in to the Red Hat OpenShift Container Platform web console as an administrator.
2. Click **Operators** > **Installed Operators**.
3. Click **IBM Cloud Pak for AIOps**.

4. On the *Operator Details* page, click the **IBM Cloud Pak for AIOps** tab, and then click the IBM Cloud Pak for AIOps installation name.
5. In the **Details** tab, right-click on the URL underneath **Launch Cloud Pak in IBM Automation**, and select **Open Link in New Tab**.
6. In the Cloud Pak for AIOps console login page, select one of the following login options:
   - **OpenShift authentication**: The `kubeadmin` user is automatically used to log in to the Cloud Pak for AIOps console. The `kubeadmin` user has the same privileges as the Cloud Pak for AIOps console admin user.
   - **IBM provided credentials (admin only)**: The default username to access the console is `admin`. To obtain the username and password, see Obtain IBM provided credentials (admin only).
   - **Enterprise LDAP**: LDAP users can log in to the Cloud Pak for AIOps console after IBM Cloud Pak for AIOps is configured with a single or multiple LDAP servers for the authentication and authorization. For more information, see Identity Management (IM).

Obtain IBM provided credentials (admin only)

1. To find the default username, run the following command:

   ```
   oc -n <project> get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_username}' | base64 -d && echo
   ```

   Where `<project>` is the project (namespace) that IBM Cloud Pak for AIOps is deployed in.

2. To get the password for the *admin* username, run the following command:

   ```
   oc -n <project> get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d
   ```

   Where `<project>` is the project (namespace) that IBM Cloud Pak for AIOps is deployed in.

   The following is a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the sample output, your password would be `EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA`.

   Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

What to do next

- Define integrations and applications with Defining.
- You can integrate with IBM Cognos® Analytics. For more information, see Integrating IBM Cognos Analytics with IBM Cloud Pak for AIOps.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/OMNIbus integrations.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/Impact integrations.
- Familiarize yourself with backup and restore procedures. It is recommended that you take regular backups of your IBM Cloud Pak for AIOps deployment. For more information, see Backup and restore.
- For more information about health checks and monitoring, see Health checks and monitoring.

Online production installation of IBM Cloud Pak for AIOps (CLI method)

If your cluster is connected to the internet, you can complete a production installation of IBM Cloud Pak® for AIOps with the Red Hat® OpenShift® Container Platform command line interface (CLI).

Before you begin

- You must know whether you are deploying a base deployment or a extended deployment of IBM Cloud Pak for AIOps. For more information, see Incremental adoption.
- Review the Planning section.
- Online installations of IBM Cloud Pak for AIOps can be run entirely as a non-root user, and do not require that user to have sudo access.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that the IBM Cloud Pak for AIOps operators need, see Permissions (IBM Cloud Pak for AIOps).
- A user with `cluster-admin` privileges is needed for the following operations:
  - Create a custom project
  - Install Cert Manager
  - Install the License Service
  - Create the catalog sources
  - Create an OperatorGroup

Important: If IBM Sales representatives and Business Partners supplied you with a custom profile ConfigMap to customize your deployment, then you must follow their instructions to apply it during installation. The custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. For more information about custom sizing, see Custom sizing.

Installation procedure

Follow these steps to install IBM Cloud Pak for AIOps.

1. Install and configure Red Hat OpenShift
2. Configure storage
3. Retrieve your entitlement key
4. Create environment variables
5. Create a custom project (namespace)
6. Create the entitlement key secret
7. Configure usage data collection
8. Create the catalog sources
9. Install Cert Manager
10. Install the License Service
11. Verify cluster readiness
12. Install the operator
13. Install IBM Cloud Pak for AIOps
14. Verify your installation
15. Create an EgressFirewall to restrict outgoing traffic
16. Access the Cloud Pak for AIOps console

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the **`IBM Cloud Container Registry`** and IBM Cloud Pak® foundational services catalog source.

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

For more information, see Configuring your firewall for OpenShift Container Platform.

1. Install and configure Red Hat OpenShift

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.
2. Install the Red Hat OpenShift command line interface (**`oc`**) on your cluster's boot node and run **`oc login`**. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.
3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a **`MachineConfig`** custom resource to configure chrony to connect to your NTP servers.
4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:
   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

2. Configure storage

Configure persistent storage for IBM Cloud Pak for AIOps that satisfies your sizing requirements. For more information about storage, see Storage.

Note: Storage classes and storage providers cannot be changed after you install IBM Cloud Pak for AIOps. OADP backup and restore requires that a ReadWriteMany (RWX) storage class must be provided. If OADP backup and restore is not needed, a ReadWriteOnce (RWO) storage class can be provided as the **`RWX-storage-class-name`** in the installation instance CR YAML file. This configuration cannot be changed after IBM Cloud Pak for AIOps is installed.

3. Retrieve your entitlement key

Obtain the IBM entitlement key that is assigned to your **IBMid**, which is needed to pull the IBM Cloud Pak for AIOps images from the IBM® Entitled Registry.

1. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Entitlement key* section, select **Copy** to copy the entitlement key to the clipboard. You will need to use this value to set the environment variable $IBM_ENTITLEMENT_KEY in the next step.

4. Create environment variables

Create and then source a shell script named *waiops_var.sh* that defines the environment variables that will be used to provide installation parameters for your deployment. Use the following codeblock as a template, replacing the brackets **< ... >** with values for your environment.

You can use the following table to find the values to set for your storage environment variables.

Table 1. Storage provider classes

Storage provider	RWX-storage-class-name	RWO-storage-class-name
IBM Cloud®	ibmc-block-gold	ibmc-block-gold
IBM Cloud® with OADP backup and restore	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
Red Hat® OpenShift® Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Global Data Platform	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc
IBM Storage Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	px-csi-aiops	px-csi-aiops
Portworx (multi-zone HA)	px-csi-aiops-mz	px-csi-aiops-mz
AWS native storage	gp3-csi	gp3-csi
AWS native storage with OADP backup and restore	efs-sc	gp3-csi

Note: To confirm that you have the storage classes for your chosen storage provider as shown in the table, run **oc get sc**.

```
#============================================================================================================================
# Cloud Pak for AIOps installation variables
#============================================================================================================================
export CP4AIOPS_NAME=ibm-cp-aiops
export CP4AIOPS_SIZE=large # Set to `small` if you only require a starter, non-production deployment.
export SECURE_TUNNEL=false # Set to `true` to install Secure Tunnel, otherwise set to `false`.
export PROJECT_CP4AIOPS=cp4aiops
export ACCEPT_LICENSE=false # Set to `true` to agree to the license terms, otherwise install will fail.
export CATALOG_SRC_CP4AIOPS=ibm-aiops-catalog

# ----------------------------------------------------------------------------------------------------------------------------
# Incremental adoption - set your deployment type
# Set to `true` to install an extended deployment with log anomaly detection and ticket analysis capabilities
# Set to `false` to install a base deployment without log anomaly detection and ticket analysis capabilities
# ----------------------------------------------------------------------------------------------------------------------------
export LOG_ANOMALY=false

# ----------------------------------------------------------------------------------------------------------------------------
# Storage
# ----------------------------------------------------------------------------------------------------------------------------
export STG_CLASS=<RWX-storage-class-name>
export STG_CLASS_BLOCK=<RWO-storage-class-name>

# ----------------------------------------------------------------------------------------------------------------------------
# Your customer details
# ----------------------------------------------------------------------------------------------------------------------------
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>

# ----------------------------------------------------------------------------------------------------------------------------
# IBM Entitled Registry
# ----------------------------------------------------------------------------------------------------------------------------
export IBM_ENTITLEMENT_KEY=<IBM-entitlement-key> # Set to the entitlement key retrieved in previous step.

# ----------------------------------------------------------------------------------------------------------------------------
# `OwnNamespace` installation mode:  leave INSTALL_MODE_NAMESPACE as it is.
# `AllNamespaces` installation mode: change to export INSTALL_MODE_NAMESPACE=openshift-operators
# ----------------------------------------------------------------------------------------------------------------------------
export INSTALL_MODE_NAMESPACE=${PROJECT_CP4AIOPS}

# ----------------------------------------------------------------------------------------------------------------------------
# Topology resource group terminology
# Specify `application` or `service` as the terminology to be used for collections of topology resource
```

```
# groups. The default is `application`.
# ----------------------------------------------------------------------------------------------------------
export TOPOLOGY_TERMINOLOGY=application
```

If you need help with deciding on the values to set for these environment variables, see the following topics.

- LOG_ANOMALY: Incremental adoption
- CP4AIOPS_SIZE: Sizing
- SECURE_TUNNEL: Secure Tunnel
- INSTALL_MODE_NAMESPACE: Operator installation mode

You can update your deployment type after installation. For more information, see Updating the deployment type.

Note: You can set a different value for $PROJECT_CP4AIOPS and $CP4AIOPS_NAME if you want. However, you must not use the **default**, **kube-system**, **kube-public**, **openshift-node**, **openshift-infra**, or **openshift** projects (namespaces) for $PROJECT_CP4AIOPS. This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects (namespaces).

Run the following command to source your script and set the environment variables:

```
. ./waiops_var.sh
```

5. Create a custom project (namespace)

Run the following command to create a project (namespace) to deploy IBM Cloud Pak for AIOps into.

```
oc create namespace "${PROJECT_CP4AIOPS}"
```

6. Create the entitlement key secret

Run the following command to create an image pull secret called *ibm-entitlement-key* to enable your deployment to pull the IBM Cloud Pak for AIOps images from the IBM Entitled Registry.

```
oc create secret docker-registry ibm-entitlement-key --docker-username=cp --docker-password="${IBM_ENTITLEMENT_KEY}" --docker-server=cp.icr.io --namespace=${PROJECT_CP4AIOPS}
```

7. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The usage data is collected by the **cp4waiops-metricsprocessor** pod, and is sent to and stored in IBM controlled GDPR-compliant systems. The collection of usage data is enabled by default, but can be disabled. For transparency, the **cp4waiops-metricsprocessor** pod's logs contain all the information that is collected. The usage data that is collected is numeric, or is about the deployment type and platform. It does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of integrations of each type (For example ServiceNow, Instana, Falcon Logscale)
- Secure tunnel enablement: whether *connection* (which controls whether you can create a secure tunnel) is enabled in the *Installation* custom resource
- Deployment type: base deployment or extended deployment
- Deployment platform: Red Hat® OpenShift® Container Platform or Linux®

Configuring the collection of usage data

If you do not want to disable the collection of usage data, run the following command to configure the usage data with your customer details.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT}
```

If you have a firewall enabled, ensure that outbound traffic to https://api.segment.io is allowed.

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must run the command that is given in *Disabling the collection of usage data*.

Disabling the collection of usage data

If you want to disable the collection of usage data, run the following command.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT} --from-literal=enableCollection=false
```

Note: You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

8. Create the catalog sources

1. Run the following command to create the catalog sources for IBM Cloud Pak for AIOps and IBM Cloud Pak® foundational services Cert Manager and License Service in the *openshift-marketplace* namespace.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ${CATALOG_SRC_CP4AIOPS}
  namespace: openshift-marketplace
spec:
  displayName: ${CATALOG_SRC_CP4AIOPS}
  publisher: IBM Content
  sourceType: grpc
  image:
icr.io/cpopen/${CATALOG_SRC_CP4AIOPS}@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cert-manager-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-cert-manager
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-cert-manager-operator-catalog
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-licensing-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM License Service Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-licensing-catalog
EOF
```

2. Verify that the *ibm-aiops-catalog*, *ibm-cert-manager-catalog* and *ibm-licensing-catalog* **CatalogSource** objects are in the output that is returned by the following command:

   ```
   oc get CatalogSources -n openshift-marketplace
   ```

   Example output:

   ```
   oc get CatalogSources -n openshift-marketplace
   NAME                     DISPLAY                     TYPE   PUBLISHER   AGE
   ibm-aiops-catalog        ibm-aiops-catalog           grpc   IBM         2m
   ibm-cert-manager-catalog ibm-cert-manager            grpc   IBM         2m
   ibm-licensing-catalog    IBM License Service Catalog grpc   IBM         2m
   ```

9. Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Run the following command to create the resource definitions that you need:

   ```
   cat << EOF | oc apply -f -
   apiVersion: v1
   kind: Namespace
   metadata:
     name: ibm-cert-manager
   ---
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: ibm-cert-manager-operator-group
     namespace: ibm-cert-manager
   ---
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-cert-manager-operator
     namespace: ibm-cert-manager
   spec:
     channel: v4.2
     installPlanApproval: Automatic
     name: ibm-cert-manager-operator
     source: ibm-cert-manager-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-cert-manager get pods
```

Example output for a successful IBM Cloud Pak® foundational services Cert Manager installation:

```
NAME                                       READY   STATUS    RESTARTS   AGE
cert-manager-cainjector-674854c49d-vstq4   1/1     Running   0          8d
cert-manager-controller-646d4bd6fd-zwmqm   1/1     Running   0          8d
cert-manager-webhook-8598787c8-s41kt       1/1     Running   0          8d
ibm-cert-manager-operator-c96957695-dkxnm  1/1     Running   0          8d
```

10. Install the License Service

Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-licensing
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-licensing-operator-group
  namespace: ibm-licensing
spec:
  targetNamespaces:
  - ibm-licensing
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-licensing-operator-app
  namespace: ibm-licensing
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-licensing-operator-app
  source: ibm-licensing-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

```
oc -n ibm-licensing get pods
```

Example output for a successful IBM Cloud Pak® foundational services License Service installation:

```
NAME                                              READY   STATUS    RESTARTS   AGE
ibm-licensing-operator-db4cd746c-xzmlf            1/1     Running   0          8d
ibm-licensing-service-instance-596b99588f-76cc5   1/1     Running   0          8d
```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

11. Verify cluster readiness

Run the prerequisite checker script to verify whether your Red Hat OpenShift cluster is correctly set up for a IBM Cloud Pak for AIOps installation.

Download the prerequisite checker script from github.com/IBM, and run it with the following command:

```
./prereq.sh -n ${PROJECT_CP4AIOPS}
```

Important: If you are installing on a multi-zone cluster, then also specify the `-m` flag to assess whether there are sufficient resources to withstand a zone outage.

Example output:

```
# ./prereq.sh -n cp4aiops

[INFO] Starting IBM Cloud Pak for AIOps prerequisite checker v4.8...

CLI: oc

[INFO] ================================Platform Version Check================================
[INFO] Checking Platform Type....
[INFO] You are using Openshift Container Platform
[INFO] OCP version 4.16.7 is compatible but only nodes with AMD64 architectures are supported at this time.
```

```
[INFO] ================================Platform Version Check================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Cert Manager Check================================
[INFO] Checking for Cert Manager operator

[INFO] Successfully functioning cert-manager found.

CLUSTERSERVICEVERSION             NAMESPACE
ibm-cert-manager-operator.v4.2.8  ibm-cert-manager

[INFO] ================================Cert Manager Check================================

[INFO] ================================Licensing Service Operator Check================================
[INFO] Checking for Licensing Service operator

[INFO] Successfully functioning licensing service operator found.

CLUSTERSERVICEVERSION          NAMESPACE
ibm-licensing-operator.v4.2.8  ibm-licensing

[INFO] ================================Licensing Service Operator Check================================

[INFO] ================================Starter or Production Install Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource Summary================================================
[INFO]                                                  Nodes     |     vCPU       |  Memory(GB)
[INFO] Starter (Non-HA) Base (available/required)       [  9 / 3 ]   [  144 / 47 ]    [  289 / 123 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 3 ]   [  144 / 55 ]    [  289 / 136 ]

[INFO] Production (HA) Base (available/required)        [  9 / 6 ]   [  144 / 136 ]   [  289 / 310 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 6 ]   [  144 / 162 ]   [  289 / 368 ]
[INFO] ================================Resource Summary================================================
[INFO] Cluster currently has resources available to create a Starter (Non-HA) install of Cloud Pak for AIOps

[INFO] ================================Prerequisite Checker Tool Summary================================
      [  PASS  ] Platform Version Check
      [  PASS  ] Storage Provider
      [  PASS  ] Starter (Non-HA) Base Install Resources
      [  FAIL  ] Production (HA) Base Install Resources
      [  PASS  ] Cert Manager Operator Installed
      [  PASS  ] Licensing Service Operator Installed
[INFO] ================================Prerequisite Checker Tool Summary================================
```

12. Install the operator

For more information about installing operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

Note: During the initial installation of Cloud Pak for AIOps, Kubernetes jobs might fail and re-run. If a job succeeds on the second or third attempt, there can be one or two pods in `Error` state and one pod in the `Completed` state. If the job fails repeatedly, the attempt is abandoned, and the logs from failed pods are used to determine the cause of the failure. When you determine the cause for the failure, you can delete the job, and the operator can recreate it to reattempt the operations.

1. Create an OperatorGroup.

   Important: Skip this step if you are installing using the 'All Namespaces' installation mode. Check that you set `INSTALL_MODE_NAMESPACE` correctly in step 4, and proceed to the next step, Install the IBM Cloud Pak for AIOps operator.

   If you are installing using the 'OwnNamespace' installation mode, then you must create an operator group in your custom project (namespace), or the IBM Cloud Pak for AIOps operator will not install. There might be an operator group for managing a namespace for given APIs. If there is an operator group for the namespace, do not create a second one.

   Create the operator group by running the following command:

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: cp4aiops-operator-group
   ```

```
  namespace: ${PROJECT_CP4AIOPS}
spec:
  targetNamespaces:
    - "${PROJECT_CP4AIOPS}"
EOF
```

2. Install the IBM Cloud Pak for AIOps operator.

   Run the following command.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-aiops-orchestrator
  namespace: $INSTALL_MODE_NAMESPACE
spec:
  channel: v4.8
  installPlanApproval: Automatic
  name: ibm-aiops-orchestrator
  source: ${CATALOG_SRC_CP4AIOPS}
  sourceNamespace: openshift-marketplace
EOF
```

   Warning: installPlanApproval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

3. After a few minutes, the IBM Cloud Pak for AIOps operator is installed. Verify that the all of the components have a state of **Succeeded** by running the following command:

```
oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator"
```

   Example output:

```
$ oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator"
ibm-aiops-orchestrator.v4.8.0         IBM Cloud Pak for AIOps                4.8.0   Succeeded
```

13. Install IBM Cloud Pak for AIOps

Create an instance of the IBM Cloud Pak for AIOps custom resource. A maximum of one IBM Cloud Pak for AIOps custom resource is allowed per cluster.

Use one of the following YAML codeblocks to create an instance of the IBM Cloud Pak for AIOps custom resource, depending on whether you are installing on a multi-zone cluster.

Installing on a non multi-zone cluster:

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ${CP4AIOPS_NAME}
  namespace: ${PROJECT_CP4AIOPS}
spec:
  size: ${CP4AIOPS_SIZE}
  storageClass: ${STG_CLASS}
  storageClassLargeBlock: ${STG_CLASS_BLOCK}
  imagePullSecret: ibm-entitlement-key
  topologyModel: ${TOPOLOGY_TERMINOLOGY}
  license:
    accept: ${ACCEPT_LICENSE}
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: ${SECURE_TUNNEL}
  - name: logAnomalyDetection
    enabled: ${LOG_ANOMALY}
EOF
```

Installing on a multi-zone cluster:

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ${CP4AIOPS_NAME}
  namespace: ${PROJECT_CP4AIOPS}
spec:
  size: ${CP4AIOPS_SIZE}
  storageClass: ${STG_CLASS}
  storageClassLargeBlock: ${STG_CLASS_BLOCK}
  imagePullSecret: ibm-entitlement-key
  topologyModel: ${TOPOLOGY_TERMINOLOGY}
```

```
  license:
    accept: ${ACCEPT_LICENSE}
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: ${SECURE_TUNNEL}
  - name: logAnomalyDetection
    enabled: ${LOG_ANOMALY}
  zones:
  - name: <zone_name1>
  - name: <zone_name2>
  - name: <zone_name3>
EOF
```

Where **<zone_name1>**, **<zone_name2>**, and **<zone_name3>** exactly match the zone labels that you applied to each of your nodes in step 1a of Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HA).

Warning: The pakModules **aiopsFoundation**, **applicationManager**, and **aiManager** must be enabled as in the preceding YAML. Do not change these values to false.

14. Verify your installation

Run the following command to check that the **PHASE** of your installation is **Updating**.

```
oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
```

Example output:

```
NAME                  PHASE      LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
ibm-cp-aiops          Updating   Accepted   rook-cephfs     rook-ceph-block           3m
```

It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to **Running**. If your installation phase does not change to **Running**, then use the following command to find out which components are not ready:

```
oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not Ready'
```

Example output:

```
lifecycleservice: Not Ready
zenservice: Not Ready
```

To see details about why a component is **Not Ready** run the following command, where **<component>** is the component that is not ready, for example **zenservice**.

```
oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
```

(Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

15. Create an EgressFirewall

There is no egress firewall policy defined when you install IBM Cloud Pak for AIOps, so outgoing traffic from workload pods to the internal and external network is unrestricted.

To create a more secure environment, use the following steps.

1. Create an **EgressFirewall** on your Red Hat OpenShift cluster to limit egress from the IBM Cloud Pak for AIOps project (namespace).

   For more information about creating an **EgressFirewall**, see Configuring an egress firewall for a project.

   Note: You can have only one **EgressFirewall** per project/namespace.

2. Configure exceptions to the **EgressFirewall**.

   Edit your **EgressFirewall** to add exceptions for the following IBM Cloud Pak for AIOps components that have egress dependencies, otherwise these components fail when they attempt egress.

   1. Allow egress to any external services, such as the following integrations:
      - Kubernetes
      - GitHub
      - Microsoft® Teams
      - ServiceNow
      - Slack
      - VMware® vCenter

2. Configure your **EgressFirewall** to allow traffic for your GitHub, Kubernetes, ServiceNow, and VMware vCenter integrations.

   Edit your **EgressFirewall** to allow or deny egress, as in the following example. Substitute values for **dnsName** and **cidrSelector** that are the DNS names and addresses of your GitHub, Kubernetes, ServiceNow, or VMware vCenter sources.

```
kind: EgressFirewall
metadata:
  name: default
spec:
  egress:
  - type: Allow
    to:
      cidrSelector: <1.2.3.0/24>
  - type: Allow
    to:
      dnsName: <www.github.com>
  - type: Allow
    to:
      dnsName: <www.developer.kubernetes.com>
  - type: Allow
    to:
      dnsName: <www.developer.servicenow.com>
  - type: Allow
    to:
      dnsName: <www.developer.vcenter.com>
  - type: Deny
    to:
      cidrSelector: <0.0.0.0/0>
```

16. Access the Cloud Pak for AIOps console

After you successfully install IBM Cloud Pak for AIOps, get the URL for accessing the Cloud Pak for AIOps console.

1. Use the following command to get the URL to access the Cloud Pak for AIOps console:

   ```
   oc get route -n ${PROJECT_CP4AIOPS} cpd -o jsonpath='{.spec.host}'
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

   Based on the sample output, your console URL would be **https://cpd-cp4aiops.apps.mycluster.mydomain**

2. Enter the URL in your browser to open the Cloud Pak for AIOps console. Log in with your username and password.

Find the IBM Cloud Pak for AIOps console username and password

The default username to access the Cloud Pak for AIOps console is **admin**. You can check the default username and their password with the following commands.

Note: This information is for the **IBM provided credentials (admin only)** authentication type.

1. Find the default username.

   ```
   oc -n ${PROJECT_CP4AIOPS} get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_username}' | base64 -d
   && echo
   ```

2. Get the password for the **admin** username.

   ```
   oc -n ${PROJECT_CP4AIOPS} get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d
   ```

   The following extract shows a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the sample output, your password would be **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA**.

Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

What to do next

- Define integrations and applications with Defining.
- You can integrate with IBM Cognos® Analytics. For more information, see Integrating IBM Cognos Analytics with IBM Cloud Pak for AIOps.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/OMNIbus integrations.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/Impact integrations.
- Familiarize yourself with backup and restore procedures. It is recommended that you take regular backups of your IBM Cloud Pak for AIOps deployment. For more information, see Backup and restore.
- For more information about health checks and monitoring, see Health checks and monitoring.

Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline)

If your cluster is not connected to the internet, you can install IBM Cloud Pak® for AIOps in your cluster by using either a bastion host, portable compute device, or a portable storage device.

The IBM Catalog Management Plug-in for IBM Cloud Pak® is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering IBM product images, uses standard tools for registry and cluster access, and extends the Red Hat OpenShift CLI (`oc`) capability to streamline the air-gapped deployment process.

Overview

It is common in production to have a cluster that does not have internet access. In these cases, you can still install IBM Cloud Pak for AIOps and Red Hat® OpenShift® Container Platform in an air-gapped (otherwise known as offline or disconnected) environment. Unlike online installations, air-gapped installations require you to enable the IBM Operator Catalog to mimic a typical online installation by using images in your own registry.

The following diagram provides an overview of air-gapped installation scenarios for portable compute, portable storage, and bastion hosts:

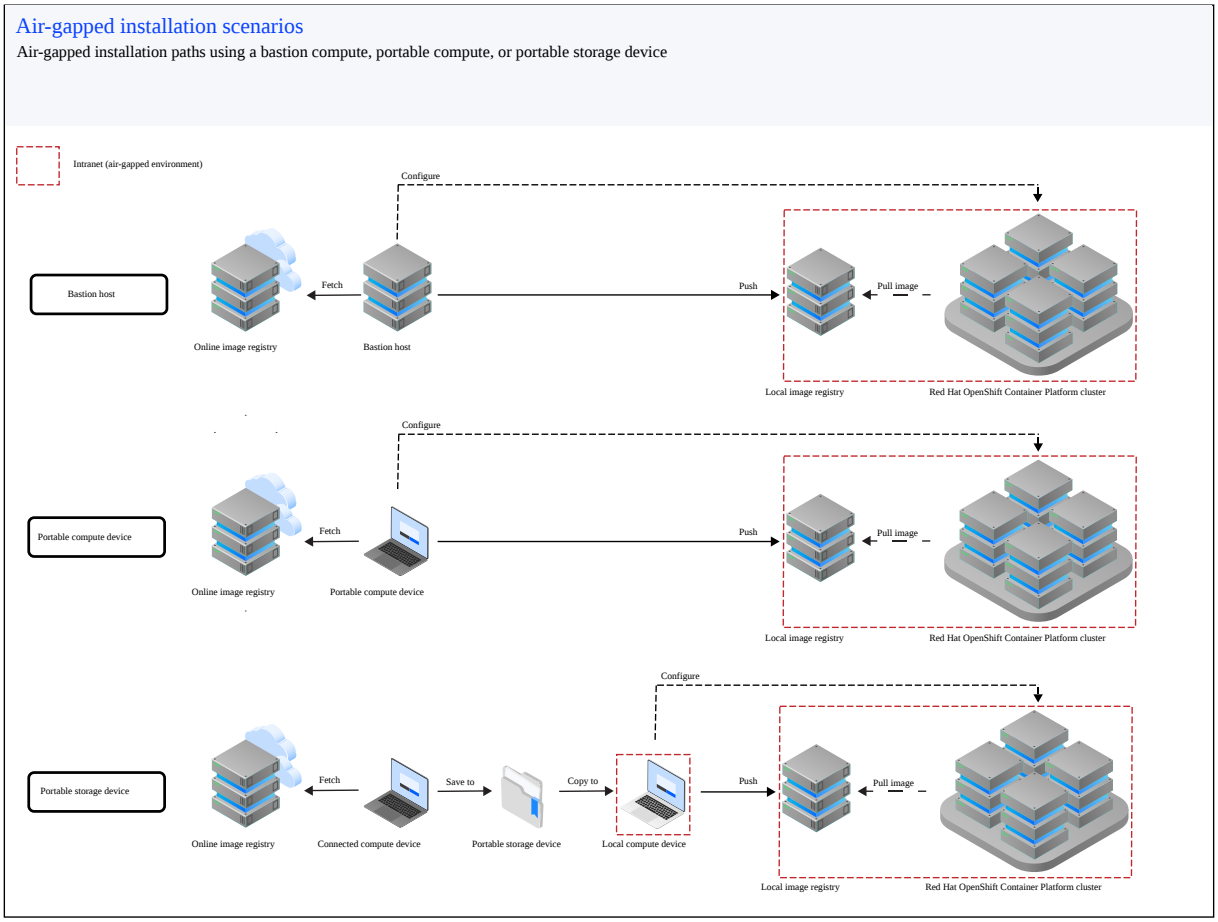


Figure 1. Air-gap scenario

All of these scenarios use Container Application Software for Enterprises (CASE) files. CASE is a specification that defines the metadata and structure for packaging, managing, and unpacking containerized applications.

You can store the product code and images to a portable compute device (like a laptop), portable storage device (like an external hard disk drive), or a bastion host and then transfer them to a local air-gapped network. The images that are stored are large, and you must ensure that each of your airgap devices have enough storage, as follows.

- Bastion host method - the local target registry must have at least 140 GB of storage space.
- Portable device methods - the portable device and the local target registry must each have at least 140 GB of storage space.

Table 1. Air-gapped installation scenarios

Air-gapped installation approach	Example	Description
Bastion host	Bastion host	The bastion host has access to the public internet and the air-gapped environment where a local target registry and a Red Hat OpenShift Container Platform cluster are. The IBM Cloud Pak for AIOps images are replicated from the IBM Entitled Registry through the bastion server directly to the local target registry behind the firewall.
Portable compute device	Laptop	IBM Cloud Pak for AIOps images are downloaded from the IBM Entitled Registry to a file system on a portable compute device. The portable compute device is then disconnected from the internet and brought into the air-gapped environment so that the images can be copied from the file system to the local target registry behind the firewall.

Air-gapped installation approach	Example	Description
Portable storage device	Portable hard disk drive	IBM Cloud Pak for AIOps images are downloaded from the IBM Entitled Registry through an internet connected device to a file system on a portable storage device. The portable storage device is then disconnected and brought into the air-gapped environment so that the images can be copied from the file system to the local target registry behind the firewall.

Installation methods

Choose one of the following deployment methods to complete an air-gapped installation.

- Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a bastion host
- Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a portable device

Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a bastion host

If your cluster is not connected to the internet, you can deploy a production installation of IBM Cloud Pak® for AIOps on your Red Hat® OpenShift® Container Platform cluster by using a bastion host.

In this scenario, your air-gapped (offline) environment has a target registry, and a Red Hat OpenShift cluster on which IBM Cloud Pak for AIOps is to be installed. The bastion host has access to the internet and to the offline environment. IBM Cloud Pak for AIOps images are mirrored from the internet through the bastion server to the target registry in the offline environment. IBM Cloud Pak for AIOps can then be installed in the offline environment by using the target registry.

Before you begin

- You must know whether you are deploying a base deployment or a extended deployment of IBM Cloud Pak for AIOps. For more information, see Incremental adoption.
- Review the Planning section. Your environment must meet the system requirements.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- The following commands must be run as a root user or by a user with sudo access: `yum install podman` and `podman login`. The rest of the procedure can be run as a non-root user.
- If you require details about the permissions that are needed by the IBM Cloud Pak for AIOps operators, see Permissions (IBM Cloud Pak for AIOps).
- A user with `cluster-admin` privileges is needed for the following operations:
  - Install Cert Manager
  - Install the License Service
  - Create a custom project
  - Configure the cluster
  - Create the catalog source
  - Create an OperatorGroup

Important: If IBM Sales representatives and Business Partners supplied you with a custom profile ConfigMap to customize your deployment, then you must follow their instructions to apply it during installation. The custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. For more information about custom sizing, see Custom sizing.

The following procedure is based on an Red Hat OpenShift 4.17 environment and includes links for that version. If your environment uses a different supported version of Red Hat OpenShift, ensure that you follow the Red Hat OpenShift documentation for that version.

Installation procedure

1. Set up the mirroring environment
2. Download the CASE
3. Mirror images
4. Configure storage
5. Install IBM Cloud Pak for AIOps

1. Set up the mirroring environment

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the `IBM Cloud Container Registry`, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.

Site	Description

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses CASE files, IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

1.1 Download documentation and scripts for offline access

Download the following documentation and scripts that you might need to access during your IBM Cloud Pak for AIOps installation, and copy them to your air-gapped environment.

1. IBM Cloud Pak for AIOps 4.8.0 documentation

   Download the Cloud Pak for AIOps 4.8.0 PDF (this documentation) so that you can access it offline.

2. IBM Cloud Pak for AIOps 4.8.0 scripts

   - The prerequisite checker script verifies whether your Red Hat OpenShift cluster is correctly set up for a IBM Cloud Pak for AIOps installation. You will need to run this script in step 5.7 Verify cluster readiness. Download this script from github.com/IBM.
   - An uninstall script can be downloaded from github.com/IBM.
   - (Optional) The status checker script can be used in step 5.9 Install IBM Cloud Pak for AIOps to give information about the status of your deployment. The use of this script is optional, as status can be found directly from the **`ibm-aiops-orchestrator`** custom resource. This script can be downloaded from github.com/IBM.

3. Red Hat OpenShift documentation

   The Red Hat OpenShift documentation can be downloaded for offline access from Red Hat. The **`Security and compliance`**, **`Installing`**, **`CLI Tools`**, **`Images`**, and **`Operators`** sections are referenced by this documentation.

1.2 Install and configure Red Hat OpenShift

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running on your target cluster. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation. Information about installing a cluster in a restricted network is given in Mirroring images for a disconnected installation.

2. Install the Red Hat OpenShift command line interface (**`oc`**) on your cluster's boot node and run **`oc login`**. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.

3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a **`MachineConfig`** custom resource to configure chrony to connect to your NTP servers.

4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:

   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

1.3 Set up a target registry

You must have a local Docker type production-grade registry available to store the IBM Cloud Pak for AIOps images in. The registry must meet the following requirements:

- supports Docker Manifest V2, Schema 2.
- supports multi-architecture images.
- is accessible from the Red Hat OpenShift cluster nodes.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.
- must have 140 GB of storage to hold all the software that is to be transferred to the target registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one. For more information, see About the mirror registry in the Red Hat OpenShift documentation.

Important: Do not use the Red Hat OpenShift image registry as your target registry. The Red Hat OpenShift registry does not support multi-architecture images or path separators in the image name.

1.4 Prepare a host

Prepare a bastion host that can connect to the internet and to the air-gapped network with access to the Red Hat OpenShift cluster and the target registry. Your host must be on a Linux® x86_64 or Mac platform with any operating system that the IBM Cloud Pak® CLI and the Red Hat OpenShift CLI support. If you are on a Windows® platform, you must run the actions in a Linux® x86_64 VM or from a Windows Subsystem for Linux (WSL) terminal.

Complete the following steps on your host.

1. Install Podman.

   To install Podman, see the Podman Installation Instructions.

   Note: Docker is not shipped or supported for Red Hat Enterprise Linux (RHEL) 8. The Podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 systems. For more information, see Running containers without Docker in the Red Hat documentation.

2. Install the Red Hat OpenShift CLI tool, **`oc`**.

   `oc` is required for Red Hat OpenShift management. For more information, see Getting started with the OpenShift CLI in the Red Hat OpenShift documentation.

1.5 Install the IBM Catalog Management Plug-in for IBM Cloud Pak®

The IBM Catalog Management Plug-in for IBM Cloud Pak (**`ibm-pak-plugin`**) is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering required IBM product images and uses standard tools for registry and cluster access. The ibm-pak-plugin also extends the Red Hat OpenShift CLI (**`oc`**) capability to streamline the process of delivering installation images to the IBM Cloud Pak in an air-gapped environment.

1. Download and install the most recent version of the **`ibm-pak-plugin`** for your host operating system from github.com/IBM.

2. Run the following command to extract the files.

   ```
   tar -xf oc-ibm_pak-linux-amd64.tar.gz
   ```

3. Run the following command to move the file to the **`/usr/local/bin`** directory.

   ```
   mv oc-ibm_pak-linux-amd64 /usr/local/bin/oc-ibm_pak
   ```

   Note: If you are installing as a non-root user, you must use sudo.

4. Confirm that the **`ibm-pak-plugin`** is installed by running the following command.

   ```
   oc ibm-pak --help
   ```

   Expected result: The **`ibm-pak-plugin`** usage is displayed.

2. Download the CASE

Set environment variables on the bastion host, and connect to the internet so that you can download the IBM Cloud Pak for AIOps CASE files.

Note: Save a copy of your environment variable values to a file that you can use as a reference when you are completing your air-gapped installation tasks.

1. Create the following environment variables.

   ```
   export CASE_NAME=ibm-cp-waiops
   export CASE_VERSION=1.14.0
   export CASE_INVENTORY_SETUP=cpwaiopsSetup
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   ```

   The target registry is the registry where the IBM Cloud Pak for AIOps images are mirrored to, and accessed from by the Red Hat OpenShift cluster, as setup in 1.3 Set up a target registry.

   If your bastion host must connect to the internet through a proxy, then also set the following environment variables.

   ```
   export https_proxy=http://proxy-server-hostname:port
   export http_proxy=http://proxy-server-hostname:port
   ```

2. Connect your host to the internet and disconnect it from the local air-gapped network.

3. Download the IBM Cloud Pak for AIOps installer and image inventory to your host.

   ```
   oc ibm-pak get ${CASE_NAME} --version ${CASE_VERSION}
   ```

   The CASE is downloaded to **`~/.ibm-pak/data/cases/$CASE_NAME/$CASE_VERSION`**. The logs files are available at **`~/.ibm-pak/logs/oc-ibm_pak.log`**.

   Note: If you do not specify the CASE version, then the latest CASE is downloaded. The root directory that is used by **`ibm-pak-plugin`** is **`~/.ibm-pak`**. If required, the root directory can be configured by setting the **`IBMPAK_HOME`** environment variable.

3. Mirror images

Complete the following steps to mirror the IBM Cloud Pak for AIOps, IBM Cloud Pak foundational services Cert Manager, and IBM Cloud Pak foundational services License Service images from the internet to the target registry in the air-gapped environment.

3.1. Generate mirror manifests

Run the following command to generate mirror manifests to be used when mirroring the images to the target registry.

```
oc ibm-pak generate mirror-manifests ${CASE_NAME} ${TARGET_REGISTRY} --version ${CASE_VERSION} --filter ibm_es_opensearch
```

A new directory named **~/.ibm-pak/mirror** is created when you issue the **oc ibm-pak generate mirror-manifests** command. The files **images-mapping.txt** and **image-content-source-policy.yaml** are generated at **~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION**.

3.2. Authenticate with the registries

Log in to the registries to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file. This file is used later to enable the **oc image mirror** command to pull the images from the IBM Entitled Registry, and push them to the target registry.

1. Get the authentication credentials for the IBM Entitled Registry.
    1. To obtain the entitlement key that is assigned to your IBMid, log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
    2. In the *Entitlement keys* section, select **Copy key** to copy the entitlement key.
2. Run the following command to create an environment variable that contains your entitlement key.

    ```
    export ENTITLED_REGISTRY_PASSWORD=<key>
    ```

    Where **<key>** is the entitlement key that you copied in the previous step.
3. Store the authentication credentials for the IBM Entitled Registry and the target registry.

    Run the following commands:

    ```
    podman login cp.icr.io -u cp -p ${ENTITLED_REGISTRY_PASSWORD}
    podman login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
    export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at **${XDG_RUNTIME_DIR}/containers/auth.json**. For more information, see the Options section in the Podman documentation.

3.3. Mirror the images

Complete these steps on your host that is connected to both the local registry and the Red Hat OpenShift cluster.

Mirror images to the target registry.

```
nohup oc image mirror \
-f ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/images-mapping.txt \
-a ${REGISTRY_AUTH_FILE} \
--filter-by-os '.*' \
--insecure \
--skip-multiple-scopes \
--max-per-registry=1 > my-mirror-progress.txt 2>&1 &
```

The UNIX command **nohup** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibility.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress.txt
```

Note: If an error occurs during mirroring, the mirror command can be rerun.

3.4 Configure the cluster

1. Log in to your Red Hat OpenShift cluster.

    You can identify your specific **oc login** command by clicking the user menu in the upper left of the Red Hat OpenShift console, and then clicking **Copy Login Command**.

    Example:

    ```
    oc login <server> -u <cluster username> -p <cluster pass>
    ```
2. Update the global image pull secret for your Red Hat OpenShift cluster.

    Follow the steps in the Red Hat OpenShift documentation topic Updating the global cluster pull secret.

    These steps enable your cluster to have authentication credentials in place to pull images from your TARGET_REGISTRY as specified in the **image-content-source-policy.yaml**, which you will apply to your cluster in the next step.
3. Create the **ImageContentSourcePolicy**.

    Run the following command to create the ImageContentSourcePolicy.

    ```
    oc apply -f  ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/image-content-source-policy.yaml
    ```
4. Verify that the **ImageContentSourcePolicy** resource is created.

```
oc get imageContentSourcePolicy
```

5. Verify your cluster node status.

```
oc get MachineConfigPool -w
```

Important: After the `ImageContentsourcePolicy` and global image pull secret are applied, the configuration of your nodes will be updated sequentially. Wait until all the `MachineConfigPools` are updated before you proceed to the next step.

6. (Optional) If you use an insecure registry, you must add the target registry to the cluster's `insecureRegistries` list.

```
oc patch image.config.openshift.io/cluster --type=merge \
-p '{"spec":{"registrySources":{"insecureRegistries":["'${TARGET_REGISTRY}'"]}}}'
```

4. Configure storage

The storage configuration must satisfy your sizing requirements. Two storage classes are needed for installing IBM Cloud Pak for AIOps. For more information, see Storage.

Note: Storage classes and storage providers cannot be changed after you install IBM Cloud Pak for AIOps. OADP backup and restore requires that a ReadWriteMany (RWX) storage class must be provided. If OADP backup and restore is not needed, a ReadWriteOnce (RWO) storage class can be provided as the `RWX-storage-class-name` in the installation instance CR YAML file. This configuration cannot be changed after IBM Cloud Pak for AIOps is installed.

5. Install IBM Cloud Pak for AIOps

Now that the images are mirrored to your air-gapped environment, you can deploy IBM Cloud Pak for AIOps to that environment. To install IBM Cloud Pak for AIOps, complete the following steps.

- 5.1 Create environment variables
- 5.2 Create the catalog source
- 5.3 Create a custom project
- 5.4 Configure usage data collection
- 5.5 Install Cert Manager
- 5.6 Install the License Service
- 5.7 Verify cluster readiness
- 5.8 Install the operator
- 5.9 Install IBM Cloud Pak for AIOps
- 5.10 Create an EgressFirewall
- 5.11 Access the Cloud Pak for AIOps console

5.1 Create environment variables

Create and then source a shell script named *waiops_var.sh* that defines the environment variables that will be used to provide installation parameters for your deployment. Use the following codeblock as a template, replacing the brackets `< ... >` with values for your environment.

You can use the following table to find the values to set for your storage environment variables.

Table 2. Storage provider classes

Storage provider	RWX-storage-class-name	RWO-storage-class-name
IBM Cloud®	ibmc-block-gold	ibmc-block-gold
IBM Cloud® with OADP backup and restore	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
Red Hat® OpenShift® Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Global Data Platform	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc
IBM Storage Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	px-csi-aiops	px-csi-aiops
Portworx (multi-zone HA)	px-csi-aiops-mz	px-csi-aiops-mz
AWS native storage	gp3-csi	gp3-csi
AWS native storage with OADP backup and restore	efs-sc	gp3-csi

```
#============================================================================================================
# Cloud Pak for AIOps installation variables
#============================================================================================================

export CP4AIOPS_NAME=ibm-cp-aiops
export CP4AIOPS_SIZE=large # Set to small if you only require a starter non-production deployment.
export SECURE_TUNNEL=false # Set to `true` to install Secure Tunnel, otherwise set to `false`.
export PROJECT_CP4AIOPS=cp4aiops
export ACCEPT_LICENSE=false # Set to `true` to agree to the license terms, otherwise install will fail.
export CATALOG_SRC_CP4AIOPS=ibm-aiops-catalog
```

```
# ------------------------------------------------------------------------------------------------------------------
# Incremental adoption - set your deployment type
# Set to `true` to install an extended deployment with log anomaly detection and ticket analysis capabilities
# Set to `false` to install a base deployment without log anomaly detection and ticket analysis capabilities
# ------------------------------------------------------------------------------------------------------------------
export LOG_ANOMALY=false

# ------------------------------------------------------------------------------------------------------------------
# Storage
# ------------------------------------------------------------------------------------------------------------------
export STG_CLASS=<RWX-storage-class-name>
export STG_CLASS_BLOCK=<RWO-storage-class-name>

# ------------------------------------------------------------------------------------------------------------------
# Your customer details
# ------------------------------------------------------------------------------------------------------------------
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>

# ------------------------------------------------------------------------------------------------------------------
# `OwnNamespace` installation mode:  leave INSTALL_MODE_NAMESPACE as it is.
# `AllNamespaces` installation mode: change to export INSTALL_MODE_NAMESPACE=openshift-operators
# ------------------------------------------------------------------------------------------------------------------
export INSTALL_MODE_NAMESPACE=${PROJECT_CP4AIOPS}

# ------------------------------------------------------------------------------------------------------------------
# Topology resource group terminology
# Specify `application` or `service` as the terminology to be used for collections of topology resource
# groups. The default is `application`.
# ------------------------------------------------------------------------------------------------------------------
export TOPOLOGY_TERMINOLOGY=application
```

You can update your deployment type after installation. For more information, see Updating the deployment type.

If you need help with deciding on the values to set for these environment variables, see the following topics.

- LOG_ANOMALY: Incremental adoption
- CP4AIOPS_SIZE: Sizing
- SECURE_TUNNEL: Secure Tunnel
- INSTALL_MODE_NAMESPACE: Operator installation mode

Note: You can set a different value for $PROJECT_CP4AIOPS, $CP4AIOPS_NAME if you want. However, you must not use the **default**, **kube-system**, **kube-public**, **openshift-node**, **openshift-infra**, or **openshift** projects (namespaces) for $PROJECT_CP4AIOPS. This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects (namespaces).

Run the following command to source your script and set the environment variables:

```
. ./waiops_var.sh
```

5.2 Create the catalog sources

1. Run the following command to create the catalog sources for IBM Cloud Pak for AIOps and IBM Cloud Pak® foundational services Cert Manager and License Service in the *openshift-marketplace* namespace.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ${CATALOG_SRC_CP4AIOPS}
  namespace: openshift-marketplace
spec:
  displayName: ${CATALOG_SRC_CP4AIOPS}
  publisher: IBM Content
  sourceType: grpc
  image:
icr.io/cpopen/${CATALOG_SRC_CP4AIOPS}@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cert-manager-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-cert-manager
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-cert-manager-operator-
catalog@sha256:c41e4d3314fac0d2cf5f08e7db85801cb2098d37d0aa8104a347e9e82526e914
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-licensing-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM License Service Catalog
  publisher: IBM
  sourceType: grpc
```

```
  image: icr.io/cpopen/ibm-licensing-catalog@sha256:9669ed5ba26ef32ad36a6784b2658553c00ac2a89cf4e9caddadf7bf46c1592e
EOF
```

2. Verify that the *ibm-aiops-catalog*, *ibm-cert-manager-catalog* and *ibm-licensing-catalog* **CatalogSource** objects are in the output that is returned by the following command:

```
oc get CatalogSources -n openshift-marketplace
```

Example output:

```
oc get CatalogSources -n openshift-marketplace
NAME                    DISPLAY                       TYPE   PUBLISHER   AGE
ibm-aiops-catalog       ibm-aiops-catalog             grpc   IBM         2m
ibm-cert-manager-catalog ibm-cert-manager             grpc   IBM         2m
ibm-licensing-catalog    IBM License Service Catalog  grpc   IBM         2m
```

5.3 Create a custom project (namespace)

Run the following command to create a project (namespace) to deploy IBM Cloud Pak for AIOps into.

```
oc create namespace ${PROJECT_CP4AIOPS}
```

5.4. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The collection of usage data is enabled by default, but can be disabled. Usage data is collected by the **cp4waiops-metricsprocessor** pod, and stored in the **cp4waiops-metricsprocessor** pod's logs. This usage data is sent to IBM when MustGather output is sent to IBM, as the MustGather includes the output from the **cp4waiops-metricsprocessor** pod's logs. The usage data is then sent to and stored in IBM controlled GDPR-compliant systems. The usage data that is collected is numeric, or is about the deployment type and platform. It does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of integrations of each type (For example ServiceNow, Instana, Falcon Logscale)
- Secure tunnel enablement: whether *connection* (which controls whether you can create a secure tunnel) is enabled in the *Installation* custom resource
- Deployment type: base deployment or extended deployment
- Deployment platform: Red Hat® OpenShift® Container Platform or Linux®

Configuring the collection of usage data

If you do not want to disable the collection of usage data, run the following command to configure the usage data with your customer details.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME} --
from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT}
```

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must run the command given in *Disabling the collection of usage data*.

Disabling the collection of usage data

If you want to disable the collection of usage data, run the following command.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME} --
from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT} --from-
literal=enableCollection=false
```

Note: You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

5.5 Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-cert-manager-operator-group
  namespace: ibm-cert-manager
---
```

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-cert-manager-operator
  namespace: ibm-cert-manager
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-cert-manager-operator
  source: ibm-cert-manager-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-cert-manager get pods
   ```

   Example output for a successful IBM Cloud Pak® foundational services Cert Manager installation:

   ```
   NAME                                          READY   STATUS    RESTARTS   AGE
   cert-manager-cainjector-674854c49d-vstq4      1/1     Running   0          8d
   cert-manager-controller-646d4bd6fd-zwmqm      1/1     Running   0          8d
   cert-manager-webhook-8598787c8-s41kt          1/1     Running   0          8d
   ibm-cert-manager-operator-c96957695-dkxnm     1/1     Running   0          8d
   ```

5.6 Install the License Service

Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

1. Run the following command to create the resource definitions that you need:

   ```
   cat << EOF | oc apply -f -
   apiVersion: v1
   kind: Namespace
   metadata:
     name: ibm-licensing
   ---
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: ibm-licensing-operator-group
     namespace: ibm-licensing
   spec:
     targetNamespaces:
     - ibm-licensing
   ---
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-licensing-operator-app
     namespace: ibm-licensing
   spec:
     channel: v4.2
     installPlanApproval: Automatic
     name: ibm-licensing-operator-app
     source: ibm-licensing-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-licensing get pods
   ```

   Example output for a successful IBM Cloud Pak® foundational services License Service installation:

   ```
   NAME                                              READY   STATUS    RESTARTS   AGE
   ibm-licensing-operator-db4cd746c-xzmlf            1/1     Running   0          8d
   ibm-licensing-service-instance-596b99588f-76cc5   1/1     Running   0          8d
   ```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

5.7 Verify cluster readiness

Run the prerequisite checker script that you downloaded in step 1.1 Download documentation and scripts for offline access to verify whether your Red Hat OpenShift cluster is correctly set up for an IBM Cloud Pak for AIOps installation.

Run the script with the following command:

```
./prereq.sh -n ${PROJECT_CP4AIOPS}
```

Important: If you are installing on a multi-zone cluster, then also specify the `-m` flag to assess whether there are sufficient resources to withstand a zone outage.

Example output:

```
# ./prereq.sh -n cp4aiops

[INFO] Starting IBM Cloud Pak for AIOps prerequisite checker v4.8...

CLI: oc

[INFO] ================================Platform Version Check================================
[INFO] Checking Platform Type....
[INFO] You are using Openshift Container Platform
[INFO] OCP version 4.16.7 is compatible but only nodes with AMD64 architectures are supported at this time.
[INFO] ================================Platform Version Check================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Cert Manager Check================================
[INFO] Checking for Cert Manager operator

[INFO] Successfully functioning cert-manager found.

CLUSTERSERVICEVERSION              NAMESPACE
ibm-cert-manager-operator.v4.2.8   ibm-cert-manager

[INFO] ================================Cert Manager Check================================

[INFO] ================================Licensing Service Operator Check================================
[INFO] Checking for Licensing Service operator

[INFO] Successfully functioning licensing service operator found.

CLUSTERSERVICEVERSION           NAMESPACE
ibm-licensing-operator.v4.2.8   ibm-licensing

[INFO] ================================Licensing Service Operator Check================================

[INFO] ================================Starter or Production Install Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource Summary================================================
[INFO]                                                Nodes     |      vCPU       |  Memory(GB)
[INFO] Starter (Non-HA) Base (available/required)       [  9 / 3 ]   [  144 / 47 ]    [  289 / 123 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 3 ]   [  144 / 55 ]    [  289 / 136 ]

[INFO] Production (HA) Base (available/required)        [  9 / 6 ]   [  144 / 136 ]   [  289 / 310 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 6 ]   [  144 / 162 ]   [  289 / 368 ]
[INFO] ================================Resource Summary================================================
[INFO] Cluster currently has resources available to create a Starter (Non-HA) install of Cloud Pak for AIOps

[INFO] ================================Prerequisite Checker Tool Summary================================
      [  PASS  ] Platform Version Check
      [  PASS  ] Storage Provider
      [  PASS  ] Starter (Non-HA) Base Install Resources
      [  FAIL  ] Production (HA) Base Install Resources
      [  PASS  ] Cert Manager Operator Installed
      [  PASS  ] Licensing Service Operator Installed
[INFO] ================================Prerequisite Checker Tool Summary================================
```

5.8 Install the operator

For more information about installing operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

1. Create an OperatorGroup.

   Important: Skip this step if you are installing using the 'All Namespaces' installation mode. Check that you set `INSTALL_MODE_NAMESPACE` correctly in step 5.2, and proceed to the next step, Install the IBM Cloud Pak for AIOps operator.

   If you are installing using the 'OwnNamespace' installation mode, then you must create an operator group in your custom project (namespace), or the IBM Cloud Pak for AIOps operator will not install. There might be an operator group for managing a namespace for given APIs. If there is an operator group for the namespace, do not create a second one.

Create the Operator group by running the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cp4aiops-operator-group
  namespace: ${PROJECT_CP4AIOPS}
spec:
  targetNamespaces:
    - "${PROJECT_CP4AIOPS}"
EOF
```

2. Install the IBM Cloud Pak for AIOps operator.

   Run the following command.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-aiops-orchestrator
  namespace: $INSTALL_MODE_NAMESPACE
spec:
  channel: v4.8
  installPlanApproval: Automatic
  name: ibm-aiops-orchestrator
  source: ${CATALOG_SRC_CP4AIOPS}
  sourceNamespace: openshift-marketplace
EOF
```

   Warning: installPlanApproval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

3. After a few minutes, the IBM Cloud Pak for AIOps operator is installed. Verify that the all of the components have a state of **Succeeded** by running the following command:

```
oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator"
```

   Example output:

```
$ oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator"
ibm-aiops-orchestrator.v4.8.0          IBM Cloud Pak for AIOps                4.8.0   Succeeded
```

5.9 Install IBM Cloud Pak for AIOps

Create an instance of the IBM Cloud Pak for AIOps custom resource. A maximum of one IBM Cloud Pak for AIOps custom resource is allowed per cluster.

1. Use one of the following YAML codeblocks to create an instance of the IBM Cloud Pak for AIOps custom resource, depending on whether you are installing on a multi-zone cluster.

   Installing on a non multi-zone cluster:

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ${CP4AIOPS_NAME}
  namespace: ${PROJECT_CP4AIOPS}
spec:
  size: ${CP4AIOPS_SIZE}
  storageClass: ${STG_CLASS}
  storageClassLargeBlock: ${STG_CLASS_BLOCK}
  imagePullSecret:
  topologyModel: ${TOPOLOGY_TERMINOLOGY}
  license:
    accept: ${ACCEPT_LICENSE}
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: ${SECURE_TUNNEL}
  - name: logAnomalyDetection
    enabled: ${LOG_ANOMALY}
EOF
```

   Installing on a multi-zone cluster:

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ${CP4AIOPS_NAME}
  namespace: ${PROJECT_CP4AIOPS}
spec:
```

```
  size: ${CP4AIOPS_SIZE}
  storageClass: ${STG_CLASS}
  storageClassLargeBlock: ${STG_CLASS_BLOCK}
  imagePullSecret:
  topologyModel: ${TOPOLOGY_TERMINOLOGY}
  license:
    accept: ${ACCEPT_LICENSE}
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: ${SECURE_TUNNEL}
  - name: logAnomalyDetection
    enabled: ${LOG_ANOMALY}
  zones:
  - name: <zone_name1>
  - name: <zone_name2>
  - name: <zone_name3>
EOF
```

Where **<zone_name1>**, **<zone_name2>**, and **<zone_name3>** exactly match the zone labels that you applied to each of your nodes in step 1a of Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HA).

Warning: The pakModules **aiopsFoundation**, **applicationManager**, and **aiManager** must be enabled as in the preceding YAML. Do not change these values to false.

2. Verify your installation.

   Run the following command to check that the **PHASE** of your installation is **Updating**.

   ```
   oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
   ```

   Example output:

   ```
   NAME           PHASE     LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
   ibm-cp-aiops   Updating  Accepted   rook-cephfs     rook-ceph-block           3m
   ```

   It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to **Running**. If your installation phase does not change to **Running**, then use the following command to find out which components are not ready:

   ```
   oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not Ready'
   ```

   Example output:

   ```
   lifecycleservice: Not Ready
   zenservice: Not Ready
   ```

   To see details about why a component is **Not Ready** run the following command, where **<component>** is the component that is not ready, for example **zenservice**.

   ```
   oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
   ```

   (Optional) If you downloaded the status checker script earlier in step 1.1 Download documentation and scripts for offline access, then you can also run this script to see information about the status of your deployment.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

5.10 Create an EgressFirewall

There is no egress firewall policy defined when you install IBM Cloud Pak for AIOps, so outgoing traffic from workload pods to the internal and external network is unrestricted.

To create a more secure environment, use the following steps.

1. Create an **EgressFirewall** on your Red Hat OpenShift cluster to limit egress from the IBM Cloud Pak for AIOps project (namespace).

   For more information about creating an **EgressFirewall**, see Configuring an egress firewall for a project.

   Note: You can have only one **EgressFirewall** per project/namespace.

2. Configure exceptions to the **EgressFirewall**.

   Edit your **EgressFirewall** to add exceptions for the following IBM Cloud Pak for AIOps components that have egress dependencies, otherwise these IBM Cloud Pak for AIOps components fail when they attempt egress.

   1. Allow egress to any external services, such as the following integrations:
      - Kubernetes
      - GitHub
      - Microsoft® Teams
      - ServiceNow

- Slack
- VMware® vCenter

2. Configure your **EgressFirewall** to allow traffic for your GitHub, Kubernetes, ServiceNow, and VMware vCenter integrations.

   Edit your **EgressFirewall** to allow or deny egress, as in the following example:

```
kind: EgressFirewall
metadata:
  name: default
spec:
  egress:
  - type: Allow
    to:
      cidrSelector: <1.2.3.0/24>
  - type: Allow
    to:
      dnsName: <www.github.com>
  - type: Allow
    to:
      dnsName: <www.developer.kubernetes.com>
  - type: Allow
    to:
      dnsName: <www.developer.servicenow.com>
  - type: Allow
    to:
      dnsName: <www.developer.vcenter.com>
  - type: Deny
    to:
      cidrSelector: <0.0.0.0/0>
```

   Substitute values for `dnsName` and `cidrSelector` that are the DNS names and addresses of your GitHub, Kubernetes, ServiceNow, or VMware vCenter sources.

5.11 Access the Cloud Pak for AIOps console

After you successfully install IBM Cloud Pak for AIOps, get the URL for accessing the Cloud Pak for AIOps console.

1. Use the following command to get the URL to access the Cloud Pak for AIOps console:

   ```
   oc get route -n ${PROJECT_CP4AIOPS} cpd -o jsonpath='{.spec.host}'
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

   Based on the sample output, your console URL would be **https://cpd-cp4aiops.apps.mycluster.mydomain**

2. Enter the URL in your browser to open the Cloud Pak for AIOps console. Log in with your username and password.

Find the IBM Cloud Pak for AIOps console username and password

The default username to access the Cloud Pak for AIOps console is **admin**. You can check the default username and their password with the following commands.

Note: This information is for the **IBM provided credentials (admin only)** authentication type.

1. Find the default username.

   ```
   oc -n ${PROJECT_CP4AIOPS} get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_username}' | base64 -d
   && echo
   ```

2. Get the password for the **admin** username.

   ```
   oc -n ${PROJECT_CP4AIOPS} get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d
   ```

   The following extract shows a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the sample output, your password would be **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA**.

Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

What to do next

- Define integrations and applications with Defining.
- You can integrate with IBM Cognos® Analytics. For more information, see Integrating IBM Cognos Analytics with IBM Cloud Pak for AIOps.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/OMNIbus integrations.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/Impact integrations.
- Familiarize yourself with backup and restore procedures. It is recommended that you take regular backups of your IBM Cloud Pak for AIOps deployment. For more information, see Backup and restore.
- For more information about health checks and monitoring, see Health checks and monitoring.

Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a portable device

If your cluster is not connected to the internet, you can deploy an installation of IBM Cloud Pak® for AIOps on your Red Hat® OpenShift® Container Platform cluster by using a portable compute device such as a laptop, or a portable storage device such as a USB device.

In this scenario, your air-gapped (offline) environment has a target registry, and a Red Hat OpenShift cluster on which IBM Cloud Pak for AIOps is to be installed. IBM Cloud Pak for AIOps images are mirrored from the internet to a file system on a portable compute device or a portable storage device. The portable device is then disconnected from the internet and connected in the offline environment, where the images are loaded to the target registry. IBM Cloud Pak for AIOps can then be installed in the offline environment by using the target registry.

Before you begin

- You must know whether you are deploying a base deployment or a extended deployment of IBM Cloud Pak for AIOps. For more information, see Incremental adoption.
- Review the Planning section. Your environment must meet the system requirements.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- The following commands must be run as a root user or by a user with sudo access: `yum install podman` and `podman login`. The rest of the procedure can be run as a non-root user.
- If you require details about the permissions that the IBM Cloud Pak for AIOps operators need, see Permissions (IBM Cloud Pak for AIOps).
- A user with `cluster-admin` privileges is needed for the following operations:
  - Install Cert Manager
  - Install the License Service
  - Create a custom project
  - Configure the cluster
  - Create the catalog source
  - Create an OperatorGroup

Important: If IBM Sales representatives and Business Partners supplied you with a custom profile ConfigMap to customize your deployment, then you must follow their instructions to apply it during installation. The custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. For more information about custom sizing, see Custom sizing.

The following procedure is based on an Red Hat OpenShift 4.17 environment and includes links for that version. If your environment uses a different supported version of Red Hat OpenShift, ensure that you follow the Red Hat OpenShift documentation for that version.

Installation procedure

1. Set up the mirroring environment
2. Download the CASE
3. Mirror images
4. Configure storage
5. Install IBM Cloud Pak for AIOps

1. Set up the mirroring environment

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

`icr.io`
`cp.icr.io`
`dd0.icr.io`
`dd2.icr.io`
`dd4.icr.io`
`dd6.icr.io`
	Allow access to these hosts on port 443 to enable access to the `IBM Cloud Container Registry`, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.

`dd1-icr.ibm-zh.com`
`dd3-icr.ibm-zh.com`
`dd5-icr.ibm-zh.com`
`dd7-icr.ibm-zh.com`
	If you are located in China, also allow access to these hosts on port 443.

`github.com`	Github houses CASE files, IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades.

1.1 Download documentation and scripts for offline access

Download the following documentation and scripts that you might need to access during your IBM Cloud Pak for AIOps installation, and copy them to your air-gapped environment.

1. IBM Cloud Pak for AIOps 4.8.0 documentation

Download the Cloud Pak for AIOps 4.8.0 PDF (this documentation) so that you can access it offline.

2. IBM Cloud Pak for AIOps 4.8.0 scripts

   - The prerequisite checker script verifies whether your Red Hat OpenShift cluster is correctly set up for a IBM Cloud Pak for AIOps installation. You will need to run this script in step 5.7 Verify cluster readiness. Download this script from github.com/IBM.
   - An uninstall script can be downloaded from github.com/IBM.
   - (Optional) The status checker script can be used in step 5.9 Install IBM Cloud Pak for AIOps to give information about the status of your deployment. The use of this script is optional, as status can be found directly from the **`ibm-aiops-orchestrator`** custom resource. This script can be downloaded from github.com/IBM.

3. Red Hat OpenShift documentation

   The Red Hat OpenShift documentation can be downloaded for offline access from Red Hat. The **`Security and compliance`**, **`Installing`**, **`CLI Tools`**, **`Images`**, and **`Operators`** sections are referenced by this documentation.

1.2 Install and configure Red Hat OpenShift

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running on your target cluster. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation. Information about installing a cluster in a restricted network is given in Mirroring images for a disconnected installation.

2. Install the Red Hat OpenShift command line interface (**`oc`**) on your cluster's boot node and run **`oc login`**. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.

3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a **`MachineConfig`** custom resource to configure chrony to connect to your NTP servers.

4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:

   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

1.3 Set up a target registry

You must have a local Docker type production-grade registry available in the air-gapped environment to store the IBM Cloud Pak for AIOps images in. The registry must meet the following requirements:

- supports Docker Manifest V2, Schema 2.
- supports multi-architecture images.
- is accessible from the Red Hat OpenShift cluster nodes.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.
- must have 140 GB of storage to hold all of the software that is to be transferred to the target registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one. For more information, see About the mirror registry in the Red Hat OpenShift documentation.

Important: Do not use the Red Hat OpenShift image registry as your target registry. The Red Hat OpenShift registry does not support multi-architecture images or path separators in the image name.

1.4 Prepare a host

Prepare a portable compute device, or if you are using a portable storage device then a connected compute device, as follows.

You must be able to connect your host to the internet. Your host must be on a Linux® x86_64 or Mac platform with any operating system that the IBM Cloud Pak CLI and the Red Hat OpenShift CLI support. If you are on a Windows® platform, you must run the actions in a Linux® x86_64 VM or from a Windows Subsystem for Linux® (WSL) terminal.

Your portable device and any intermediary devices must have 140 GB of storage to hold all the software that is to be transferred to the local target registry.

Complete the following steps on your host.

1. Install Podman.

   To install Podman, see the Podman Installation Instructions.

   Note: Docker is not shipped or supported for Red Hat Enterprise Linux (RHEL) 8. The Podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 systems. For more information, see Running containers without Docker in the Red Hat documentation.

2. Install the Red Hat OpenShift CLI tool, **`oc`**.

`oc` is required for Red Hat OpenShift management. For more information, see Getting started with the Red Hat OpenShift CLI in the Red Hat OpenShift documentation.

1.5 Install the IBM Catalog Management Plug-in for IBM Cloud Pak®

The IBM Catalog Management Plug-in for IBM Cloud Pak (`ibm-pak-plugin`) is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering required IBM product images and uses standard tools for registry and cluster access. The `ibm-pak-plugin` also extends the Red Hat OpenShift CLI (`oc`) capability to streamline the process of delivering installation images to the IBM Cloud Pak in an air-gapped environment.

1. Download and install the most recent version of the IBM Catalog Management Plug-in for IBM Cloud Pak for your host operating system from github.com/IBM.
2. Run the following command to extract the files.

   ```
   tar -xf oc-ibm_pak-linux-amd64.tar.gz
   ```

3. Run the following command to move the file to the `/usr/local/bin` directory.

   ```
   mv oc-ibm_pak-linux-amd64 /usr/local/bin/oc-ibm_pak
   ```

   Note: If you are installing as a non-root user, you must use sudo.

4. Confirm that `ibm-pak-plugin` is installed by running the following command.

   ```
   oc ibm-pak --help
   ```

   Expected result: The `ibm-pak-plugin` usage is displayed.

2. Download the CASE

Set environment variables on the portable device, and connect it to the internet so that you can download the IBM Cloud Pak for AIOps CASE files.

Note: Save a copy of your environment variable values to a file that you can use as a reference when you are completing your air-gapped installation tasks.

1. Create the following environment variables.

   ```
   export CASE_NAME=ibm-cp-waiops
   export CASE_VERSION=1.14.0
   export CASE_INVENTORY_SETUP=cpwaiopsSetup
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   ```

   The target registry is the registry where the IBM Cloud Pak for AIOps images are mirrored to, and accessed from by the Red Hat OpenShift cluster, as setup in 1.3 Set up a target registry

   If your portable device must connect to the internet through a proxy, then also set the following environment variables.

   ```
   export https_proxy=http://proxy-server-hostname:port
   export http_proxy=http://proxy-server-hostname:port
   ```

2. Connect your portable device to the internet and disconnect it from the air-gapped environment.
3. Download the IBM Cloud Pak for AIOps installer and image inventory to your portable device.

   ```
   oc ibm-pak get ${CASE_NAME} --version ${CASE_VERSION}
   ```

   The CASE is downloaded to `~/.ibm-pak/data/cases/$CASE_NAME/$CASE_VERSION`. The logs files are available at ~/.ibm-pak/logs/oc-ibm_pak.log.

   Note: If you do not specify the CASE version, then the latest CASE is downloaded. The root directory that is used by the `ibm-pak-plugin` is `~/.ibm-pak`. If required, the root directory can be configured by setting the `IBMPAK_HOME` environment variable.

3. Mirror images

Complete the following steps to mirror the IBM Cloud Pak for AIOps, IBM Cloud Pak foundational services Cert Manager, and IBM Cloud Pak foundational services License Service images from the internet to the file system on your portable device, and then from the file system to the target registry in the air-gapped environment.

3.1. Generate mirror manifests

Run the following command to generate mirror manifests to be used when mirroring the images to the target registry.

```
oc ibm-pak generate mirror-manifests ${CASE_NAME} file://cpfs --version ${CASE_VERSION} --filter ibm_es_opensearch --final-registry ${TARGET_REGISTRY}/cpfs
```

Table 2. Parameter description

Argument	Description
`file://cpfs`	Specifies the path extension where images are mirrored to. Images are mirrored to `$IMAGE_PATH/cpfs` when the `oc image mirror` command is run with `images-mapping-to-filesystem.txt`. For more information, see Mirror the images to the file system.

Argument	Description
`$TARGET_REGISTRY/cpfs`	This argument generates an image-mapping file that is used by the `oc image mirror` commands to mirror images to the `TARGET_REGISTRY` at namespace `cpfs`. For example, if the CASE you are installing needs the image, `quay.io/opencloudio/ibm-events-kafka-2.6.0@sha256:10d422dddd29ff19c87066fc6510eee05f5fa4ff608b87a06e898b3b6a3a13c7`, then its final URL in your target registry will be `$TARGET_REGISTRY/cpfs/opencloudio/ibm-events-kafka-2.6.0`. Note the new namespace of `cpfs` in the URL. The namespace path can be multi level if your target registry supports it.

The command generates the following files at `~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION`:

- images-mapping-to-filesystem.txt
- images-mapping-from-filesystem.txt
- image-content-source-policy.yaml

3.2. Authenticate with the IBM Entitled Registry

Log in to the IBM Entitled Registry to generate an authentication file containing the IBM Entitled Registry credentials, and then create an environment variable that has the location of the authentication file. This file is used later to enable the `oc image mirror` command to pull the images from the IBM Entitled Registry.

1. Get the authentication credentials for the IBM Entitled Registry.
    1. To obtain the entitlement key that is assigned to your IBMid, log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
    2. In the *Entitlement keys* section, select **Copy key** to copy the entitlement key.
2. Run the following command to create an environment variable that contains your entitlement key.

    ```
    export ENTITLED_REGISTRY_PASSWORD=<key>
    ```

    Where `<key>` is the entitlement key that you copied in the previous step.
3. Store the authentication credentials for the IBM Entitled Registry.

    Run the following commands:

    ```
    podman login cp.icr.io -u cp -p ${ENTITLED_REGISTRY_PASSWORD}
    export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at `${XDG_RUNTIME_DIR}/containers/auth.json`. For more information, see the Options section in the Podman documentation.

3.3. Mirror the images to the file system

Complete these steps to mirror the images from the internet to a file system on your portable device.

1. Create an environment variable to store the location of the file system where the images are to be stored.

    ```
    export IMAGE_PATH=<image-path>
    ```

    Where `<image-path>` is the directory where you want the images to be stored.
2. Run the following command to mirror the images from the IBM Entitled Registry to the file system.

    ```
    nohup oc image mirror \
    -f ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/images-mapping-to-filesystem.txt \
    -a ${REGISTRY_AUTH_FILE} \
    --filter-by-os '.*' \
    --insecure \
    --skip-multiple-scopes \
    --dir "${IMAGE_PATH}" \
    --max-per-registry=1 > my-mirror-progress.txt 2>&1 &
    ```

    The UNIX® command `nohup` is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibility.

    Run the following command if you want to see the progress of the mirroring:

    ```
    tail -f my-mirror-progress.txt
    ```

    Note: If an error occurs during mirroring, the mirror command can be rerun.

3.4 Setup the file system in the air-gapped environment

1. Copy files to the air-gapped environment (portable storage device only)

    If you are using a portable storage device, you must copy the files from the portable storage device to a local compute device in the air-gapped environment that has access to the target registry. If you are using a portable compute device, then these items are already present, and you can proceed to the next step.

    Copy the following items to your local compute device:

    - the file system located at $IMAGE_PATH, which you specified earlier
    - `~/.ibm-pak` directory
2. Disconnect the device that has your file system, (the portable compute device or the local compute device) from the internet and connect it to the air-gapped environment.
3. Ensure that environment variables are set on the device in the air-gapped environment that has access to the target registry.

If you are using a portable storage device, then set the following environment variables on your local compute device within the air-gapped environment.

If you are using a portable compute device that you have restarted since mirroring the images, then your environment variables will have been lost and you will need to set the following environment variables on your portable compute device again.

```
export CASE_NAME=ibm-cp-waiops
export CASE_VERSION=1.14.0
export CASE_INVENTORY_SETUP=cpwaiopsSetup
export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
export TARGET_REGISTRY=${TARGET_REGISTRY_HOST}:${TARGET_REGISTRY_PORT}
export TARGET_REGISTRY_USER=<username>
export TARGET_REGISTRY_PASSWORD=<password>
export IMAGE_PATH=<image_path>
```

3.5 Authenticate with the target registry

Authenticate with the target registry in the air-gapped environment that you are mirroring the images into.

Run the following commands:

```
podman login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
```

Note: The authentication file is usually at **`${XDG_RUNTIME_DIR}/containers/auth.json`**. For more information, see the Options section in the Podman documentation.

3.6 Mirror the images to the target registry from the file system

Complete the steps in this section on the device that has your file system (the portable compute device or the local compute device) to copy the images from the file system to the $TARGET_REGISTRY. Your device with the file system must be connected to both the target registry and the Red Hat OpenShift cluster.

Run the following command to copy the images referenced in the **`images-mapping-from-filesystem.txt`** from the $IMAGE_PATH file system to the final target registry.

```
nohup oc image mirror \
-f ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/images-mapping-from-filesystem.txt \
--from-dir "${IMAGE_PATH}" \
-a ${REGISTRY_AUTH_FILE} \
--filter-by-os '.*' \
--insecure  \
--skip-multiple-scopes \
--max-per-registry=1 > my-mirror-progress2.txt 2>&1 &
```

The UNIX command **`nohup`** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibility.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress2.txt
```

Note: If an error occurs during mirroring, the mirror command can be rerun.

3.7 Configure the cluster

1. Log in to your Red Hat OpenShift cluster.

   You can identify your specific **`oc login`** command by clicking the user menu in the upper left of the Red Hat OpenShift console, and then clicking **Copy Login Command**.

   Example:

   ```
   oc login <server> -u <cluster username> -p <cluster pass>
   ```

2. Update the global image pull secret for your Red Hat OpenShift cluster.

   Follow the steps in the Red Hat OpenShift documentation topic Updating the global cluster pull secret.

   These steps enable your cluster to have authentication credentials in place to pull images from your TARGET_REGISTRY as specified in the **`image-content-source-policy.yaml`**, which you will apply to your cluster in the next step.

3. Create the **`ImageContentSourcePolicy`**.

   Run the following command to create an ImageContentSourcePolicy (ICSP).

   ```
   oc apply -f  ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/image-content-source-policy.yaml
   ```

4. Verify that the **`ImageContentSourcePolicy`** resource is created.

   ```
   oc get imagecontentsourcepolicy
   ```

   Example output, showing a newly created **`ibm-cp-waiops`** ICSP.

   ```
   oc get imagecontentsourcepolicy
   NAME                        AGE
   ibm-cp-waiops               95s
   redhat-operator-index-0     5d18h
   ```

5. Verify your cluster node status.

```
oc get MachineConfigPool -w
```

Important: After the `ImageContentsourcePolicy` and global image pull secret are applied, the configuration of your nodes will be updated sequentially. Wait until all the `MachineConfigPools` are updated before proceeding to the next step.

6. (Optional) If you use an insecure registry, you must add the target registry to the cluster's `insecureRegistries` list.

```
oc patch image.config.openshift.io/cluster --type=merge \
-p '{"spec":{"registrySources":{"insecureRegistries":["'${TARGET_REGISTRY}'"]}}}'
```

4. Configure storage

The storage configuration must satisfy your sizing requirements. Two storage classes are needed for installing IBM Cloud Pak for AIOps. For more information, see Storage.

Note: Storage classes and storage providers cannot be changed after you install IBM Cloud Pak for AIOps. OADP backup and restore requires that a ReadWriteMany (RWX) storage class must be provided. If OADP backup and restore is not needed, a ReadWriteOnce (RWO) storage class can be provided as the `RWX-storage-class-name` in the installation instance CR YAML file. This configuration cannot be changed after IBM Cloud Pak for AIOps is installed.

5. Install IBM Cloud Pak for AIOps

Now that the images are mirrored to your air-gapped environment, you can deploy IBM Cloud Pak for AIOps to that environment. To install IBM Cloud Pak for AIOps, complete the following steps.

- 5.1 Create environment variables
- 5.2 Create the catalog source
- 5.3 Create a custom project
- 5.4 Configure usage data collection
- 5.5 Install Cert Manager
- 5.6 Install the License Service
- 5.7 Verify cluster readiness
- 5.8 Install the operator
- 5.9 Install IBM Cloud Pak for AIOps
- 5.10 Create an EgressFirewall
- 5.11 Access the Cloud Pak for AIOps console

5.1 Create environment variables

Create and then source a shell script named *waiops_var.sh* that defines the environment variables which will be used to provide installation parameters for your deployment. Use the following codeblock as a template, replacing the brackets **< ... >** with values for your environment.

You can use the following table to find the values to set for your storage environment variables.

Table 3. Storage provider classes

Storage provider	RWX-storage-class-name	RWO-storage-class-name
IBM Cloud®	ibmc-block-gold	ibmc-block-gold
IBM Cloud® with OADP backup and restore	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
Red Hat® OpenShift® Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation	ocs-storagecluster-ceph-rbd	ocs-storagecluster-ceph-rbd
IBM Fusion Data Foundation with OADP backup and restore	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Fusion Global Data Platform	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc	If you are using IBM Storage Fusion, use ibm-spectrum-scale-sc. If you are using IBM Storage Fusion HCI System, use ibm-storage-fusion-cp-sc
IBM Storage Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	px-csi-aiops	px-csi-aiops
Portworx (multi-zone HA)	px-csi-aiops-mz	px-csi-aiops-mz
AWS native storage	gp3-csi	gp3-csi
AWS native storage with OADP backup and restore	efs-sc	gp3-csi

```
#============================================================================================================
# Cloud Pak for AIOps installation variables
#============================================================================================================

export CP4AIOPS_NAME=ibm-cp-aiops
export CP4AIOPS_SIZE=large # Set to small if you only require a starter non-production deployment.
export SECURE_TUNNEL=false # Set to `true` to install Secure Tunnel, otherwise set to `false`.
export PROJECT_CP4AIOPS=cp4aiops
export ACCEPT_LICENSE=false # Set to `true` to agree to the license terms, otherwise install will fail.
export CATALOG_SRC_CP4AIOPS=ibm-aiops-catalog

# ------------------------------------------------------------------------------------------------------------
```

```
# Incremental adoption - set your deployment type.
# Set to `true` to install an extended deployment with log anomaly detection and ticket analysis capabilities
# Set to `false` to install a base deployment without log anomaly detection and ticket analysis capabilities
# ------------------------------------------------------------------------------------------------------------------
export LOG_ANOMALY=false

# ------------------------------------------------------------------------------------------------------------------
# Storage
# ------------------------------------------------------------------------------------------------------------------
export STG_CLASS=<RWX-storage-class-name>
export STG_CLASS_BLOCK=<RWO-storage-class-name>

# ------------------------------------------------------------------------------------------------------------------
# Your customer details
# ------------------------------------------------------------------------------------------------------------------
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>

# ------------------------------------------------------------------------------------------------------------------
# `OwnNamespace` installation mode:  leave INSTALL_MODE_NAMESPACE as it is.
# `AllNamespaces` installation mode: change to export INSTALL_MODE_NAMESPACE=openshift-operators
# ------------------------------------------------------------------------------------------------------------------
export INSTALL_MODE_NAMESPACE=${PROJECT_CP4AIOPS}

# ------------------------------------------------------------------------------------------------------------------
# Topology resource group terminology
# Specify `application` or `service` as the terminology to be used for collections of topology resource
# groups. The default is `application`.
# ------------------------------------------------------------------------------------------------------------------
export TOPOLOGY_TERMINOLOGY=application
```

If you need help with deciding on the values to set for these environment variables, see the following topics.

- LOG_ANOMALY: Incremental adoption
- CP4AIOPS_SIZE: Sizing
- SECURE_TUNNEL: Secure Tunnel
- INSTALL_MODE_NAMESPACE: Operator installation mode

You can update your deployment type after installation. For more information, see Updating the deployment type.

Note: You can set a different value for $PROJECT_CP4AIOPS and $CP4AIOPS_NAME if you want. However, you must not use the **default**, **kube-system**, **kube-public**, **openshift-node**, **openshift-infra**, or **openshift** projects (namespaces) for $PROJECT_CP4AIOPS. This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects (namespaces).

Run the following command to source your script and set the environment variables:

```
. ./waiops_var.sh
```

5.2 Create the catalog sources

1. Run the following command to create the catalog sources for IBM Cloud Pak for AIOps and IBM Cloud Pak® foundational services Cert Manager and License Service in the *openshift-marketplace* namespace.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ${CATALOG_SRC_CP4AIOPS}
  namespace: openshift-marketplace
spec:
  displayName: ${CATALOG_SRC_CP4AIOPS}
  publisher: IBM Content
  sourceType: grpc
  image:
icr.io/cpopen/${CATALOG_SRC_CP4AIOPS}@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cert-manager-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-cert-manager
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-cert-manager-operator-
catalog@sha256:c41e4d3314fac0d2cf5f08e7db85801cb2098d37d0aa8104a347e9e82526e914
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-licensing-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM License Service Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-licensing-catalog@sha256:9669ed5ba26ef32ad36a6784b2658553c00ac2a89cf4e9caddadf7bf46c1592e
EOF
```

2. Verify that the *ibm-aiops-catalog*, *ibm-cert-manager-catalog* and *ibm-licensing-catalog* **CatalogSource** objects are in the output that is returned by the following command:

```
oc get CatalogSources -n openshift-marketplace
```

Example output:

```
oc get CatalogSources -n openshift-marketplace
NAME                     DISPLAY                       TYPE   PUBLISHER   AGE
ibm-aiops-catalog        ibm-aiops-catalog             grpc   IBM         2m
ibm-cert-manager-catalog ibm-cert-manager              grpc   IBM         2m
ibm-licensing-catalog    IBM License Service Catalog   grpc   IBM         2m
```

5.3 Create a custom project (namespace)

Run the following command to create a project (namespace) to deploy IBM Cloud Pak for AIOps into.

```
oc create namespace ${PROJECT_CP4AIOPS}
```

5.4. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The collection of usage data is enabled by default, but can be disabled. Usage data is collected by the **cp4waiops-metricsprocessor** pod, and stored in the **cp4waiops-metricsprocessor** pod's logs. This usage data is sent to IBM when MustGather output is sent to IBM, as the MustGather includes the output from the **cp4waiops-metricsprocessor** pod's logs. The usage data is then sent to and stored in IBM controlled GDPR-compliant systems. The usage data that is collected is numeric, or is about the deployment type and platform. It does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of integrations of each type (For example ServiceNow, Instana, Falcon Logscale)
- Secure tunnel enablement: whether *connection* (which controls whether you can create a secure tunnel) is enabled in the *Installation* custom resource
- Deployment type: base deployment or extended deployment
- Deployment platform: Red Hat® OpenShift® Container Platform or Linux®

This usage data is sent to IBM only if the MustGather tool is run and sent to IBM. The MustGather includes the output from the **cp4waiops-metricsprocessor** pod's logs.

Configuring the collection of usage data

If you do not want to disable the collection of usage data, run the following command to configure the usage data with your customer details.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME} --
from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT}
```

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must run the command given in *Disabling the collection of usage data*.

Disabling the collection of usage data

If you want to disable the collection of usage data, run the following command.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME} --
from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT} --from-
literal=enableCollection=false
```

Note: You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

5.5 Install Cert Manager

Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

1. Run the following command to create the resource definitions that you need:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Namespace
metadata:
  name: ibm-cert-manager
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ibm-cert-manager-operator-group
  namespace: ibm-cert-manager
---
```

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-cert-manager-operator
  namespace: ibm-cert-manager
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-cert-manager-operator
  source: ibm-cert-manager-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-cert-manager get pods
   ```

   Example output for a successful IBM Cloud Pak® foundational services Cert Manager installation:

   ```
   NAME                                       READY   STATUS    RESTARTS   AGE
   cert-manager-cainjector-674854c49d-vstq4   1/1     Running   0          8d
   cert-manager-controller-646d4bd6fd-zwmqm   1/1     Running   0          8d
   cert-manager-webhook-8598787c8-s41kt       1/1     Running   0          8d
   ibm-cert-manager-operator-c96957695-dkxnm  1/1     Running   0          8d
   ```

5.6 Install the License Service

Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

1. Run the following command to create the resource definitions that you need:

   ```
   cat << EOF | oc apply -f -
   apiVersion: v1
   kind: Namespace
   metadata:
     name: ibm-licensing
   ---
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: ibm-licensing-operator-group
     namespace: ibm-licensing
   spec:
     targetNamespaces:
     - ibm-licensing
   ---
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-licensing-operator-app
     namespace: ibm-licensing
   spec:
     channel: v4.2
     installPlanApproval: Automatic
     name: ibm-licensing-operator-app
     source: ibm-licensing-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-licensing get pods
   ```

   Example output for a successful IBM Cloud Pak® foundational services License Service installation:

   ```
   NAME                                              READY   STATUS    RESTARTS   AGE
   ibm-licensing-operator-db4cd746c-xzmlf            1/1     Running   0          8d
   ibm-licensing-service-instance-596b99588f-76cc5   1/1     Running   0          8d
   ```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

5.7 Verify cluster readiness

Run the prerequisite checker script that you downloaded in step 1.1 Download documentation and scripts for offline access to verify whether your Red Hat OpenShift cluster is correctly set up for an IBM Cloud Pak for AIOps installation.

Run the script with the following command:

```
./prereq.sh -n ${PROJECT_CP4AIOPS}
```

Important: If you are installing on a multi-zone cluster, then also specify the `-m` flag to assess whether there are sufficient resources to withstand a zone outage.

Example output:

```
# ./prereq.sh -n cp4aiops

[INFO] Starting IBM Cloud Pak for AIOps prerequisite checker v4.8...

CLI: oc

[INFO] ================================Platform Version Check================================
[INFO] Checking Platform Type....
[INFO] You are using Openshift Container Platform
[INFO] OCP version 4.16.7 is compatible but only nodes with AMD64 architectures are supported at this time.
[INFO] ================================Platform Version Check================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Cert Manager Check================================
[INFO] Checking for Cert Manager operator

[INFO] Successfully functioning cert-manager found.

CLUSTERSERVICEVERSION              NAMESPACE
ibm-cert-manager-operator.v4.2.8   ibm-cert-manager

[INFO] ================================Cert Manager Check================================

[INFO] ================================Licensing Service Operator Check================================
[INFO] Checking for Licensing Service operator

[INFO] Successfully functioning licensing service operator found.

CLUSTERSERVICEVERSION           NAMESPACE
ibm-licensing-operator.v4.2.8   ibm-licensing

[INFO] ================================Licensing Service Operator Check================================

[INFO] ================================Starter or Production Install Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource Summary================================================
[INFO]                                                Nodes     |      vCPU       |  Memory(GB)
[INFO] Starter (Non-HA) Base (available/required)       [  9 / 3 ]   [  144 / 47 ]    [  289 / 123 ]
[INFO]      (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 3 ]   [  144 / 55 ]    [  289 / 136 ]

[INFO] Production (HA) Base (available/required)        [  9 / 6 ]   [  144 / 136 ]   [  289 / 310 ]
[INFO]      (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 6 ]   [  144 / 162 ]   [  289 / 368 ]
[INFO] ================================Resource Summary================================================
[INFO] Cluster currently has resources available to create a Starter (Non-HA) install of Cloud Pak for AIOps

[INFO] ================================Prerequisite Checker Tool Summary================================
      [  PASS  ] Platform Version Check
      [  PASS  ] Storage Provider
      [  PASS  ] Starter (Non-HA) Base Install Resources
      [  FAIL  ] Production (HA) Base Install Resources
      [  PASS  ] Cert Manager Operator Installed
      [  PASS  ] Licensing Service Operator Installed
[INFO] ================================Prerequisite Checker Tool Summary================================
```

5.8 Install the operator

For more information about installing operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

1. Create an OperatorGroup.

   Important: Skip this step if you are installing using the 'All Namespaces' installation mode. Check that you set **`INSTALL_MODE_NAMESPACE`** correctly in step 5.2, and proceed to the next step, Install the IBM Cloud Pak for AIOps operator.

   If you are installing using the 'OwnNamespace' installation mode, then you must create an operator group in your custom project (namespace), or the IBM Cloud Pak for AIOps operator will not install. There might be an operator group for managing a namespace for given APIs. If there is an operator group for the namespace, do not create a second one.

Create the Operator group by running the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cp4aiops-operator-group
  namespace: ${PROJECT_CP4AIOPS}
spec:
  targetNamespaces:
    - "${PROJECT_CP4AIOPS}"
EOF
```

2. Install the IBM Cloud Pak for AIOps operator.

   Run the following command.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-aiops-orchestrator
  namespace: $INSTALL_MODE_NAMESPACE
spec:
  channel: v4.8
  installPlanApproval: Automatic
  name: ibm-aiops-orchestrator
  source: ${CATALOG_SRC_CP4AIOPS}
  sourceNamespace: openshift-marketplace
EOF
```

   Warning: installPlanApproval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

3. After a few minutes, the IBM Cloud Pak for AIOps operator is installed. Verify that the all of the components have a state of **Succeeded** by running the following command:

```
oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator"
```

   Example output:

```
$ oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator"
ibm-aiops-orchestrator.v4.8.0          IBM Cloud Pak for AIOps                4.8.0   Succeeded
```

5.9 Install IBM Cloud Pak for AIOps

Create an instance of the IBM Cloud Pak for AIOps custom resource. A maximum of one IBM Cloud Pak for AIOps custom resource is allowed per cluster.

1. Use one of the following YAML codeblocks to create an instance of the IBM Cloud Pak for AIOps custom resource, depending on whether you are installing on a multi-zone cluster.

   Installing on a non multi-zone cluster:

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ${CP4AIOPS_NAME}
  namespace: ${PROJECT_CP4AIOPS}
spec:
  size: ${CP4AIOPS_SIZE}
  storageClass: ${STG_CLASS}
  storageClassLargeBlock: ${STG_CLASS_BLOCK}
  imagePullSecret:
  topologyModel: ${TOPOLOGY_TERMINOLOGY}
  license:
    accept: ${ACCEPT_LICENSE}
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: ${SECURE_TUNNEL}
  - name: logAnomalyDetection
    enabled: ${LOG_ANOMALY}
EOF
```

   Installing on a multi-zone cluster:

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ${CP4AIOPS_NAME}
  namespace: ${PROJECT_CP4AIOPS}
spec:
```

```
  size: ${CP4AIOPS_SIZE}
  storageClass: ${STG_CLASS}
  storageClassLargeBlock: ${STG_CLASS_BLOCK}
  imagePullSecret:
  topologyModel: ${TOPOLOGY_TERMINOLOGY}
  license:
    accept: ${ACCEPT_LICENSE}
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: ${SECURE_TUNNEL}
  - name: logAnomalyDetection
    enabled: ${LOG_ANOMALY}
  zones:
  - name: <zone_name1>
  - name: <zone_name2>
  - name: <zone_name3>
EOF
```

Where **<zone_name1>**, **<zone_name2>**, and **<zone_name3>** exactly match the zone labels that you applied to each of your nodes in step 1a of Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HA).

Warning: The pakModules **aiopsFoundation**, **applicationManager**, and **aiManager** must be enabled as in the preceding YAML. Do not change these values to false.

2. Verify your installation.

   Run the following command to check that the **PHASE** of your installation is **Updating**.

   ```
   oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
   ```

   Example output:

   ```
   NAME                  PHASE     LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK   AGE
   ibm-cp-aiops          Updating  Accepted   rook-cephfs     rook-ceph-block          3m
   ```

   It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to **Running**. If your installation phase does not change to **Running**, then use the following command to find out which components are not ready:

   ```
   oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not Ready'
   ```

   Example output:

   ```
   lifecycleservice: Not Ready
   zenservice: Not Ready
   ```

   To see details about why a component is **Not Ready** run the following command, where **<component>** is the component that is not ready, for example **zenservice**.

   ```
   oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
   ```

   (Optional) If you downloaded the status checker script earlier in step 1.1 Download documentation and scripts for offline access, then you can also run this script to see information about the status of your deployment.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

5.10 Create an EgressFirewall

There is no egress firewall policy defined when you install IBM Cloud Pak for AIOps, so outgoing traffic from workload pods to the internal and external network is unrestricted.

To create a more secure environment, use the following steps.

1. Create an **EgressFirewall** on your Red Hat OpenShift cluster to limit egress from the IBM Cloud Pak for AIOps project (namespace).

   For more information about creating an **EgressFirewall**, see Configuring an egress firewall for a project.

   Note: There must be only one **EgressFirewall** per project/namespace.

2. Configure exceptions to the **EgressFirewall**.

   Edit your **EgressFirewall** to add exceptions for the following IBM Cloud Pak for AIOps components that have egress dependencies, otherwise these IBM Cloud Pak for AIOps components fail when they attempt egress.

   1. Allow egress to any external services, such as the following integrations:
      - Kubernetes
      - GitHub
      - Microsoft® Teams
      - ServiceNow

- Slack
- VMware® vCenter

2. Configure your **EgressFirewall** to allow traffic for your GitHub, Kubernetes, ServiceNow, and VMware vCenter integrations.

   Edit your **EgressFirewall** to allow or deny egress, as in the following example:

```
kind: EgressFirewall
metadata:
  name: default
spec:
  egress:
  - type: Allow
    to:
      cidrSelector: <1.2.3.0/24>
  - type: Allow
    to:
      dnsName: <www.github.com>
  - type: Allow
    to:
      dnsName: <www.developer.kubernetes.com>
  - type: Allow
    to:
      dnsName: <www.developer.servicenow.com>
  - type: Allow
    to:
      dnsName: <www.developer.vcenter.com>
  - type: Deny
    to:
      cidrSelector: <0.0.0.0/0>
```

   Substitute values for `dnsName` and `cidrSelector` that are the DNS names and addresses of your GitHub, Kubernetes, ServiceNow, or VMware vCenter sources.

5.11 Access the Cloud Pak for AIOps console

After you successfully install IBM Cloud Pak for AIOps, get the URL for accessing the Cloud Pak for AIOps console.

1. Use the following command to get the URL to access the Cloud Pak for AIOps console:

   ```
   oc get route -n ${PROJECT_CP4AIOPS} cpd -o jsonpath='{.spec.host}'
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

   Based on the sample output, your console URL would be **https://cpd-cp4aiops.apps.mycluster.mydomain**

2. Enter the URL in your browser to open the Cloud Pak for AIOps console. Log in with your username and password.

Find the IBM Cloud Pak for AIOps console username and password

The default username to access the Cloud Pak for AIOps console is **admin**. You can check the default username and their password with the following commands.

Note: This information is for the **IBM provided credentials (admin only)** authentication type.

1. Find the default username.

   ```
   oc -n ${PROJECT_CP4AIOPS} get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_username}' | base64 -d
   && echo
   ```

2. Get the password for the **admin** username.

   ```
   oc -n ${PROJECT_CP4AIOPS} get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d
   ```

   The following extract shows a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the sample output, your password would be **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA**.

Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

What to do next

- Define integrations and applications with Defining.
- You can integrate with IBM Cognos® Analytics. For more information, see Integrating IBM Cognos Analytics with IBM Cloud Pak for AIOps.
- If yo- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/OMNIbus integrations.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect it to IBM Cloud Pak for AIOps through an integration. For more information, see Creating IBM Tivoli Netcool/Impact integrations.
- Familiarize yourself with backup and restore procedures. It is recommended that you take regular backups of your IBM Cloud Pak for AIOps deployment. For more information, see Backup and restore.
- For more information about health checks and monitoring, see Health checks and monitoring.

Installing on cloud platforms

When you are installing IBM Cloud Pak® for AIOps on specific cloud platforms, there are some suggested and best practices that you can follow to help you complete the installation process.

While the following suggested and required configurations are only for deployment on Azure Red Hat OpenShift (ARO), Google Cloud Platform (GCP), Red Hat® OpenShift® on IBM Cloud® service (ROKS), Red Hat OpenShift Service on AWS (ROSA), and AWS GovCloud, IBM Cloud Pak for AIOps can be installed on any platform that has a supported version of Red Hat® OpenShift® Container Platform installed that meets all other prerequisities.

View the following topics for guidance on installing IBM Cloud Pak for AIOps on cloud platforms:

- Installing IBM Cloud Pak for AIOps on ARO
- Installing IBM Cloud Pak for AIOps on GCP
- Installing IBM Cloud Pak for AIOps on ROKS
- Installing IBM Cloud Pak for AIOps on ROSA
- Installing IBM Cloud Pak for AIOps on AWS GovCloud

View the following topics for details on the main installation options for installing IBM Cloud Pak for AIOps on Red Hat OpenShift.

- Starter installation (CLI)
- Production installation

Installing IBM Cloud Pak for AIOps on Azure Red Hat OpenShift (ARO)

If you are installing IBM Cloud Pak® for AIOps on Azure Red Hat OpenShift (ARO), then you must complete some additional steps before you rejoin the main installation procedure for installing IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform.

Before you begin

Ensure that you meet the following prerequisites:

- Access to the Azure portal (requires email invitation).
- Red Hat subscription, with username and password.
- Azure account permissions to deploy an ARO cluster. You must have the following permissions directly on the virtual network, on the resource group, or on the subscription containing it:
  - Contributor and User Access Administrator permissions, or
  - Owner permissions

  For example, an Azure user with assigned roles of **`Application administrator`** and **`Conditional Access administrator`** has adequate permission.

Installation procedure

Follow these steps to install IBM Cloud Pak for AIOps on ARO.

1. Configure Azure and Red Hat access
2. Create an ARO cluster
3. Configure storage
4. Install IBM Cloud Pak for AIOps

1. Configure Azure and Red Hat access

1. Install the Azure command-line interface (CLI), **`az`**.

   Follow the instructions Install the Azure CLI on Linux in the Azure documentation.

2. Get an OpenShift pull secret to enable access to the OpenShift container registries.

   1. Log in to the OpenShift cluster manager portal.

   2. Click **Download pull secret** to download the pull secret, and save it to a file called **`pull-secrets.json`**. This file is used in step 2.2

3. Log in to the Azure CLI.

   Run the following command to log in to the Azure CLI, and to find your subscription-id.

   ```
   az login
   az account show | jq .id
   ```

4. Register providers to your Azure account.

   ```
   az account set --subscription <subscription-id>
   az provider register -n Microsoft.RedHatOpenShift --wait
   ```

```
az provider register -n Microsoft.Compute --wait
az provider register -n Microsoft.Storage --wait
az provider register -n Microsoft.Authorization --wait
az feature register --namespace Microsoft.RedHatOpenShift --name preview
```

Where **<subscription-id>** is the subscription-id that you found in the previous step.

2. Create an ARO cluster

1. Create dependant assets.

   Run the following commands to create a virtual network that contains two empty subnets.

   ```
   # Set environment variables for use by the `az` commands
   export SUFFIX=waiops4test
   export LOCATION=eastus
   export RESOURCEGROUP=aro-$SUFFIX
   export CLUSTER=cluster-$SUFFIX

   # Create a resource group
   az group create -g $RESOURCEGROUP -l $LOCATION

   # Create a new virtual network in the resource group that you created above:
   az network vnet create -g $RESOURCEGROUP \
     -n aro-vnet-$SUFFIX \
     --address-prefixes 10.0.0.0/22

   # Add an empty subnet for the master nodes:
   az network vnet subnet create -g $RESOURCEGROUP \
     -n master-subnet-$SUFFIX \
     --vnet-name aro-vnet-$SUFFIX \
     --address-prefixes 10.0.0.0/23 \
     --service-endpoints Microsoft.ContainerRegistry \
     --disable-private-link-service-network-policies true

   # Add an empty subnet for the worker nodes:
   az network vnet subnet create -g $RESOURCEGROUP \
     -n worker-subnet-$SUFFIX \
     --vnet-name aro-vnet-$SUFFIX \
     --address-prefixes 10.0.2.0/23 \
     --service-endpoints Microsoft.ContainerRegistry
   ```

2. Run the following command to create an ARO cluster.

   Note: Use worker-vm-size **Standard_D16s_v3** for a small, starter deployment of IBM Cloud Pak for AIOps and a worker-vm-size of **Standard_D48s_v3** for a large, production deployment of IBM Cloud Pak for AIOps.

   ```
   az aro create -g $RESOURCEGROUP \
     -n $CLUSTER \
     --vnet aro-vnet-$SUFFIX \
     --master-subnet master-subnet-$SUFFIX \
     --worker-subnet worker-subnet-$SUFFIX \
     --worker-count 6 \
     --worker-vm-size Standard_D16s_v3 \
     --master-vm-size Standard_D8s_v3 \
     --pull-secret @pull-secrets.json
   ```

   Note: **pull-secrets.json** must be the file that the Red Hat OpenShift secret was stored in, in step 1.2.

   It takes approximately 35 minutes for the ARO cluster to be created. Progress can be viewed from the Azure console or CLI.

3. When the ARO cluster is created, run the following command to log in to it.

   ```
   oc login \
    -u kubeadmin -p $(az aro list-credentials  -n $CLUSTER -g $RESOURCEGROUP | jq -r .kubeadminPassword) \
    -s $(az aro show  -n $CLUSTER -g $RESOURCEGROUP | jq -r .apiserverProfile.url)
   ```

3. Configure storage

Azure's native storage is not compatible with some IBM Cloud Pak for AIOps components. Configure one of the following supported storage providers for IBM Cloud Pak for AIOps:

- Red Hat® OpenShift® Data Foundation (ODF)
- Portworx Enterprise

Note: Storage classes and storage providers cannot be changed after you install IBM Cloud Pak for AIOps. OADP backup and restore requires that a ReadWriteMany (RWX) storage class must be provided. If OADP backup and restore is not needed, a ReadWriteOnce (RWO) storage class can be provided as the **RWX-storage-class-name** in the installation instance CR YAML file. This configuration cannot be changed after IBM Cloud Pak for AIOps is installed.

3.1 Configuring Red Hat OpenShift Data Foundation

To configure Red Hat OpenShift Data Foundation storage for your IBM Cloud Pak for AIOps on ARO deployment, use the instructions in Deploying OpenShift Data Foundation on Azure Red Hat OpenShift in the Red Hat documentation.

3.2 Configuring Portworx Enterprise

To configure Portworx storage for your IBM Cloud Pak for AIOps on ARO deployment, use the following instructions.

3.2.1 Create secret for Portworx

1. Use a browser to access the Azure portal, **portal.azure.com**, click **Azure Active Directory** and find the value of **Tenant ID** in the returned page.
2. From the command line, run the following command to set an environment that contains the tenant ID.

   ```
   export AZURE_TENANT_ID=<value of the tenant ID>
   ```

3. Use a browser to access the Azure portal, **portal.azure.com**, and then click **Virtual machines**.
4. In the search bar on the **Virtual machines** page, enter the value of your $SUFFIX environment variable as a filter and click the **Name** of any of the returned VMs.
5. Click **Access control (IAM)** on the menu.
6. On the **cluster name | Access control (IAM)** page, click **View** in the **View deny assignments** section.
7. On the **cluster name | Access control (IAM)** page's **Deny assignments** tab, click the link beneath **Name**.
8. On the **Deny assignment | Users** page, click the link in the **Name** column in the **Deny assignment excludes** section.
9. In the **enterprise application | Overview** page, locate the **Application ID** field and copy it.
10. From the command line, run the following command to set an environment variable that contains the application ID.

    ```
    export AZURE_CLIENT_ID=<value of the application ID>
    ```

11. Use a browser to navigate to the Azure portal, **portal.azure.com**, and then click **Azure Active Directory**.
12. On the **Azure Active Directory | Overview** page, click **App registrations** on the menu.
13. On the **Azure Active Directory | App registrations** page, enter the cluster's client-id (that you just exported) in the filter field, and then click the link in the **Display name** column.
14. On the returned **Application registration details** page, click **Certificates and secrets** on the menu.
15. In the **Certficates and secrets** page, select **+ New client secret** to create a new client secret.
16. In the **Add a client secret** page, enter a **Description** and an **Expiry**, and select **Add**.
17. On the **Certificates and secrets** page, locate the secret that you just added in the **Client secrets** section. Then select the copy icon next to the **Value** column to get the secret.
18. From the command line, run the following command to set an environment variable that contains the client secret.

    ```
    export AZURE_CLIENT_SECRET=<value of the client secret>
    ```

19. Create a secret

    Run the following command to create the secret

    ```
    oc delete secret px-azure -n kube-system
    oc create secret generic px-azure \
    -n kube-system \
    --from-literal=AZURE_TENANT_ID=${AZURE_TENANT_ID} \
    --from-literal=AZURE_CLIENT_ID=${AZURE_CLIENT_ID} \
    --from-literal=AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
    ```

3.2.2 Create a Portworx spec

A Portworx Enterprise spec is required to set up a Portworx storage cluster.

1. From a browser, go to https://portworx.com, select **LOGIN**, and log in to your Portworx account. Select to use the Portworx Enterprise edition. Then, click **Continue**, and **Continue** again on the Portworx product comparison page.
2. On the **Spec Generator - Enterprise** page, enter or select the following settings and then click **Next**.
   1. Select the checkbox for **Use the Portworx Operator**.
   2. Select *2.10* for the **Portworx Version**.
   3. Select the *Built-in* radio button for **ETCD**.
3. On the next page, set the following values and then click **Next**.
   1. For **Select your environment**, select the *Cloud* radio button.
   2. Under **Select Cloud Platform**, select *AZURE*.
   3. Under **Configure storage devices**, select *Create Using a Spec*
   4. Under **Size(GB)**, select *2000 GB*
4. Click **Next** to skip the Network configuration page.
5. On the **Customize** page, select **Azure Kubernetes Service (AKS)**, then click **Finish**, and click **Agree** to accept the license agreement.
6. In the **spec** page, highlight and select the live download URL.

   For example, **https://install.portworx.com/2.10?operator=true&mc=false&kbver=&b=true&kd=type%3DPremium_LRS%2Csize%3D150&s=%22type%3DPremium_LRS%2Csize%3D2000%22&c=p**

```
x-cluster-e0fb19ba-4e45-4175-b92e-18003948b965&aks=true&stork=true&csi=true&mon=true&tel=false&st=k8s&promop=true
```

3.2.3 Installing the Portworx operator from Red Hat OpenShift Container Platform

1. Open the Red Hat OpenShift Container Platform console for your Red Hat OpenShift Service on AWS cluster.

   If you need to find your ARO cluster credentials and console URL, you can use the following commands.

   ```
   az aro list-credentials  -n $CLUSTER -g $RESOURCEGROUP
   az aro show  -n $CLUSTER -g $RESOURCEGROUP | jq -r .consoleProfile.url
   ```

2. Go to **Operators** > **OperatorHub**.
3. Search for *Portworx Enterprise*. Then, click **Install** > **Install**.

   Warning: Do not click *Create StorageCluster*. You will create the storage cluster in the next step, with the spec that you saved.

4. *Portworx Enterprise* is now listed under **Operators** > **Installed Operators**.

3.2.4 Create the Portworx storage cluster

1. From the command line, run the following command to load the Portworx spec that you created into the ARO cluster.

   ```
   export PORTWORX_SPEC_URL=<spec_URL>
   curl -ks "${PORTWORX_SPEC_URL}&osft=true" | oc apply -f -
   ```

   Where **<spec_URL>** is the spec URL that you copied earlier.

2. From the console, go to **Operators** > **Installed Operators** > **Portworx Enterprise** and select the **StorageClusters** tab. The storage cluster is ready when the **Status** changes from *Phase: Initializing* to *Phase: Online*.
3. Create the Portworx storage classes. Follow the **Define a custom Portworx storage class** instructions in Installing recommended storage providers and configuring storage classes: Portworx.

4. Install IBM Cloud Pak for AIOps

You are now ready to install IBM Cloud Pak for AIOps.

Follow the procedure for installing IBM Cloud Pak for AIOps from step 3 in one of the following topics, according to your requirements:

- Starter installation (CLI)
- Online production installation of IBM Cloud Pak for AIOps (console method)
- Online production installation of IBM Cloud Pak for AIOps (CLI method)

Deleting the ARO cluster

If you no longer require your ARO cluster, you can remove it with the following command.

```
az aro delete -y -g $RESOURCEGROUP -n $CLUSTER
```

Installing IBM Cloud Pak for AIOps on Google Cloud Platform

Follow this procedure if you want to install IBM Cloud Pak® for AIOps on a Red Hat® OpenShift® Container Platform cluster on Google Cloud Platform (GCP).

Installation procedure

1. Create a Red Hat OpenShift cluster on GCP
2. Configure storage
3. Install IBM Cloud Pak for AIOps

1. Create a Red Hat OpenShift cluster on GCP

Install a Red Hat OpenShift cluster on GCP. For more information, see the Red Hat documentation Installing on Google Cloud Platform.

2. Configure storage

Configure one of the following storage providers for IBM Cloud Pak for AIOps:

- Red Hat® OpenShift® Data Foundation (ODF)
- Portworx Enterprise

Note: Storage classes and storage providers cannot be changed after you install IBM Cloud Pak for AIOps. OADP backup and restore requires that a ReadWriteMany (RWX) storage class must be provided. If OADP backup and restore is not needed, a ReadWriteOnce (RWO) storage class can be provided as the **RWX-storage-class-name** in the installation instance CR YAML file. This configuration cannot be changed after IBM Cloud Pak for AIOps is installed.

2.1 Configuring ODF

To configure ODF storage for your IBM Cloud Pak for AIOps deployment, use the instructions in the Red Hat documentation Deploying and managing OpenShift Data Foundation using Google Cloud.

Important: Read the statement about the technology preview status of Red Hat® OpenShift® Data Foundation on GCP in the previous link.

2.2 Configuring Portworx Enterprise

To configure Portworx storage for your IBM Cloud Pak for AIOps on GCP deployment, use the following instructions.

2.2.1 Create a Portworx spec

A Portworx Enterprise spec is required to set up a Portworx storage cluster. For more information, see GCP in the Portworx documentation.

1. From a browser, go to https://portworx.com, select **LOGIN**, and log in to your Portworx account. Select to use the Portworx Enterprise edition. Then, click **Continue**, and **Continue** again on the Portworx product comparison page.
2. On the **Spec Generator - Enterprise** page, enter or select the following settings and then click **Next**.
    1. Select the checkbox for **Use the Portworx Operator**.
    2. Select *2.10* for the **Portworx Version**.
    3. Select the *Built-in* radio button for **ETCD**.
3. On the next page, set the following values and then click **Next**.
    1. For **Select your environment**, select the *Cloud* radio button.
    2. Under **Select Cloud Platform**, select *Google cloud/GKE*.
    3. Under **Configure storage devices**, select *Create Using a Spec*
    4. Under **Size(GB)**, select *2000 GB*
4. Click **Next** to skip the Network configuration page.
5. On the **Customize** page, select **Google Kubernetes Engine (GKE)**, then click **Finish**, and click **Agree** to accept the license agreement.
6. In the **spec** page, highlight and select the live download URL.

    For example,

    ```
    https://install.portworx.com/2.10?
    operator=true&mc=false&kbver=&b=true&kd=type%3DPremium_LRS%2Csize%3D150&s=%22type%3DPremium_LRS%2Csize%3D2000%22&c=px-cluster-e0fb19ba-4e45-4175-b92e-18003948b965&aks=true&stork=true&csi=true&mon=true&tel=false&st=k8s&promop=true
    ```

2.2.2 Configure GCP for Portworx

1. Open the worker node ports 17001-17020 to the master node network. For more information, see Firewall policy rules in the GCP documentation.
2. The worker node's service account (created by the Red Hat OpenShift installer) must have the following roles:

    ```
    Compute Admin
    Service Account User
    Kubernetes Engine Cluster Viewer
    ```

    You can view and edit the roles by opening the GCP console, and selecting **IAM & Admin** > **Service Accounts**. For more information, see in the GCP documentation.

2.2.3 Installing the Portworx operator from Red Hat OpenShift Container Platform

1. Open the Red Hat OpenShift Container Platform console for your GCP cluster.
2. Go to **Operators** > **OperatorHub**.
3. Search for *Portworx Enterprise*. Then, click **Install** > **Install**.

    Warning: Do not click *Create StorageCluster*. You will create the storage cluster in the next step, with the spec that you saved.
4. Portworx Enterprise should now be listed under **Operators** > **Installed Operators**.

2.2.4 Create the Portworx storage cluster

1. From the command line, run the following command to load the Portworx spec that you created into the GCP cluster.

    ```
    export PORTWORX_SPEC_URL=<spec_URL>
    curl -ks "${PORTWORX_SPEC_URL}" | oc apply -f -
    ```

    Where **<spec_URL>** is the spec URL that you copied earlier.
2. From the console, go to **Operators** > **Installed Operators** > **Portworx Enterprise** and select the **StorageClusters** tab. The storage cluster is ready when the **Status** changes from *Phase: Initializing* to *Phase: Online*.
3. Create the Portworx storage classes. Follow the **`Define a custom Portworx storage class`** instructions in Installing recommended storage providers and configuring storage classes: Portworx.

3. Install IBM Cloud Pak for AIOps

You are now ready to install IBM Cloud Pak for AIOps.

Follow the procedure for installing IBM Cloud Pak for AIOps from step 3 in one of the following topics, according to your requirements:

- Starter installation (CLI)
- Online production installation of IBM Cloud Pak for AIOps (console method)
- Online production installation of IBM Cloud Pak for AIOps (CLI method)

Installing IBM Cloud Pak for AIOps on Red Hat OpenShift on IBM Cloud (ROKS)

If you are installing on Red Hat OpenShift on IBM Cloud (ROKS), you can follow any of the main online installation options for installing IBM Cloud Pak® for AIOps. For installing on Red Hat OpenShift on IBM Cloud, you need to replace any storage class configuration with your configured storage class for Red Hat OpenShift on IBM Cloud.

- Starter installation (CLI)
- Production installation(online installation only)

Storage

Installing on a classic infrastructure:

When you are configuring storage as part of the installation, consider using the default storage that is available with Red Hat OpenShift on IBM Cloud (ROKS). This storage has been tested with a IBM Cloud Pak for AIOps deployment on Red Hat OpenShift on IBM Cloud. You can choose to deploy and use your own storage if needed or preferred.

The following storage classes are the default storage classes for Red Hat OpenShift on IBM Cloud:

- `storageClass: ibmc-file-gold-gid`
- `storageClassLargeBlock: ibmc-block-gold`

Note: `ibmc-file-gold-gid` does not have volume expansion enabled by default, which is required by IBM Cloud Pak for AIOps. For more information, see Storage class requirements.

Installing on VPC infrastructure (using Red Hat OpenShift Data Foundation storage):

Consider using Red Hat OpenShift Data Foundation storage. This storage has been tested with a IBM Cloud Pak for AIOps deployment on Red Hat OpenShift on IBM Cloud. To use this storage provider, you need to first deploy Red Hat OpenShift Data Foundation on your CPV cluster. For more information, see Deploying OpenShift Data Foundation on VPC clusters.

When you do use this storage, the following storage classes are the default storage classes to use for Red Hat OpenShift on IBM Cloud:

- `storageClass: ocs-storagecluster-cephfs`
- `storageClassLargeBlock: ocs-storagecluster-ceph-rbd`

For more information about these storage classes, see the IBM Cloud Docs Storage class reference.

Installing IBM Cloud Pak for AIOps on Red Hat OpenShift Service on AWS (ROSA)

If you are installing IBM Cloud Pak® for AIOps on Red Hat OpenShift Service on AWS (ROSA), then you must complete some additional steps before you rejoin the main installation procedure for installing IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform.

Before you begin

Ensure that you meet the following prerequisites:

- You have an AWS account.
- You have a Red Hat subscription.
- Ensure that your environment meets the prerequisites for Red Hat OpenShift Service on AWS.

Installation procedure

Follow these steps to install IBM Cloud Pak for AIOps on Red Hat OpenShift Service on AWS

1. Configure AWS and Red Hat access
2. Create a Red Hat OpenShift Service on AWS Red Hat OpenShift Container Platform cluster
3. Configure storage
4. Install IBM Cloud Pak for AIOps

1. Configure AWS and Red Hat access

1. Log in to the AWS console to verify that you have access to the console. If needed, the *Login* page provides a link to register for an account.

2. From the AWS console, deploy Red Hat OpenShift Container Platform into a VPC within your AWS environment if it is not already deployed. For more information see Red Hat OpenShift Service on AWS quickstart guide.
3. Log in to the Red Hat console to verify that you have access to the console. If needed, the *Login* page provides a link to register for an account.
4. Retrieve your Red Hat Red Hat OpenShift Cluster Manager API Token by entering the following URL:
   - https://console.redhat.com/openshift/token/rosa/show
5. Configure the ROSA CLI to work with your AWS account. For more information, see Install and configure the latest ROSA CLI.
6. (Optional) Create a test cluster to validate your environment configuration
   1. Log in to Red Hat OpenShift Service on AWS.

      ```
      rosa login --token="<token>"
      ```

   2. Configure your AWS account to allow a IAM (non-STS) ROSA cluster.

      ```
      rosa init
      ```

   3. Create the test cluster.

      ```
      rosa create cluster --cluster-name=mytest
      ```

      During the cluster creation, you can review the installation logs to watch the progress.

      ```
      rosa logs install -c mytest --watch
      ```

   4. After the installation completes, list the cluster and note the cluster ID.

      ```
      rosa list clusters
      ```

   5. Describe your cluster using the cluster ID.

      ```
      rosa describe cluster -c 1ab23de4fghijk5lmno6p78q9r1stu2v
      ```

   6. Add an identity provider.

      ```
      rosa create idp --cluster 1ab23de4fghijk5lmno6p78q9r1stu2v --interactive
      ```

      You need to use the noninteractive mode and get some constant values to use

   7. Create your initial admin account.

      ```
      rosa create admin --cluster=1ab23de4fghijk5lmno6p78q9r1stu2v
      ```

   8. Log in to your test cluster by using the admin and password.

      ```
      oc login https://api.jgtest01.dj5a.p1.openshiftapps.com:6443 --username cluster-admin --password XXXXXX
      ```

      If your login is successful, your test cluster is working and your environment is configured.

   9. Clean up and delete the test cluster to proceed with creating your main cluster.

      ```
      rosa delete cluster -c 1ab23de4fghijk5lmno6p78q9r1stu2v
      ```

2. Create a Red Hat OpenShift Service on AWS Red Hat OpenShift Container Platform cluster

1. Log in to the Red Hat OpenShift Service on AWS CLI.

   ```
   rosa login
   ```

   Log in to the site https://cloud.redhat.com/openshift/token/rosa to retrieve your token. Then, copy and paste the token into the CLI prompt.

2. Create your cluster.

   ```
   rosa create cluster --cluster-name=<myclustername> --compute-machine-type=m5.8xlarge  --compute-nodes=<Number of Compute Nodes>  --version <Red Hat OpenShift_version>
   ```

   Example:

   ```
   rosa create cluster --cluster-name=cluster-test1  --compute-machine-type=m5.8xlarge  --compute-nodes=6 --version 4.16
   ```

3. After your cluster is ready, create your cluster administrator account.
   1. Run the following command:

      ```
      rosa create admin -c <myclustername>
      ```

      Important: Record the admin username (cluster-admin) and password for future use.

   2. Run the `oc login` command with the cluster administrator credentials.
4. Verify that all nodes are in `Ready` state before proceeding. Your cluster may take 40+ minutes to create and for you to be able to successfully log in with the 'oc login' command.

1. Run the following command:

   ```
   oc get nodes
   ```

2. Run the following command:

   ```
   rosa describe cluster -c <myclustername>
   ```

   Record the console URL for the Red Hat OpenShift Console and the Details Page for viewing the cluster details.

3. Verify that you can access the Red Hat OpenShift console by logging in to the provided Red Hat OpenShift console URL using the **cluster-admin** role and credentials.

3. Configure storage

AWS native storage and Portworx are tested and supported storage options for installing IBM Cloud Pak for AIOps on ROSA. Use the following instructions to configure AWS native storage or Portworx as your storage solution.

Note: Storage classes and storage providers cannot be changed after you install IBM Cloud Pak for AIOps. OADP backup and restore requires that a ReadWriteMany (RWX) storage class must be provided. If OADP backup and restore is not needed, a ReadWriteOnce (RWO) storage class can be provided as the **RWX-storage-class-name** in the installation instance CR YAML file. This configuration cannot be changed after IBM Cloud Pak for AIOps is installed.

AWS native storage

The AWS native storage class that is required by IBM Cloud Pak for AIOps is **gp3-csi**. If you want to use OADP backup and restore, the **efs-sc** AWS native storage class is required. Storage class configuration cannot be changed after installing IBM Cloud Pak for AIOps.

Amazon Elastic Block Store (EBS) provides block storage. The storage class is **gp3-csi**, and is created when ROSA is installed.

Amazon Elastic File System (EFS) provides file storage. The storage class is **efs-sc**, and you must create it. For more information, see **Setting up the AWS EFS CSI Driver Operator** and **Creating the AWS EFS storage class** in the Red Hat OpenShift documentation. You can also review the information in the Red Hat article Enabling the AWS EFS CSI Driver Operator on ROSA.

Portworx

To use Portworx for storage you must have a Portworx account and license. If needed, register for a Portworx account.

3.1 Configure the Portworx services spec

1. Log in to your Portworx account. Select to use the Portworx Enterprise edition. Then, click **Next**.
2. On the *Spec Generator - Enterprise* page, enter or select the following settings for your storage:
   1. Select the checkbox for *Use the Portworx Operator*.
   2. Select **2.10** or higher for the Portworx Version.
   3. Select the *Built-in* radio button for ETCD. Click **Next**.
   4. Select the *Cloud* radio button. Then, select *AWS* for Cloud Platform.
   5. Select the type of disk: *Create Using a Spec*
   6. Select the EBS volume type: **GP2**, Size (GB): **2000**. Click **Next**.
   7. Click **Next** to skip the Network configuration page.
   8. On the *Customize* page, select *OpenShift 4+*. Click **Finish**.
   9. Click *Agree* to accept the license agreement.
   10. Choose your own values to enter under the *Spec Name* and *Spec Tags* fields. Then, click **Save Spec**.
3. From *Spec List* page, find your Spec name and expand the Actions column menu and select **Copy to Clipboard**. Save the kubectl command. You use this command later.

3.2 Configure the AWS infrastructure for Portworx storage

1. Edit the Inbound Rules for both your master and worker nodes to allow for Network File System (NFS).
   1. Log in to the AWS EC2 Console.
   2. Under *EC2*, select *Instances*. Click an Instance ID for one of your worker nodes.
   3. Click the **Security** tab, and click the Security Group Name for the node.
   4. Click **Edit Inbound Rules**. Scroll to the bottom, and click **Add Rule**.
   5. Update the following settings for the rule:
      - Expand the first drop-down menu, and change the setting from *Custom TCP* to *All TCP*.
      - In the *CIDR blocks Source* field, enter **10.0.0.0/8**.
      - For the ports, open the following ports: 111, 2049, 20048, 17001-17020, 27017.
      - If you encounter issues during the installation, considering opening all inbound ports from **10.0.0.0/8**.
   6. Click **Save rules**.
   7. Repeat the above steps for your other nodes.
2. Create your Portworx policy in the AWS Identity and Access Management (IAM) tool.
   1. Go to **IAM** > **Access management** > **Policies**.
   2. Choose **Create policy**.
   3. Choose the **JSON** tab.
   4. Replace all the text with the following content:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PortworxPolicy",
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:ModifyVolume",
        "ec2:DetachVolume",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:DeleteTags",
        "ec2:DeleteVolume",
        "ec2:DescribeTags",
        "ec2:DescribeVolumeAttribute",
        "ec2:DescribeVolumesModifications",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVolumes",
        "ec2:DescribeInstances",
        "autoscaling:DescribeAutoScalingGroups"
      ],
    "Resource": [
      "*"
    ]
  }
]
}
```

5. Click **Next: Tags**. Then, click **Next: Review**.
6. Enter the name for your new policy. Then, click **Create policy**.

3. Attach the policy.
    1. Go to **IAM** > **Roles**.
    2. Click the worker-role name for your cluster.
    3. Click **Attach policy**. Search for, and select, the policy that you created. Then, select to attach the policy.

3.3 Install the Portworx operator from Red Hat OpenShift Container Platform

1. Open the Red Hat OpenShift Container Platform console for your Red Hat OpenShift Service on AWS cluster.
2. Go to **Operators** > **OperatorHub**.
3. Search for Portworx Enterprise. Then, click **Install** > **Install**.
4. Portworx Enterprise is now be listed under *Installed Operators*.

3.4 Configure the Portworx operator

1. Import the Spec from Portworx.
    1. Run `oc login` to log in to your Red Hat OpenShift Service on AWS cluster.
    2. Run `kubectl apply -f <command>` where `<command>` is the command that you saved earlier from the Portworx console.
2. Verify that Portworx Enterprise shows as one of the Installed Operators in `Project: kube-system`. Click Portworx Enterprise tile.

   In the **Storage Cluster** tab, you can see the storage cluster being initialized. Wait until Status is `Phase: Online`.

3. Create the Portworx storage classes. Follow the `Define a custom Portworx storage class` instructions in Installing recommended storage providers and configuring storage classes: Portworx.

4. Install IBM Cloud Pak for AIOps

You are now ready to install IBM Cloud Pak for AIOps.

Follow the procedure for installing IBM Cloud Pak for AIOps from step 3 in one of the following topics, according to your requirements:

- Starter installation (CLI)
- Online production installation of IBM Cloud Pak for AIOps (console method)
- Online production installation of IBM Cloud Pak for AIOps (CLI method)

Deleting the Red Hat OpenShift Service on AWS cluster

If you no longer require your Red Hat OpenShift Service on AWS cluster, you can remove it with the following steps.

1. Run `oc login` to log in to your Red Hat OpenShift Service on AWS cluster.
2. Run the following command to delete the cluster:

   ```
   rosa delete cluster -c <clustername> --watch
   ```

3. Run the following command to verify that the cluster is deleted:

   ```
   rosa list clusters
   ```

   Ensure that your cluster no longer is listed.

4. Delete any Identity and Access Management (IAM) policies for the cluster.

Installing IBM Cloud Pak for AIOps on AWS GovCloud

If you are installing IBM Cloud Pak® for AIOps on AWS GovCloud, then you must complete some additional steps before you rejoin the main installation procedure for installing IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform.

Getting Started with IBM Cloud Pak for AIOps and AWS Marketplace

When you purchase IBM Cloud Pak for AIOps through the Amazon Web Services (AWS) Marketplace, IBM begins processing your order immediately. To complete your deployment, use the following steps:

- Obtain your IBM Cloud Pak for AIOps entitlement key by following the prompts that are provided in the **`Order Processed Successfully`** email that you received from IBM. The entitlement key is needed to install IBM Cloud Pak for AIOps.
- Obtain your Red Hat® OpenShift® Container Platform entitlement by following the prompts that are provided in the **`ACTION NEEDED: Your Red Hat Account Update`** email that you received from IBM, or in the **`Your Red Hat order has been processed`** email that you received from Red Hat. The sender and content of this email depends on the product purchased.
- Review the information about enabling FIPS compliance for IBM Cloud Pak for AIOps: Federal Information Processing Standards (FIPS).
- Review the storage requirements for IBM Cloud Pak for AIOps: Hardware requirements - Storage.
- Review the information about viewing your audit logs: Audit logging. Audit logging is a FISMA requirement. When you complete your deployment, audit logging is configured and enabled by default.

Before you begin

Ensure that you meet the following prerequisites:

- You have an AWS GovCloud account.
- You have a Red Hat subscription.
- You have a Portworx account and license. The following steps are tested with, and require, Portworx storage. If needed, register for a Portworx account.
- Ensure that your environment meets the prerequisites for AWS GovCloud.

Installation procedure

Follow these steps to install IBM Cloud Pak for AIOps on AWS GovCloud:

1. Create a Red Hat OpenShift Container Platform cluster on AWS GovCloud
2. Configure storage
3. Install IBM Cloud Pak for AIOps

1. Create a Red Hat OpenShift cluster on AWS GovCloud

To create your Red Hat OpenShift cluster, follow the instructions in the Red Hat OpenShift documentation for Installing a Cluster on AWS into a Government Region.

These steps include provisioning a Virtual Private Cloud (VPC) in your AWS account that conforms to the requirements for Red Hat OpenShift Container Platform.

Important:

- When you are defining the **`install-config.yaml`** for your Red Hat OpenShift deployment, select worker and master nodes that meet your resource requirements for IBM Cloud Pak for AIOps as detailed in the topic Hardware requirements - Storage.
- If your deployment must be FIPS compliant, then you must comply with the steps in Enabling Federal Information Processing Standards (FIPS) support for IBM Cloud Pak for AIOps when creating your Red Hat OpenShift cluster and configuring your IBM Cloud Pak for AIOps installation.

2. Configure storage

Portworx is the tested and supported storage option for installing IBM Cloud Pak for AIOps on AWS GovCloud.

Note: Storage classes and storage providers cannot be changed after you install IBM Cloud Pak for AIOps. OADP backup and restore requires that a ReadWriteMany (RWX) storage class must be provided. If OADP backup and restore is not needed, a ReadWriteOnce (RWO) storage class can be provided as the **`RWX-storage-class-name`** in the installation instance CR YAML file. This configuration cannot be changed after IBM Cloud Pak for AIOps is installed.

2.1 Configure the Portworx services spec

1. Log in to your Portworx account. Select to use the Portworx Enterprise edition. Then, click **Next**.
2. On the *Spec Generator - Enterprise* page, enter or select the following settings for your storage:
    1. Select the checkbox for *Use the Portworx Operator*.
    2. Select **`2.10`** or higher for the Portworx Version.
    3. Select the *Built-in* radio button for ETCD. Click **Next**.
    4. Select the *Cloud* radio button. Then, select *AWS* for Cloud Platform.

5. Select the type of disk: *Create Using a Spec*
6. Select the EBS volume type: `GP2`, Size (GB): `2000`. Click **Next**.
7. Click **Next** to skip the *Network configuration* page.
8. On the *Customize* page, select *OpenShift 4+*. Click **Finish**.
9. Click *Agree* to accept the license agreement.
10. Choose your own values to enter under the *Spec Name* and *Spec Tags* fields. Then, click **Save Spec**.

3. From the *Spec List* page, find your Spec name and expand the Actions column menu and select **Copy to Clipboard**. Save the kubectl command, which is needed later.

2.2 Configure the AWS infrastructure for Portworx storage

1. Edit the Inbound Rules for both your master and worker nodes to allow for Network File System (NFS).

    1. Log in to the AWS EC2 Console.
    2. Under *EC2*, select *Instances*. Click an Instance ID for one of your worker nodes.
    3. Click the **Security** tab, and click the Security Group Name for the node.
    4. Click **Edit Inbound Rules**. Scroll to the bottom, and click **Add Rule**.
    5. Update the following settings for the rule:
        - Expand the first drop-down menu, and change the setting from *Custom TCP* to *All TCP*.
        - In the *CIDR blocks Source* field, enter `10.0.0.0/8`.
        - For the ports, open the following ports: 111, 2049, 20048, 17001-17020, 27017.
        - If you encounter issues during the installation, considering opening all inbound ports from `10.0.0.0/8`.
    6. Click **Save rules**.
    7. Repeat the preceding steps for your other nodes.

2. Create your Portworx policy in the AWS Identity and Access Management (IAM) tool.

    1. Go to **IAM** > **Access management** > **Policies**.
    2. Choose **Create policy**.
    3. Choose the **JSON** tab.
    4. Replace all the text with the following content:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PortworxPolicy",
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:ModifyVolume",
        "ec2:DetachVolume",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:DeleteTags",
        "ec2:DeleteVolume",
        "ec2:DescribeTags",
        "ec2:DescribeVolumeAttribute",
        "ec2:DescribeVolumesModifications",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVolumes",
        "ec2:DescribeInstances",
        "autoscaling:DescribeAutoScalingGroups"
      ],
    "Resource": [
      "*"
    ]
  }
]
}
```

    5. Click **Next: Tags**. Then, click **Next: Review**.
    6. Enter the name for your new policy. Then, click **Create policy**.

3. Attach the policy.

    1. Go to **IAM** > **Roles**.
    2. Click the worker-role name for your cluster.
    3. Click **Attach policy**. Search for, and select, the policy that you created. Then, select to attach the policy.

2.3 Install the Portworx operator from Red Hat OpenShift Container Platform

1. Open the Red Hat OpenShift Container Platform console for your AWS GovCloud cluster.
2. Go to **Operators** > **OperatorHub**.
3. Search for Portworx Enterprise. Then, click **Install** > **Install**.
4. Portworx Enterprise is now be listed under *Installed Operators*.

2.4. Configure the Portworx operator

1. Import the Spec from Portworx.

    1. Run `oc login` to log in to your AWS GovCloud cluster.

2. Run `kubectl apply -f <command>` where `<command>` is the command that you saved earlier from the Portworx console.

2. Verify that Portworx Enterprise shows as one of the Installed Operators in `Project: kube-system`. Click Portworx Enterprise tile.

   In the **Storage Cluster** tab, you can see the storage cluster initialization. Wait until the Status is `Phase: Online`.

3. Create the Portworx storage classes. Follow the `Define a custom Portworx storage class` instructions in Installing recommended storage providers and configuring storage classes: Portworx.

3. Install IBM Cloud Pak for AIOps

You are now ready to install IBM Cloud Pak for AIOps.

Follow the procedure for installing IBM Cloud Pak for AIOps from step 3 in one of the following topics, according to your requirements:

- Starter installation (CLI)
- Online production installation of IBM Cloud Pak for AIOps (console method)
- Online production installation of IBM Cloud Pak for AIOps (CLI method)

Upgrading IBM Cloud Pak for AIOps on OpenShift

Upgrade an existing online or offline deployment of IBM Cloud Pak® for AIOps 4.7.0 or later to 4.8.0.

You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 4.6.1 or earlier.

Important: If IBM® Netcool® Operations Insight® is deployed on the same cluster as IBM Cloud Pak for AIOps, ensure that Netcool Operations Insight is at version 1.6.13 or later before you upgrade IBM Cloud Pak for AIOps. Failure to do so may result in a broken Netcool Operations Insight deployment, as the IBM Cloud Pak for AIOps upgrade process updates a shared component that is used by both applications.

Upgrade paths

IBM Cloud Pak for AIOps uses a versioning system of X.Y.Z, where X=version, Y=release, and Z=patch. For example, IBM Cloud Pak for AIOps 4.7.1 is version 4, release 7, patch 1. IBM Cloud Pak for AIOps upgrades can be release upgrades or patch upgrades.

You can upgrade by a maximum of one release at a time. If the release that you want to upgrade from is more than one release behind the release that you are upgrading to, then you must upgrade sequentially. For example, you cannot upgrade directly from IBM Cloud Pak for AIOps 4.4.0 to v4.8.0, you must use the following sequence:

1. Use the v4.5.1 documentation to upgrade from IBM Cloud Pak for AIOps 4.4.0 to v4.5.1.
2. Use the v4.6.1 documentation to upgrade from IBM Cloud Pak for AIOps 4.5.1 to v4.6.1.
3. Use the v4.7.1 documentation to upgrade from IBM Cloud Pak for AIOps 4.6.1 to v4.7.1.
4. Use this documentation, the v4.8.0 documentation to upgrade from IBM Cloud Pak for AIOps 4.7.1 to v4.8.0.

Controlling upgrade

From IBM Cloud Pak for AIOps 4.7.0 onwards, all upgrades must be initiated manually. Automatic upgrade of IBM Cloud Pak for AIOps does not occur. For more information, see Whats New in the IBM Cloud Pak for AIOps 4.7.0 documentation.

Guidance for upgrades that require an Red Hat OpenShift upgrade

Before you upgrade IBM Cloud Pak for AIOps, ensure that you are on a version of Red Hat® OpenShift® Container Platform that your current and target versions of IBM Cloud Pak for AIOps both support. This helps to ensure compatibility and avoid potential issues caused by platform API changes.

If you already have a version of Red Hat OpenShift that your current and target versions of IBM Cloud Pak for AIOps both support but you want to upgrade Red Hat OpenShift, then complete the IBM Cloud Pak for AIOps upgrade before you upgrade Red Hat OpenShift.

Example upgrade scenario

You have IBM Cloud Pak for AIOps 4.7.0 deployed on Red Hat OpenShift 4.13, and you want to upgrade to IBM Cloud Pak for AIOps 4.8.0 on Red Hat OpenShift 4.17.

The following table shows the supported Red Hat OpenShift versions for your current and target versions of IBM Cloud Pak for AIOps:

Table 1. Supported Red Hat OpenShift versions for current and target application versions

IBM Cloud Pak for AIOps version	Supported Red Hat OpenShift versions
4.7.0	4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16
4.8.0	4.14, 4.15, 4.16, 4.17

The upgrade steps for this example upgrade scenario are:

1. Upgrade Red Hat OpenShift 4.13 to v4.14, v4.15, or v4.16.
2. Upgrade IBM Cloud Pak for AIOps 4.7.0 to v4.8.0.
3. Upgrade Red Hat OpenShift to v4.17.

Upgrade procedure

The upgrade procedure depends on whether your deployment is an online or offline (air-gapped) deployment:

- Online upgrade of IBM Cloud Pak for AIOps (console method)
- Online upgrade of IBM Cloud Pak for AIOps (CLI method)
- Offline upgrade of IBM Cloud Pak for AIOps

For more information about online and offline deployments, see Installing IBM Cloud Pak for AIOps.

Online upgrade of IBM Cloud Pak for AIOps (console method)

Use these instructions to upgrade an online deployment of IBM Cloud Pak® for AIOps 4.7.0 or later to 4.8.0.

Overview

This procedure is for deployments of IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform. This procedure can be used on an online deployment of IBM Cloud Pak for AIOps 4.7.0 or later, and can still be used if the deployment has had hotfixes applied.

You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 4.6.1 or earlier. For more information, see Upgrade paths.

If you have an offline deployment of IBM Cloud Pak for AIOps on Red Hat OpenShift, follow the instructions in Upgrading IBM Cloud Pak for AIOps (offline).

If you have a deployment of IBM Cloud Pak for AIOps on Linux, follow the instructions in Online upgrade of IBM Cloud Pak for AIOps on Linux.

Important: If IBM® Netcool® Operations Insight® is deployed on the same cluster as IBM Cloud Pak for AIOps, ensure that Netcool Operations Insight is at version 1.6.13 or later before you upgrade IBM Cloud Pak for AIOps. Failure to do so may result in a broken Netcool Operations Insight deployment, as the IBM Cloud Pak for AIOps upgrade process updates a shared component that is used by both applications.

Before you begin

Notes:

- Ensure that you are on a version of Red Hat OpenShift that your current and target versions of IBM Cloud Pak for AIOps both support. If you already have a qualifying version of Red Hat OpenShift but you want to upgrade it, then complete the IBM Cloud Pak for AIOps upgrade first. For more information, see Guidance for upgrades that require an Red Hat OpenShift upgrade.
- Some steps must still be performed with the Red Hat® OpenShift® Container Platform command line interface (CLI). Ensure that you are logged in to your Red Hat OpenShift Container Platform cluster with `oc login` for any steps that use the OpenShift command-line interface (CLI).
- Red Hat OpenShift Container Platform requires a user with `cluster-admin` privileges for the following operations:
  - Update the catalog
  - Update the operator subscription

Warnings:

- Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources are lost when IBM Cloud Pak for AIOps is upgraded, and must be manually reapplied after upgrade. For more information, see Manual adjustments are not persisted.
- The upgrade cannot be removed or rolled back.
- If you previously increased the size of a PVC directly, then you must follow the correct procedure that is supplied in Resizing storage to ensure that the size is updated by the operator. Failure to do so before upgrading IBM Cloud Pak for AIOps causes the operator to attempt to restore a lower default value for the PVC, and causes an error in your IBM Cloud Pak for AIOps deployment.

Database migration during upgrade

The IBM Cloud Pak for AIOps 4.7.x to v4.8 upgrade process migrates the Elasticsearch database to an OpenSearch database.

Extra resources are needed during the upgrade, and are shown in the following table. These resources can be released after the upgrade is complete.

Table 1. Extra resources required during upgrade

Size	vCPU	Memory	Disk
Starter	2	8Gi	100Gi
Production	6	24Gi	300Gi

If IBM Sales representatives and Business Partners supplied you with a custom profile, then you must calculate the additional resource requirements as follows:

1. Run the following commands to view your custom profile ConfigMap.

   ```
   export PROJECT_CP4AIOPS=<project>
   oc get configmap -n "${PROJECT_CP4AIOPS}" $(oc get installation -n "${PROJECT_CP4AIOPS}" -o
   jsonpath='{.items[0].status.customProfileConfigmap}') -o yaml
   ```

   Where `<project>` is the namespace (project) that IBM Cloud Pak for AIOps is deployed in.

2. Locate the section for `elasticsearch` in the output, and calculate the additional resources that are needed for the duration of upgrade.

   The additional vCPU needed is `replicas` multiplied by `requests.cpu`, and the additional memory needed is `replicas` multiplied bu `requests.memory`.

   Example output for an `elasticsearch` section, where upgrade will need 15 additional vCPU (5 x 3) and 40960 Mi memory (5 x 8192).

```
elasticsearch:
     replicas: 5
     resources:
       requests:
         cpu: 3000m
         memory: 8192Mi
       limits:
         cpu: 6000m
         memory: 8192Mi
```

The migration causes approximately 10 minutes of downtime for the following IBM Cloud Pak for AIOps components: log integrations, ticket integrations, log anomaly model training, and the Insights Dashboard. If alert archiving to Elasticsearch is enabled on your deployment, then the downtime for the migration might be extended depending on the quantity of alerts stored.

During upgrade, you can run the following command to monitor the status of the migration:

```
oc describe installation -n <project>
```

Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.

Example output for a migration in progress:

```
Conditions:
    Last Transition Time:            2024-10-29T16:58:55Z
    Message:                         Foreground Elasticsearch migration in progress
    Observed Generation:             2
    Reason:                          ForegroundMigrationInProgress
    Status:                          True
    Type:                            MigratingElastic
```

When migration is complete, **Message** has a value of **`Elasticsearch migration completed`**.

Note: Some integrations might be marked as **`Not Running`** while the migration is in progress. This is expected behavior, and the integrations must not be altered until the migration is complete.

After all services resume function, historic log data is migrated in the background. Historic log data is not accessible for training new models while this migration occurs. The duration of this background migration depends on the quantity of historic log data stored.

Upgrade procedure

Follow these steps to upgrade your online IBM Cloud Pak for AIOps deployment.

1. Ensure cluster readiness
2. Update the catalog
3. Update the operator subscription
4. Verify the deployment
5. Post upgrade actions

1. Ensure cluster readiness

Recommended: Take a backup before upgrading. For more information, see Backup and restore.

1. Ensure that your cluster still meets all of the prerequisites for deployment. For more information, see Planning.

   Ensure that your cluster has the additional resources that are required for upgrade, as outlined in Database migration during upgrade.

2. Run the IBM Cloud Pak for AIOps prerequisite checker script.

   Run the prerequisite checker script to ensure that your Red Hat OpenShift Container Platform cluster is correctly set up for an IBM Cloud Pak for AIOps upgrade.

   Download the prerequisite checker script from github.com/IBM, and run it with the following command:

   ```
   ./prereq.sh -n <project> --ignore-allocated
   ```

   Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.

   Important: If you are installing on a multi-zone cluster, then also specify the **`-m`** flag to assess whether there are sufficient resources to withstand a zone outage.

   Note: The prerequisite checker script might show inadequate resources in the *Resource Summary* because the script does not account for resources already being in use by the upgrading deployment. This can be ignored, as can the following message: *[FAIL] Small or Large Profile Install Resources*.

   Example output:

   ```
   # ./prereq.sh -n cp4aiops --ignore-allocated
   [INFO] Starting IBM Cloud Pak for AIOps prerequisite checker v4.8...

   CLI: oc

   [INFO] ================================Platform Version Check================================
   [INFO] Checking Platform Type....
   ```

```
[INFO] You are using Openshift Container Platform
[INFO] OCP version 4.16.7 is compatible but only nodes with AMD64 architectures are supported at this time.
[INFO] ================================Platform Version Check================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping configuration check for
Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Cert Manager Check================================
[INFO] Checking for Cert Manager operator

[INFO] Successfully functioning cert-manager found.

CLUSTERSERVICEVERSION             NAMESPACE
ibm-cert-manager-operator.v4.2.8  ibm-cert-manager

[INFO] ================================Cert Manager Check================================

[INFO] ================================Licensing Service Operator Check================================
[INFO] Checking for Licensing Service operator

[INFO] Successfully functioning licensing service operator found.

CLUSTERSERVICEVERSION          NAMESPACE
ibm-licensing-operator.v4.2.8  ibm-licensing

[INFO] ================================Licensing Service Operator Check================================

[INFO] ================================Starter or Production Install Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource Summary================================================
[INFO]                                                Nodes     |     vCPU       |  Memory(GB)
[INFO] Starter (Non-HA) Base (available/required)       [  9 / 3 ]   [  144 / 47 ]    [  289 / 123 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 3 ]   [  144 / 55 ]    [  289 / 136 ]

[INFO] Production (HA) Base (available/required)        [  9 / 6 ]   [  144 / 136 ]   [  289 / 310 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 6 ]   [  144 / 162 ]   [  289 / 368 ]
[INFO] ================================Resource Summary================================================
[INFO] Cluster currently has resources available to create a Starter (Non-HA) install of Cloud Pak for AIOps

[INFO] ================================Prerequisite Checker Tool Summary================================
   [  PASS  ] Platform Version Check
   [  PASS  ] Storage Provider
   [  PASS  ] Starter (Non-HA) Base Install Resources
   [  FAIL  ] Production (HA) Base Install Resources
   [  PASS  ] Cert Manager Operator Installed
   [  PASS  ] Licensing Service Operator Installed
[INFO] ================================Prerequisite Checker Tool Summary================================
```

3. Delete any evicted **connector-orchestrator** pods.
    1. From the **Project** list, select the project (namespace) that IBM Cloud Pak for AIOps is deployed in if your deployment is namespace scoped, or **openshift-operators** if your deployment has a cluster-wide scope.
    2. From the left menu, select **Workloads** > **Pods**, and search for **connector-orchestrator**.
    3. If there are any **connector-orchestrator** pods with a **STATUS** of **Evicted**, then select **Delete Pod** from the three dots menu at the end of the row.
4. Run the following command from the command line to ensure the safety of the Postgres database during upgrade.

    Important: Failure to run this step might lead to a complete loss of data.

    ```
    oc label clusters.postgresql.k8s.enterprisedb.io -n <project> $(oc get installations.orchestrator.aiops.ibm.com -n
    <project> -o jsonpath='{.items[].metadata.name}')-edb-postgres operator.ibm.com/opreq-do-not-uninstall='true'
    ```

    Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.

2. Update the catalog

Update the **ibm-aiops-catalog** to use the v4.8.0 digest.

1. Go to **Administration** > **Cluster Settings** > **Configuration** > **OperatorHub** > **Sources**.

2. Select **ibm-aiops-catalog**, and then click the **YAML** tab.
3. Update the value of *image* to be **icr.io/cpopen/ibm-aiops-catalog@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753**
4. Go to **Administration** > **Cluster Settings** > **Configuration** > **OperatorHub** > **Sources**.
5. Verify that the *ibm-aiops-catalog*, *ibm-cert-manager-catalog* and *ibm-licensing-catalog* CatalogSource objects are present, and that the image for *ibm-aiops-catalog* is **icr.io/cpopen/ibm-aiops-catalog@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753**.

Note: During the upgrade of Cloud Pak for AIOps, Kubernetes jobs might fail and re-run. If a job succeeds on the second or third attempt, there can be one or two pods in **Error** state and one pod in the **Completed** state. If the job fails repeatedly, the attempt is abandoned, and the logs from failed pods are used to determine the cause of the failure. When you determine the cause for the failure, you can delete the job, and the operator can recreate it to reattempt the operations.

3. Update the operator subscription

Use the following steps to update the **spec.channel** value of the IBM Cloud Pak for AIOps subscription to v4.8.

1. Log in to your OpenShift cluster's console.
2. Select **Home** > **Search**.
3. From the **Project** list, select the project (namespace) that your IBM Cloud Pak for AIOps subscription is deployed in. This is your IBM Cloud Pak for AIOps project if your deployment is namespace scoped, or **openshift-operators** if your deployment has a cluster wide scope.
4. In the **Resources** list, select **SUB Subscription**. A list of subscriptions is displayed.
5. Click the subscription that has a **Name** of **ibm-aiops-orchestrator**. A new window with the subscription details for **ibm-aiops-orchestrator** is displayed.
6. Click the value in the **Update channel** box. A new window called **Change Subscription update channel** is displayed.
7. Change the channel to **v4.8** and click **Save**.
8. If you are installing with a cluster wide scope (**AllNamespaces** mode), use the following steps to refresh the connectors' secret:

   For more information about installation modes, see Operator installation mode.

   1. Select **Home** > **Search**.
   2. In the **Resources** list, select **Secret**. A list of secrets is displayed.
   3. Click the three dots at the end of the row that has a name of **cp4waiops-connectors-deploy-cert-secret**, and click **Delete Secret**.

4. Verify the deployment

Use the following procedure to check the status of your upgraded IBM Cloud Pak for AIOps deployment. It takes around 60-90 minutes for the upgrade to complete (subject to the speed with which images can be pulled).

1. Log in to your OpenShift cluster's console.
2. Click **Operators** > **Installed Operators**.
3. From the **Project** list, select the project (namespace) that IBM Cloud Pak for AIOps is deployed in if your deployment is namespace scoped, or **openshift-operators** if your deployment has a cluster-wide scope.
4. Locate **IBM Cloud Pak for AIOps** in the list, and verify that the annotation underneath it shows **4.8.0**.
5. Select **IBM Cloud Pak for AIOps** and then click the **IBM Cloud Pak for AIOps** tab.
6. Under **Installations**, look for the entry with the name that you specified for your IBM Cloud Pak for AIOps instance, and verify that it has a **Status** of *Phase: Running*, which means that your deployment is complete and successful.

   (Optional): If you want to see more detail about the status of your deployment's components, select the entry with the name that you specified for your IBM Cloud Pak for AIOps instance, and then switch to the **YAML** view. Scroll down to the **Status** section near the end of the YAML. A component's installation is complete and successful when the component has a value of **Ready**.

   Example YAML:

```
status:
  size: small
  customProfileConfigmap: aiops-custom-size-profile
  customProfileValidationStatus: >-
    Custom profile configmap not found, continue installation process without
    customization
  storageclasslargeblock: rook-ceph-rbd
  componentstatus:
    issueresolutioncore: Ready
    kafka: Ready
    aiopsanalyticsorchestrator: Ready
    aiopsedge: Ready
    tunnel: Ready
    lifecycleservice: Ready
    zenservice: Ready
```

```
    flinkcluster: Ready
    cluster: Ready
    elasticsearchcluster: Ready
    aiopsui: Ready
    redissentinel: Ready
    <...>
```

(Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the upgrade fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

Important: Wait for the deployment to enter a **Running** phase before continuing to the next step.

7. Verify that the database migration was successful.

    1. Switch to the **YAML** view. Scroll down to the **Status** section near the end of the YAML. Review the **conditions** section and verify that the **MigratingElastic** condition has a status of **False** and a reason of **MigrationComplete**.

        Example YAML for a successful migration:

        ```
        conditions:
        - lastTransitionTime: '2024-11-28T06:51:58Z'
          message: Elasticsearch migration completed
          observedGeneration: 3
          reason: MigrationComplete
          status: 'False'
          type: MigratingElastic
        ```

    2. Verify that all data is available in IBM Cloud Pak for AIOps as expected.

    Important: If database migration is not complete, review the troubleshooting entry Database migration is stuck. If this does not resolve the problem, then do not continue to the next step and contact IBM Support.

5. Post upgrade actions

1. Delete the persistent volumes that were used for storing data in Elasticsearch, the database that was migrated from.

    Run the following command:

    ```
    oc delete pvc -l app.kubernetes.io/managed-by=ibm-elastic-operator -n <project>
    ```

    Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.

2. If you previously set up backup or restore on your deployment, then you must follow the instructions in Upgrading IBM Cloud Pak for AIOps backup and restore artifacts.

3. If the *EXPIRY_SECONDS* environment variable was set for configuring log anomaly alerts, the environment variable was not retained in the upgrade. After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly alerts.

4. If you have a metric integration configured that stops working after upgrade, then you must follow the instructions in After upgrade, a metric integration goes into a failed state.

Online upgrade of IBM Cloud Pak for AIOps (CLI method)

Use these instructions to upgrade an online deployment of IBM Cloud Pak® for AIOps 4.7.0 or later to 4.8.0.

Overview

This procedure is for deployments of IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform. This procedure can be used on an online deployment of IBM Cloud Pak for AIOps 4.7.0 or later, and can still be used if the deployment has had hotfixes applied.

You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 4.6.1 or earlier. For more information, see Upgrade paths.

If you have an offline deployment of IBM Cloud Pak for AIOps on Red Hat OpenShift, follow the instructions in Upgrading IBM Cloud Pak for AIOps (offline).

If you have a deployment of IBM Cloud Pak for AIOps on Linux, follow the instructions in Online upgrade of IBM Cloud Pak for AIOps on Linux.

Important: If IBM® Netcool® Operations Insight® is deployed on the same cluster as IBM Cloud Pak for AIOps, ensure that Netcool Operations Insight is at version 1.6.13 or later before you upgrade IBM Cloud Pak for AIOps. Failure to do so may result in a broken Netcool Operations Insight deployment, as the IBM Cloud Pak for AIOps upgrade process updates a shared component that is used by both applications.

Before you begin

Notes:

- Ensure that you are on a version of Red Hat OpenShift that your current and target versions of IBM Cloud Pak for AIOps both support. If you already have a qualifying version of Red Hat OpenShift but you want to upgrade it, then complete the IBM Cloud Pak for AIOps upgrade first. For more information, see Guidance for upgrades that require an Red Hat OpenShift upgrade.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- Red Hat OpenShift requires a user with `cluster-admin` privileges for the following operations:
  - Update the catalog
  - Update the operator subscription

Warnings:

- Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources are lost when IBM Cloud Pak for AIOps is upgraded, and must be manually reapplied after upgrade. For more information, see Manual adjustments are not persisted.
- The upgrade cannot be removed or rolled back.
- If you previously increased the size of a PVC directly, then you must follow the correct procedure that is supplied in Resizing storage to ensure that the size is updated by the operator. Failure to do so before upgrading IBM Cloud Pak for AIOps causes the operator to attempt to restore a lower default value for the PVC, and causes an error in your IBM Cloud Pak for AIOps deployment.

Database migration during upgrade

The IBM Cloud Pak for AIOps 4.7.x to v4.8 upgrade process migrates the Elasticsearch database to an OpenSearch database.

Extra resources are needed during the upgrade, and are shown in the following table. These resources can be released after the upgrade is complete.

Table 1. Extra resources required during upgrade

Size	vCPU	Memory	Disk
Starter	2	8Gi	100Gi
Production	6	24Gi	300Gi

If IBM Sales representatives and Business Partners supplied you with a custom profile, then you must calculate the additional resource requirements as follows:

1. Run the following commands to view your custom profile ConfigMap.

   ```
   export PROJECT_CP4AIOPS=<project>
   oc get configmap -n "${PROJECT_CP4AIOPS}" $(oc get installation -n "${PROJECT_CP4AIOPS}" -o
   jsonpath='{.items[0].status.customProfileConfigmap}') -o yaml
   ```

   Where `<project>` is the namespace (project) that IBM Cloud Pak for AIOps is deployed in.

2. Locate the section for `elasticsearch` in the output, and calculate the additional resources that are needed for the duration of upgrade.

   The additional vCPU needed is `replicas` multiplied by `requests.cpu`, and the additional memory needed is `replicas` multiplied bu `requests.memory`.

   Example output for an `elasticsearch` section, where upgrade will need 15 additional vCPU (5 x 3) and 40960 Mi memory (5 x 8192).

   ```
   elasticsearch:
        replicas: 5
        resources:
          requests:
            cpu: 3000m
            memory: 8192Mi
          limits:
            cpu: 6000m
            memory: 8192Mi
   ```

The migration causes approximately 10 minutes of downtime for the following IBM Cloud Pak for AIOps components: log integrations, ticket integrations, log anomaly model training, and the Insights Dashboard. If alert archiving to Elasticsearch is enabled on your deployment, then the downtime for the migration might be extended depending on the quantity of alerts stored.

During upgrade, you can run the following command to monitor the status of the migration:

```
oc describe installation -n ${PROJECT_CP4AIOPS}
```

Example output for a migration in progress:

```
Conditions:
    Last Transition Time:            2024-10-29T16:58:55Z
    Message:                         Foreground Elasticsearch migration in progress
    Observed Generation:             2
    Reason:                          ForegroundMigrationInProgress
    Status:                          True
    Type:                            MigratingElastic
```

When migration is complete, **Message** has a value of `Elasticsearch migration completed`.

Note: Some integrations might be marked as `Not Running` while the migration is in progress. This is expected behavior, and the integrations must not be altered until the migration is complete.

After all services resume function, historic log data is migrated in the background. Historic log data is not accessible for training new models while this migration occurs. The duration of this background migration depends on the quantity of historic log data stored.

Upgrade procedure

Follow these steps to upgrade your online IBM Cloud Pak for AIOps deployment.

1. Ensure cluster readiness
2. Update the catalog
3. Update the operator subscription
4. Verify the deployment
5. Post upgrade actions

1. Ensure cluster readiness

Recommended: Take a backup before upgrading. For more information, see Backup and restore.

1. Ensure that your cluster still meets all of the prerequisites for deployment. For more information, see Planning.

   Ensure that your cluster has the additional resources that are required for upgrade, as outlined in Database migration during upgrade.

2. If you still have **waiops_var.sh** from when you installed IBM Cloud Pak for AIOps, then run the following command from the directory that the script is in, to set the environment variables that are used later.

   ```
   . ./waiops_var.sh
   ```

   If you do not have **waiops_var.sh**, then run the following commands to set the environment variables that you need for upgrade.

   ```
   export PROJECT_CP4AIOPS=<project>
   export INSTALL_MODE_NAMESPACE=<install_namespace>
   ```

   Where

   - **<project>** is the namespace (project) that your IBM Cloud Pak for AIOps subscription is deployed in.
   - **<install_namespace>** is ${PROJECT_CP4AIOPS} if your deployment is namespace scoped, or **openshift-operators** if your deployment has a cluster-wide scope.

3. Run the IBM Cloud Pak for AIOps prerequisite checker script.

   Run the prerequisite checker script to ensure that your Red Hat OpenShift Container Platform cluster is correctly set up for an IBM Cloud Pak for AIOps upgrade.

   Download the prerequisite checker script from github.com/IBM, and run it with the following command:

   ```
   ./prereq.sh -n ${PROJECT_CP4AIOPS} --ignore-allocated
   ```

   Important: If you are installing on a multi-zone cluster, then also specify the **-m** flag to assess whether there are sufficient resources to withstand a zone outage.

   Note: The prerequisite checker script might show inadequate resources in the *Resource Summary* because the script does not account for resources already being in use by the upgrading deployment. This can be ignored, as can the following message: *[FAIL] Small or Large Profile Install Resources*.

   Example output:

   ```
   # ./prereq.sh -n cp4aiops --ignore-allocated
   [INFO] Starting IBM Cloud Pak for AIOps prerequisite checker v4.8...

   CLI: oc

   [INFO] ================================Platform Version Check================================
   [INFO] Checking Platform Type....
   [INFO] You are using Openshift Container Platform
   [INFO] OCP version 4.16.7 is compatible but only nodes with AMD64 architectures are supported at this time.
   [INFO] ================================Platform Version Check================================

   [INFO] ================================Storage Provider================================
   [INFO] Checking storage providers
   [INFO] No IBM Storage Fusion Found... Skipping configuration check.

   [INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping configuration check for
   Portworx.
   [INFO] Openshift Data Foundation found.
   [INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

   Checking Openshift Data Foundation Configuration...
   Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
   [INFO] Pods in openshift-storage project are "Running" or "Completed"
   [INFO] ocs-storagecluster-ceph-rbd exists.
   [INFO] ocs-storagecluster-cephfs exists.
   [INFO] No warnings or failures found when checking for Storage Providers.
   [INFO] ================================Storage Provider================================

   [INFO] ================================Cert Manager Check================================
   [INFO] Checking for Cert Manager operator

   [INFO] Successfully functioning cert-manager found.

   CLUSTERSERVICEVERSION             NAMESPACE
   ibm-cert-manager-operator.v4.2.8  ibm-cert-manager

   [INFO] ================================Cert Manager Check================================

   [INFO] ================================Licensing Service Operator Check================================
   ```

```
[INFO] Checking for Licensing Service operator

[INFO] Successfully functioning licensing service operator found.

CLUSTERSERVICEVERSION          NAMESPACE
ibm-licensing-operator.v4.2.8  ibm-licensing

[INFO] ================================Licensing Service Operator Check================================

[INFO] ================================Starter or Production Install Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource Summary============================================================
[INFO]                                                  Nodes     |     vCPU       |  Memory(GB)
[INFO] Starter (Non-HA) Base (available/required)       [  9 / 3 ]   [  144 / 47 ]    [  289 / 123 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 3 ]   [  144 / 55 ]    [  289 / 136 ]

[INFO] Production (HA) Base (available/required)        [  9 / 6 ]   [  144 / 136 ]   [  289 / 310 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 6 ]   [  144 / 162 ]   [  289 / 368 ]
[INFO] ================================Resource Summary============================================================
[INFO] Cluster currently has resources available to create a Starter (Non-HA) install of Cloud Pak for AIOps

[INFO] ================================Prerequisite Checker Tool Summary================================
      [  PASS  ] Platform Version Check
      [  PASS  ] Storage Provider
      [  PASS  ] Starter (Non-HA) Base Install Resources
      [  FAIL  ] Production (HA) Base Install Resources
      [  PASS  ] Cert Manager Operator Installed
      [  PASS  ] Licensing Service Operator Installed
[INFO] ================================Prerequisite Checker Tool Summary================================
```

4. Delete any evicted **connector-orchestrator** pods.

    1. Run the following command to check if there are any evicted **connector-orchestrator** pods.

        ```
        oc get pods -n ${PROJECT_CP4AIOPS} | grep connector-orchestrator
        ```

    2. Cleanup any evicted connector-orchestrator pods.

        If the previous command returned any pods with a *STATUS* of **Evicted**, then run the following command to delete each of them.

        ```
        oc delete pod -n ${PROJECT_CP4AIOPS} <connector_orchestrator>
        ```

        Where **<connector_orchestrator>** is a pod returned in the previous step.

5. Run the following command to ensure the safety of the Postgres database during upgrade.

    Important: Failure to run this step might lead to a complete loss of data.

    ```
    oc label clusters.postgresql.k8s.enterprisedb.io -n $PROJECT_CP4AIOPS $(oc get
    installations.orchestrator.aiops.ibm.com -n $PROJECT_CP4AIOPS -o jsonpath='{.items[].metadata.name}')-edb-postgres
    operator.ibm.com/opreq-do-not-uninstall='true'
    ```

2. Update the catalog

Update the **ibm-aiops-catalog** to use the v4.8.0 digest.

1. Run the following command:

    ```
    cat << EOF | oc apply -f -
    apiVersion: operators.coreos.com/v1alpha1
    kind: CatalogSource
    metadata:
      name: ibm-aiops-catalog
      namespace: openshift-marketplace
    spec:
      displayName: ibm-aiops-catalog
      publisher: IBM Content
      sourceType: grpc
      image: icr.io/cpopen/ibm-aiops-catalog@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753
    EOF
    ```

2. Verify that the *ibm-aiops-catalog*, *ibm-cert-manager-catalog* and *ibm-licensing-catalog* CatalogSource objects are in the output that is returned by the following command:

    ```
    oc get CatalogSources -n openshift-marketplace
    ```

    Example output:

    ```
    oc get CatalogSources -n openshift-marketplace
    NAME                    DISPLAY                       TYPE   PUBLISHER   AGE
    ibm-aiops-catalog       ibm-aiops-catalog             grpc   IBM         2m
    ibm-cert-manager-catalog ibm-cert-manager             grpc   IBM         2m
    ibm-licensing-catalog    IBM License Service Catalog  grpc   IBM         2m
    ```

Note: During the upgrade of Cloud Pak for AIOps, Kubernetes jobs might fail and re-run. If a job succeeds on the second or third attempt, there can be one or two pods in `Error` state and one pod in the `Completed` state. If the job fails repeatedly, the attempt is abandoned, and the logs from failed pods are used to determine the cause of the failure. When you determine the cause for the failure, you can delete the job, and the operator can recreate it to reattempt the operations.

3. Update the operator subscription

1. Update the `spec.channel` value of the IBM Cloud Pak for AIOps subscription to the release that you want to upgrade to, v4.8.

   ```
   oc patch subscription.operators.coreos.com ibm-aiops-orchestrator -n ${INSTALL_MODE_NAMESPACE} --type=json -
   p='[{'op': 'replace', 'path': '/spec/channel', 'value': 'v4.8'}]'
   ```

2. If you are installing in `AllNamespaces` mode, run the following command to refresh the connectors' secret:

   ```
   oc delete secret cp4waiops-connectors-deploy-cert-secret -n "${PROJECT_CP4AIOPS}"
   ```

   For more information about installation modes, see Operator installation mode.

4. Verify the deployment

4.1 Check the deployment

Run the following command to check that the **PHASE** of your deployment is `Updating`.

```
oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
```

Example output:

```
NAME           PHASE     LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
ibm-cp-aiops   Updating  Accepted   rook-cephfs     rook-ceph-block           3m
```

It takes around 60-90 minutes for the upgrade to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to `Running`. If your installation phase does not change to `Running`, then use the following command to find out which components are not ready:

```
oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not Ready'
```

Example output:

```
lifecycleservice: Not Ready
zenservice: Not Ready
```

To see details about why a component is `Not Ready` run the following command, where `<component>` is the component that is not ready, for example `zenservice`.

```
oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
```

(Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the upgrade fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

Important: Wait for the deployment to enter a `Running` phase before continuing to the next step.

4.2 Verify database migration

1. Verify that the database migration was successful.

   Run the following command, and check that the `MigratingElastic` condition has a status of `false` and a reason of `MigrationCompleted`.

   ```
   oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'MigratingElastic' -B 5
   ```

   Example output for a successful migration:

   ```
   Conditions:
    - lastTransitionTime: "2024-10-29T17:16:55Z"
        message: Elasticsearch migration completed
        observedGeneration: 2
        reason: MigrationComplete
        status: "False"
        type: MigratingElastic
   ```

2. Verify that all data is available in IBM Cloud Pak for AIOps as expected.

Important: If database migration is not complete, review the troubleshooting entry Database migration is stuck. If this does not resolve the problem, then do not continue to the next step and contact IBM Support.

4.3 Check the version

Run the following command and check that the *VERSION* that is returned is `4.8.0`.

```
oc get csv -l operators.coreos.com/ibm-aiops-orchestrator.${INSTALL_MODE_NAMESPACE} -n ${INSTALL_MODE_NAMESPACE}
```

Example output:

```
oc get csv -l operators.coreos.com/ibm-aiops-orchestrator.cp4aiops -n cp4aiops
```

```
NAME                              DISPLAY                    VERSION  REPLACES                          PHASE
ibm-aiops-orchestrator.v4.8.0  IBM Cloud Pak for AIOps  4.8.0    ibm-aiops-orchestrator.v4.7.1  Succeeded
```

5. Post upgrade actions

1. Delete the persistent volumes that were used for storing data in Elasticsearch, the database that was migrated from.

   Run the following command:

   ```
   oc delete pvc -l app.kubernetes.io/managed-by=ibm-elastic-operator -n ${PROJECT_CP4AIOPS}
   ```

2. If you previously set up backup or restore on your deployment, then you must follow the instructions in Upgrading IBM Cloud Pak for AIOps backup and restore artifacts.

3. If the *EXPIRY_SECONDS* environment variable was set for configuring log anomaly alerts, the environment variable was not retained in the upgrade. After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly alerts.

4. If you have a metric integration configured that stops working after upgrade, then you must follow the instructions in After upgrade, a metric integration goes into a failed state.

Offline upgrade of IBM Cloud Pak for AIOps

Use these instructions to upgrade an offline deployment of IBM Cloud Pak® for AIOps 4.7.0 or later to 4.8.0.

Overview

This procedure is for deployments of IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform. This procedure can be used on an offline deployment of IBM Cloud Pak for AIOps 4.7.0 or later, and can still be used if the deployment has had hotfixes applied.

You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 4.6.1 or earlier. For more information, see Upgrade paths.

If you have an online deployment of IBM Cloud Pak for AIOps on Red Hat OpenShift, follow the instructions in Upgrading IBM Cloud Pak for AIOps (online).

If you have a deployment of IBM Cloud Pak for AIOps on Linux®, follow the instructions in Offline upgrade of IBM Cloud Pak for AIOps on Linux.

Important: If IBM® Netcool® Operations Insight® is deployed on the same cluster as IBM Cloud Pak for AIOps, ensure that Netcool Operations Insight is at version 1.6.13 or later before you upgrade IBM Cloud Pak for AIOps. Failure to do so may result in a broken Netcool Operations Insight deployment, as the IBM Cloud Pak for AIOps upgrade process updates a shared component that is used by both applications.

Before you begin

Notes:

- Ensure that you are on a version of Red Hat OpenShift that your current and target versions of IBM Cloud Pak for AIOps both support. If you already have a qualifying version of Red Hat OpenShift but you want to upgrade it, then complete the IBM Cloud Pak for AIOps upgrade first. For more information, see Guidance for upgrades that require an Red Hat OpenShift upgrade.
- Ensure that you are logged in to your Red Hat® OpenShift® Container Platform cluster with `oc login` for any steps that use the OpenShift command-line interface (CLI).
- Red Hat OpenShift Container Platform requires a user with `cluster-admin` privileges for the following operations:
  - Mirror images
  - Update the catalog
  - Update the operator subscription

Warnings:

- Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources are lost when IBM Cloud Pak for AIOps is upgraded, and must be manually reapplied after upgrade. For more information, see Manual adjustments are not persisted.
- The upgrade cannot be removed or rolled back.
- If you previously increased the size of a PVC directly, then you must follow the correct procedure that is supplied in Resizing storage to ensure that the size is updated by the operator. Failure to do so before upgrading IBM Cloud Pak for AIOps causes the operator to attempt to restore a lower default value for the PVC, and causes an error in your IBM Cloud Pak for AIOps deployment.

Database migration during upgrade

The IBM Cloud Pak for AIOps 4.7.x to v4.8 upgrade process migrates the Elasticsearch database to an OpenSearch database.

Extra resources are needed during the upgrade, and are shown in the following table. These resources can be released after the upgrade is complete.

Table 1. Extra resources required during upgrade

Size	vCPU	Memory	Disk

Size	vCPU	Memory	Disk
Starter	2	8Gi	100Gi
Production	6	24Gi	300Gi

If IBM Sales representatives and Business Partners supplied you with a custom profile, then you must calculate the additional resource requirements as follows:

1. Run the following commands to view your custom profile ConfigMap.

   ```
   export PROJECT_CP4AIOPS=<project>
   oc get configmap -n "${PROJECT_CP4AIOPS}" $(oc get installation -n "${PROJECT_CP4AIOPS}" -o
   jsonpath='{.items[0].status.customProfileConfigmap}') -o yaml
   ```

   Where **<project>** is the namespace (project) that IBM Cloud Pak for AIOps is deployed in.

2. Locate the section for **elasticsearch** in the output, and calculate the additional resources that are needed for the duration of upgrade.

   The additional vCPU needed is **replicas** multiplied by **requests.cpu**, and the additional memory needed is **replicas** multiplied bu **requests.memory**.

   Example output for an **elasticsearch** section, where upgrade will need 15 additional vCPU (5 x 3) and 40960 Mi memory (5 x 8192).

   ```
   elasticsearch:
        replicas: 5
        resources:
          requests:
            cpu: 3000m
            memory: 8192Mi
          limits:
            cpu: 6000m
            memory: 8192Mi
   ```

The migration causes approximately 10 minutes of downtime for the following IBM Cloud Pak for AIOps components: log integrations, ticket integrations, log anomaly model training, and the Insights Dashboard. If alert archiving to Elasticsearch is enabled on your deployment, then the downtime for the migration might be extended depending on the quantity of alerts stored.

During upgrade, you can run the following command to monitor the status of the migration:

```
oc describe installation -n ${PROJECT_CP4AIOPS}
```

Example output for a migration in progress:

```
Conditions:
    Last Transition Time:            2024-10-29T16:58:55Z
    Message:                         Foreground Elasticsearch migration in progress
    Observed Generation:             2
    Reason:                          ForegroundMigrationInProgress
    Status:                          True
    Type:                            MigratingElastic
```

When migration is complete, **Message** has a value of **Elasticsearch migration completed**.

Note: Some integrations might be marked as **Not Running** while the migration is in progress. This is expected behavior, and the integrations must not be altered until the migration is complete.

After all services resume function, historic log data is migrated in the background. Historic log data is not accessible for training new models while this migration occurs. The duration of this background migration depends on the quantity of historic log data stored.

Upgrade procedure

Follow these steps to upgrade your offline IBM Cloud Pak for AIOps deployment.

1. Ensure cluster readiness
2. Download CASE files
3. Mirror images
4. Update the catalog
5. Update the operator subscription
6. Verify the deployment
7. Post upgrade actions

1. Ensure cluster readiness

Recommended: Take a backup before upgrading. For more information, see Backup and restore.

1. Ensure that your cluster still meets all of the prerequisites for an air-gapped deployment.

   Ensure that your cluster has the additional resources that are required for upgrade, as outlined in Database migration during upgrade.

   Review the steps for your installation approach:

   - Bastion host: Prerequisites
   - Portable device: Prerequisites
2. Ensure that the IBM Catalog Management Plug-in for IBM Cloud Pak® (**ibm-pak-plugin**) is at the required level.

Important: `ibm-pak-plugin` v1.9 and lower cannot process the IBM Cloud Pak for AIOps 4.8.0 catalog. You must ensure that you have `ibm-pak-plugin` v1.10 or higher installed.

1. Check which version of `ibm-pak-plugin` you have installed.

   Run the following command on your bastion host, portable compute device, or connected compute device if you are using a portable storage device.

   ```
   oc ibm-pak --version
   ```

   Example output:

   ```
   oc ibm-pak --version
   v1.13.0
   ```

2. If the `ibm-pak-plugin` version is lower than v1.10.0, then you must download and install the most recent version.

   Follow the steps for your installation approach:

   - Bastion host: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.
   - Portable device: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.

3. Download scripts.

   1. Download the prerequisite checker script and copy it to your air-gapped environment.

      For more information about the script, including how to download and run it, see github.com/IBM.

   2. Download the IBM Cloud Pak for AIOps uninstall script, and copy it to your air-gapped environment.

      For more information about the script, including how to download and run it, see github.com/IBM.

   3. (Optional) Download the status checker script, and copy it to your air-gapped environment.

      For more information about the script, including how to download and run it, see github.com/IBM. The status checker script can be used in step 9. Verify the deployment to give information about the status of your deployment. The use of this script is optional, as status can be found directly from the `ibm-aiops-orchestrator` custom resource.

4. If you still have `waiops_var.sh` from when you installed IBM Cloud Pak for AIOps, then run the following command from the directory that the script is in, to set the environment variables that are used later.

   ```
   . ./waiops_var.sh
   ```

   If you do not have `waiops_var.sh`, then run the following commands to set the environment variables that you need for upgrade.

   ```
   export PROJECT_CP4AIOPS=<project>
   export INSTALL_MODE_NAMESPACE=<install_namespace>
   export CATALOG_SRC_CP4AIOPS=ibm-aiops-catalog
   ```

   Where

   - `<project>` is the namespace (project) that your IBM Cloud Pak for AIOps subscription is deployed in.
   - `<install_namespace>` is ${PROJECT_CP4AIOPS} if your deployment is namespace scoped, or `openshift-operators` if your deployment has a cluster-wide scope.

5. Run the IBM Cloud Pak for AIOps prerequisite checker script.

   Run the prerequisite checker script to ensure that your Red Hat OpenShift Container Platform cluster is correctly set up for an IBM Cloud Pak for AIOps upgrade.

   Download the prerequisite checker script from github.com/IBM, and run it with the following command:

   ```
   ./prereq.sh -n ${PROJECT_CP4AIOPS} --ignore-allocated
   ```

   Important: If you are installing on a multi-zone cluster, then also specify the `-m` flag to assess whether there are sufficient resources to withstand a zone outage.

   Note: The prerequisite checker script might show inadequate resources in the *Resource Summary* because the script does not account for resources already being in use by the upgrading deployment. This can be ignored, as can the following message: *[FAIL] Small or Large Profile Install Resources*.

   Example output:

   ```
   # ./prereq.sh -n cp4aiops --ignore-allocated

   [INFO] Starting IBM Cloud Pak for AIOps prerequisite checker v4.8...

   CLI: oc

   [INFO] ================================Platform Version Check================================
   [INFO] Checking Platform Type....
   [INFO] You are using Openshift Container Platform
   [INFO] OCP version 4.16.7 is compatible but only nodes with AMD64 architectures are supported at this time.
   [INFO] ================================Platform Version Check================================

   [INFO] ================================Storage Provider================================
   [INFO] Checking storage providers
   [INFO] No IBM Storage Fusion Found... Skipping configuration check.
   ```

```
[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping configuration check for
Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Cert Manager Check================================
[INFO] Checking for Cert Manager operator

[INFO] Successfully functioning cert-manager found.

CLUSTERSERVICEVERSION             NAMESPACE
ibm-cert-manager-operator.v4.2.8  ibm-cert-manager

[INFO] ================================Cert Manager Check================================

[INFO] ================================Licensing Service Operator Check================================
[INFO] Checking for Licensing Service operator

[INFO] Successfully functioning licensing service operator found.

CLUSTERSERVICEVERSION          NAMESPACE
ibm-licensing-operator.v4.2.8  ibm-licensing

[INFO] ================================Licensing Service Operator Check================================

[INFO] ================================Starter or Production Install Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource Summary================================================
[INFO]                                                  Nodes     |     vCPU       |  Memory(GB)
[INFO] Starter (Non-HA) Base (available/required)       [  9 / 3 ]   [  144 / 47 ]    [  289 / 123 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 3 ]   [  144 / 55 ]    [  289 / 136 ]

[INFO] Production (HA) Base (available/required)        [  9 / 6 ]   [  144 / 136 ]   [  289 / 310 ]
[INFO]     (+ Log Anomaly Detection & Ticket Analysis)  [  9 / 6 ]   [  144 / 162 ]   [  289 / 368 ]
[INFO] ================================Resource Summary================================================
[INFO] Cluster currently has resources available to create a Starter (Non-HA) install of Cloud Pak for AIOps

[INFO] ================================Prerequisite Checker Tool Summary================================
      [  PASS  ] Platform Version Check
      [  PASS  ] Storage Provider
      [  PASS  ] Starter (Non-HA) Base Install Resources
      [  FAIL  ] Production (HA) Base Install Resources
      [  PASS  ] Cert Manager Operator Installed
      [  PASS  ] Licensing Service Operator Installed
[INFO] ================================Prerequisite Checker Tool Summary================================
```

6. Delete any evicted **connector-orchestrator** pods.

    1. Run the following command to check if there are any evicted **connector-orchestrator** pods.

        ```
        oc get pods -n ${PROJECT_CP4AIOPS} | grep connector-orchestrator
        ```

    2. Cleanup any evicted connector-orchestrator pods.

        If the previous command returned any pods with a *STATUS* of **Evicted**, then run the following command to delete each of them.

        ```
        oc delete pod -n ${PROJECT_CP4AIOPS} <connector_orchestrator>
        ```

        Where **<connector_orchestrator>** is a pod returned in the previous step.

7. Run the following command to ensure the safety of the Postgres database during upgrade.

    Important: Failure to run this step might lead to a complete loss of data.

    ```
    oc label clusters.postgresql.k8s.enterprisedb.io -n $PROJECT_CP4AIOPS $(oc get
    installations.orchestrator.aiops.ibm.com -n $PROJECT_CP4AIOPS -o jsonpath='{.items[].metadata.name}')-edb-postgres
    operator.ibm.com/opreq-do-not-uninstall='true'
    ```

2. Download CASE files

On your OpenShift cluster, rerun step 2 of the air-gap installation procedure **Download CASE files** to download the latest CASE files.

Follow the steps for your installation approach:

- Bastion host: Download the CASE
- Portable device: Download the CASE

3. Mirror images

Rerun step 3 of the air-gap installation procedure to mirror the updated images to the offline registry.

Follow the steps for your installation approach:

- Bastion host: Mirror images
- Portable device: Mirror images

4. Update the catalog

1. Rerun step 5.2 of the air-gap installation procedure **Create the catalog source** to update your catalog source.

   Follow the steps for your installation approach:

   - Bastion host: Create the catalog source
   - Portable device: Create the catalog source

2. From IBM Cloud Pak for AIOps 4.8.0, IBM Cloud Pak® foundational services Cert Manager and License Service are deployed using their own independent catalogs, and a new catalog called **ibm-aiops-catalog** is used for the remaining IBM Cloud Pak for AIOps operators. You must migrate your subscriptions to use the new catalogs.

   Run the following script from the command line:

```
# New catalog name to migrate to
target="ibm-aiops-catalog"

# Previous catalog names to migrate from
catalogs="""aiops-analytics-operator-catalog
aiops-lifecycle-operator-catalog
cloud-native-postgresql-catalog
ibm-aiops-appconnect-catalog
ibm-aiops-ir-core-operators
ibm-aiops-opencontent-flink-catalog
ibm-asm-catalog ibm-cp-waiops-catalog
ibm-cp-waiops-elastic-catalog
ibm-cp-waiops-flink-catalog
ibm-aiopsedge-case
ibm-redis-cp-operator-catalog
ibm-secure-tunnel-operator-catalog
ibm-watson-ai-manager-catalog
ibm-watson-aiops-ui-catalog
opencloud-operators"""

echo "Getting all subscriptions..."
all_subs=$(oc get subscriptions.operators.coreos.com --no-headers -n ${PROJECT_CP4AIOPS})

for catalog in $catalogs; do
  subs=$(echo "$all_subs" | awk -v c=$catalog '$3 == c {print $1}' -)
  if [[ -n $subs ]]; then
    echo "Updating catalog source for subscription $subs from $catalog to $target"
    oc patch subscriptions.operators.coreos.com $subs -n ${PROJECT_CP4AIOPS} --type=json --patch='[{"op": "replace",
"path": "/spec/source", "value": '$target'}]'
  fi

  oc delete catalogsource $catalog -n openshift-marketplace
done
```

5. Update the operator subscription

1. Run the following command to update the IBM Cloud Pak for AIOps subscription to use the new channel, v4.8.

```
oc patch subscription.operators.coreos.com ibm-aiops-orchestrator -n ${INSTALL_MODE_NAMESPACE} --type=json -
p='[{'op': 'replace', 'path': '/spec/channel', 'value': 'v4.8'}]'
```

2. If you are installing in **AllNamespaces** mode, run the following command to refresh the connectors' secret:

```
oc delete secret cp4waiops-connectors-deploy-cert-secret -n "${PROJECT_CP4AIOPS}"
```

   For more information about installation modes, see Operator installation mode.

6. Verify the deployment

6.1 Check the deployment

Run the following command to check that the **PHASE** of your deployment is **Updating**.

```
oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
```

Example output:

```
NAME           PHASE     LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
ibm-cp-aiops   Updating  Accepted   rook-cephfs     rook-ceph-block           3m
```

It takes around 60-90 minutes for the upgrade to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to `Running`. If your installation phase does not change to `Running`, then use the following command to find out which components are not ready:

```
oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not Ready'
```

Example output:

```
lifecycleservice: Not Ready
zenservice: Not Ready
```

To see details about why a component is `Not Ready` run the following command, where `<component>` is the component that is not ready, for example `zenservice`.

```
oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
```

(Optional) If you downloaded the status checker script earlier in step 1.5 Ensure cluster readiness, then you can also run this script to see information about the status of your deployment.

If the upgrade or migration fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

Important: Wait for the deployment to enter a `Running` phase before continuing to the next step.

6.2 Verify database migration

1. Verify that the database migration was successful.

   Run the following command, and check that the `MigratingElastic` condition has a status of `false` and a reason of `MigrationCompleted`.

   ```
   oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'MigratingElastic' -B 5
   ```

   Example output for a successful migration:

   ```
   Conditions:
    - lastTransitionTime: "2024-10-29T17:16:55Z"
        message: Elasticsearch migration completed
        observedGeneration: 2
        reason: MigrationComplete
        status: "False"
        type: MigratingElastic
   ```

2. Verify that all data is available in IBM Cloud Pak for AIOps as expected.

Important: If database migration is not complete, review the troubleshooting entry Database migration is stuck. If this does not resolve the problem, then do not continue to the next step and contact IBM Support.

6.3 Check the version

Run the following command and check that the *VERSION* that is returned is `4.8.0`.

```
oc get csv -l operators.coreos.com/ibm-aiops-orchestrator.${INSTALL_MODE_NAMESPACE} -n ${INSTALL_MODE_NAMESPACE}
```

Example output:

```
oc get csv -l operators.coreos.com/ibm-aiops-orchestrator.cp4aiops -n cp4aiops
NAME                           DISPLAY                  VERSION  REPLACES                       PHASE
ibm-aiops-orchestrator.v4.8.0  IBM Cloud Pak for AIOps  4.8.0    ibm-aiops-orchestrator.v4.7.1  Succeeded
```

7. Post upgrade actions

1. Delete the persistent volumes that were used for storing data in Elasticsearch, the database that was migrated from.

   Run the following command:

   ```
   oc delete pvc -l app.kubernetes.io/managed-by=ibm-elastic-operator -n ${PROJECT_CP4AIOPS}
   ```

2. If you previously set up backup or restore on your deployment, then you must follow the instructions in Upgrading IBM Cloud Pak for AIOps backup and restore artifacts.

3. If the *EXPIRY_SECONDS* environment variable was set for configuring log anomaly alerts, the environment variable was not retained in the upgrade. After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly alerts.

4. If you have a metric integration configured that stops working after upgrade, then you must follow the instructions in After upgrade, a metric integration goes into a failed state.

Administering IBM Cloud Pak for AIOps on OpenShift

After you install IBM Cloud Pak® for AIOps, you can access several options for administering and customizing your deployment.

- Backup and restore
- Configuring topology resources for production
- Health checks and monitoring
- Managing certificates
- Multi-zone rebalancing
- Resizing storage
- Restarting the environment
- Secure Tunnels
- Updating the deployment type

See also Cloud Pak administration

To gather data about your deployment for support or troubleshooting purposes, you can install and run the `MustGather` script:

- MustGather tool

Back up and restore (IBM Cloud Pak for AIOps on OpenShift)

You can use Red Hat OpenShift APIs for Data Protection (OADP) or IBM Fusion to backup and restore a deployment of IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform.

Backup and restore with IBM Fusion offers an incremental online backup with a console based user experience, which backup and restore with OADP does not. Backup and restore with IBM Fusion requires an IBM Fusion license. For more information, see the following topics:

- Backup and restore with OADP
- Backup and restore with Fusion (Cold Standby DR)

Backup and restore Cloud Pak for AIOps with OADP

Learn how to back up and restore IBM Cloud Pak for AIOps, such as for recovering from a disaster like a complete data center outage.

- Overview
- Planning
- Backup process
- Restore process
- Components

Overview

IBM Cloud Pak for AIOps backup and restore is based on the storage and database backup of critical datastores, cluster resources, and component data and settings. With the backup and restore feature, you can recover your services to a previous point in time backup to recover from data corruption, system failures, or user errors.

You can also use the backup and restore feature as a means of copying data from one cluster to another, or from one environment into another, such as for disaster recovery purposes.

If you are using backup data to restore environments on a new cluster in another data center, your recovery time objective (RTO) and recovery point objective (RPO) must be sufficient for this usage.

IBM Cloud Pak for AIOps uses the *OpenShift APIs for Data Protection* (OADP) to backup data to object storage. In the case of a failure, the APIs and OADP are used to restore data from the backups.

For more information about these tools, see:

- OpenShift APIs for Data Protection (OADP) features
- Velero documentation

Planning for backup and restore

When you are planning your IBM Cloud Pak for AIOps backup and restore strategy, consider what hardware requirements are needed for backing up your environment:

- You need to install the backup and restore tools in the Red Hat OpenShift Container Platform cluster. These tools include the OpenShift APIs for Data Protection (OADP) which creates required CRs such as `DataProtectionApplication` and `BackupStorageLocation`. You need to configure the OpenShift APIs and OADP with the appropriate object storage for storing your backups. This object storage must support RWX (read-write-execute) mode.
- You can backup both smaller starter and larger production sized cluster deployments.
- You can use backed up data to restore data to the existing cluster where the data was backed up, or to a new cluster.
- A restored cluster should have the same size and high availability level as the original cluster.
- Data is copied and stored in object Storage as part of the backup process. During a restoration, the same data is fetched and restored.

Notes:

- The backup and restore processes can take time. Using a backup to restore data or configurations is most useful when your recover time objective (RTO) and recovery point objectives (RPO) are met.
- As your data grows, the size of your backup storage might need to grow.
- A backup and restore of Red Hat OpenShift Container Platform or ETCD is not completed as part of the IBM Cloud Pak for AIOps backup and restore feature.
- Configure the duration between your backups to be less than your RPO. By default, the RPO is 12 hours, which requires backups to run once every 12 hours. The default RTO is 4 hours.
- The Flink state for log anomaly is not backed up and any training is lost during a failure or disaster. You need to run the training again after you recover from the disaster.
- If you upgrade your deployment, you must setup back and restore again. For more information, see Upgrading IBM Cloud Pak for AIOps backup and restore artifacts.
- A backup and restore of the optional product IBM Cognos® Analytics is not completed as part of the IBM Cloud Pak for AIOps backup and restore feature.
- If you perform backup and restore for Cloud Pak for AIOps with OADP, then IBM Spectrum Fusion must not be installed on source and target clusters or else the backup and restore procedures might fail.

Backup process

The IBM Cloud Pak® for AIOps backup process is capable of backing up data for both IBM Cloud Pak for AIOps and Infrastructure Automation. The backup of data is completed by backing up the datastores that include IBM Cloud Pak for AIOps data, and if available, Infrastructure Automation data. Some datastores are backed up and restored in their entirety. Other datastores are backed up and restored depending on a particular component need.

The following table shows the IBM Cloud Pak® for AIOps datastores and persistent volume claims (PVCs) that are backed up:

Table. Backed up datastores

Datastores	Backup type	Example backup PVC name	Access mode
`CouchDB`	Volume backup	data-c-example-couchdbcluster-m-0	ReadWriteOnce
`Cassandra`	Data export	back-aiops-topology-cassandra-0	ReadWriteOnce
`EDB Postgres`	Data export	postgres-backup-data	ReadWriteOnce
`Minio`	Volume backup	export-aimanager-ibm-minio-0	ReadWriteOnce
`ElasticSearch`	Partial (Elastic native volume backup used)	aiops-ibm-elasticsearch-es-server-snap	ReadWriteMany
`Kafka`	None		
`Flink`	None		
`metastore`	Data export	metastore-backup-data	ReadWriteOnce
Other resources like `Tunnel` custom resource	Volume export	other-resources-backup-data	ReadWriteOnce

The following table shows the backed up Kubernetes resources:

Table. Backed up resources

Backup resource name	Resource type
dummy-db	pod
es-backup	pod
backup-back-aiops-topology-cassandra	pod
backup-data-c-example-couchdbcluster-m	pod
backup-export-aimanager-ibm-minio	pod
backup-metastore	pod
backup-postgres	pod
backup-other-resources	pod
aiops-ir-core-model-secret	secret
aimanager-ibm-minio-access-secret	secret
zen-secrets-aes-key	secret
BundleManifest	custom resource
ConnectorConfiguration	custom resource
ConnectorSchemas	custom resource

The backup process is run by custom backup-jobs. When the backup job runs, pods can be scaled down to ensure consistent backups are created. The backup job calls OADP to back up Kubernetes resources and volumes. Native backup scripts are run to back up datastores. The backup job then scales up pods to match their state before the backup process began. Datastores that have a selective backup types are backed up only when the datastore is updated since the previous backup.

For more information about the backup process, see:

- Backing up IBM Cloud Pak for AIOps data.
- Backing up Infrastructure Automation data.

Notes: For Elasticsearch, the backup and restore process completes incremental backup snapshots of Elasticsearch. While creating the snapshot, Elasticsearch analyses the list of the index files that are stored in the repository that is being backed up and copies only the files that were created or changed since the last snapshot occurred. This process allows you to have multiple snapshots stored in a compact form.

Component data backups

When the backup process runs for IBM Cloud Pak for AIOps, data for the following components are backed up. This data includes cluster resources, such as Secrets, Pods, CRDs, CRs, PVCs, and PVs.

- **AI model management**

log anomaly detection and metric anomaly detection templates and models are backed up. In addition, model and training metadata, secrets, and configurations are backed up. If this data is not backed up and is lost during a disaster, you cannot run training until you collect training data again. In addition your models and templates can be lost, which can prevent principal component analysis (PCA) from running until you run training to rebuild the models. You might not be able to recreate historical model if data loss occurs.

For similar ticket analysis, the trained models are backed up.

Notes

- Component data, such as for log anomaly detections or similar ticket analysis, is not encrypted when it is backed up.
- Raw data and data-in-transit, such as Kafka topics, are not backed up.
- logtrain indices and other training data is not backed up other than training metadata.
- Before you run the backup process, stop all training that is in progress. Do not start or restart training until the backup process is completed

- **Change risk**

  For change risk, risk models, risk assessments, incidents, and ticket data assets are backed up.

  Notes

  - Data-in-transit is not backed up.
  - If change or incident ticket data loss occurs, you might need to run data collection again from your ticketing system.
  - In change risk model loss occurs, you need to recreate the lost models by using the IBM Cloud Pak for AIOps IBM Cloud Pak for AIOps console. Historical models can be difficult to re-create.
  - In the case of risk assessments loss, there is no way to reconstruct the assessments.
- **Automations** (Runbooks)

  For Automations, runbooks, automations, execution records, and configurations are backed up.

- **Integrations**

  Integration CRs, Secrets, and configurations, are backed up.

- **Event & Topology analytics**

  Probable cause data, Topology data, configurations and indexed data, is backed up.

- **Policy management**

  For policy management, operational policies and policy run time statistics (last run, count of executions) are backed up.

- **Secure Tunnel**

  Tunnel CRs, which represent connections, application mappings, and templates are backed up.

Restore process

To run the restore process, you first need to create a new cluster and install the required prerequisites on the cluster. The prerequisites include installing the required CLI tools, creating the storage classes, and updating the **`bcdr/common/aiops-config.json`** and **`bcdr/restore/restore-data.json`** files with the required configuration values. Then you need to configure OADP to point to the same s3 storage where the backup is located.

You then need to create the required namespaces for IBM Cloud Pak® for AIOps by running the **`restore-namespace.sh`** job. This script creates only the namespaces and namespace metadata, not the contents of the namespace. This namespace restore is needed because the metadata contains SELinux settings that must match the settings from the corresponding namespaces on the cluster where you took the backup.

Then, you need to complete the installation of IBM Cloud Pak for AIOps and create an IBM Cloud Pak for AIOps instance. With the instance created and OADP configured, you can run the OADP restore jobs to populate the instance. When the jobs complete you need to run any other post restore tasks that you need to complete.

Note: OADP does not support the overriding of resources during the restore. The restoration of resources already existing in the cluster are skipped. If this occurs, sync the existing resources and create only the resources that were not restored instead of creating a new resources Operator.

For more information about the restore process, see Restoring IBM Cloud Pak for AIOps.

Procedures for backing up and restoring IBM Cloud Pak for AIOps

- Installing the backup and restore tools
- Configuring ODF as backup storage for OADP
- Backing up IBM Cloud Pak for AIOps
- Restoring IBM Cloud Pak for AIOps
- Upgrading IBM Cloud Pak for AIOps backup and restore artifacts

Installing the backup and restore tools

Learn how to install the tools that are required for the backup and restore of IBM Cloud Pak for AIOps.

You need to install the Red Hat OpenShift APIs for Data Protection (OADP). OADP provides default Velero plug-ins that are integrated with storage providers to support backup operations.

Note: The backup and restore tools for IBM Cloud Pak for AIOps are the same as the tools for backing up and restoring Infrastructure Automation. If you have the tools installed for Infrastructure Automation, you do not need to complete the following procedure.

Prerequisites

- Provision the object storage (S3 bucket) for storing your backups. The backup process moves backup files to an S3 bucket, which must be in an S3 compatible location, such as Amazon Web Services (AWS).
- Ensure that the following command line tools are installed on your workstation:
  - `oc`
  - `jq`
  - `git`
- Ensure that you are logged in to your Red Hat® OpenShift® Container Platform cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).

Procedure

1. Install the Velero client CLI
2. Install OpenShift API for Data Protection (OADP)
3. (Offline deployments only) Mirror the backup and restore image

1. Install the Velero client CLI

Install the Velero client CLI on the workstation that you plan to use to run the scripts for backing up and restoring the cluster.

1. Run the following command to download the Velero CLI client:

   ```
   curl -LO https://github.com/vmware-tanzu/velero/releases/download/v1.12.3/velero-v1.12.3-linux-amd64.tar.gz
   ```

2. Extract the tarball:

   ```
   tar -xzf velero-v1.12.3-linux-amd64.tar.gz
   ```

3. Move the extracted velero binary to `/usr/local/bin`:

   ```
   mv velero-v1.12.3-linux-amd64/velero /usr/local/bin/velero
   ```

4. Run the following command to verify the installation.

   ```
   velero version
   ```

   Example output:

   ```
   Client:
    Version: v1.12.3
    Git commit: 06d12dec4030703cb46b2699ca434cdabd9edbb0
   velero version
   ```

2. Install Red Hat OpenShift APIs for Data Protection (OADP)

For more information, see the Red Hat OpenShift documentation About installing OADP.

As a convenience, if you are using S3 buckets on AWS, you can use the installation script `install-oadp.sh` in this section to simplify the process.

If you are using Red Hat® OpenShift® Data Foundation for your backup storage, follow the instructions in Configuring ODF as backup storage for OADP.

If you are not using Red Hat® OpenShift® Data Foundation for your backup storage, use the following steps:

1. Clone the IBM Cloud Pak for AIOps samples GitHub repository that includes the backup and restore scripts. Clone this repository by running the following command:

   ```
   git clone https://github.com/IBM/cp4waiops-samples.git
   ```

   If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps, then copy the cloned scripts to your air-gapped environment.

2. Update the installation files.

   1. Change to the `install` directory where the backup and restore scripts are downloaded.

      ```
      cd <Path>/bcdr/4.8.0/install
      ```

      Where `<Path>` is the path to where you cloned the IBM Cloud Pak for AIOps backup and restore scripts.

   2. Edit the `install-oadp-config.json` script. Configure the following settings:

      - `aws_access_key_id` - Access key ID to connect to your object storage S3 bucket.
      - `aws_secret_access_key` - Secret access key to connect to your S3 bucket.
      - `bucket_name` - Name of the S3 bucket where backup data will be stored.
      - `bucket_region` - Region where the S3 bucket is deployed.

- `namespace` - Namespace name in which you want to deploy OADP. If the namespace does not exist, it is created when you run the OADP installation script.
- `backup_label` - Label for creating a folder structure to organize sets of backups together.

3. Edit the `install-oadp.sh` script.

   If your version of Red Hat OpenShift is higher than v4.15, update `channel: stable-1.3` to `channel: stable-1.4`.

   Example excerpt:

   ```
   spec:
     channel: stable-1.4
     installPlanApproval: Automatic
     name: redhat-oadp-operator
     source: redhat-operators
     sourceNamespace: openshift-marketplace
   ```

3. Run the OADP installation script:

   ```
   nohup ./install-oadp.sh > install-oadp.log &
   ```

4. Check the logs by running the following command:

   ```
   tail -f install-oadp.log
   ```

5. Export an environment variable with the Velero namespace.

   ```
   export OADP_NAMESPACE=<oadpNamespace>
   ```

   Where `<oadpNamespace>` is the namespace where OADP is installed.

6. Check the Velero pods status by running the following command:

   ```
   oc get pods -n ${OADP_NAMESPACE}
   ```

   The Velero and Restic pods should be in a running state.

7. Check the status of the `backupStorageLocation` by running the following command:

   ```
   oc get backupStorageLocation -n ${OADP_NAMESPACE}
   ```

   The `backupStorageLocation` needs to be in an available status.

3. (Offline deployments only) Mirror the backup and restore image

If the deployment that you are backing up or restoring to is an offline (air-gapped) deployment, then you must copy the backup and restore image to a registry that is available to your air-gapped environment. You can do this by using either a bastion host or a portable device. For background information on offline (air-gapped) deployments, see Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline).

Prerequisites:

- The host that you are running the mirror from must allow access to `icr.io` on port 443.
- You must have a target registry available to mirror the backup and restore image into. The target registry must adhere to the requirements in Set up a target registry.

Follow the instructions in Mirroring with a bastion host or Mirroring with a portable device.

Mirroring with a bastion host

1. Set environment variables.

   ```
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export BCDR_IMAGE=<bcdr_image>
   export BCDR_UTILS_IMAGE=cp4waiops-bcdr-utils@sha256:1db594ad10da9c7049367b5dd4373e86af7e6bcb875758022e2e743c83cf486f
   export AIOPS_NAMESPACE=<aiops_namespace>
   ```

   Where

   - `<bcdr_image>` is the name of the backup and restore image, as given in the backup helm chart `values.yaml` file, in the form `cp4waiops-bcdr@{digest}`. An example value for BCDR_IMAGE is `cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355`.
   - `<aiops_namespace>` is the namespace that IBM Cloud Pak for AIOps is deployed in.

   If your bastion host must connect to the internet through a proxy, then also set the following environment variables:

   ```
   export https_proxy=http://proxy-server-hostname:port
   export http_proxy=http://proxy-server-hostname:port
   ```

2. Authenticate with the target registry.

   You do not need to authenticate with the IBM Entitled Registry, as the image that you require for backup and restore is freely available.

You need to authenticate with the target registry to enable the `oc image mirror` command to push the image to your target registry. Log in to your target registry to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file.

Run the following commands:

```
podman login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
```

Note: The authentication file is usually at `${XDG_RUNTIME_DIR}/containers/auth.json`. For more information, see the Options section in the Podman documentation.

3. Mirror the image.

   Connect the bastion host to the internet and disconnect if from the air-gapped environment. Then run the following command to mirror the backup and restore image to the target registry:

   ```
   oc image mirror icr.io/cpopen/cp4waiops/${BCDR_IMAGE} ${TARGET_REGISTRY}/cpopen/cp4waiops/${BCDR_IMAGE} -a
   ${REGISTRY_AUTH_FILE} --insecure
   oc image mirror icr.io/cpopen/cp4waiops/${BCDR_UTILS_IMAGE} ${TARGET_REGISTRY}/cpopen/cp4waiops/${BCDR_UTILS_IMAGE}
   -a ${REGISTRY_AUTH_FILE} --insecure
   ```

4. Create the `ImageContentSoucePolicy`.

   1. Switch to the namespace that IBM Cloud Pak for AIOps is deployed in.

      ```
      oc project ${AIOPS_NAMESPACE}
      ```

   2. Run the following command to create the `ImageContentSourcePolicy`.

      ```
      cat << EOF | oc apply -f -
      apiVersion: operator.openshift.io/v1alpha1
      kind: ImageContentSourcePolicy
      metadata:
       name: bcdr-mirrors
      spec:
       repositoryDigestMirrors:
        - mirrors:
          - ${TARGET_REGISTRY}/cpopen/cp4waiops/
          source: icr.io/cpopen/cp4waiops/
      EOF
      ```

Mirroring with a portable device

If you are using a portable storage device, then connect it to a host with access to the internet. If you are using a portable compute device, then connect it to the internet. Run the following steps.

1. Set environment variables.

   ```
   export IMAGE_PATH=<image-path>
   export BCDR_IMAGE=<bcdr_image>
   export BCDR_UTILS_IMAGE=cp4waiops-bcdr-utils@sha256:1db594ad10da9c7049367b5dd4373e86af7e6bcb875758022e2e743c83cf486f
   ```

   Where

   - `<image_path>` is the directory where you want the image to be stored.
   - `<bcdr_image>` is the name of the backup and restore image, as given in the backup helm chart `values.yaml` file, in the form `cp4waiops-bcdr@{digest}`. An example value for BCDR_IMAGE is `cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355`.

   If your device must connect to the internet through a proxy, then also set the following environment variables:

   ```
   export https_proxy=http://proxy-server-hostname:port
   export http_proxy=http://proxy-server-hostname:port
   ```

2. Mirror the images to the file system on your portable device.

   You do not need to authenticate with the IBM Entitled Registry, as the image that you require for backup and restore is freely available. Run the following command to mirror the image from the IBM Entitled Registry to the file system on your portable device.

   ```
   oc image mirror \
   icr.io/cpopen/cp4waiops/${BCDR_IMAGE} file://cpopen/cp4waiops/${BCDR_IMAGE} \
   --filter-by-os '.*' \
   --insecure \
   --skip-multiple-scopes \
   --dir "${IMAGE_PATH}" \
   --max-per-registry=1

   oc image mirror \
   icr.io/cpopen/cp4waiops/${BCDR_UTILS_IMAGE} file://cpopen/cp4waiops/${BCDR_UTILS_IMAGE} \
   --filter-by-os '.*' \
   --insecure \
   --skip-multiple-scopes \
   --dir "${IMAGE_PATH}" \
   --max-per-registry=1
   ```

3. Setup the file system in the air-gapped environment.

   1. (Portable storage device only) Copy files to the air-gapped environment

If you are using a portable storage device, you must copy the files at $IMAGE_PATH on the portable storage device to a local compute device in the air-gapped environment that has access to the target registry. If you are using a portable compute device, then these files are already present, and you can proceed to the next step.

2. Disconnect the device that has your file system (the portable compute device or the local compute device) from the internet and connect it to the air-gapped environment.

3. Set environment variables on the device in the air-gapped environment that has access to the target registry.

   If you are using a portable compute device, then set the following environment variables on it. If you are using a portable storage device, then set the following environment variables on your local compute device within the air-gapped environment.

   ```
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export IMAGE_PATH=<image_path>
   export BCDR_IMAGE=<bcdr_image>
   export BCDR_UTILS_IMAGE=cp4waiops-bcdr-
   utils@sha256:1db594ad10da9c7049367b5dd4373e86af7e6bcb875758022e2e743c83cf486f
   export AIOPS_NAMESPACE=<aiops_namespace>
   ```

   Where

   - **<bcdr_image>** is the name of the backup and restore image, as given in the backup helm chart **values.yaml** file, in the form **cp4waiops-bcdr@{digest}**. An example value for BCDR_IMAGE is **cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355**.
   - **<aiops_namespace>** is the namespace that IBM Cloud Pak for AIOps is deployed in.

4. Authenticate with the target registry.

   Authenticate with the target registry in the air-gapped environment that you are mirroring the image into. You need to authenticate with the target registry to enable the **oc image mirror** command to push the image to your target registry. Log in to your target registry to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file.

   Run the following commands:

   ```
   podman login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
   export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
   ```

   Note: The authentication file is usually at **${XDG_RUNTIME_DIR}/containers/auth.json**. For more information, see the Options section in the Podman documentation.

5. Mirror the images to the target registry from the file system.

   Run the following command on the device that has your file system (the portable compute device or the local compute device) to copy the image from the file system to the $TARGET_REGISTRY. Your device with the file system must be connected to both the target registry and the Red Hat® OpenShift® Container Platform cluster that you want to backup or restore IBM Cloud Pak for AIOps on.

   ```
   oc image mirror \
   file://cpopen/cp4waiops/${BCDR_IMAGE} ${TARGET_REGISTRY}/cpopen/cp4waiops/${BCDR_IMAGE} \
   --from-dir "${IMAGE_PATH}" \
   -a ${REGISTRY_AUTH_FILE} \
   --filter-by-os '.*' \
   --insecure  \
   --skip-multiple-scopes \
   --max-per-registry=1

   oc image mirror \
   file://cpopen/cp4waiops/${BCDR_UTILS_IMAGE} ${TARGET_REGISTRY}/cpopen/cp4waiops/${BCDR_UTILS_IMAGE} \
   --from-dir "${IMAGE_PATH}" \
   -a ${REGISTRY_AUTH_FILE} \
   --filter-by-os '.*' \
   --insecure  \
   --skip-multiple-scopes \
   --max-per-registry=1
   ```

6. Create the **ImageContentSoucePolicy**.

   1. Switch to the namespace that IBM Cloud Pak for AIOpsis deployed in.

      ```
      oc project ${AIOPS_NAMESPACE}
      ```

   2. Run the following command to create the **ImageContentSourcePolicy**.

      ```
      cat << EOF | oc apply -f -
      apiVersion: operator.openshift.io/v1alpha1
      kind: ImageContentSourcePolicy
      metadata:
       name: bcdr-mirrors
      spec:
       repositoryDigestMirrors:
        - mirrors:
          - ${TARGET_REGISTRY}/cpopen/cp4waiops/
          source: icr.io/cpopen/cp4waiops/
      EOF
      ```

Troubleshooting

- Red Hat OpenShift APIs for Data Protection (OADP) installation failed
- Backups are not showing after installation

Red Hat OpenShift APIs for Data Protection (OADP) installation failed

If your installation of OADP fails for any reason, delete any related files and reattempt the installation.

For example, the installation script **install-oadp.sh** might not complete successfully. If this occurs an error similar to the following message can display:

```
error: unable to recognize "STDIN": no matches for kind "DataProtectionApplication" in version
"oadp.openshift.io/v1alpha1"
```

This can occur if the network or cluster is running slow and taking too long to pull the required images. This delay causes the OADP operator to take too long to create the required custom resource.

If the preceding issue occurs, or the installation fails for another reason, run the following **oc** commands to delete any existing OADP installation files before attempting to install again.

1. Delete any cluster service versions (CSV) in the project:

   ```
   oc delete csv --all -n ${OADP_NAMESPACE}
   ```

2. Delete any subscriptions in the project:

   ```
   oc delete subscription --all -n ${OADP_NAMESPACE}
   ```

3. Delete any deployments in the project:

   ```
   oc delete deploy --all -n ${OADP_NAMESPACE}
   ```

4. Delete any daemonsets in the project:

   ```
   oc delete ds --all -n ${OADP_NAMESPACE}
   ```

5. Delete the project (namespace):

   ```
   oc delete project ${OADP_NAMESPACE}
   ```

Any existing installation is deleted and you can reattempt the installation.

Backups are not showing after installation

If your backups do not show, complete the following steps:

1. Correct the **bucket** and **prefix** property within the **backupstoragelocation** resource that is named **bcdr-s3-location**. The **bucket** and **prefix** name should match the value that is defined within the **install-oadp-config.json** file during the OADP installation.

   Run the **oc edit** command to edit the **backupstoragelocation** resource:

   ```
   oc edit backupstoragelocation bcdr-s3-location -n ${OADP_NAMESPACE}
   ```

2. Delete the Velero pod by running the following command:

   ```
   oc delete pod <pod> -n ${OADP_NAMESPACE}
   ```

   The backups should now be available.

3. Run the following command to list the backups:

   ```
   velero get backup
   ```

Configuring ODF as backup storage for OADP

Learn how to create an object storage bucket in Red Hat OpenShift Data Foundation that Red Hat OpenShift APIs for Data Protection (OADP) can use to store backups in.

Overview

Use these instructions if you want to use Red Hat® OpenShift® Data Foundation as the backup storage for an OADP backup of IBM Cloud Pak® for AIOps. For more information about backing up IBM Cloud Pak for AIOps using OADP, see Installing the backup and restore tools.

Procedure

1. Create an S3-compatible object storage bucket
2. Retrieve the S3 bucket information and credentials
3. Prepare the DataProtectionApplication configuration
4. Verify that the backupStorageLocation object is created
5. Configure the volume snapshot classes for OADP

6. Back up a project to validate the OADP configuration

1. Create an S3-compatible object storage bucket

1. Create a YAML file called **obc-backup.yml** with the following content:

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: backup
  namespace: openshift-adp
spec:
  storageClassName: openshift-storage.noobaa.io
  generateBucketName: backup
```

2. Run the following command to create an ObjectBucketClaim object called **backup** that uses the **openshift-storage.noobaa.io** storage class.

```
oc apply -f obc-backup.yml
```

2. Retrieve the S3 bucket information and credentials

When Red Hat OpenShift Data Foundation creates the **backup** ObjectBucketClaim instance, it also creates a secret and a configuration map named **backup**. The **backup** secret contains the bucket credentials, and the **backup** configuration map contains the bucket information.

1. Install **s3cmd**.

   s3cmd is a command-line tool and client for uploading, retrieving, and managing data in Amazon S3 and other cloud storage service providers.

   Install **s3cmd** from Amazon S3 Tools.

2. Retrieve the bucket name and bucket host from the **backup** configuration map.

```
oc extract --to=- cm/backup
```

   Example output:

```
# BUCKET_HOST
s3.openshift-storage.svc
# BUCKET_NAME
backup-7d9...f4c
# BUCKET_PORT
443
# BUCKET_REGION
# BUCKET_SUBREGION
```

3. Encode the certificate in Base64 format, and make a note of the returned value for use in step 3.1.

```
oc get cm/openshift-service-ca.crt -o jsonpath='{.data.service-ca\.crt}' | base64 -w0; echo
```

4. Retrieve the bucket credentials from the **backup** secret.

```
oc extract --to=- secret/backup
```

   Example output:

```
# AWS_ACCESS_KEY_ID
xxxxxKey_IDzzzzz
# AWS_SECRET_ACCESS_KEY
xxxxxxSecretxxxxx
```

5. Identify the public URL for the S3 endpoint from the s3 route in the **openshift-storage** namespace.

```
oc get route s3 -n openshift-storage
```

   Example output:

```
NAME   HOST/PORT                                                    PATH
SERVICES   PORT       TERMINATION       WILDCARD
s3     [s3-openshift-storage.apps.gocpprd1.emn.gdps.gov.sa] (https://s3-openshift-
storage.apps.gocpprd1.emn.gdps.gov.sa/)   s3         s3-https    reencrypt/Allow    None
```

6. Create a file called **.s3cfg** in your home directory with the S3 credentials and S3 endpoint URL.

```
access_key = <AWS_ACCESS_KEY_ID>
secret_key = <AWS_SECRET_ACCESS_KEY>
host_base = s3-openshift-storage.apps.gocpprd1.emn.gdps.gov.sa
host_base = <HOST_BASE>
host_bucket = <HOST_BASE>/%(bucket)s
signature_v2 = True
```

   Where

   - **<AWS_ACCESS_KEY_ID>** and **<AWS_SECRET_ACCESS_KEY>** are the values returned in step 2.4
   - **<HOST_BASE>** is the value returned in step 2.5. For example, **s3-openshift-storage.apps.gocpprd1.emn.gdps.gov.sa**

7. Validate the configuration

   Run the following command to list the contents of the bucket, and verify that the command returns no output because the bucket is empty. If the configuration is incorrect, then the command returns an error message.

   ```
   s3cmd la
   ```

3. Prepare the DataProtectionApplication configuration

1. Create a YAML file called **dpa-backup.yml**. Use the S3 bucket information from the previous steps to configure the default backup storage location.

   Example configuration:

   ```
   apiVersion: [oadp.openshift.io/v1alpha1](https://oadp.openshift.io/v1alpha1)
   kind: DataProtectionApplication
   metadata:
     name: oadp-backup
     namespace: openshift-adp
   spec:
     features:
       dataMover:
     configuration:
       velero:
         defaultPlugins:
           - aws
           - openshift
     backupLocations:
       - velero:
           config:
             profile: "default"
             region: "us-east-1"
             s3Url: https://s3.openshift-storage.svc/
             s3ForcePathStyle: "true"
             insecureSkipTLSVerify: "true"
           provider: aws
           default: true
           credential:
             key: cloud
             name:  cloud-credentials
           objectStorage:
             bucket: backup-6bce9ba8-59e1-4b2c-9c17-a88498954a56
             prefix: oadp
             caCert: <LS0tLS1.........>
   ```

2. Create the **cloud-credentials** secret in the **openshift-adp namespace** with the **backup** ObjectBucketClaim credentials.

   Create a file called **cloud-credentials** with the following content:

   ```
   [default]
   aws_access_key_id=<AWS_ACCESS_KEY_ID>
   aws_secret_access_key=<AWS_SECRET_ACCESS_KEY>
   ```

   Where **<AWS_ACCESS_KEY_ID>** and **<AWS_SECRET_ACCESS_KEY>** are the values retrieved in step 2.4.

   ```
   oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=cloud-credentials
   ```

3. Create the **restic-enc-key** secret in the **openshift-adp** namespace, with an encryption key to use for application data.

   Run the following command, which uses openssl to generate a random password to use as the encryption key.

   ```
   oc create secret generic restic-enc-key -n openshift-adp --from-literal=RESTIC_PASSWORD=$(openssl rand -base64 24)
   ```

4. Run the following command to apply the OADP configuration.

   ```
   oc apply -f dpa-backup.yml
   ```

4 Verify that the backupStorageLocation object is created

Run the following command to verify that the backupStorageLocation object is created, and has a phase of **Available**.

```
oc get backupStorageLocation
```

Example output:

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
oadp-backup    Available   1s               3d16h   true
```

5. Configure the volume snapshot classes for OADP

1. Run the following command to list all the available volume snapshot classes.

   ```
   oc get volumesnapshotclass
   ```

2. Set the **deletionPolicy** attribute to *Retain* for each volume snapshot class.

   Run the following command:

   ```
   for class in \
   $(oc get volumesnapshotclass -oname); do
   oc patch $class --type=merge -p '{"deletionPolicy": "Retain"}'
   done
   ```

3. Set the **velero.io/csi-volumesnapshot-class** label to true for each volume snapshot class.

   Run the following command:

   ```
   oc label volumesnapshotclass velero.io/csi-volumesnapshot-class="true" --all
   ```

6. (Optional) Back up a project to validate the OADP configuration

Optionally test your configuration by backing up the resources in a test namespace. This example uses a test namespace called **production**.

1. Create a file called up **backup.yml** with the following content.

   ```
   apiVersion: velero.io/v1
   kind: Backup
   metadata:
     name: backup-production
     namespace: openshift-adp
   spec:
     includedNamespaces:
     - production
   ```

2. Run the following command to apply the backup configuration.

   ```
   oc apply -f backup.yml
   ```

3. Wait a couple of minutes and check the backup completion status. Verify that the backup object has a phase of *Completed*.

   ```
   oc describe backup backup-production
   ```

   Example output:

   ```
   Status:
     Completion Timestamp: 2024-05-22T13:52:25Z
     Expiration:           2024-06-21T13:51:33Z
     Format Version:       1.1.0
     Phase:                Completed
     Progress:
       Items Backed Up:  55
       Total Items:      55
   ```

4. Run the following command to review the contents of the S3 storage.

   ```
   s3cmd la -r
   ```

   Example output:

   ```
   $ s3cmd la -r
   2024-05-22 13:52   2597  s3://backup-6bce9ba8-59e1-4b2c-9c17-
   a88498954a56/docker/registry/v2/blobs/sha256/05/057c885bbca68570905d4a9e26948b6fa07e06ecc80951aa0def58cd227a2513/dat
   a
   2024-05-22 13:51   4512  s3://backup-6bce9ba8-59e1-4b2c-9c17-
   a88498954a56/docker/registry/v2/blobs/sha256/07/07dc0ebf5f604feb9222bc7187a18dff7cb8d48b56cdacf0d1e564f20301b292/dat
   a
   2024-05-22 13:52   455  s3://backup-6bce9ba8-59e1-4b2c-9c17-
   a88498954a56/docker/registry/v2/blobs/sha256/13/1367c2cdbbe1a194f979ff4298091d41ab445934892cb325f8cc715812bc9f2d/dat
   a
   ```

Backing up IBM Cloud Pak for AIOps data

Learn how to back up data for IBM Cloud Pak for AIOps components.

The backup process uses Red Hat OpenShift APIs for Data Protection (OADP) for cluster object backups and volume backups. The backup script must be the same version as your deployment.

Note: If you also need to back up data for Infrastructure Automation components, the process and tools to use are the same. The only change that is required is when you are packaging the Helm chart. During this step you need to configure the value of the **enabledComponents** parameter to include **'"IA"'** as well as **'"AIOPS"'**. All other steps remain the same.

Prerequisites

- Ensure that you are logged in to your Red Hat OpenShift cluster with **oc login** for any steps that use the Red Hat OpenShift command-line interface (CLI).

- If they are not already installed, install the backup and restore tools. You need to install the Red Hat® OpenShift® Container Platform APIs for Data Protection (OADP) in the Red Hat OpenShift Container Platform cluster. For more information, see Installing the backup and restore tools.
- Provision S3 compliant object storage for your backups. You must have sufficient storage to backup the data that is contained in the datastores and persistent volume claims (PVCs) of the components that are backed up, as detailed in Backup process. The storage must also be accessible by any environment that a backup might need to be restored to. The backup process moves backup files to an S3 bucket, which must be in an S3 compatible location. For instance, you can use the following public or private cloud options for provisioning storage for your backups:
  - MinIO
  - Red Hat OpenShift Data Foundation object store
  - IBM Cloud Object Storage
  - AWS Cloud

  Important: Ensure that OADP is configured to point to the same object storage (S3 bucket) that includes the backup that you plan to use.

Note: As your data grows, the capacity of your backup storage might need to grow, and you must ensure that sufficient storage for backups is provisioned to accommodate any growth in backup data.

Backup procedure

Follow these steps to backup IBM Cloud Pak for AIOps.

1. Prepare to backup IBM Cloud Pak for AIOps
2. Backup IBM Cloud Pak for AIOps
3. Setup on-demand backups of IBM Cloud Pak for AIOps

1. Prepare to backup IBM Cloud Pak for AIOps

1. Clone the IBM Cloud Pak for AIOps samples GitHub repository to retrieve the latest IBM Cloud Pak for AIOps backup and restore scripts by running the following command:

   ```
   git clone https://github.com/IBM/cp4waiops-samples.git
   ```

   If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps, then copy the cloned scripts to your air-gapped environment.

2. Export the environment variables that you will need for the backup procedure.

   If you are backing up an online deployment, set the following:

   ```
   export OADP_NAMESPACE=<oadpNamespace>
   export PATH=<path>
   export AIOPS_NAMESPACE=<aiops_namespace>
   ```

   If you are backing up an offline deployment, set the following:

   ```
   export TARGET_REGISTRY_HOST=<target_registry_host>
   export TARGET_REGISTRY_PORT=<port>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export EMAIL=<email>
   export PATH=<path>
   export OADP_NAMESPACE=<oadpNamespace>
   export AIOPS_NAMESPACE=<aiops_namespace>
   ```

   Where:

   - `<target_registry_host>` is the IP address or FQDN of the target registry that holds the backup and restore images, from Offline deployments only: Mirror the backup and restore images
   - `<port>` is the port_number of the target registry
   - `<username>` is the username for the target registry
   - `<password>` is the password for the target registry
   - `<email>` is the email for the target registry
   - `<path>` is the path to where you downloaded and extracted the IBM Cloud Pak for AIOps backup and restore files.
   - `<oadpNamespace>` is the OADP namespace
   - `<aiops_namespace>` is the namespace where IBM Cloud Pak for AIOps is installed.

3. Package the Helm Chart.

   1. Change to the directory where you have the backup and recovery files downloaded.

      ```
      cd ${PATH}/bcdr/4.8.0/backup
      ```

   2. Set environment variables

      1. Run the following command to find the name of your IBM Cloud Pak for AIOps installation.

         ```
         oc get installation -n ${AIOPS_NAMESPACE}
         ```

         Example output, where `ibm-cp-aiops` is the name of the IBM Cloud Pak for AIOps installation:

         ```
         NAME          PHASE    LICENSE    STORAGECLASS  STORAGECLASSLARGEBLOCK  AGE
         ibm-cp-aiops  Running  Accepted   rook-cephfs   rook-ceph-rbd           34h
         ```

2. Run the following commands to find the name of your Elasticsearch backup PVC:

```
elasticsearch_cluster=$(oc get elasticsearchclusters.elasticsearch.opencontent.ibm.com -n
${AIOPS_NAMESPACE} -o jsonpath='{.items[0].metadata.name}')
esbackupPVC="$elasticsearch_cluster-ibm-elasticsearch-es-server-snap"
```

3. Update the following parameters in the **values.yaml** file. The file is located in the **./helm** directory.

   - **repository** - Do not edit this parameter.
   - **pullPolicy** - The policy for determining when to pull the image from the image registry server. For example, to force pull the image, use the **Always** policy.
   - **schedule** - (optional when on-demand backups are activated) The Cron expression for the automated backup. For example, to take a backup once a day, use the **0 0 * * *** Cron expression.
   - **backupStorageLocation** - The storage location where backed up data is stored. For example, **bcdr-s3-location**.

     Use the **oc get backupstoragelocation -n ${OADP_NAMESPACE}** command to get the **backupStorageLocation** on the OpenShift cluster.
   - **backupNameSuffix** - The prefix for the backup name when the backup is created by using the job. Generally, it can be name of source cluster itself. For example, **aiops-cluster-backup-106**.
   - **aiopsNamespace** - The namespace where IBM Cloud Pak for AIOps is installed.
   - **csNamespace** - The namespace where IBM Cloud Pak foundational services is installed. From IBM Cloud Pak for AIOps v4.4.0 this is the same as the namespace where IBM Cloud Pak for AIOps is installed.
   - **oadpNamespace** - The namespace where OADP is installed.
   - **redisBackupPod** - set this to *backup-data-<installation_name>-redis-server* where <installation_name> is the name that you found in the previous step.
   - **redisPVC** - set this to *data-<installation_name>-redis-server-* where <installation_name> is the name that you found in the previous step.
   - **redisSecret** - set this to *<installation_name>-redis-secret* where <installation_name> is the name that you found in the previous step.
   - **esBackupPVC**- set this to the value of **$esbackupPV** that you found in the previous step.
   - **ttl** - The time-to-live setting for the backup. The backed up data is retained until the **ttl** value is reached (expires). For example **720h0m0s**.
   - **enabledNamespaces** The namespaces of the components that need to be backed up. You can delete any unused namespaces from the list to reduce the time that is required for the backup process. If you have installed only IBM Cloud Pak for AIOps, then you require the namespaces that IBM Cloud Pak for AIOps and IBM Cloud Pak foundational services are installed in. From IBM Cloud Pak for AIOps v4.4.0, the IBM Cloud Pak foundational services namespace is the same as the IBM Cloud Pak for AIOps namespace, but this namespace must be listed twice, as in the following example.

     ```
     enabledNamespaces:
     - '"cp4aiops"'
     - '"cp4aiops"'
     ```
   - **enabledComponents** - Required. The list of components to back up.

     The backup and restore processes support backing up and restoring both the IBM Cloud Pak for AIOps and Infrastructure Automation components. These components can be installed together or independtly. You can specify to back up both, or only one of these components. Any other value is ignored and a corresponding error message is generated.

     To specify Infrastructure Automation, set the value **'"IA"'**.

     To specify IBM Cloud Pak for AIOps, set the value **'"AIOPS"'**.

     The following configuration shows both components listed:

     ```
     enabledComponents:
     - '"IA"'
     - '"AIOPS"'
     ```

4. Package the Helm Chart.

   ```
   helm package ./helm
   ```

4. Create a network policy.

   For more information about creating a **NetworkPolicy**, see Creating a network policy.

   1. Create a network policy file called **policy-bcdr.yaml** with the following contents:

      ```
      apiVersion: networking.k8s.io/v1
      kind: NetworkPolicy
      metadata:
       name: bcdr-np
       namespace: <aiopsNamespace>
      spec:
       podSelector:
        matchLabels:
          ibm-es-server: aiops-ibm-elasticsearch-es-server
       ingress:
      ```

```
  - from:
    - namespaceSelector:
        matchLabels:
          kubernetes.io/metadata.name: <oadpNamespace>
  policyTypes:
  - Ingress
```

Where:

- **<aiopsNamespace>** is the value of $AIOPS_NAMESPACE
- **<oadpNamespace>** is the value of $OADP_NAMESPACE

2. Apply the network policy to your cluster to allow the required ingress traffic.

```
oc apply -f policy-bcdr.yaml -n ${AIOPS_NAMESPACE}
```

2. Setup an automated backup of IBM Cloud Pak for AIOps

Run the following steps to trigger an automated backup.

1. Change to the **backup** directory where the script is located:

   ```
   cd ${PATH}/bcdr/4.8.0/backup
   ```

2. Deploy the backup job by running the following command:

   ```
   helm install backup-job clusterbackup-0.1.0.tgz
   ```

3. Optional - update the **schedule** parameter in the **values.yaml** file (located in the **./helm** directory). This uses a cron expression to schedule the automated backup. For example, to take a backup once a day, use the **0 0 * * *** Cron expression.

3. Setup on-demand backups of IBM Cloud Pak for AIOps

This step is optional. Use only when you do not want to wait untill the next scheduled backup job.

Prerequisite: The deployment of an automated backup job is a prerequisite for the on-demand job. Only after you initiate an automated backup job can you then trigger an on-demand backup.

1. Deploy the on-demand backup job by running the following command:

   ```
   oc create job --from=cronjob/backup-job on-demand-backup-job -n ${OADP_NAMESPACE}
   ```

2. Check the on-demand backup pods status by running the following command:

   ```
   oc get pods -n ${OADP_NAMESPACE}
   ```

3. Check the on-demand backup job logs by running the following command:

   ```
   oc logs -f <on-demand-backup-job-***> -n ${OADP_NAMESPACE}
   ```

4. Find and export the name of your backup.

   You can see the backup name after the on-demand backup job is complete. For example, you might see the backup name **aiops-cluster-backup-106-1622193915** in the on-demand backup job log as follows:

   ```
   Waiting for backup aiops-cluster-backup-106-1622193915 to complete
   ```

   Export the name of your backup by running the following command:

   ```
   export BACKUP_NAME=<backup_name>
   ```

   Where **<backup_name>** is the name of the backup.

5. Check the backup status by running the following command:

   ```
   velero get backup ${BACKUP_NAME} -n ${OADP_NAMESPACE}
   ```

   When the backup is finished, the status is **Completed**, such as in the following sample output:

   ```
   aiops-backup-1654171690-testfest34   Completed   2022-06-02 05:08:11 -0700 PDT   29d   bcdr-data-protection-app-1
   cp4aiops.ibm.com/backup=t
   ```

   To see more details, you can also:

   - Check the backup pods' status by running the following command:

     ```
     oc get pods -n ${OADP_NAMESPACE}
     ```

   - Check the backup job logs by running the following command:

     ```
     oc logs -f <backup-job-***>
     ```

6. Review the backup result summary that is included within the logs of the backup job.

   Note: A **backup-result-config** ConfigMap is created to include this result. To view this ConfigMap, run the following command:

```
oc get cm backup-result-config -n ${AIOPS_NAMESPACE} -o yaml
```

Troubleshooting

- Backup process is stuck in an *In progress* state
- Classifier and layout pods remain not ready after running a backup
- Helm install backup job command failed

Backup process is stuck in an *In progress* state

If your backup process remain stuck in an *In progress* state for an unexpected duration, complete the following steps. This procedure stops the backup process so that you can try the backup again.

1. Delete the Velero pod by running the following command:

   ```
   oc delete pod <pod> -n ${OADP_NAMESPACE}
   ```

2. Delete the backup that is stuck in the *In progress* state:

   ```
   velero delete backup <backup> -n ${OADP_NAMESPACE}
   ```

   Where **<backup>** is the backup that you want to delete. The backup process should begin again.

3. Wait for the process to complete and verify that the backup is created.

Classifier and layout pods remain not ready after running a backup

After you complete a backup, you might notice that the classifier (**aiops-ir-analytics-classifier**) and layout (**aiops-topology-layout**) pods do not change to a ready state, which can cause **AIOpsAnalyticsOrchestrator** to be stuck in an updating phase. This issue can occur after most backups. To resolve this issue, you need to either restart the pods or increase the resource limits for the pods.

- For the classifier (**aiops-ir-analytics-classifier**) pod, restart the pod.
- For the layout (**aiops-topology-layout**) pod, restart the pod. If the restart does not result in the pod becoming ready, increase the resource limits for the pod. Doubling the **cpu** and **memory** limits and request allocations for the pods can result in the pod starting and entering a ready state.

Helm install backup job command failed

When you are running the **helm install backup-job clusterbackup-0.1.0.tgz** command, you might encounter the command failing with an error that is similar to the following error:

```
Error: admission webhook "trust.hooks.securityenforcement.admission.cloud.ibm.com" denied the request:
Deny "icr.io/cpopen/cp4waiops/cp4aiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355", no
matching repositories in ClusterImagePolicy and no ImagePolicies in the "velero" namespace
```

If you encounter this error, complete the following steps to resolve the issue:

1. Uninstall the **backup-job** job by running the following command:

   ```
   helm uninstall backup-job -n ${OADP_NAMESPACE}
   ```

2. Export an environment variable for the image.

   For an online deployment:

   ```
   export REGISTRY=icr.io/cpopen/cp4waiops/cp4aiops-bcdr
   export BCDR_IMAGE=${REGISTRY}/<bcdr_image>
   ```

   For an offline deployment:

   ```
   export REGISTRY=$TARGET_REGISTRY
   export BCDR_IMAGE=${REGISTRY}/<bcdr_image>
   ```

   Where **<bcdr_image>** is the name of the backup and restore image, as given in the backup helm chart **values.yaml** file, in the form **cp4waiops-bcdr@{digest}**. An example value for BCDR_IMAGE is **icr.io/cpopen/cp4waiops/cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355**.

3. Create a **backup-image-policy.yaml** file and add the following content within the file:

   ```
   apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
   kind: ClusterImagePolicy
   metadata:
     name: backup-image-policy
   spec:
    repositories:
     - name: ${BCDR_IMAGE}
       policy:
   ```

4. Apply the policy by running the following command:

   ```
   oc apply -f backup-image-policy.yaml
   ```

5. Deploy the backup job by running the following command:

   ```
   helm install backup-job clusterbackup-0.1.0.tgz
   ```

Restoring IBM Cloud Pak for AIOps data

Learn how to restore data for IBM Cloud Pak for AIOps components to a cluster, such as for disaster recovery.

The following procedure restores all backed up data that exists in the specified backup for IBM Cloud Pak for AIOps components. The steps in the following procedure restore IBM Cloud Pak for AIOps in a new cluster.

Before you begin

- All required storage classes must be created prior to running the restore process. The storage classes must have the same name as the backup cluster.
- You can restore a backup only within an environment that has the same version of IBM Cloud Pak for AIOps as the environment where the backup was created. For example, a backup of an IBM Cloud Pak for AIOps 4.8.0 environment must be restored within a cluster that has IBM Cloud Pak for AIOps 4.8.0 installed. If you need to upgrade as well as restore data, complete the restore process before you upgrade.
- If you are also restoring Infrastructure Automation data, the overall procedure is the same with the following additional steps required:
  - After you install IBM Cloud Pak for AIOps on the cluster where you are restoring data, you need to install Infrastructure Automation.
  - After you restore your IBM Cloud Pak for AIOps data, you need to restore your Infrastructure Automation data. For more information, see Restoring Infrastructure Automation.

Restore procedure

Follow the steps to restore IBM Cloud Pak for AIOps from backup.

1. Set up your new cluster for backup and restore
2. Prepare the backup data for restoring
3. Restore the cluster namespaces and install IBM Cloud Pak for AIOps
4. Restore the IBM Cloud Pak for AIOps data
5. (Optional) Restore Infrastructure Automation data
6. Post-restore tasks

If you encounter any issues with the restore process, see Troubleshooting

1. Set up your new cluster for backup and restore

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.

   IBM Cloud Pak for AIOps requires OpenShift to be installed and running. You must have administrative access to your OpenShift cluster.

   Important: Ensure that the version of Red Hat OpenShift Container Platform that you install is the same as the version that was installed in the backed up cluster.

   For information on supported versions of OpenShift, see Supported Red Hat OpenShift Container Platform versions.

   Note: IBM Cloud Pak for AIOps uses the OpenShift image registry when it builds images in real time. If the OpenShift image registry is not persistent and the registry respawns, then workloads can temporarily fail until respawning is complete. A persistent OpenShift image registry is recommended to avoid this. For more information, see Setting up and configuring the registry in the Red Hat OpenShift Container Platform documentation.

2. Install the OpenShift command line interface (`oc`) on your cluster's boot node and run `oc login`, using the instructions in Getting started with the OpenShift CLI.

3. Configure storage

   The storage configuration must satisfy your sizing requirements. For more information on the storage classes that are needed for installing IBM Cloud Pak for AIOps, see Storage.

   Important: All required storage classes must be created prior to running the restore process. The storage classes must have the same name as the backup cluster.

4. Install the backup and restore tools

   Install the Red Hat OpenShift APIs for Data Protection (OADP) in the Red Hat OpenShift Container Platform cluster. For more information, see Installing the backup and restore tools.

   Important: Ensure that the OpenShift APIs for Data Protection (OADP) is configured to point to the same object storage (S3 bucket) that includes the backup that you plan to use.

5. Export the environment variables that you will need for the restore procedure.

   If you are restoring to an online deployment, set the following:

   ```
   export PATH=<path>
   export OADP_NAMESPACE=<oadpNamespace>
   export AIOPS_NAMESPACE=<aiops_namespace>
   ```

   If you are restoring to an offline deployment, set the following:

   ```
   export TARGET_REGISTRY_HOST=<target_registry_host>
   export TARGET_REGISTRY_PORT=<port>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   ```

```
export TARGET_REGISTRY_USER=<username>
export TARGET_REGISTRY_PASSWORD=<password>
export EMAIL=<email>
export PATH=<path>
export OADP_NAMESPACE=<oadpNamespace>
export AIOPS_NAMESPACE=<aiops_namespace>
```

Where:

- **<target_registry_host>** is the IP address or FQDN of the target registry that holds the backup and restore images, from Offline deployments only: Mirror the backup and restore images
- **<port>** is the port_number of the target registry
- **<username>** is the username for the target registry
- **<password>** is the password for the target registry
- **<email>** is the email for the target registry
- **<path>** is the path to where you downloaded and extracted the IBM Cloud Pak for AIOps backup and restore files.
- **<oadpNamespace>** is the OADP namespace
- **<aiops_namespace>** is the namespace for the IBM Cloud Pak for AIOps deployment.

2. Prepare the backup data for restoring

Verify your backed up data and prepare the data for restoring.

1. Check the backup status

   Check the backup status to ensure that the backup that you want to restore in your cluster is complete. Run the following command to check the contents of the backup:

   ```
   velero describe backup <backup-name> --details
   ```

   The output should list the backed-up data for IBM Cloud Pak for AIOps (**cp4aiops/***) and for IBM Cloud Pak foundational services. If you also backed up Infrastructure Automation data, this data (**cp4aiops/***) should also be listed.

   The output should resemble the following sample output:

   ```
   v1/PersistentVolumeClaim:
     - cp4aiops/back-aiops-topology-cassandra-0
     - cp4aiops/data-c-example-couchdbcluster-m-0
     - cp4aiops/export-aimanager-ibm-minio-0
     - cp4aiops/aiops-ibm-elasticsearch-es-server-snap
     - cp4aiops/postgres-backup-data
     - cp4aiops/metastore-backup-data
   v1/Pod:
     - cp4aiops/backup-back-aiops-topology-cassandra-0
     - cp4aiops/backup-data-c-example-couchdbcluster-m-0
     - cp4aiops/backup-export-aimanager-ibm-minio-0
     - cp4aiops/backup-metastore
     - cp4aiops/es-backup
     - cp4aiops/backup-metastore
     - cp4aiops/dummy-db
   v1/Secret:
     - cp4aiops/aimanager-ibm-minio-access-secret
     - cp4aiops/aiops-ir-core-model-secret
     - cp4aiops/icp-serviceid-apikey-secret

   Velero-Native Snapshots: <none included>

   Restic Backups:
    Completed:
     cp4aiops/backup-back-aiops-topology-cassandra-0: backup
     cp4aiops/backup-data-c-example-couchdbcluster-m-0: backup
     cp4aiops/backup-export-aimanager-ibm-minio-0: backup
     cp4aiops/backup-metastore: data
     cp4aiops/es-backup: elasticsearch-backups
     cp4aiops/backup-postgres: backup
   ```

2. Package and install the Helm Chart

   1. Change to the **restore** directory where you need to package the Helm Chart:

      ```
      cd ${PATH}/bcdr/4.8.0/restore
      ```

   2. Update the following parameters in the **values.yaml** file. The file is located in the **./helm** directory:

      - **backupName** - The name of the backup that you are restoring.
      - **aiopsNamespace** - The namespace where IBM Cloud Pak for AIOps is installed.
      - **csNamespace** - The namespace where IBM Cloud Pak foundational services is installed. In IBM Cloud Pak for AIOps v4.8.0 this is the same as the namespace where IBM Cloud Pak for AIOps is installed.
      - **oadpNamespace** - The namespace where OADP is installed.

   3. Package the Helm Chart.

      ```
      helm package ./helm
      ```

   4. Install the Helm Chart for restoring data by running the following job:

      ```
      helm install restore-job clusterrestore-0.1.0.tgz
      ```

3. Restore the cluster namespaces and install IBM Cloud Pak for AIOps

Since the restore job does not install IBM Cloud Pak for AIOps, you need to first install IBM Cloud Pak for AIOps before you can run the restore jobs for restoring database and component data. A restore job for restoring the cluster namespaces is available and must be run before you install IBM Cloud Pak for AIOps.

1. Restore the cluster namespaces

   You need to restore the projects (namespaces) of the backed up cluster so that your new cluster includes the metadata with the SELinux settings that need to match the settings for the backup data that you plan to restore.

   These steps only restore the namespaces and namespace metadata. The commands do not restore the contents of the namespaces.

   Note: In IBM Cloud Pak for AIOps v4.8.0, the IBM Cloud Pak foundational services namespace is the same as the IBM Cloud Pak for AIOps namespace.

   1. Change to the **restore** directory where the restore script is located:

      ```
      cd ${PATH}/bcdr/4.8.0/restore
      ```

   2. Optional. Delete any existing namespace restore jobs:

      ```
      oc delete -f ns-restore-job.yaml
      ```

   3. Create a job to restore the cluster namespaces:

      ```
      oc create -f ns-restore-job.yaml
      ```

   4. Check the restore job logs by running the following command:

      ```
      oc logs -f <ns-restore-job-***>
      ```

   5. Check the velero-restore status for the namespace by running the following command:

      ```
      velero get restore <RESTORE_NAME>
      ```

      Where **<RESTORE_NAME>** is the name of the restore for namespace. You can see the restore name after the restore job is completed. For example, you might see the restore name **aiops-namespace-restore-20221006054710** within the restore job log as follows:

      ```
      Restore request "aiops-namespace-restore-20221006054710" submitted successfully.
      ```

      Ensure that the projects (namespaces) are restored before you proceed.

2. Create a network policy.

   For more information about creating a **NetworkPolicy**, see Creating a network policy.

   1. Create a network policy file called **policy-bcdr.yaml** with the following contents:

      ```
      apiVersion: networking.k8s.io/v1
      kind: NetworkPolicy
      metadata:
       name: bcdr-np
       namespace: <aiopsNamespace>
      spec:
       podSelector:
        matchLabels:
          ibm-es-server: aiops-ibm-elasticsearch-es-server
       ingress:
       - from:
         - namespaceSelector:
             matchLabels:
               kubernetes.io/metadata.name: <oadpNamespace>
       policyTypes:
       - Ingress
      ```

      Where:

      - **<aiopsNamespace>** is the value of $AIOPS_NAMESPACE
      - **<oadpNamespace>** is the value of $OADP_NAMESPACE

   2. Apply the network policy to your cluster to allow the required ingress traffic.

      ```
      oc apply -f policy-bcdr.yaml -n ${AIOPS_NAMESPACE}
      ```

3. Install IBM Cloud Pak for AIOps

   For more information, see Installing IBM Cloud Pak for AIOps.

   Important:

   - Ensure that the version of IBM Cloud Pak for AIOps that you are installing is the same as the version that was installed in the backed up cluster.
   - The backup includes keys and certificates from the backed up cluster. Ensure that your new cluster is configured to support the use of these keys and certificates so that the restored data can be accessed.
   - Wait until the installation is complete and all pods in the IBM Cloud Pak for AIOps project (namespace) are running before you proceed.

4. Optional. If you also need to restore Infrastructure Automation data, you need to install the Infrastructure Automation operators and create the required custom resources.

   - For more information about installing Infrastructure Automation, see Installing Infrastructure Automation.

- For more information about restoring Infrastructure Automation, see Restoring Infrastructure Automation.

4. Restore the IBM Cloud Pak for AIOps data

This procedure restore the data for all backed up IBM Cloud Pak for AIOps databases and components. Optionally, you can choose to restore only individual components. For details, see Restoring individual components.

1. Change to the **restore** directory where the restore script is located:

   ```
   cd ${PATH}/bcdr/4.8.0/restore
   ```

2. Optional. Delete any existing IBM Cloud Pak for AIOps restore jobs:

   ```
   oc delete -f aiops-restore-job.yaml
   ```

3. Create a job to restore IBM Cloud Pak for AIOps:

   ```
   oc create -f aiops-restore-job.yaml
   ```

4. Check the restore job logs by running the following command:

   ```
   oc logs -f <aiops-restore-job-***>
   ```

5. Check the velero-restore status by running the following command:

   ```
   velero get restore <RESTORE_NAME>
   ```

   Where **<RESTORE_NAME>** is the name of the restore for IBM Cloud Pak for AIOps. You can see the this restore name after the restore job is completed. For example, you might see the restore name **cassandra-restore-20221006054710** for the Cassandra restore in the restore job log as follows:

   ```
   Restore request "cassandra-restore-20221006054710" submitted successfully.
   ```

   Similarly, the OADP restore name for other IBM Cloud Pak for AIOps components can display in the restore job log.

   Note: You might notice that the **aiops-configmaps** resource group shows as failed under the **Restore sequence** section of the IBM Cloud Pak for AIOps restore recipe but the overall restore is successful and works as intended.

5. (Optional) Restore the Infrastructure Automation data

If you are also restoring Infrastructure Automation data, run the commands to restore the Infrastructure Automation data. For more information, see Restoring Infrastructure Automation.

6. Post-restore tasks

1. Update your integrations.

   If you have an integration that you are restoring, the status for these integrations can be in error after the restore process completes. To resolve this status, you need to edit and save your integrations with the Integrations feature in the IBM Cloud Pak for AIOps console. Editing the integration regenerates the associated Flink job for the integration, which updates the status. For more information about editing these integrations, see Defining integrations. Sometimes an integration can fail to gather data but not show an error status. To resolve this, edit and save the integration.

2. If you are restoring data for any integrations that are remotely deployed, complete the following steps to redeploy the integrations:

   1. On the remote cluster, delete any existing integrations.
   2. For each remotely deployed integration that you need to redeploy, Copy or download a new bootstrap command for the deployment script from cluster where the restore process ran.

      You can click the **Copy to clipboard** button to copy the command, or click **Download as a sh-file** to download the command as a **bootstrap.sh** file.

      Note: If you do not obtain the command now, you can copy or download the command later by completing the following steps:

      1. Log in to IBM Cloud Pak for AIOps console.
      2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
      3. For each integration, click the integration type on the *Manage integrations* tab of the Integrations page.
      4. On the page for the integration type, click the **Download** link in the **Remote deployment script** column for the integrations.
      5. Either copy or download the bootstrap command for the deployment script.

   3. Run the remote deployment scripts on the remote cluster to redeploy the integrations.

      1. Get the OpenShift CLI **oc login** command from the remote cluster.
      2. From a command line, log in to the remote cluster with the **oc login** command.
      3. Switch to the target project (namespace).
      4. Run the deployment script as a **sh** (script file), such as the downloadable **bootstap.sh** file.

3. If you are resoring secure tunnels data, install the secure tunnel connections for any restored secure tunnel data.

1. Optional. If the tunnel exists on your cluster, uninstall the existing tunnel connection.
2. Install the Secure Tunnel connector on your restored cluster.
3. Create the secure tunnel connection on your restored cluster.

4. Enable authentication with the restored JWT certificate.
   1. Retrieve the new **initial_admin_password** from the **admin-user-details** secret.

      ```
      export PROJECT=<project>
      PASS=$(oc get secret admin-user-details -n ${PROJECT} -o jsonpath='{.data.initial_admin_password}' | base64 -d)
      ```

   2. Disable the **admin** user.

      ```
      oc rsh -n ${PROJECT} $(oc get pod -l component=usermgmt -n ${PROJECT} | tail -1 | cut -f1 -d\ )
      /usr/src/server-src/scripts/manage-user.sh --disable-user admin
      ```

   3. Enable the **admin** user with the new password.

      ```
      echo "$PASS" | oc rsh -n ${PROJECT} $(oc get pod -l component=usermgmt -n ${PROJECT} | tail -1 | cut -f1 -d\ )
      /usr/src/server-src/scripts/manage-user.sh --enable-user admin
      ```

   4. Delete the secret **app-api-user-jwt**. It will be recreated automatically after a few seconds.

      ```
      oc delete secret app-api-user-jwt -n ${PROJECT}
      ```

Restoring individual components

If needed, you can choose to restore data for specific individual databases and components instead of for all databases and components at once.

1. Change to the **restore** directory where the restore script is located:

   ```
   cd ${PATH}/bcdr/4.8.0/restore
   ```

2. Copy the **aiops-restore-job.yaml** file. Rename your new file **<component>-restore-job.yaml**, where **<component>** is the name of the component that you are restoring.

   For example, if you are restoring Cassandra, rename the file to be **cassandra-restore-job.yaml**.

3. Open the new **<component>-restore-job.yaml** file for editing. Update the **name** and **command** sections to match the values for the component that you are restoring:

   - Update the **name** of the restore job in the metadata section to be the individual component job, such as **cassandra-restore-job**.
   - Update the command section **command: ["/bin/bash", "restore.sh","-aiops"]** to replace **-aiops** with the respective argument for the component that you are restoring. For the list of component arguments, see the table that follows this procedure.

4. Create a job to restore the individual component.

   ```
   oc create -f <component>-restore-job.yaml
   ```

5. Check the restore job logs by running the following command:

   ```
   oc logs -f <cp4waiops-component-restore-job-***>
   ```

6. Check the velero-restore status by running the following command:

   ```
   velero get restore <RESTORE_NAME>
   ```

   Where **<RESTORE_NAME>** is one of the restore names for the component. You can view the names for the component in the restore job log when the restore job is completed. For example, the restore name for a Cassandra restore can be **cassandra-restore-20221006054710**, which can display in an entry similar to the following example log entry:

   ```
   Restore request "cassandra-restore-20221006054710" submitted successfully.
   ```

Component or database arguments for restore job command configuration

Table. Component or database arguments for restore job commands

Component or Database	Argument
Cassandra	**-cassandra**
CouchDB	**-couchdb**
Elasticsearch	**-es**
Metastore	**-metastore**
Minio	**-minio**
Postgres	**-postgres**
IBM Cloud Pak foundational services	**-cs**
Intergration CR	**-connectioncr**
Secure Tunnel CR	**-tunnelcr**

Troubleshooting

- Stale alert data exists after restoration
- Restore process for ElasticSearch failed
- Restore fails with CouchDB pod in CrashLoopBackOff
- LDAP user login is not working after a restore
- Data is not being processed after restoration
- Applications still show non-existent active incidents and alerts notations
- Topology observer jobs do not run after restore
- Restore process terminated mid-process with partial data available
- OADP restore is stuck in an In progress state
- Helm install restore job command failed
- Cassandra restore fails with `org.apache.cassandra.io.FSReadError` exception
- Integrations missing after a restore
- Netcool integration is missing details after restore
- Restore fails with lifecycletrigger: Not Ready
- Cannot login to the Cloud Pak for AIOps console
- Policy configuration is not retained on the restored deployment
- Troubleshooting the Infrastructure Automation restore

Stale alert data exists for topology resources after restoration

After you complete a restore, you might notice notice that stale alert data exists for topology resources. For instance, if you search for a resource that had alert data, you might see outdated alert data. To clear the historical alert data, you can run the **`statusClear`** crawler.

You can run the crawler to clear stale events by using the topology service Swagger pages:

1. Open the Swagger API for the topology service. For more information, see Accessing Topology service Swagger UI.
2. Go to the Crawlers section.
3. Open the statusClear crawler.
4. Run the crawler with the default message body.

This crawler runs asynchronously over the topology data. The crawler POST response returns an EntityId header, which you can use to check the progress of the crawler. To check the progress include the EntityId header in a **`GET /mgmt_artifacts/{id}`** call. The response shows the status of the asynchronous crawl.

For more information about the topology service Swagger, see Application and topology APIs.

Restore process for ElasticSearch failed

If the restore of a particular data store, such as ElasticSearch, or custom resource fails, complete the following steps before you can attempt the restore again.

1. Optional. If this cluster was previously configured to enable a backup of ElasticSearch, or you need to rerun restore, complete the following steps:

    1. Remove the backup path and Snapshot location configurations from the AutomationBase CR.

    2. Delete the ElasticSearch backup snapshot PVC by running the following command. This deletion is needed since this PCV is replaced with the ElasticSearch backup data and the **`es-restore1`**.

        ```
        elasticsearch_cluster=$(kubectl get elasticsearchclusters.elasticsearch.opencontent.ibm.com -n
        ${AIOPS_NAMESPACE} -o jsonpath='{.items[0].metadata.name}')
        esbackupPVC="$elasticsearch_cluster-ibm-elasticsearch-es-server-snap"

        oc delete pvc -n ${AIOPS_NAMESPACE} $esbackupPVC
        ```

    3. Run the following command to delete the snapshot and **`es-restore1`** if it exists:

        ```
        oc delete restore es-restore1 -n $OADP_NAMESPACE
        ```

2. Repeat the process to restore the data store or custom resource to reattempt the failed restore.

Restore fails with CouchDB pod in CrashLoopBackOff

This problem can occur if you restore an IBM Cloud Pak for AIOps backup to a different cluster or namespace and the database files in the persistent storage are not readable or writeable by the root user group.

When this issue occurs, the pod **`c-example-couchdbcluster-m`** has a status of *CrashLoopBackOff*, as in the following example.

```
oc get pods | grep couchdb
c-example-couchdbcluster-m-0                                      1/2     CrashLoopBackOff        533 (4m13s ago)   2d5h
c-example-couchdbcluster-m-0-debug                                1/2     Running                 0                 2m39s
```

The logs from the database container in the failing pods contain the message **`Could not open file /data/db/_nodes.couch: permission denied`**, as in the following example:

```
oc logs c-example-couchdbcluster-m-0 -c db --tail=100
[error] 2022-11-10T10:39:34.788855Z couchdb@c-example-couchdbcluster-m-0.c-example-couchdbcluster-m <0.322.0> --------
Could not open file /data/db/_nodes.couch: permission denied
```

To resolve this issue, grant the root group write permissions to the folders in the persistent storage that contain the database files.

1. Get the name of the persistent volume that contains the CouchDB database files.

    ```
    oc get pv | grep couch
    ```

    Example output:

```
oc get pv | grep couch
pvc-d6599cfb-aa38-47be-857e-a2efa9ee0407   5Gi        RWO            Delete           Bound    katamari/data-c-
example-couchdbcluster-m-0                              rook-cephfs              2d5h
```

2. Find the worker node for the failing CouchDB pod.

```
oc get pod -o wide c-example-couchdbcluster-m-0
```

Example output:

```
oc get pod -o wide c-example-couchdbcluster-m-0
NAME                           READY   STATUS             RESTARTS         AGE    IP              NODE
NOMINATED NODE   READINESS GATES
c-example-couchdbcluster-m-0   1/2     CrashLoopBackOff   537 (76s ago)   2d5h   10.254.15.54   worker1.bcdr-test-
12345678.mysite.com   <none>           <none>
```

3. Debug the worker node that you identified in the previous step.

```
oc debug node/<worker_node>
```

Where **<worker_node>** is the value returned for *NODE* in the previous step.

Example output:

```
oc debug node/worker1.bcdr-test-12345678.mysite.com
Starting pod/worker1bcdr-rtp-26101516cpmyclustercom-debug ...
To use host binaries, run `chroot /host`
Pod IP: 10.22.1.1
If you don't see a command prompt, try pressing enter.
```

4. Find the mount point of the persistent volume.

```
mount | grep <pv>
```

Where **<pv>** is the persistent volume name returned in step 1.

Example output:

```
sh-4.4# mount | grep pvc-d6599cfb-aa38-47be-857e-a2efa9ee0407
172.30.2.127:6789,172.30.43.40:6789,172.30.11.17:6789:/volumes/csi/csi-vol-0bb3b8e3-5f26-11ed-97a6-
0a580afe2807/af21d8e8-f29e-4111-8fcd-c4366d604c99 on /host/var/lib/kubelet/plugins/kubernetes.io/csi/pv/pvc-
d6599cfb-aa38-47be-857e-a2efa9ee0407/globalmount type ceph (rw,relatime,seclabel,name=csi-cephfs-node,secret=
<hidden>,acl,mds_namespace=rook-cephfilesystem)
172.30.2.127:6789,172.30.43.40:6789,172.30.11.17:6789:/volumes/csi/csi-vol-0bb3b8e3-5f26-11ed-97a6-
0a580afe2807/af21d8e8-f29e-4111-8fcd-c4366d604c99 on /host/var/lib/kubelet/pods/26fc2b39-aaf0-4add-8649-
6cfd56fc4a3b/volumes/kubernetes.io~csi/pvc-d6599cfb-aa38-47be-857e-a2efa9ee0407/mount type ceph
(rw,relatime,seclabel,name=csi-cephfs-node,secret=<hidden>,acl,mds_namespace=rook-cephfilesystem)
```

The mount point is the directory ending with **mount**. In the above example output this is **/host/var/lib/kubelet/pods/26fc2b39-aaf0-4add-8649-6cfd56fc4a3b/volumes/kubernetes.io~csi/pvc-d6599cfb-aa38-47be-857e-a2efa9ee0407/mount**

5. Change the permissions on the mount directory by changing to the directory and running **chmod**.

```
cd <mount_dir>
chmod -R g+w
ls -lah db/
exit
```

Where **<mount_dir>** is the directory you identified in the previous step.

Example output:

```
sh-4.4# cd /host/var/lib/kubelet/pods/26fc2b39-aaf0-4add-8649-6cfd56fc4a3b/volumes/kubernetes.io~csi/pvc-d6599cfb-
aa38-47be-857e-a2efa9ee0407/mount
sh-4.4# ls
db  view_index
sh-4.4# ls -lah db/
total 37K
drwxr-xr-x. 4 1000650000 root    4 Nov  7 04:55 .
drwxrwxrwx. 5 root       root    3 Nov 10 10:52 ..
drwxr-xr-x. 2 1000650000 root    0 Nov  7 04:54 .delete
-rw-r--r--. 1 1000650000 root  29K Nov  7 05:31 _dbs.couch
-rw-r--r--. 1 1000650000 root 8.3K Nov  7 04:54 _nodes.couch
drwxr-xr-x. 4 1000650000 root    2 Nov  7 04:55 shards
sh-4.4# chmod -R g+w .
sh-4.4# ls -lah db/
total 37K
drwxrwxr-x. 4 1000650000 root    4 Nov  7 04:55 .
drwxrwxrwx. 5 root       root    3 Nov 10 10:52 ..
drwxrwxr-x. 2 1000650000 root    0 Nov  7 04:54 .delete
-rw-rw-r--. 1 1000650000 root  29K Nov  7 05:31 _dbs.couch
-rw-rw-r--. 1 1000650000 root 8.3K Nov  7 04:54 _nodes.couch
drwxrwxr-x. 4 1000650000 root    2 Nov  7 04:55 shards
sh-4.4# exit
```

6. Run the following command to verify that the CouchDB pod has a status of **Running**.

```
oc get pods | grep couchdb
```

Example output:

```
c-example-couchdbcluster-m-0                                  2/2     Running                 539 (87s ago)
2d5h
```

LDAP user login is not working after a restore

Follow the steps to solve the problem:

1. Log in to the console as the default admin user.
2. From the main navigation menu, click **Administer** > **Identify and access**.
3. Select the LDAP connection, and click **Edit connection**. Edit the LDAP connection with the correct information.
4. Click **Test connection**.
5. Click **Save** once the connection is success.
6. log in to the console with the LDAP user's credentials.

Data is not being processed after restoration

After you complete a restore, you might notice that data for some AI modeling features, such as log anomaly detection, change risk, similar tickets, might not be processed.

This can occur when expected Kafka topics for the AI modeling is not present after the restore completes. To resolve this problem, restart the **`aimanager-aio-controller`** pod. This restart can result in the creation of the expected Kafka topics.

Applications still show non-existent active incidents and alerts notations

After you complete a restore, you might notice notice that active incidents and alerts notations display for applications when the incidents and notations no longer exist. When the topology data store is backed up and restored, it contains restored data which is out of sync with the rest of the system due to the way the data is modelled against groups, applications, and resources within the topology.

If your data is outdated, you need to manually delete the topology association between the incidents and applications. Use the topology API to clear out any groups or events. Once removed, the changes are incorporated into the topology view.

To remove groups (incident groups of entity **`waiopsStory`**) use the API:

- **`GET /topology/groups?_type=waiopsStory`**
- **`DELETE /topology/groups?_type=waiopsStory`**

To remove events, remove the incorrect status from the topology with the following API:

- **`GET /topology/status`**
- **`DELETE /topology/status`**

Alternatively, you can use the *Resource management* tool in the UI to remove non-existant alert indiciations from a topology. When you are viewing a topology in the UI, open the **Settings** menu and select **Topology configuration**. From the**Data administration routines** page, run the **Status clear** routine to remove the alert indications.

Topology observer jobs do not run after restore

After running the restore procedure, some observer jobs are offline, have a **Status** of **`Error`**, or have a **Status** of **`Running`** even though job runs are being skipped.

For jobs with a job type of **`Repeating Schedule`**, this problem can occur when a restore ran from a backup that was taken while the observer job was waiting for data processing to finish. After the restore completes, the observer then starts with the job still in a FINISHING state. The observer waits for its job to finish, but the job cannot finish because the referenced data was not backed up, and so the job becomes stuck. Messages similar to the following might be seen in the observer log:

```
execution skipped because current job  had state: FINISHING
```

For other jobs that do not have a job type of **`Repeating Schedule`**, the observer log shows that it is looking for a vertex that no longer exists:

```
WARN   [2022-09-13 16:06:24,265] [pool-5-thread-1] c.i.i.t.o.t.ObserverVertex -  Failed to poll observer vertex
cp4waiops-cartridge.github-observer.  Response : InboundJaxrsResponse{context=ClientResponse{method=POST,
uri=https://aiops-topology-topology:8080/1.0/topology/mgmt_artifacts/MKSrcRpkRC6WmWVGGdSukA, status=404, reason=Not
Found}}
```

To resolve these problems, complete the following steps:

1. Locate jobs that are stuck in the **`FINISHING`** state.

   For jobs with a **Job type** of **`Repeating schedule`**, check if the job is stuck by waiting for the normal job duration or by checking the job history. If the job has been stuck for a long time, subsequent runs will be skipped. Use the following steps to navigate to the Observer configuration user interface (UI), view the observer job history and look for job runs that have been continuously skipped since the restore completed.

   1. Log in to IBM Cloud Pak for AIOps console.
   2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
   3. On the *Integrations* page, click **Add integration**.
   4. On the *Add integrations** page, click **Topology** in the *Category* list that is next to the list of all integrations.
   5. Click the **configure, schedule, and manage other observer jobs** link in the description for the topology integrations.
   6. Click the three dots icon at the end of the line for the observer job, and select **View history**.
   7. Click **All**, and look for runs with **`Skipped`** status and a **Details** entry of **`execution skipped because current job had state: FINISHING`**.

   For more information on viewing observer jobs, see the section **`To access the Observer Configuration UI`** in Defining observer jobs for application and topology data.

2. Attempt to restart the observer job.

1. For each observer job with a **Job type** of `Repeating schedule` that you identified in FINISHING state, edit the job in the Observer configuration UI by nominally changing the **Optional description** field, and saving it.

2. For other observer jobs that do not have a **Job type** of `Repeating schedule` and have a missing vertex issue, delete the related observer pod.

   ```
   oc delete pod <pod_name> -n <namespace>
   ```

   Where

   - `<pod_name>` is the observer pod to be deleted.
   - `<namespace>` is the project (namespace) where IBM Cloud Pak for AIOps is installed.

   For example, for an Instana observer job that has a missing vertex issue, delete the Instana observer pod:

   ```
   oc get pods -n | grep instana
   aiopsedge-instana-topology-integrator-6b777bd6b4-6nvqx 1/1 Running 0 8d

   oc delete pod aiopsedge-instana-topology-integrator-6b777bd6b4-6nvqx -n <cp4aiops-namespace>
   ```

   This causes the job to stop and restart, and to be shown as **Running**, **Scheduled**, or **Ready**.

   A message similar to the following is seen in the observer log:

   ```
   WARN   [2022-08-22 16:19:34,706] [pool-10-thread-1] c.i.i.t.o.a.JobManager -  stop - Stopped job for
   tenantId:cfd95b7e-3bc7-4006-a4a8-a73a79c71255, uniqueId:1100
   INFO   [2022-08-22 16:19:34,712] [cfd95b7e-3bc7-4006-a4a8-a73a79c71255:1100] c.i.i.t.o.f.j.LoadFileJob -
   cfd95b7e-3bc7-4006-a4a8-a73a79c71255:5eb27ee0-772c-4d65-bc0f-4e56b6b7913b Interrupted while waiting for state
   JobState [state=FINISHED, reason=Job finished after restart, date=Mon Aug 22 16:09:45 GMT 2022]
   INFO   [2022-08-22 16:19:34,944] [cfd95b7e-3bc7-4006-a4a8-a73a79c71255:1100] c.i.i.t.o.f.j.LoadFileJob -
   cfd95b7e-3bc7-4006-a4a8-a73a79c71255:1100 Observer job finished with state STOPPED
   ```

   Some jobs might still be shown with `Error` if an attempt was made to run the stuck job before the previous remediating steps were attempted. This causes the observer to fail the job, as a job already exists with the same name. In this scenario, a warning similar to the following is seen in the observer log:

   ```
   WARN   [2022-08-23 15:58:50,431] [pool-10-thread-1] c.i.i.t.o.a.o.RESTApiException -  pool-10-thread-1 - APIMessage:
   { httpCode=422, _error={ message=Job Creation Failure, level=warning, description=Cannot create observation job,
   causes=[{ message=Duplicate unique id, level=error, description=A job with the same unique id '1100' is already
   being processed., field=unique_id }] } }
   ```

   If there are no longer any observer jobs with a **Status** of `Error`, then exit this troubleshooting.

3. Complete the following steps only if one or more of your observer jobs are still shown with a **Status** of `Error`.

   1. Find the name of the topology pod.

      ```
      oc get pods | grep topology-topology
      ```

   2. Remote shell into the topology container.

      ```
      oc rsh <topology_pod> -c <release_name>-topology-topology
      ```

      Where

      - `<topology_pod>` is the pod that was returned in the previous step.
      - `<release_name>` is the name of your IBM Cloud Pak for AIOps instance.

4. Find observers that still have a job that is stuck in the `FINISHING` state.

   Run the following command, and note down the returned `observerName` and `_id`.

   ```
   curl -X GET "https://localhost:8080/1.0/topology/mgmt_artifacts?
   _filter=hasState%3DFINISHING&_field=keyIndexName&_field=hasState&_field=observerName&_type=ASM_OBSERVER_JOB" -H
   "accept: application/json" -H  "X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255" --insecure -u
   ${ASM_USER}:${ASM_PASS} | jq
   ```

   Note: ASM_USER and ASM_PASS are environment variables available within the topology container.

   Example output:

   ```
   {
    "_executionTime": 13,
    "_offset": 0,
    "_limit": 50,
    "_items": [
      {
        "keyIndexName": "dns-observer:bbc",
        "_id": "z7ovGeNrSKSpMlWY8n7Mfw",
        "observerName": "dns-observer",
        "hasState": "FINISHING"
      }
    ]
   }
   ```

5. Run the following steps for each observer job returned by the previous step.

   1. Change the state of the job by running the following command.

      ```
      curl -X POST "https://localhost:8080/1.0/topology/mgmt_artifacts/<id>" -H  "accept: application/json" -H  "X-
      TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255" -H  "Content-Type: application/json" -d "{  \"hasState\":
      \"FINISHED\"}" --insecure -u ${ASM_USER}:${ASM_PASS}
      ```

      Where `<id>` is the `_id` of the stuck job that you identified in step 4.

2. Find the pods for the observer that you need to restart.

   ```
   oc get pods -n <namespace> | grep <stuck_observer_name>
   ```

   Where

   - **<stuck_observer_name>** is the name of the stuck observer that you identified in step 4.
   - **<namespace>** is the project (namespace) where IBM Cloud Pak for AIOps is installed.
3. Delete the pods for the observer.

   ```
   oc delete po <observername_pods> -n <namespace>
   ```

   Where

   - **<observername_pods>** are the pods returned in the previous step.
   - **<namespace>** is the project (namespace) where IBM Cloud Pak for AIOps is installed.
4. Rerun the observer job.

   On the observer configuration UI, click **Run** on any job for the observer that is in the **Error** state. The observer job now runs successfully, and has a status of **Scheduled** or **Ready**.

Restore process terminated mid-process with partial data available

If the restore process does not complete as expected for a data store, such as if it is aborted or terminated during its run, data might not be restored correctly. This incomplete restored data neeeds to be removed before you run the restore process again. To remove the data, you need to run a post-restore cleanup script to clean up the data store.

To run a script, complete the following steps:

1. Define the following environment variable on your workstation:

   ```
   export WORKDIR="${PATH}/bcdr/4.8.0/"
   ```

2. Change to the **restore** directory where the post-restore cleanup script is located:

   ```
   cd <Path>/bcdr/4.8.0/restore/<data_store>/
   ```

   Where

   - **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.
   - **<data_store>** is the directory for the data store or custom resource that needs to be cleaned up. For example, **couchdb** is the directory for CouchDb.
3. Run the post-restore cleanup script:

   ```
   nohup ./<data-store>-post-restore.sh.sh > <data-store>-post-restore.log &
   ```

   Where **<data_store>** is the data store or custom resource that needs to be cleaned up.

4. Run the restore process for that data store or resource again.

For example, if the restore job for restoring the Cassandra data store aborted, you need to run the **cassandra-post-restore.sh** post-restore script that is stored in the **bcdr/restore/cassandra** directory to clean the data. Then, run the **cassandra-native-post-restore.sh** post-restore script.

The following table lists the cleanup script to run for each data store:

Table. Cleanup script for components

Component or Database	Cleanup script
Cassandra	Run first: **bcdr/4.8.0/restore/cassandra/cassandra-post-restore.sh** Then, run: **bcdr/4.8.0/restore/cassandra/cassandra-native-post-restore.sh**
CouchDB	**bcdr/4.8.0/restore/couchdb/couchdb-post-restore.sh**
Elasticsearch	**bcdr/4.8.0/restore/elasticsearch/es-post-restore.sh**
Metastore	**bcdr/4.8.0/restore/metastore/metastore-post-restore.sh**
Minio	**bcdr/4.8.0/restore/minio/minio-post-restore.sh**
Postgres	**bcdr/4.8.0/restore/postgres/postgres-post-restore.sh**
IBM Cloud Pak foundational services	**bcdr/4.8.0/restore/common-services/cs-post-restore.sh**
Connection CR	N/A
Secure Tunnel CR	**bcdr/4.8.0/restore/other-resources/tunnel-cr-post-restore.sh**

OADP restore is stuck in an `In progress` state

If you notice that the Velero pod is stuck in an **In progress** state, complete the following steps to stop the restore process:

1. Delete the velero pod by running the following command:

   ```
   oc delete pod <velero pod name> -n <OADP installed namespace>
   ```

2. Delete the restore that is stuck in progress by running the following command:

   ```
   velero delete restore <restore name>
   ```

   Wait for the completion of the restore job.

Helm install restore job command failed

When you are running the **helm install restore-job clusterrestore-0.1.0.tgz** command, you might encounter the command failing with an error that is similar to the following error:

```
Error: admission webhook "trust.hooks.securityenforcement.admission.cloud.ibm.com" denied the request:
Deny "icr.io/cpopen/cp4waiops/cp4aiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355", no
matching repositories in ClusterImagePolicy and no ImagePolicies in the "velero" namespace
```

If you encounter this error, complete the following steps to resolve the issue:

1. Uninstall the **restore-job** job by running the following command:

   ```
   helm uninstall restore-job -n $OADP_NAMESPACE
   ```

2. Export an environment variable for the image.

   For an online deployment:

   ```
   export REGISTRY=icr.io/cpopen/cp4waiops/cp4aiops-bcdr
   export BCDR_IMAGE=${REGISTRY}/<bcdr_image>
   ```

   For an offline deployment:

   ```
   export REGISTRY=$TARGET_REGISTRY
   export BCDR_IMAGE=${REGISTRY}/<bcdr_image>
   ```

   Where **<bcdr_image>** is the name of the backup and restore image, as given in the backup helm chart **values.yaml** file, in the form **cp4waiops-bcdr@{digest}**. An example value for BCDR_IMAGE is **icr.io/cpopen/cp4waiops/cp4waiops-bcdr@sha256:294a42a851a2717ebbc68528ab3c6bcb1ba48114ff058f1c1b537dc6aa167355**.

3. Create a **restore-image-policy.yaml** file and add the following content within the file:

   ```
   apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
   kind: ClusterImagePolicy
   metadata:
     name: restore-image-policy
   spec:
    repositories:
     - name: ${BCDR_IMAGE}
       policy:
   ```

4. Apply the policy by running the following command:

   ```
   oc apply -f restore-image-policy.yaml
   ```

5. Deploy the restore job by running the following command:

   ```
   helm install restore-job clusterrestore-0.1.0.tgz
   ```

Cassandra restore fails with `org.apache.cassandra.io.FSReadError` exception

Policies are not restored after restore completed. Although there were a couple of key repair failures, the log indicated the Cassandra restore completed. However, the policies from the backup cluster were not restored.

You can encounter an issue where the Cassandra restore fails. If this issue occurs, the **aiops-topology-cassandra-2** pod is in **0/1** status and the pod log contains an error message that is similar to the following error message:

```
ERROR [CompactionExecutor:8] 2022-11-29 21:39:17,324 CassandraDaemon.java:244 - Exception in thread
Thread[CompactionExecutor:8,1,main]
org.apache.cassandra.io.FSReadError: java.io.IOException: Channel not open for writing - cannot extend file to required
size
```

If this error occurs, complete the following steps for the Cassandra statefulset:

1. Scale down the statefulset **aiops-topology-cassandra** to **0**.

   Important: Make a note of the current scaling of the Cassandra statefulset before you scale down.

   ```
   oc scale statefulsets aiops-topology-cassandra --replicas 0 -n <namespace>
   ```

   Where: **<namespace>** is the namespace where IBM Cloud Pak for AIOps is installed.

2. Increase the statefulset memory limit to 32 GB.

   1. Access the resource for editing:

      ```
      oc edit statefulset aiops-topology-cassandra -n <namespace>
      ```

   2. Increase the memory limit:

3. Scale the statefulset back to the initial number of replicas.

   ```
   oc scale statefulsets aiops-topology-cassandra --replicas=<number_of_replicas> -n <namespace>
   ```

   Where:

   - **<namespace>** is the namespace where IBM Cloud Pak for AIOps is installed.
   - **<number_of_replicas>** is the number of replicas the StatefulSet it to be scaled up to.

4. Run the restore again.

5. Scale down the statefulset **aiops-topology-cassandra** to **0** again.

   ```
   oc scale statefulsets aiops-topology-cassandra --replicas 0 -n <namespace>
   ```

6. Change the statefulset memory limit to 16 GB .

   1. Access the resource for editing:

      ```
      oc edit statefulset aiops-topology-cassandra -n <namespace>
      ```

   2. Decrease the memory limit to 16 GB.

7. Scale the statefulset back to the initial number of replicas.

   ```
   oc scale statefulsets aiops-topology-cassandra --replicas=<number_of_replicas> -n <namespace>
   ```

   Where:

   - **<namespace>** is the namespace where IBM Cloud Pak for AIOps is installed.
   - **<number_of_replicas>** is the number of replicas the StatefulSet it to be scaled up to.

Integrations missing after a restore

After you complete a full restore, if you do not see all of the data integrations that you expect, run the command to restore data for the **Metastore** component to restore the integrations. For instructions, see Restoring individual components.

Netcool integration is missing details after restore

If you had a Netcool integration defined, then the Netcool integration might be missing some details after restore, such as the ObjectServer details. If you try to edit these details then the test still fails. To resolve this problem, delete the Netcool integration, and then create a new Netcool integration.

Restore fails with lifecycletrigger: Not Ready

If **lifecycletrigger** is showing *Not Ready* after the restore, then run the following command:

```
oc patch lifecycletrigger aiops --type=json --patch="$(oc get lifecycletrigger aiops -o jsonpath='{"
[{"}"op":"add","path":"/spec/cancelJobs","value":[{range .status.jobs[*]}"{.jid}",{end}]}]' | sed 's/,]/]/g')"
```

Cannot login to the Cloud Pak for AIOps console on the restore cluster

After the restore, the IBM Cloud Pak for AIOps console is inaccessible and there is an error similar to the following:

```
`CWOAU0061E: The OAuth service provider could not find the client because the client name is not valid`
```

Solution: Run the following steps on the restore cluster:

1. Export an environment variable containing your project's name.

   ```
   export PROJECT=<project>
   ```

   Where **<project>** is the namespace (project) that IBM Cloud Pak for AIOps is deployed in on the restore cluster.

2. Delete the platform pods.

   ```
   oc delete pod -n ${PROJECT} -l component=platform-auth-service
   oc delete pod -n ${PROJECT} -l component=platform-identity-management
   oc delete pod -n ${PROJECT} -l component=platform-identity-provider
   ```

3. Update and delete the **iam-config-job** YAML.

   ```
   oc get  -n ${PROJECT} job iam-config-job  -o json > /tmp/iam-config-job.json
   oc -n ${PROJECT} delete job iam-config-job
   jq 'del(.metadata.creationTimestamp) | del(.metadata.managedFields) | del(.metadata.resourceVersion) |
   del(.metadata.uid) | del(.spec.selector) | del(.spec.template.metadata.labels) | del(.status)' /tmp/iam-config-
   job.json > /tmp/updated-iam-config-job.json
   ```

4. Wait for a few seconds and then apply the modified YAML to rerun **iam-config-job**.

   ```
   oc -n ${PROJECT} apply -f /tmp/updated-iam-config-job.json
   oc -n ${PROJECT} get pods | grep iam-config-job
   ```

Policy configuration is not retained on the restored deployment

Incident policies that previously had a ticket integration are missing the ticket integration on the restored deployment. If GitHub or ServiceNow are configured, then external tickets are not created in them. The policy page has a **Warning** similar to the following example:

```
The previously selected ticket connection Github -mygit is no longer available. Please check the status of your
connector.
```

Solution: From the IBM Cloud Pak for AIOps console, edit the policy and add the connection that is shown in the warning.

Troubleshooting the Infrastructure Automation restore

If you are also restoring Infrastructure Automation data and encounter an issues with the restore process for Infrastructure Automation or encounter an issue with data not being available or processed after the restore, see Troubleshooting the Infrastructure Automation restore

In addition, review the Known issues with the backup and restore process.

Known issues with the backup and restore process

- Elasticsearch health status yellow after restore

Elasticsearch health status yellow after restore

When you are restoring an Elasticsearch backup to a new single node Elasticsearch cluster, the Elasticsearch database might not work as expected. Instead, the Elasticsearch cluster health shows a yellow status after the restore completes.

Upgrading IBM Cloud Pak for AIOps backup and restore artifacts

Use this procedure if you want to run backup or restore on an upgraded IBM Cloud Pak® for AIOps deployment.

If you run a backup or restore on an IBM Cloud Pak for AIOps deployment, the backup and restore scripts must always be the same version as your deployment. For example, if you run backup on a IBM Cloud Pak for AIOps v4.8.0 deployment, you must use a v4.8.0 backup script.

If you previously set up backup or restore on a deployment that you have now upgraded, then you must clean up any existing backup or restore artifacts. If you then want to run a new backup or restore, you must get the new version of the backup or restore scripts.

Note: If you have never set up backup or restore and you want to run backup or restore, then use the instructions in IBM Cloud Pak for AIOps backup and restore.

Set up backup on an upgraded IBM Cloud Pak for AIOps deployment

Use the following steps to upgrade your backup artifacts.

1. Uninstall the backup helm chart.

   Run the following command in the namespace that **Red Hat OpenShift APIs for Data Protection** (OADP) is installed in.

   ```
   helm uninstall backup-job
   ```

2. Set up backup for IBM Cloud Pak for AIOps v4.8.0.

   If you need to run backup for IBM Cloud Pak for AIOps v4.8.0, follow the instructions in Backing up IBM Cloud Pak for AIOps data.

   Important: The **cp4waiops-bcdr** image must be the same version as your IBM Cloud Pak for AIOps deployment. Follow these instructions to ensure that your **cp4waiops-bcdr** image is v4.8.0 for your IBM Cloud Pak for AIOps v4.8.0 deployment.

Set up restore on an upgraded IBM Cloud Pak for AIOps deployment

Use the following steps to upgrade your restore artifacts.

1. Uninstall the restore helm chart.

   Run the following command in the namespace that **Red Hat OpenShift APIs for Data Protection** (OADP) is installed in.

   ```
   helm uninstall restore-job
   ```

2. Set up restore for IBM Cloud Pak for AIOps v4.8.0.

   If you need to run restore for IBM Cloud Pak for AIOps v4.8.0, follow the instructions in Restoring IBM Cloud Pak for AIOps.

   Important: The **cp4waiops-bcdr** image must be the same version as your IBM Cloud Pak for AIOps deployment. Follow these instructions to ensure that your **cp4waiops-bcdr** image is v4.8.0 for your IBM Cloud Pak for AIOps v4.8.0 deployment.

Multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps

Multi-region cold standby disaster recovery (DR) enables the restoration of your IBM Cloud Pak for AIOps deployment if an unrecoverable failure impacts the deployment or its hosting cluster, data center, or region.

Overview

The multi-region cold standby disaster recovery solution for IBM Cloud Pak for AIOps uses IBM Fusion to create and restore backups of your IBM Cloud Pak for AIOps deployment and its data. For more information about IBM Fusion, see the Fusion documentation.

IBM Fusion uses the concept of a hub and spoke. A Red Hat® OpenShift® Container Platform cluster that has an IBM Fusion backup & restore service installation on it is the hub cluster. The hub manages the backups, and backups can also be taken or restored on a hub cluster. A Red Hat OpenShift cluster that has an IBM Fusion backup & restore agent installation on it is a spoke cluster. Backups can be taken or restored on a spoke, but the spoke cannot manage the backups.

IBM Fusion takes a backup of your IBM Cloud Pak for AIOps deployment, and stores it in external object storage. If a problem occurs that renders your IBM Cloud Pak for AIOps deployment unusable, then the backup can be restored to a Red Hat OpenShift spoke cluster. The entire IBM Cloud Pak for AIOps deployment is restored, along with its data.

Entitlement to IBM Cloud Pak for AIOps provides you with entitlement to the IBM Fusion Data Foundation service in internal deployment mode only. To implement multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps, you must purchase a full IBM Fusion license for access to the backup and disaster recovery features. For more information, see the license terms.

Restrictions

- IBM Fusion and the IBM Cloud Pak for AIOps deployment to be backed up must be installed on the same Red Hat OpenShift cluster.
- The backup and restore procedures cannot be used on offline (air gapped) deployments of IBM Cloud Pak for AIOps.
- The backup and restore procedures cannot be used if IBM Cloud Pak for AIOps is installed with Infrastructure Automation.
- The backup and restore procedures cannot be used on `AllNamespace` deployments of IBM Cloud Pak for AIOps. For more information about installation modes, see Operator installation mode.
- If IBM Cognos® Analytics is installed, it is not backed up or recovered.
- User interface customizations are not backed up or restored.

Before you begin

Red Hat OpenShift

Ensure that you meet the following prerequisites for the hub cluster, and also for the spoke cluster if you are using one:

- The Red Hat OpenShift version is supported by IBM Cloud Pak for AIOps and IBM Fusion v2.8.x or v2.9.x.
  - IBM Cloud Pak for AIOps: For more information, see Supported Red Hat OpenShift versions.
  - IBM Fusion: For more information, see Prerequisites in the IBM Fusion documentation.
- The Red Hat OpenShift command-line interface (oc) is installed. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI in the Red Hat OpenShift documentation.
- You are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).

Storage

Ensure that you meet the following prerequisites for your persistent and external storage.

- In addition to your IBM Cloud Pak for AIOps deployment's persistent storage, you must have external object storage available for IBM Fusion to use as the backup storage location. This is where the IBM Cloud Pak for AIOps volume backups will be stored. This can be any S3 compatible storage, or Microsoft® Azure Blob Storage.
- The amount of external object storage that is required is affected by the amount of data to be stored, the change rate of the data, and the storage efficiencies that can be made from compression and deduplication. As a guide, for a 30-day retention period, you must have object storage capacity equivalent to a minimum of 125% of your IBM Cloud Pak for AIOps deployment's persistent storage. For example, if you have 100 GB of persistent storage for IBM Cloud Pak for AIOps, then you must have 125 GB of object storage available for backups.
- The persistent storage that IBM Cloud Pak for AIOps uses must be Container Storage Interface (CSI) compatible, and must support volume snapshot and restore. The CSI driver must be v1 or higher.
- If you are using Portworx as your storage provider, then IBM Fusion must be at v2.9.0 or higher, or backup and restore will fail.
- PersistentVolumes (PVs) must be created with `volumeMode` set to *Filesystem*.
- The storage class that is used by IBM Cloud Pak for AIOps must have `volumeBindingMode` set to *Immediate*, and `allowVolumeExpansion` set to *true*.

For more information, see the `Backup & Restore` section in System requirements in the IBM Fusion documentation.

Backup and restore procedures

Use the instructions in the following sections to backup or restore an IBM Cloud Pak for AIOps deployment.

- Multi-region cold standby disaster recovery: Backing up IBM Cloud Pak for AIOps
- Multi-region cold standby disaster recovery: Restoring IBM Cloud Pak for AIOps
- Troubleshooting multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps

Multi-region cold standby disaster recovery: Backing up IBM Cloud Pak for AIOps

Use IBM Fusion to create a backup of your IBM Cloud Pak for AIOps deployment.

Backup procedure

Ensure that you meet the prerequisites in Before you begin.

Use the following instructions to install IBM Fusion on the same Red Hat OpenShift cluster as IBM Cloud Pak for AIOps, and then create a backup of IBM Cloud Pak for AIOps.

1. Ensure that you have a VolumeSnapshotClass
2. Install IBM Fusion
3. Deploy the IBM Fusion Backup & Restore service
4. Configure the IBM Fusion Backup & Restore service for IBM Cloud Pak for AIOps

1. Ensure that you have a VolumeSnapshotClass

To enable IBM Fusion backups, you must have a **VolumeSnapshotClass** defined for one of the storage classes that host IBM Cloud Pak for AIOps. For more information, see Prerequisites in the IBM Fusion documentation.

Run the following command to check whether a **VolumeSnapshotClass** is defined already:

```
oc get volumesnapshotclass
```

If the preceding command does not return anything, then you do not have a **VolumeSnapshotClass** defined. Refer to your storage provider's documentation to create one.

2. Install IBM Fusion

You must install IBM Fusion v2.8.x or v2.9.x.

Follow the *Installing* instructions for your Red Hat OpenShift deployment type in Deploying IBM Fusion in the IBM Fusion documentation. Accept the default values.

Note: As you are installing IBM Fusion on the same Red Hat OpenShift cluster as IBM Cloud Pak for AIOps, you already have an entitlement pull secret configured.

3. Deploy the IBM Fusion Backup & Restore service

1. Install the IBM Fusion Backup & Restore service.

   Follow the instructions in Backup and restore hub in the IBM Fusion documentation.

   The **ibm-backup-restore** project (namespace) is created, and the backup and restore service is installed in that project.

2. Configure the backup storage location for IBM Fusion to store the IBM Cloud Pak for AIOps backups in.

   The backup storage location must be external object storage, and can be any S3 compatible storage or Microsoft® Azure Blob Storage.

   Follow the instructions in Adding a backup storage location in the IBM Fusion documentation.

3. Create a backup policy.

   The backup policy specifies the backup location, frequency, and retention period of the backup. Specify the backup storage location that you created in the previous step as the backup location.

   Follow the instructions in Creating backup policy in the IBM Fusion documentation.

4. Select the IBM Cloud Pak for AIOps application that you want to backup, and assign the backup policy to it.

   Follow the instructions in Selecting applications for protection in the IBM Fusion documentation.

   Important: Before you select **Assign** in the user interface, you must deselect **Run backup now** to prevent the backup from running immediately.

5. Find the name of your policy assignment.

   Your policy assignment name is the IBM Cloud Pak for AIOps installation namespace concatenated with the backup policy name. Run the following command to find it, and then make a note of it:

   ```
   oc get fpa -n ibm-spectrum-fusion-ns
   ```

4. Configure the IBM Fusion Backup & Restore service for IBM Cloud Pak for AIOps

1. Run the following commands to set the environment variables that you need.

   ```
   export PROJECT_CP4AIOPS=<namespace>
   export CUSTOM_PROFILE=<custom_profile>
   export POLICY_ASSIGNMENT_NAME=<policy_assignment_name>
   ```

   Where:

   - **<namespace>** is the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in, for example **cp4aiops**.
   - **<custom_profile>** is blank unless IBM Sales representatives and Business Partners used the custom sizing tool to supply you with a custom profile ConfigMap to customize your deployment. If you were supplied with a custom profile ConfigMap, then this is the name of your custom profile ConfigMap, usually **aiops-custom-size-profile**.
   - **<policy_assignment_name>** is the policy assignment name that you found in the previous step.

2. Find your installation name and set an environment variable for it.

   1. Find the name of your IBM Cloud Pak for AIOps installation.

      ```
      oc get installation -n ${PROJECT_CP4AIOPS}
      ```

      Example output, where **ibm-cp-aiops** is the name of the IBM Cloud Pak for AIOps installation:

      ```
      NAME          PHASE    LICENSE   STORAGECLASS STORAGECLASSLARGEBLOCK AGE
      ibm-cp-aiops  Running  Accepted  rook-cephfs  rook-ceph-rbd          34h
      ```

   2. Export an environment variable with the name of your installation.

      ```
      export INSTALLATION_NAME=<name>
      ```

Where **<name>** is the name returned by the previous command, for example **ibm-cp-aiops**.

3. Run the following commands to add labels to IBM Cloud Pak for AIOps resources so that they are correctly identified.

```
oc -n $PROJECT_CP4AIOPS label secret ${INSTALLATION_NAME}-edb-secret manual=${INSTALLATION_NAME}-edb-secret
oc -n $PROJECT_CP4AIOPS label secret ${INSTALLATION_NAME}-redis-secret manual=${INSTALLATION_NAME}-redis-secret
oc -n $PROJECT_CP4AIOPS label secret ibm-zen-secret-jwks manual=ibm-zen-secret-jwks
oc -n $PROJECT_CP4AIOPS label secret ibm-zen-secret-jwt manual=ibm-zen-secret-jwt
oc -n $PROJECT_CP4AIOPS label secret ibm-zen-secret-jwt-private manual=ibm-zen-secret-jwt-private
oc -n $PROJECT_CP4AIOPS label secret aiopsedge-config-encryption-history manual=aiopsedge-config-encryption-history
oc -n $PROJECT_CP4AIOPS label secret platform-auth-ldaps-ca-cert manual=platform-auth-ldaps-ca-cert
oc -n $PROJECT_CP4AIOPS label secret ibm-entitlement-key manual=ibm-entitlement-key-secret
oc -n $PROJECT_CP4AIOPS label subscriptions.operators.coreos.com ibm-aiops-orchestrator manual=ibm-aiops-
orchestrator
```

4. Download the IBM Cloud Pak for AIOps backup and restore IBM Fusion recipe.

   The IBM Fusion recipe defines the sequence of actions and checks that IBM Fusion must use for the backup and restore of IBM Cloud Pak for AIOps resources and PVCs.

   Download the recipe **aiops-backup-restore.yaml**, from github.com/IBM.

5. Adjust the recipe if you have a non-default installation name.

   If your installation name is not **ibm-cp-aiops**, then edit the recipe that you downloaded, and alter all occurences of **ibm-cp-aiops** to be your installation name.

6. Adjust the recipe for custom sizing.

   IBM Sales representatives and Business Partners may have used a custom sizing tool to supply you with a custom profile to customize your IBM Cloud Pak for AIOps deployment at installation time.

   If IBM Sales representatives and Business Partners did not supply you with a custom profile, then skip this step.

   If IBM Sales representatives and Business Partners supplied you with a custom profile, then add a label to the custom profile ConfigMap in your IBM Cloud Pak for AIOps deployment. Run the following command:

```
oc label cm ${CUSTOM_PROFILE} cm=aiops-custom-size-profile
```

7. Create the IBM Cloud Pak for AIOps recipe custom resource.

   Run the following command to create the recipe custom resource.

```
oc apply -f aiops-backup-restore.yaml
```

8. Configure your policy assignment to use the IBM Cloud Pak for AIOps back and restore recipe.

   Patch the policy assignment with the details of the IBM Cloud Pak for AIOps back and restore recipe.

```
oc -n ibm-spectrum-fusion-ns patch policyassignment ${POLICY_ASSIGNMENT_NAME} --type merge  -p '{"spec":{"recipe":
{"name":"aiops-backup-restore-recipe", "namespace":"ibm-spectrum-fusion-ns"}}}'
```

What to do next

Backups occur according to the schedule that you set in your policy. If you want to run a backup immediately, follow the instructions in Backup now.

You can monitor the progress of the backup from **Backup & restore** > **Jobs** in the IBM Fusion user interface.

To view the IBM Fusion logs, or to see the resources that were backed up, see Troubleshooting multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps.

Multi-region cold standby disaster recovery: Restoring IBM Cloud Pak for AIOps

Use IBM Fusion to restore your IBM Cloud Pak for AIOps deployment from backup.

Restore procedure

IBM Fusion recovers the IBM Cloud Pak for AIOps instance to a working state on a target cluster, along with its data. Your target cluster can be the hub cluster, or a different Red Hat OpenShift cluster (a spoke).

Ensure that you meet the prerequisites in Before you begin. User interface customizations are not restored.

Follow the steps for Restoring to the hub cluster or Restoring to a spoke cluster.

Restoring to the hub cluster

If you are restoring IBM Cloud Pak for AIOps to the same Red Hat OpenShift cluster that you created your backup on, then you must completely uninstall your IBM Cloud Pak for AIOps deployment, and remove the deployment namespaces from it (usually **cp4aiops**). The Red Hat OpenShift cluster must not have any residual IBM Cloud Pak for AIOps artifacts. For more information, see Uninstalling IBM Cloud Pak for AIOps.

1. Get the entitlement key.

   Follow the instructions in Obtaining the entitlement key in the IBM Fusion documentation.

2. Create a global pull secret.

   Follow the instructions in Creating image pull secret in the IBM Fusion documentation.

3. Create the required catalogs.

   Follow the instructions in Create the catalog sources.

4. Install Cert Manager

   Skip this step if a certificate manager is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not have a certificate manager, then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

   For more information about IBM Cloud Pak® foundational services Cert Manager hardware requirements, see IBM Certificate Manager (cert-manager) hardware requirements in the IBM Cloud Pak foundational services documentation.

   1. Run the following command to create the resource definitions that you need:

      ```
      cat << EOF | oc apply -f -
      apiVersion: v1
      kind: Namespace
      metadata:
        name: ibm-cert-manager
      ---
      apiVersion: operators.coreos.com/v1
      kind: OperatorGroup
      metadata:
        name: ibm-cert-manager-operator-group
        namespace: ibm-cert-manager
      ---
      apiVersion: operators.coreos.com/v1alpha1
      kind: Subscription
      metadata:
        name: ibm-cert-manager-operator
        namespace: ibm-cert-manager
      spec:
        channel: v4.2
        installPlanApproval: Automatic
        name: ibm-cert-manager-operator
        source: ibm-cert-manager-catalog
        sourceNamespace: openshift-marketplace
      EOF
      ```

   2. Run the following command to ensure that the IBM Cloud Pak® foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

      ```
      oc -n ibm-cert-manager get pods
      ```

      Example output for a successful IBM Cloud Pak® foundational services Cert Manager installation:

      ```
      NAME                                         READY   STATUS    RESTARTS   AGE
      cert-manager-cainjector-674854c49d-vstq4     1/1     Running   0          8d
      cert-manager-controller-646d4bd6fd-zwmqm     1/1     Running   0          8d
      cert-manager-webhook-8598787c8-s41kt         1/1     Running   0          8d
      ibm-cert-manager-operator-c96957695-dkxnm    1/1     Running   0          8d
      ```

5. Install the License Service

   Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

   IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

   1. Run the following command to create the resource definitions that you need:

      ```
      cat << EOF | oc apply -f -
      apiVersion: v1
      kind: Namespace
      metadata:
        name: ibm-licensing
      ---
      apiVersion: operators.coreos.com/v1
      kind: OperatorGroup
      metadata:
        name: ibm-licensing-operator-group
        namespace: ibm-licensing
      spec:
        targetNamespaces:
        - ibm-licensing
      ---
      apiVersion: operators.coreos.com/v1alpha1
      kind: Subscription
      ```

```
metadata:
  name: ibm-licensing-operator-app
  namespace: ibm-licensing
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-licensing-operator-app
  source: ibm-licensing-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-licensing get pods
   ```

   Example output for a successful IBM Cloud Pak® foundational services License Service installation:

   ```
   NAME                                                  READY   STATUS    RESTARTS   AGE
   ibm-licensing-operator-db4cd746c-xzmlf                1/1     Running   0          8d
   ibm-licensing-service-instance-596b99588f-76cc5       1/1     Running   0          8d
   ```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

6. Update the IBM Fusion Backup & Restore transaction manager cluster roles.

   Run the following command:

   ```
   oc get clusterrole transaction-manager-ibm-backup-restore -o json | jq '.rules += [{"verbs":
   ["get","list"],"apiGroups":["orchestrator.aiops.ibm.com"],"resources":["installations"]},{"verbs":
   ["get","list"],"apiGroups":["ai-manager.watson-aiops.ibm.com"],"resources":["aimanagers"]},{"verbs":
   ["get","list"],"apiGroups":["lifecycle.ir.aiops.ibm.com"],"resources":["lifecycleservices"]}]' | oc apply -f -
   ```

7. Follow the instructions in Restoring the application in the IBM Fusion documentation.

   Select **Same cluster**, and then for **Project destination** select *Use the same project the application is already using*. Accept the default values for the other fields.

   Note: You might notice that the `aiops-configmaps` resource group shows as failed under the **Restore sequence** section of the IBM Cloud Pak for AIOps restore recipe but the overall restore is successful and works as intended.

8. Delete any duplicate observer jobs on the restored cluster.

   1. Log in to IBM Cloud Pak for AIOps console.
   2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
   3. For any duplicate jobs, click the options menu (three vertical dots) and click **Delete**.

9. From the IBM Cloud Pak for AIOps console, select and edit each `incident` policy that utilises a `ticket` integration, and manually reassign the relevant ticketing system (Github/ServiceNow).

Restoring to a spoke cluster

If you are restoring IBM Cloud Pak for AIOps to a different cluster, ensure that your target cluster meets the prerequisites for an IBM Cloud Pak for AIOps deployment, and then install IBM Fusion and the IBM Fusion backup agent on it.

1. Spoke cluster prerequisites

- You must have the same storage provider and storage classes available for your target spoke cluster as you did for your IBM Cloud Pak for AIOps deployment on the hub cluster.
- The target spoke cluster must have the same deployment platform and Red Hat OpenShift version as the source hub cluster. Review the Planning section and ensure that the environment that you are restoring to meets the system requirements.
- The Red Hat OpenShift cluster that you restore on must not have a previous IBM Cloud Pak for AIOps deployment on it, and must not have any residual IBM Cloud Pak for AIOps artifacts or deployment namespaces (usually `cp4aiops`). If you need to uninstall IBM Cloud Pak for AIOps, see the instructions in Uninstalling IBM Cloud Pak for AIOps.

2. Install IBM Fusion on the target spoke cluster

   1. Get the entitlement key.

      Follow the instructions in Obtaining the entitlement key in the IBM Fusion documentation.

   2. Create a global pull secret.

      Follow the instructions in Creating image pull secret in the IBM Fusion documentation.

   3. Create the required catalogs.

      Follow the instructions in Create the catalog sources.

   4. Install IBM Fusion.

      You must install IBM Fusion v2.8.x or v2.9.x.

Follow the **Installing** instructions for your Red Hat OpenShift deployment type in Deploying IBM Fusion in the IBM Fusion documentation. Accept the default values.

3. Install Cert Manager

   Skip this step if you already have a certificate manager installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on. If you do not have a certificate manager then you must install one. The IBM Cloud Pak® foundational services Cert Manager is recommended, and can be installed with the following steps.

   For more information about IBM Cloud Pak® foundational services Cert Manager, see Certificate Management in the IBM Cloud Pak foundational services documentation.

   1. Run the following command to create the resource definitions that you need:

      ```
      cat << EOF | oc apply -f -
      apiVersion: v1
      kind: Namespace
      metadata:
        name: ibm-cert-manager
      ---
      apiVersion: operators.coreos.com/v1
      kind: OperatorGroup
      metadata:
        name: ibm-cert-manager-operator-group
        namespace: ibm-cert-manager
      ---
      apiVersion: operators.coreos.com/v1alpha1
      kind: Subscription
      metadata:
        name: ibm-cert-manager-operator
        namespace: ibm-cert-manager
      spec:
        channel: v4.2
        installPlanApproval: Automatic
        name: ibm-cert-manager-operator
        source: ibm-cert-manager-catalog
        sourceNamespace: openshift-marketplace
      EOF
      ```

   2. Run the following command to ensure that the IBM Cloud Pak® foundational services Cert Manager pods have a **STATUS** of *Running* before proceeding to the next step.

      ```
      oc -n ibm-cert-manager get pods
      ```

      Example output for a successful IBM Cloud Pak® foundational services Cert Manager installation:

      ```
      NAME                                       READY   STATUS    RESTARTS   AGE
      cert-manager-cainjector-674854c49d-vstq4   1/1     Running   0          8d
      cert-manager-controller-646d4bd6fd-zwmqm   1/1     Running   0          8d
      cert-manager-webhook-8598787c8-s4lkt       1/1     Running   0          8d
      ibm-cert-manager-operator-c96957695-dkxnm  1/1     Running   0          8d
      ```

4. Install the License Service

   Skip this step if the IBM Cloud Pak® foundational services License Service is already installed on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

   IBM Cloud Pak for AIOps requires the installation of the IBM Cloud Pak foundational services License Service. You must install the IBM Cloud Pak foundational services License Service on the Red Hat OpenShift cluster that you are installing IBM Cloud Pak for AIOps on.

   1. Run the following command to create the resource definitions that you need:

      ```
      cat << EOF | oc apply -f -
      apiVersion: v1
      kind: Namespace
      metadata:
        name: ibm-licensing
      ---
      apiVersion: operators.coreos.com/v1
      kind: OperatorGroup
      metadata:
        name: ibm-licensing-operator-group
        namespace: ibm-licensing
      spec:
        targetNamespaces:
        - ibm-licensing
      ---
      apiVersion: operators.coreos.com/v1alpha1
      kind: Subscription
      metadata:
        name: ibm-licensing-operator-app
        namespace: ibm-licensing
      spec:
        channel: v4.2
        installPlanApproval: Automatic
        name: ibm-licensing-operator-app
        source: ibm-licensing-catalog
        sourceNamespace: openshift-marketplace
      EOF
      ```

2. Run the following command to ensure that the IBM Cloud Pak® foundational services License Server pods have a **STATUS** of *Running* before proceeding to the next step.

   ```
   oc -n ibm-licensing get pods
   ```

   Example output for a successful IBM Cloud Pak® foundational services License Service installation:

   ```
   NAME                                          READY   STATUS    RESTARTS   AGE
   ibm-licensing-operator-db4cd746c-xzmlf        1/1     Running   0          8d
   ibm-licensing-service-instance-596b99588f-76cc5   1/1     Running   0          8d
   ```

For more information about the IBM Cloud Pak® foundational services License Service, see License Service in the IBM Cloud Pak foundational services documentation.

5. Get connection YAML from the hub cluster.

   On the hub cluster, follow the **`Use the Fusion UI`** instructions in Connect clusters in the IBM Fusion documentation.

   When you have copied the connection snippet, return to these steps. The connection snippet is used in the next step to connect the hub and spoke clusters.

6. Install the **`Backup & Restore Agent`** on the spoke cluster, and connect the spoke cluster to the hub cluster.

   On the spoke cluster, follow the procedure in Backup & Restore spoke in the IBM Fusion documentation.

   For the **Hub Connection Snippet** field, use the connection snippet that you copied in the previous step. The storage classes must be the same storage classes that the IBM Cloud Pak for AIOps deployment on the hub cluster used.

7. On the spoke cluster, update the IBM Fusion Backup & Restore transaction manager cluster roles.

   Run the following command:

   ```
   oc get clusterrole transaction-manager-ibm-backup-restore -o json | jq '.rules += [{"verbs":
   ["get","list"],"apiGroups":["orchestrator.aiops.ibm.com"],"resources":["installations"]},{"verbs":
   ["get","list"],"apiGroups":["ai-manager.watson-aiops.ibm.com"],"resources":["aimanagers"]},{"verbs":
   ["get","list"],"apiGroups":["lifecycle.ir.aiops.ibm.com"],"resources":["lifecycleservices"]}]' | oc apply -f -
   ```

8. Follow the instructions in Restoring the application in the IBM Fusion documentation, and wait for the restore to complete.

   Select **New cluster**, and then accept the default values for the other fields.

   Note: You might notice that the **`aiops-configmaps`** resource group shows as failed under the **Restore sequence** section of the IBM Cloud Pak for AIOps restore recipe but the overall restore is successful and works as intended.

9. Redeploy remote integrations.

   Re-run the bootstrap procedure for any remote connections that are configured.

   1. Log in to IBM Cloud Pak for AIOps console.
   2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
   3. For each remote integration, click the integration type on the **Manage integrations** tab of the Integrations page, then:
      1. Click the **Download** link in the **Remote deployment script** column for the integration.
      2. Click **Download as a sh-file** to download the deployment script as a **`bootstrap.sh`** file.
      3. From the Red Hat OpenShift CLI, switch to the target project (namespace).

         ```
         oc project <namespace>
         ```

         Where **`<namespace>`** is the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in, for example *cp4aiops*.

      4. Run the **`bootstrap.sh`** deployment script.

10. If you have secure tunnel connections, then re-create them. For more information, see Create the secure tunnel connection on your restored cluster.

11. Delete any duplicate observer jobs on the restored cluster.

    1. Log in to IBM Cloud Pak for AIOps console.
    2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
    3. For any duplicate jobs, click the options menu (three vertical dots) and click **Delete**.

12. From the IBM Cloud Pak for AIOps console, select and edit each **`incident`** policy that utilises a **`ticket`** integration, and manually reassign the relevant ticketing system (Github/ServiceNow).

What to do next

The restore takes a few hours to run. You are notified on the IBM Fusion user interface (UI) when the restore is complete. You can also monitor the progress of the restore from **Backup & restore** > **Jobs** in the IBM Fusion UI.

Important: If you restored to a spoke cluster, the URL and password for accessing the IBM Cloud Pak for AIOps console are different from the URL and password that you used on the hub cluster. Follow the steps in Access the Cloud Pak for AIOps console to find the new URL and password.

(Optional) After the restore is complete, you can verify the status of your deployment by using the steps in Verify your installation (CLI) or Verify your installation (console).

To view the IBM Fusion logs, or to see the resources that were backed up or restored, see Troubleshooting multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps.

Troubleshooting multi-region cold standby disaster recovery for IBM Cloud Pak for AIOps

Troubleshoot your IBM Fusion backup or restore of IBM Cloud Pak for AIOps.

General troubleshooting

- Retrieve backup or restore logs
- List resources that were backed up or restored

Restore issues

- Policy configuration is not retained on the restored deployment
- PostgresSQL instance is out of sync after fusion restore

General troubleshooting tips

Retrieve backup or restore logs

1. Download **`getRecipeWorkflow.sh`** from github.com/IBM.
2. Obtain the job UID for the backup or restore job. The job UID can be obtained from the IBM Fusion user interface, or from the command-line.

   From the IBM Fusion user interface:

   1. Select **Applications** from the left navigation menu.
   2. Select the Application from the table, for example **`filebrowser`**.
   3. Select **Backups**.
   4. For the backup or restore job of interest, select the right ... menu and select **Details**.
   5. Click the job name, for example **`filebrowser-demo-daily-apps.test.stglabs.ibm.com-202311220800`**.
   6. Note the **Job ID** in the **Details** window, for example: **`63eb6ea5-dc24-41ef-9fb7-199c48df1508`**.`

   From the command-line:

   1. Open the YAML file for the backup custom resource. For example:

      ```
      oc get fbackup filebrowser-demo-daily-apps.test.stglabs.ibm.com-202311220800 -n ibm-spectrum-fusion-ns -o yaml
      ```

   2. Locate the **`uid`** in **`spec: metatdata`**, for example: **`uid: 63eb6ea5-dc24-41ef-9fb7-199c48df1508`**.
3. Run **`getRecipeWorkflow.sh`** with the job UID.

   ```
   getRecipeWorkflow.sh <job_type> <job_uid>
   ```

   Where

   - **`<job_type>`** is **`backup`** or **`restore`**.
   - **`<job_uid>`** is the job UID that you obtained.

   Example command:

   ```
   getRecipeWorkflow.sh backup 63eb6ea5-dc24-41ef-9fb7-199c48df1508
   ```

List resources that were backed up or restored

1. Download **`getResources.sh`** from github.com/IBM.
2. Use step 2 in Retrieve backup or restore logs to get the job UID.
3. Run **`getResources.sh`** with the job UID.

   ```
   getReesources.sh <job_type> <job_uid>
   ```

   Where

   - **`<job_type>`** is **`backup`** or **`restore`**.

- `<job_uid>` is the job UID that you obtained.

Example command:

```
getResources.sh backup 63eb6ea5-dc24-41ef-9fb7-199c48df1508
```

Troubleshooting restore issues

Policy configuration is not retained on the restored deployment

Incident policies that previously had a ticket integration are missing the ticket integration on the restored deployment. If GitHub or ServiceNow are configured, then external tickets are not created in them. The policy page has a **Warning** similar to the following example:

```
The previously selected ticket connection Github -mygit is no longer available. Please check the status of your
connector.
```

Solution: From the IBM Cloud Pak for AIOps console, edit the policy and add the connection that is shown in the warning.

PostgresSQL instance is out of sync after fusion restore

After restore, an EDB Postgres instance has a different value for the Log Sequence Number (LSN), which means that it is not synchronized with the other instances.

1. If it is not already installed, install the `kubectl` plug-in.

   Run the following command:

   ```
   curl -sSfL \
   https://github.com/EnterpriseDB/kubectl-cnp/raw/main/install.sh | \
   sudo sh -s -- -b /usr/local/bin
   ```

2. Run the following command to check whether all the EDB Postgres instances are synchronized.

   ```
   oc cnp status ibm-cp-aiops-edb-postgres
   ```

   If a Postgres instance is not synchronized, then the value for `Current LSN` is not the same as the `Current LSN` for the other instances.

   Example output where a Postgres instance is not synchronized: `worker9.ir-backend-backup-acme.com` has a different Log Sequence Number (LSN):

   ```
    Name                          Database  Current     Replication           Status  QoS         Manager  Node
                                  Size      LSN         role                                      Version
    ibm-cp-aiops-edb-postgres-1   83 MB     6/C214B440  Primary               OK      Burstable   1.18.13  worker7.ir-
   backend-backup-acme.com
    ibm-cp-aiops-edb-postgres-2   82 MB     6/C214B440  Standby (async)       OK      Burstable   1.18.13  worker2.ir-
   backend-backup-acme.com
    ibm-cp-aiops-edb-postgres-3   82 MB     6/A124B330  Standby (file based)  OK      Burstable   1.18.13  worker9.ir-
   backend-backup-acme.com
   ```

Solution: If one or more of the Postgres instances are not synchronized, follow the instructions in Postgres EDB database crashes.

Configuring topology resources for production (IBM Cloud Pak for AIOps on OpenShift)

Learn how to configure the File observer for more than 600,000 resources.

Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).

Configuring the File observer

Additional configuration is required if you are using the File observer for more than 600,000 resources.

If IBM Sales representatives and Business Partners did not use the custom sizing tool to supply you with a custom profile ConfigMap to customize your deployment, then run the following command:

```
cat <<EOF | oc apply -f -
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ibm-aiops-orchestrator
    app.kubernetes.io/name: custom-sized-profiles
    app.kubernetes.io/part-of: ibm-aiops
    aiops-custom-size-profile-version: 4.2.0-2023
  name: aiops-custom-size-profile
data:
  # WARNING: Modifications to this ConfigMap may cause your AIOPs installation to become unstable.
  profiles: |
    generatedfor: HA
```

```
    configmaps:
    - name: aiops-topology-sizing
      data:
        asm: |
          file-observer:
            specs:
              replicas: 1
            containers:
              file-observer:
                resources:
                  requests:
                    cpu: 100m
                    memory: 750Mi
                  limits:
                    cpu: 1000m
                    memory: 1750Mi
                env:
                  - name: JVM_ARGS
                    value: -Xms300M -Xmx1400M
EOF
```

If IBM Sales representatives and Business Partners used the custom sizing tool to supply you with a custom profile ConfigMap to customize your deployment, then use the following steps to adjust the value of *JVM_ARGS* for the File observer to *-Xms300M -Xmx1400M*:

1. Log in to your Red Hat OpenShift cluster's console, and then from the **Project** list, select the project (namespace) that IBM Cloud Pak for AIOps is deployed in.
2. Navigate to **Workloads** > **ConfigMaps**.
3. Select the name of your custom profile, usually **aiops-custom-size-profile**, and then click **YAML** to switch to the YAML view. Adjust the value of *JVM_ARGS* for the File observer to *-Xms300M -Xmx1400M* and then click **Save**.

   Example excerpt:

   ```
   - name: aiops-topology-sizing
     data:
       asm: |
         file-observer:
           specs:
             replicas: 1
           containers:
             file-observer:
               resources:
                 requests:
                   cpu: 100m
                   memory: 750Mi
                 limits:
                   cpu: 1000m
                   memory: 1750Mi
               env:
                 - name: JVM_ARGS
                   value: -Xms300M -Xmx1400M
   ```

Health checks and monitoring (IBM Cloud Pak for AIOps on OpenShift)

To prepare your cluster for major operations, such as installing or upgrading IBM Cloud Pak for AIOps, and to maintain your system as you complete Day 2 operations, run health checks and use the available monitoring options to review the health of your IBM Cloud Pak for AIOps deployment and your Red Hat OpenShift Container Platform cluster.

By completing initial and routine health checks of your IBM Cloud Pak for AIOps deployment and cluster, you can gain the following benefits:

- **Ensure installation or upgrade readiness**

  The good health condition of your Red Hat OpenShift Container Platform cluster, storage, and services are prerequisites to ensure the success of any installation or upgrade of IBM Cloud Pak for AIOps. The prerequisites check can help to ensure the readiness of your cluster to proceed with an installation or upgrade.

- **Ensure your system stability**

  Routine health checks monitor your Red Hat OpenShift Container Platform cluster and services to ensure that your system is functioning properly and to identify any potential issues.

- **Maintain your system performance**

  Routine health checks can help to ensure that your IBM Cloud Pak for AIOps deployment and cluster runs optimally by detecting and addressing performance bottlenecks. Your administrators can maintain a historical record to support requests for resource increases.

- **Improve your system reliability**

  By monitoring the health of IBM Cloud Pak for AIOps, you can proactively identify and resolve issues that might potentially cause downtime or data loss.

For more information about running health checks and logging, see the following topics:

- Tools
- Running health checks
- Cluster logging

Tools

To help you with running initial and routine health checks, use the following tools and scripts:

- **Red Hat OpenShift Container Platform web console Monitoring dashboards**

  Red Hat OpenShift Container Platform provides different dashboards and capabilities that are accessible for the web console that can use to monitor and manage your system health.

- **IBM Cloud Pak for AIOps prerequisite checker tool**

  The prerequisite checker tool can be used to validate whether a Red Hat OpenShift Container Platform cluster has the required resources and prerequisites available to enable a successful installation or upgrade of IBM Cloud Pak for AIOps. This tool completes the following checks and generates an installation readiness report:

- **IBM Cloud Pak for AIOps custom sizing tool**

  IBM Sales representatives and Business Partners have access to a custom sizing tool that can assess your runtime requirements and provide a custom profile that scales IBM Cloud Pak for AIOps components. The custom profile is applied when you install IBM Cloud Pak for AIOps. This custom profile cannot be applied after installation, and attempting to do so can break your IBM Cloud Pak for AIOps deployment. By requesting your sales representative use this tool for helping you develop your custom profile, you can help to ensure that you have the allocated resources for handling your workload volumes.

- **IBM Cloud Pak for AIOps MustGather tool**

  The MustGather tool gathers data for IBM Cloud Pak for AIOps deployments. With this tool, you can complete data collection tasks and view analytics on the gathered data. The tool generates a report that shows critical data about the status and health of the cluster and installation. This report allows your to view an overview of your cluster to help you spot and solve problems. This report can provide you with basic cluster analysis to help identify cluster node resource utilization, warnings, and problematic resources. The report can also provide some basic incident analysis.

Running health checks

For guidance on running health checks for different use cases, review the following documentation:

- Installation and upgrade health checks

  Run prerequisite checks and health checks to ensure that your cluster is ready to begin an installation or upgrade of IBM Cloud Pak for AIOps.

- Proactive health checks

  Complete routine health checks and use the available monitoring capabilities to track the ongoing status of your deployment and cluster to maintain the health of your system.

- Reactive health checks

  When you encounter any issues with the health of your IBM Cloud Pak for AIOps deployment or Red Hat OpenShift Container Platform cluster, run the MustGather tool to gather data about the health and details of your deployment and cluster:

Self monitoring

Review the following different ways to monitor your IBM Cloud Pak for AIOps installation:

- Self-monitoring with alerts

  Since Cloud Pak for AIOps runs on OpenShift Container Platform, you can use the inbuilt Alertmanager feature of OpenShift Container Platform to monitor and send alerts to Cloud Pak for AIOps when a problem occurs in the underlying infrastructure.

- Self-monitoring with topology

  Create a topology visualization of your Cloud Pak for AIOps instance to monitor all the resources and their associated errors and warnings.

- Monitoring with Prometheus and Grafana

  Prometheus can serve as a monitoring system, while Grafana acts as a visualization tool to oversee your current installation. Prometheus and Grafana can be installed and configured on a Red Hat OpenShift Container Platform cluster.

- Monitoring data processed

  Monitor the volume of data that is processed by Cloud Pak for AIOps to see whether the OpenShift Container Platform cluster is sized properly.

Circuit breaker for Kafka robustness and stability

Equip IBM Cloud Pak for AIOps with the ability to protect itself and stay operational during an event storm by implementing a circuit breaker pattern. This helps prevent system failure while ensuring events, alerts, and incidents are still processed.

For more information, see Circuit breaker for Kafka robustness and stability.

Cluster logging

As a cluster administrator, you can deploy centralized logging on an OpenShift Container Platform cluster, and use it to collect and aggregate node system audit logs, application container logs, and infrastructure logs. You can forward logs to your chosen log solution, including on-cluster, and Red Hat managed log storage. You can also visualize your log data in the OpenShift Container Platform web console, the Kibana web console, or 3rd party log aggregator depending on your deployed log storage solution.

Note: The Kibana web console is deprecated.

OpenShift Container Platform cluster administrators can deploy logging by using Operators.

For information, see About logging.

Installation and upgrade health checks

To prepare your cluster for installing or upgrading IBM Cloud Pak for AIOps, and to ensure the health of your deployment afterwards, you can run prerequisite checks and health checks.

Install readiness checks

To help ensure that your cluster is ready to begin an installation of IBM Cloud Pak for AIOps, run the following health check:

- Request that your IBM Sales representative (or Business Partner) uses the IBM Cloud Pak for AIOps custom sizing tool to create a sizing profile to meet your operation workload needs. By having this profile before you begin an installation, you can help to ensure that you have the allocated resources for handling your workload volumes. Contact your IBM Sales representative (or Business Partner) for more information about this tool. For more information about custom sizing, see Hardware requirements - Custom sizing.
- After you prepare your storage and Red Hat OpenShift Container Platform cluster, install and run the IBM Cloud Pak for AIOps prerequisite checker tool. Running this tool verifies whether your Red Hat OpenShift cluster is correctly set up for an IBM Cloud Pak for AIOps installation. For more information about the script, and how to download and run it, see github.com/IBM.

  For more information about running the prerequisite checker during an installation, see:

  - Online production installation of IBM Cloud Pak for AIOps (CLI method)
  - Online production installation of IBM Cloud Pak for AIOps (console method)
  - Online starter installation of IBM Cloud Pak for AIOps (CLI)

Post-installation health check

After you complete your installation of IBM Cloud Pak for AIOps and some time is passed, run the MustGather tool with the **`healthcheck`** option to check the current health of your deployment and Red Hat OpenShift Container Platform cluster.

For more information about installing and running this tool, see:

- Installing the MustGather tool

  Important: Ensure that your installed version of the tool is **`v1.17.6`** or newer. Some healthcheck data is only gathered with version **`v1.17.6`** and newer.

- Running the MustGather tool (healthcheck)

  When running the tool, use the following command:

  ```
  waiops-mustgather.sh -DO healthcheck
  ```

Upgrade readiness checks

To help ensure that your cluster is ready to begin an upgrade of IBM Cloud Pak for AIOps, run the following health check:

- Run the IBM Cloud Pak for AIOps prerequisite checker tool. Running this tool verifies whether your Red Hat OpenShift cluster is correctly set up for upgrading IBM Cloud Pak for AIOps. For more information about the script, and how to download and run it, see github.com/IBM.

  For more information about running the prerequisite checker during an upgrade, see:

  - Online upgrade of IBM Cloud Pak for AIOps (console method)
  - Online upgrade of IBM Cloud Pak for AIOps (CLI method)
  - Offline upgrade of IBM Cloud Pak for AIOps

Post-upgrade health check

- After you complete your upgrade of IBM Cloud Pak for AIOps and some time is passed, run the MustGather tool with the **`healthcheck`** option to check the current health of your deployment and Red Hat OpenShift Container Platform cluster.

For more information about installing and running this tool, see:

- Installing the MustGather tool

Important: Ensure that your installed version of the tool is `v1.17.6` or newer. Some healthcheck data is only gathered with version `v1.17.6` and newer.

- Running the MustGather tool (healthcheck)

  When running the tool, use the following command:

  ```
  waiops-mustgather.sh -DO healthcheck
  ```

Proactive health checks and monitoring

To maintain your system as you complete Day 2 operations, review the following guidance for completing routine health checks and for monitoring the ongoing status and health of your IBM Cloud Pak for AIOps deployment and your Red Hat OpenShift Container Platform cluster.

Monitor Red Hat OpenShift Container Platform

Routinely check the Red Hat OpenShift Container Platform Monitoring dashboards in the web console. Red Hat OpenShift Container Platform includes a preconfigured, preinstalled, and self-updating monitoring stack that provides monitoring for core platform components. Red Hat OpenShift Container Platform also provides a comprehensive set of monitoring dashboards that help you understand the state of cluster components and user-defined workloads.

- Use the Administrator perspective to access the monitoring dashboards to monitor the following components:
  - API performance
  - etcd
  - Kubernetes compute resources
  - Kubernetes network resources
  - Prometheus
  - Cluster and node performance (USE method dashboards)

- Use the Developer perspective to access Kubernetes compute resources dashboards to view application metrics for monitoring the following status:
  - CPU usage
  - Memory usage
  - Bandwidth information
  - Packet rate information

For more information about Red Hat OpenShift Container Platform Monitoring and the monitoring dashboards, see:

- About Red Hat OpenShift Container Platform Monitoring.
- Reviewing monitoring dashboards
- Configure and use the Red Hat OpenShift Container Platform Alerting capabilities to monitor your cluster and identify potential problems. With the web console Alerting UI you can manage alerts, silences, and alerting rules to monitor the health of your cluster. For more information, see Managing alerts.

Monitor IBM Cloud Pak for AIOps

For more information about self-monitoring with alerts, see Self-monitoring with alerts.

Run the MustGather health check

Run the MustGather tool with the `healthcheck` option to check the current health of your deployment and Red Hat OpenShift Container Platform cluster.

For more information about installing and running this tool, see the following topics:

- Installing and running the MustGather tool (Linux)
- Installing the MustGather tool (OpenShift)

  Important: If you are using the (OpenShift) version, ensure that your installed version of the tool is `v1.17.6` or newer. Some healthcheck data is only gathered with version `v1.17.6` and newer.

- Running the MustGather tool (healthcheck)

  When running the tool, use the following command:

  ```
  waiops-mustgather.sh -DO healthcheck
  ```

Reactive health checks and troubleshooting

When you encounter any issues with the health of your IBM Cloud Pak for AIOps deployment or Red Hat OpenShift Container Platform cluster, run a health check with the MustGather tool to gather data to analyze to help you identify problems.

Run the MustGather tool

For more information about installing and running this tool, see:

- Installing the MustGather tool
- Running the MustGather tool (healthcheck)

  When running the tool, use the following command:

  ```
  waiops-mustgather.sh -DO healthcheck
  ```

Keep a copy of the generated logs in case you need to open a support case with IBM Support. As part of opening a case, provide the MustGather logs to IBM Support.

Review known issues and troubleshooting

In addition to running the MustGather health check, review the troubleshooting and known issues for IBM Cloud Pak for AIOps to determine whether your problem is already known and if a solution or workaround is identified.

- Known issues and limitations
- Troubleshooting

Self-monitoring with alerts

You can use Red Hat® OpenShift® Container Platform inbuilt Alertmanager to monitor and send alerts to Cloud Pak for AIOps when a problem occurs in the underlying infrastructure.

Cloud Pak for AIOps can ingest events from multiple sources and carry out analytics on them for you to reduce the noise and focus on the events that matter the most in your network. What if Cloud Pak for AIOps itself or the infrastructure that it is running on has a looming problem that might result in losing your event management system?

Since Cloud Pak for AIOps runs on OpenShift Container Platform, you can use the inbuilt Alertmanager feature of OpenShift Container Platform to do the monitoring and configure it to send alerts to Cloud Pak for AIOps when it sees a problem with any of the underlying infrastructure. Examples of problems include pod restarts, low storage, or latency issues, which can cause the loss of the event management system.

Note: Alertmanager sends you initial warning alerts and then escalates those to Critical as the issue becomes more serious. For example, when storage passes a threshold, you initially get a warning alert, but as the threshold goes to 90% an alert becomes Critical.

Complete the following steps to configure the monitoring by using the Alertmanager:

1. Create a Generic Webhook integration.

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, search **Generic Webhook** and click the tile to set up a new webhook ingestion endpoint.

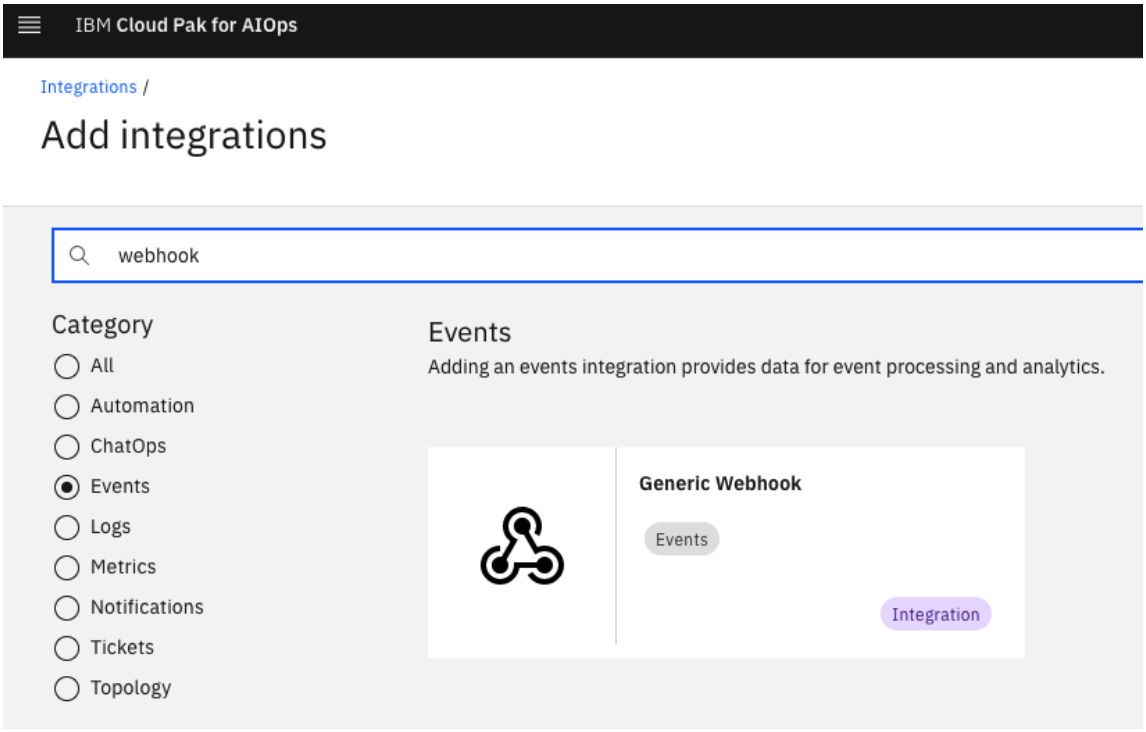


Figure. Generic Webhook integration

5. In the *Add Integration* dialog box, provide a name for the integration and select **None** as the **Authentication type** and click **Next**.

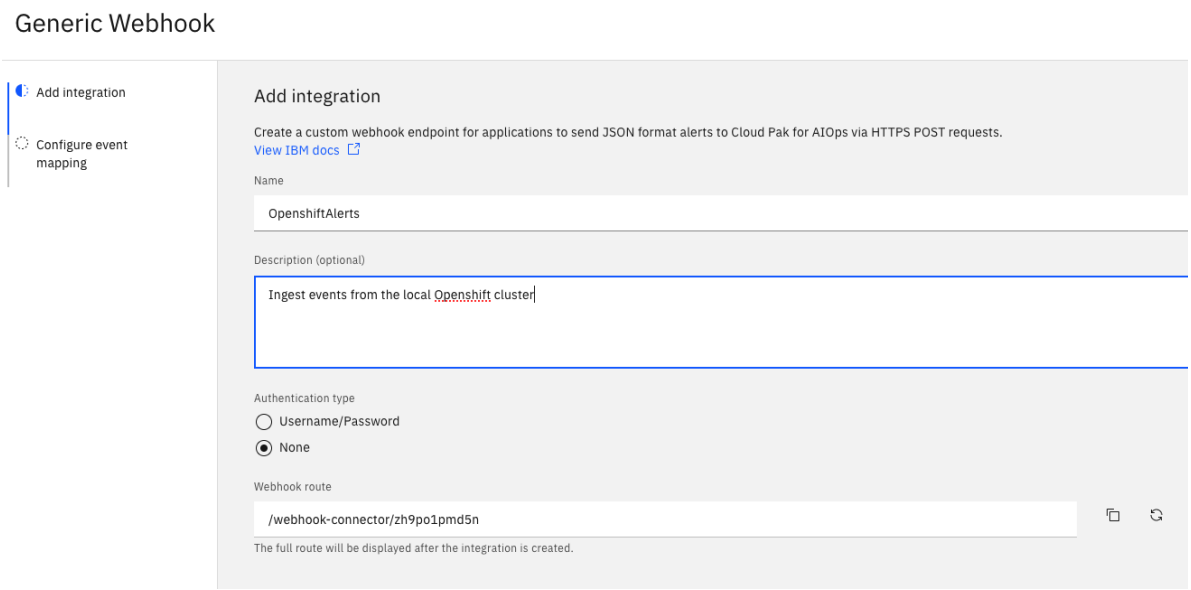


Figure. Add integration page

6. Click **Load sample mapping** and select **Prometheus Alert Manager** to configure the integration with the correct schema mapping from Prometheus to Cloud Pak for AIOps alerts. Click **Done**.

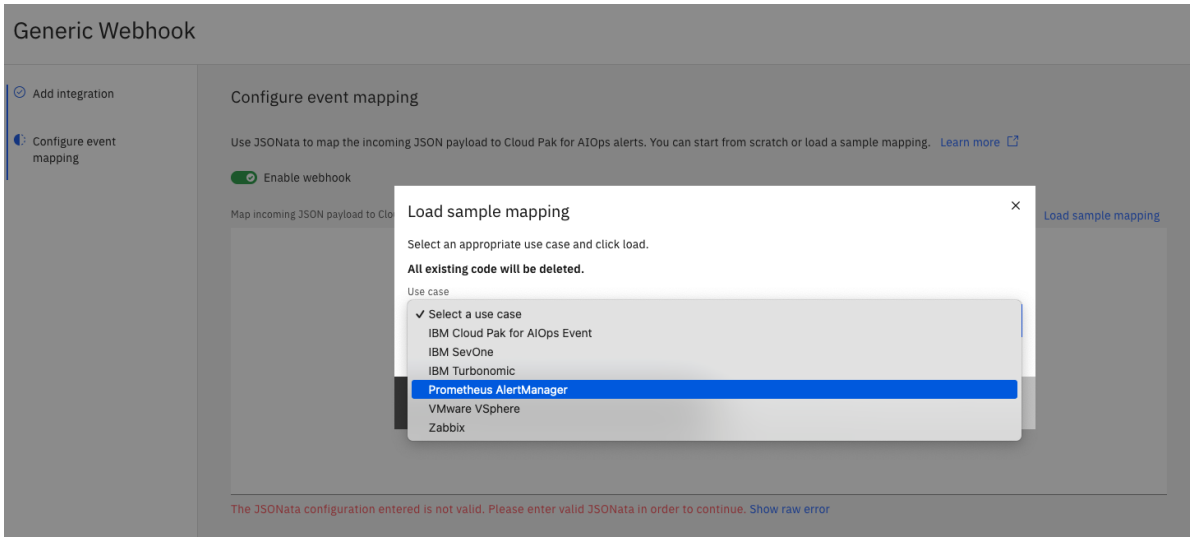


Figure. Configure event mapping

2. Copy the webhook URL to the Red Hat® OpenShift® Alertmanager user interface.

1. From the new Generic Webhook integration that you created, copy the full webhook URL and switch to the OpenShift Container Platform console. An example URL looks like **`https://whconn-6492ea38-cf72-4f3d-b7d3-9fbbc2b1065c-aiops.apps.example.com/webhook-connector/ezca4fgo2tj`**.
2. In the OpenShift Container Platform console go to **Administration** > **Cluster Settings** > **Configuration** > **Alertmanager**.
3. Edit the *Receivers* section for *Critical* and *Default* **Receiver name**.

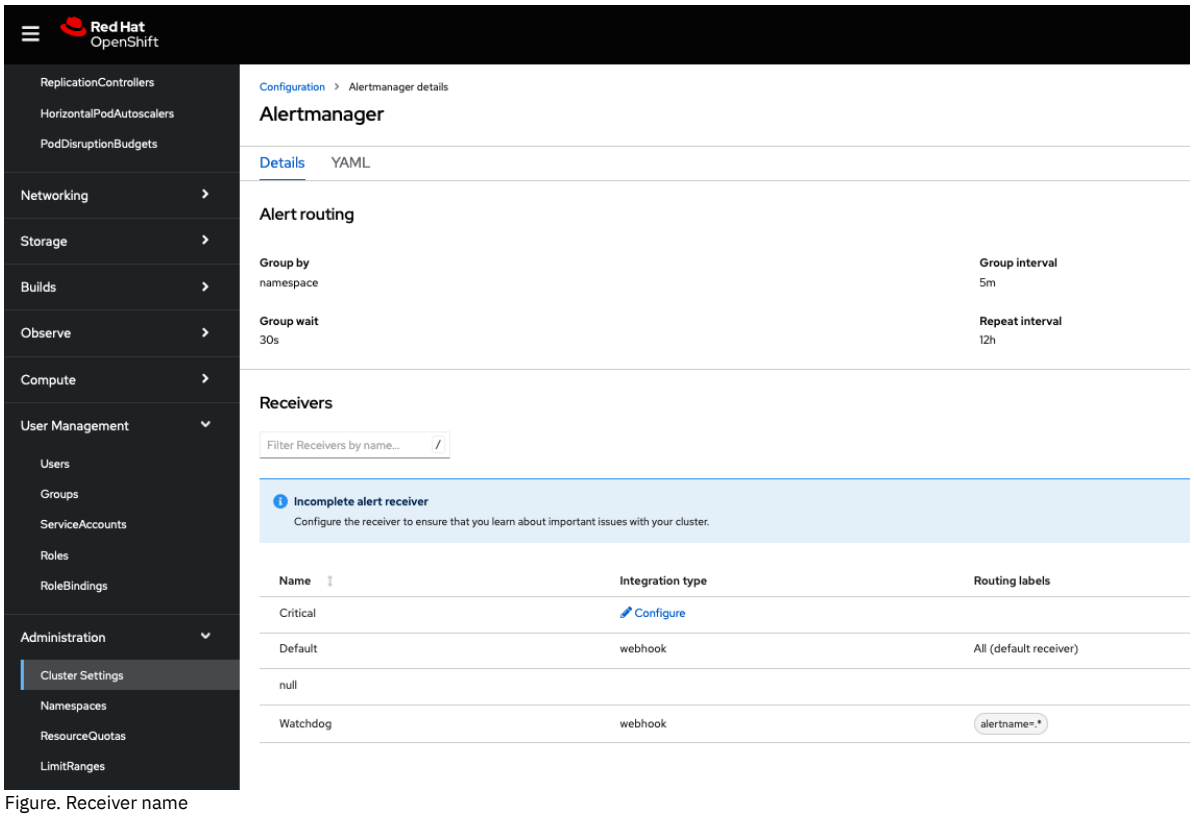


Figure. Receiver name

4. Select **Webhook** for the **Receiver Type** field and paste the full URL into the **URL** field.

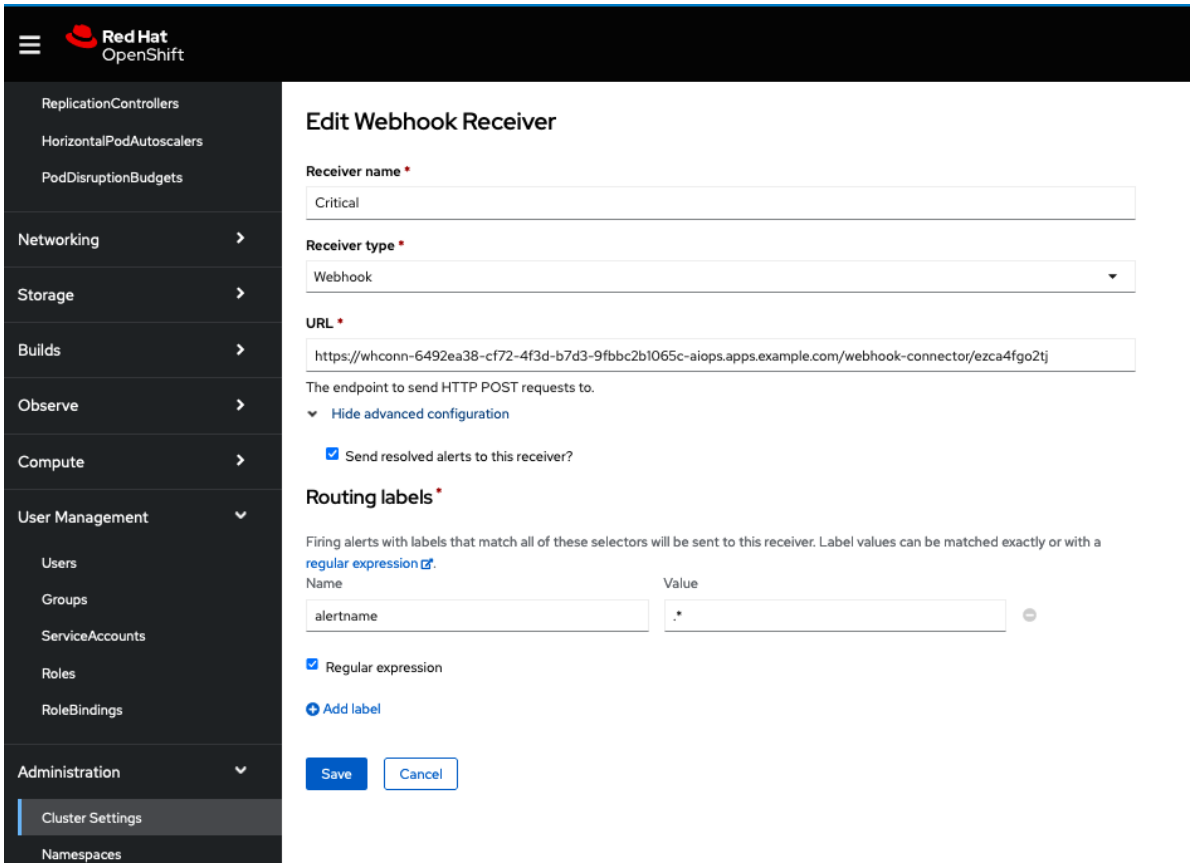


Figure. Webhook receiver

3. Configure Alertmanager to send the alerts you are interested in.

Use the Routing labels section to further filter the alerts that get sent to Cloud Pak for AIOps.

The preceding sample figure (with caption Webhook receiver) sends all alerts by using the value of `.*`.

In the *Routing labels* section of the Webhook receiver, you can use **`alertname`** and a value of `.*` to allow all alerts through, or you can filter by using, for instance, alertname with a value of **SystemMemoryExceedsReservation** or any other regular expression that gives you the set of alerts that you want to manage in Cloud Pak for AIOps. When an OpenShift Container Platform alert is generated it contains *Labels* attached to it as shown in the following example output:

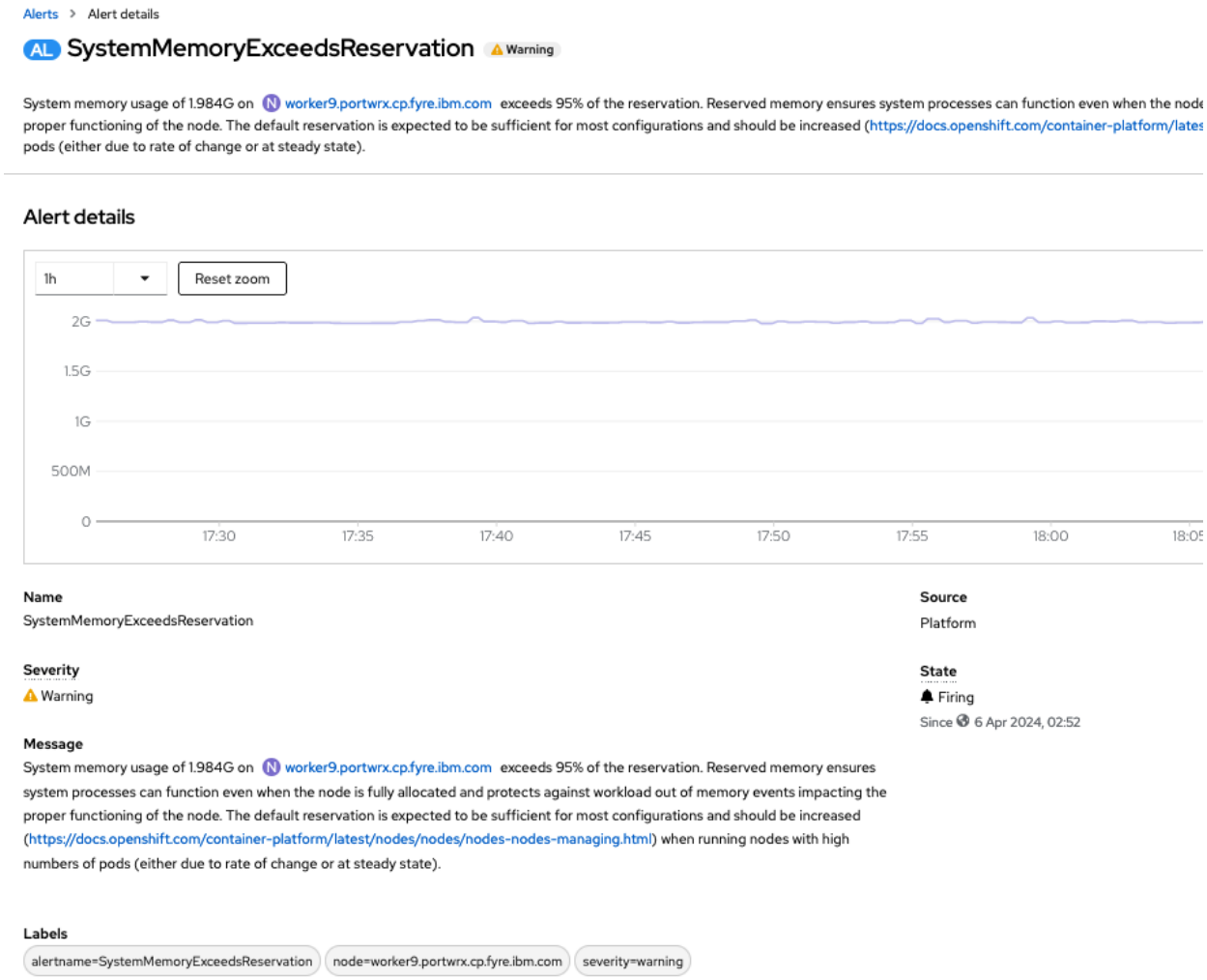


Figure. Labels

When you save the Alertmanager configuration, any existing active alerts appear in your Alert Viewer.

Optionally, you can also create an Cloud Pak for AIOps policy to scope group these Prometheus based alerts if you want those to be grouped in one Cloud Pak for AIOps incident.

The following simple example policy provides a scope-based grouping:

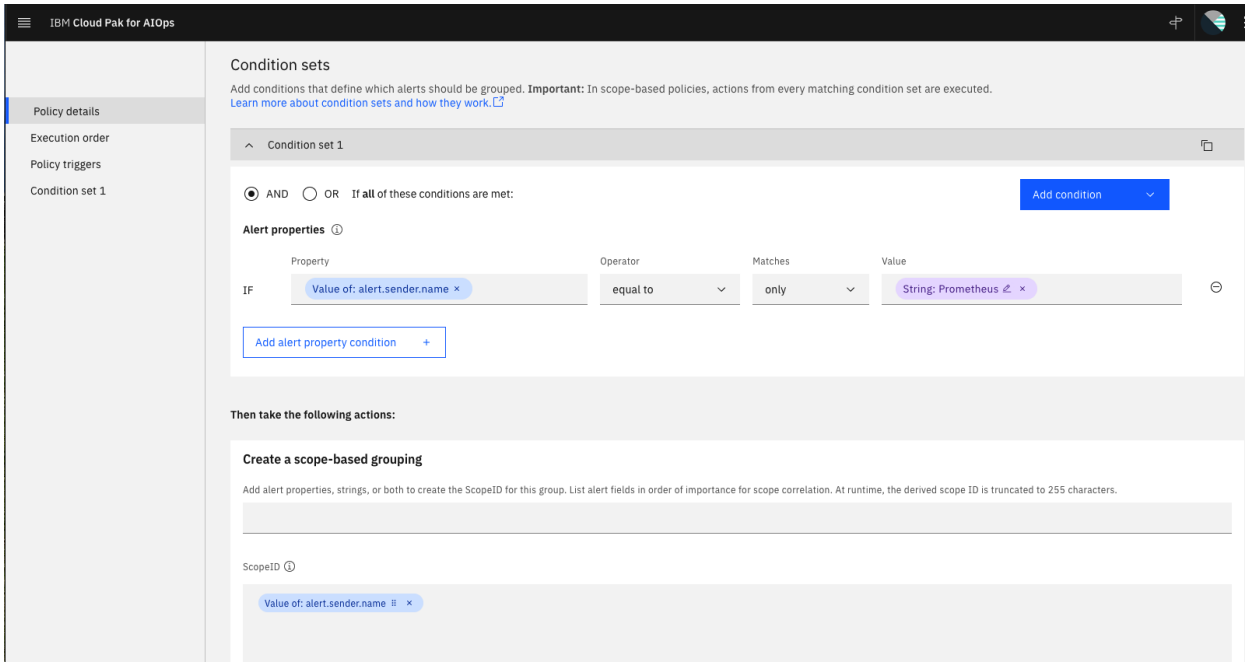


Figure. Scope-based grouping

Note: Alertmanager requires a trusted webhook endpoint. If your Cloud Pak for AIOps deployment is not using certs that are trusted by OpenShift itself, it cannot send the alerts to Cloud Pak for AIOps. You need to use trusted certs or add the Cloud Pak for AIOps CA signing cert to the OpenShift CA bundle. For more information about certs, see Updating the CA bundle.

Self-monitoring with topology

You can create the topology visualization of your Cloud Pak for AIOps instance to monitor the errors and warnings and to view all the resources, which make up the Cloud Pak for AIOps installation.

With the topology, see how different aspects of the estate are related to one another. Also, use it for more advanced analytics, such as topology-based alert correlation, probable cause determination, inventory management, and more.

Since Cloud Pak for AIOps runs on Red Hat OpenShift Container Platform, you need a single observer to get started, namely the *Kubernetes observer*. The Kubernetes observer works with many of the Kubernetes-based platforms, such as Red Hat OpenShift and Google Kubernetes Engine.

To configure a Kubernetes integration, you need the following information:

- Kubernetes cluster or data center name
- Namespace where Cloud Pak for AIOps is installed
- Kubernetes master IP address
- Kubernetes API port
- Kubernetes token

Complete the following steps to configure the self-monitoring by using the Kubernetes integration:

1. Create and use a Service Account.

To create and use a Service Account, follow the steps that are listed under the topic Kubernetes Observer Load job prerequisites.

2. Configure the observer.

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars). Then, click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, search **Kubernetes** and click the **Kubernetes** tile.
5. Click **Get started**.
6. Select **Load** as **job type**.
7. Enter the values for the **Unique ID**, **data_center**, **Kubernetes Master IP Address**, **Kubernetes API port**, and **Kubernetes token** fields.

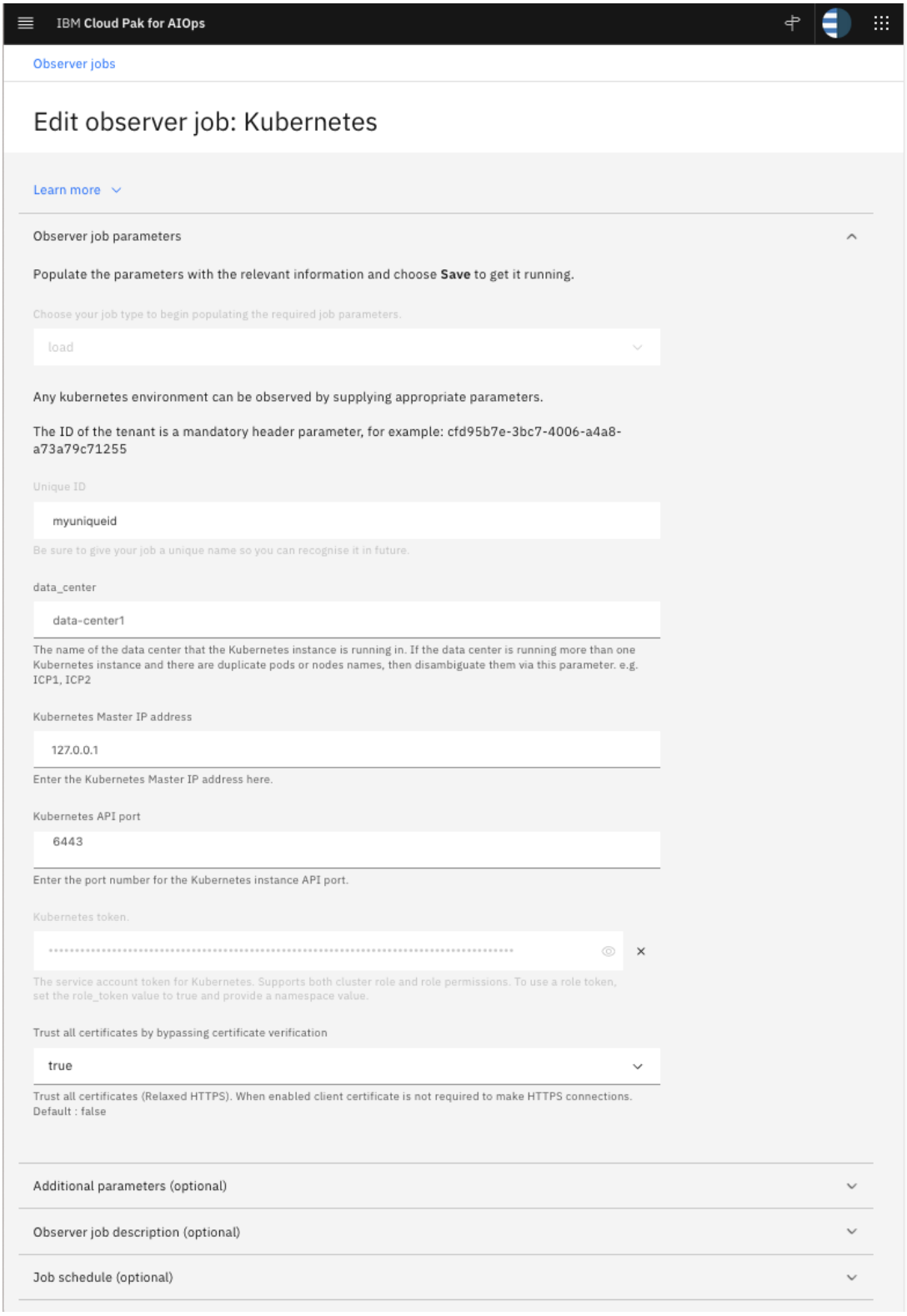


Figure. Kubernetes observer job

8. Under the *additional parameters (optional)* tab, enter the namespace where Cloud Pak for AIOps is installed so you observe only the namespace and not the other applications installed on the cluster.

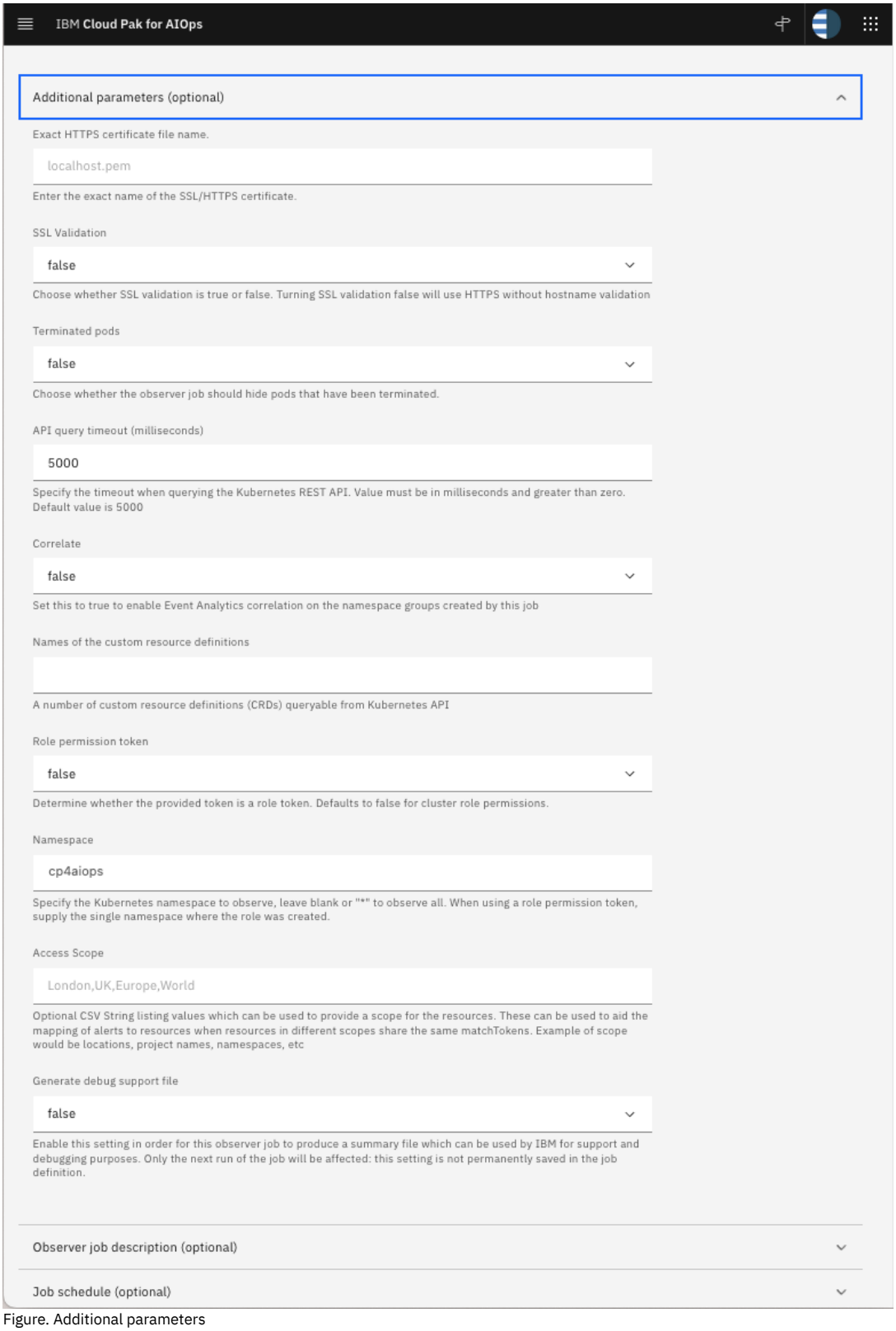


Figure. Additional parameters

9. Under the *Job schedule (optional)* tab, indicate how frequently you want to refresh the information. For example, refresh every 5 minutes.

For more information about all the fields, see Configuring Kubernetes Observer jobs.

3. View the topology.

When the observer job runs for the first time, it creates a resource group by using the name that is provided in the observer **data_center** field.

The observer brings in the alerts, errors, and warnings, which are displayed under the *Resource groups* tab.

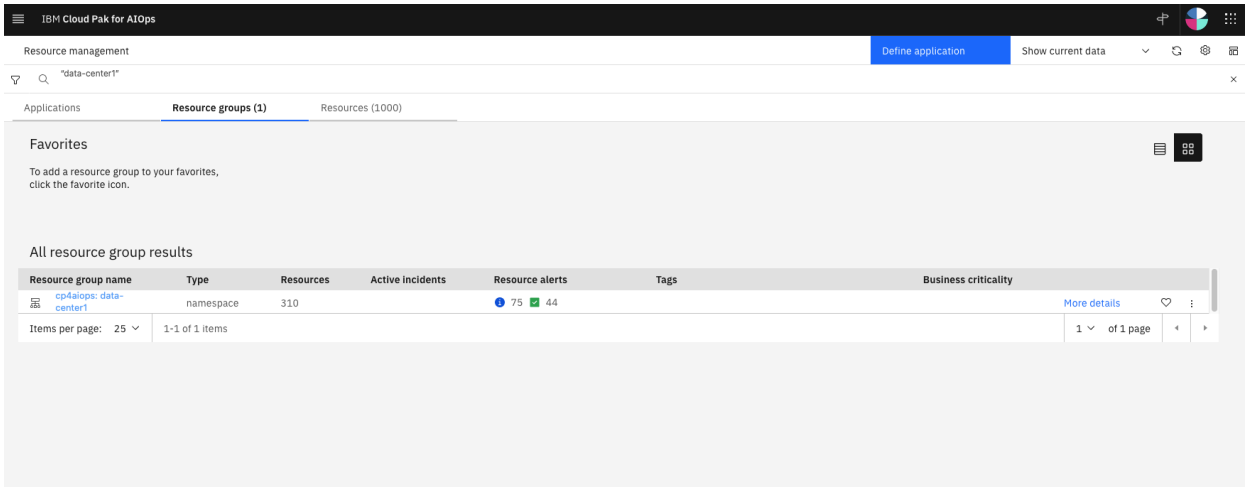


Figure. Resource group results

When you click the resource group name that is created by the observer job, you can view all the resources, which make up the Cloud Pak for AIOps installation. If the cluster is updated, or the Cloud Pak for AIOps is updated, the changes are shown as displayed in the following example.

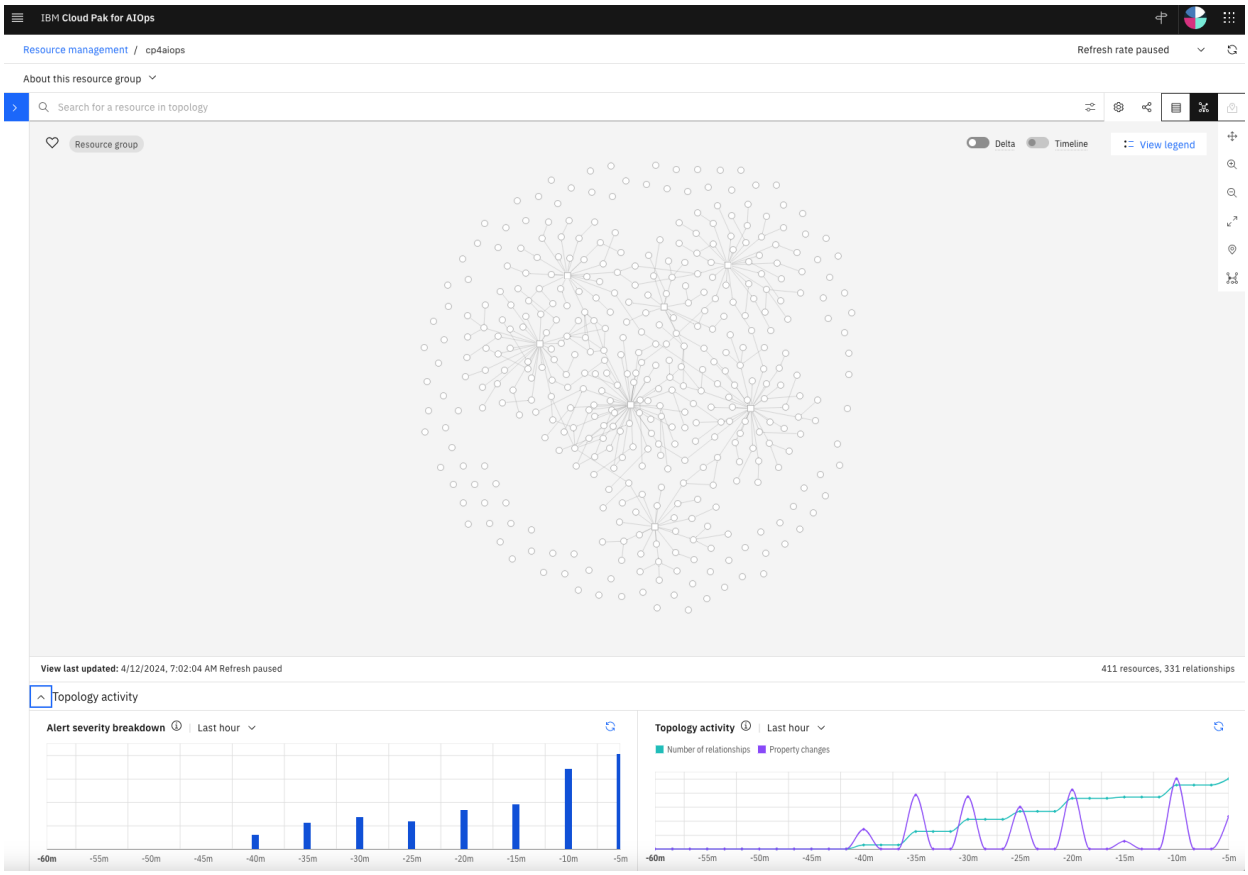


Figure. Resource group in Resource management

4. Create topology resource groups.

The view of the entire set of resources for Cloud Pak for AIOps is useful, but it might be useful to look at specific portions. For example, you can view each node to see the workload of the node and any issues associated with that node.

To create groups for each of the nodes, you can use the dynamic resource group template. For more information, see Building resource group templates.

Use the following steps to create topology resource groups:

1. From the main navigation of the Cloud Pak for AIOps console, click **Resource management**.
2. Click **Resource group templates** icon.
3. Click **Create a new template** > **Dynamic template**.
4. Click **Start**. The Template builder page appears.

5. In the search box, enter the resource name, for example, **`worker1.data-center1.example.com`**, and click the desired resource.

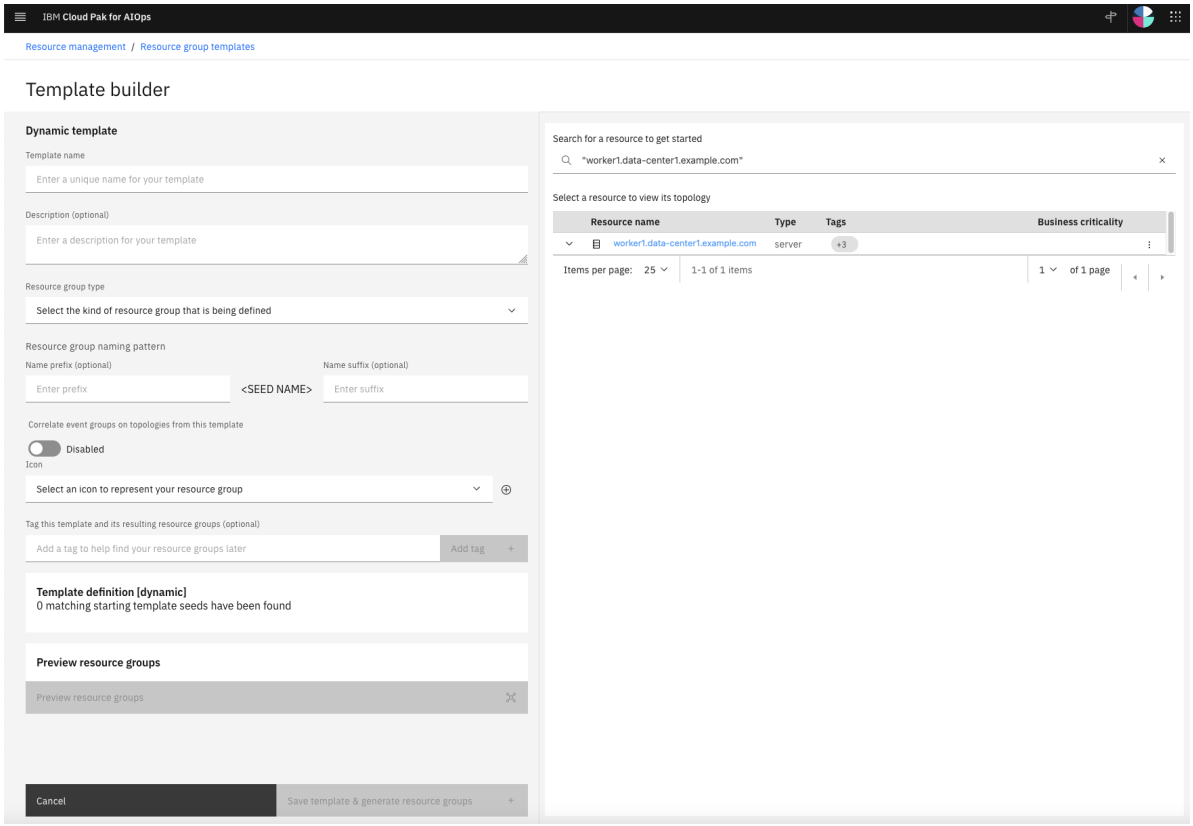


Figure. Template builder

6. To see both the node and the associated workload, right-click the large node image as shown in the following example, and select **Get Neighbors**.

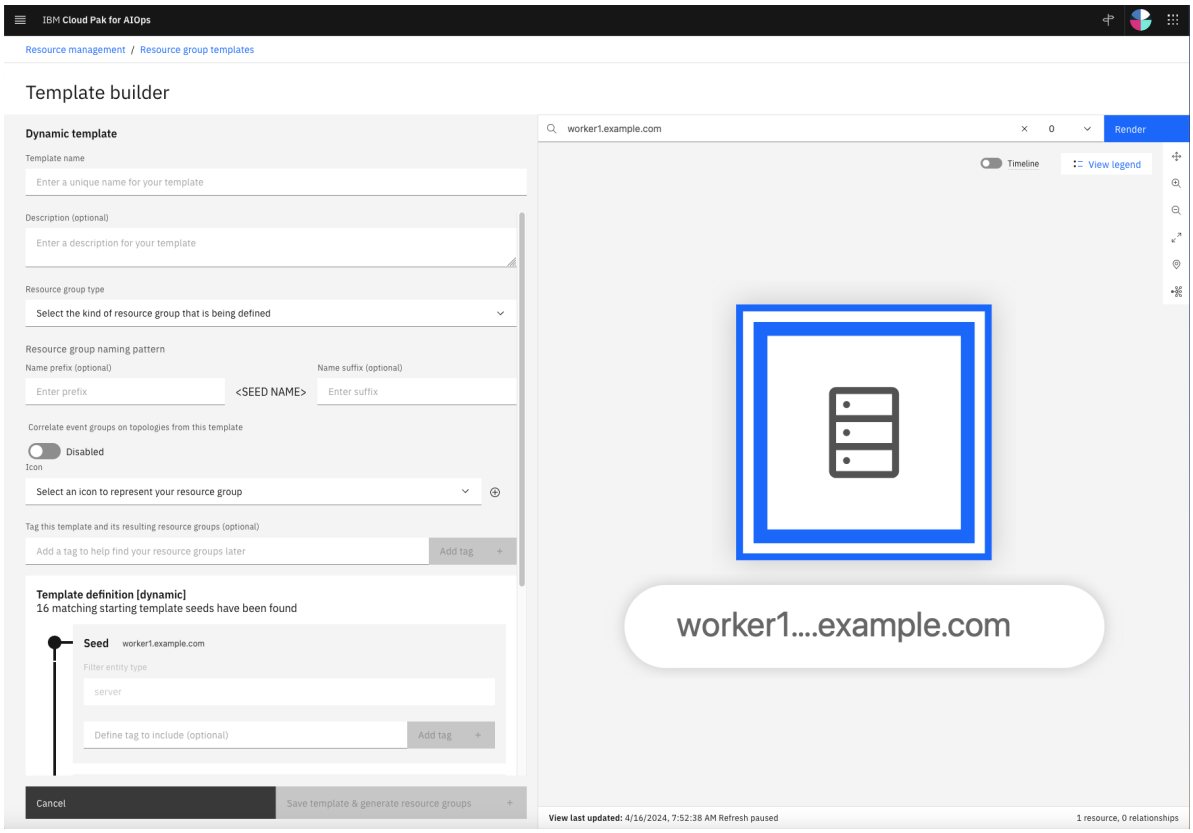


Figure. Render template

When you hover over the large node image, you see the full name of the resource and the type of the resource.

7. Click **All**. This option shows the pods that are running on the node and the information such as the IP address.

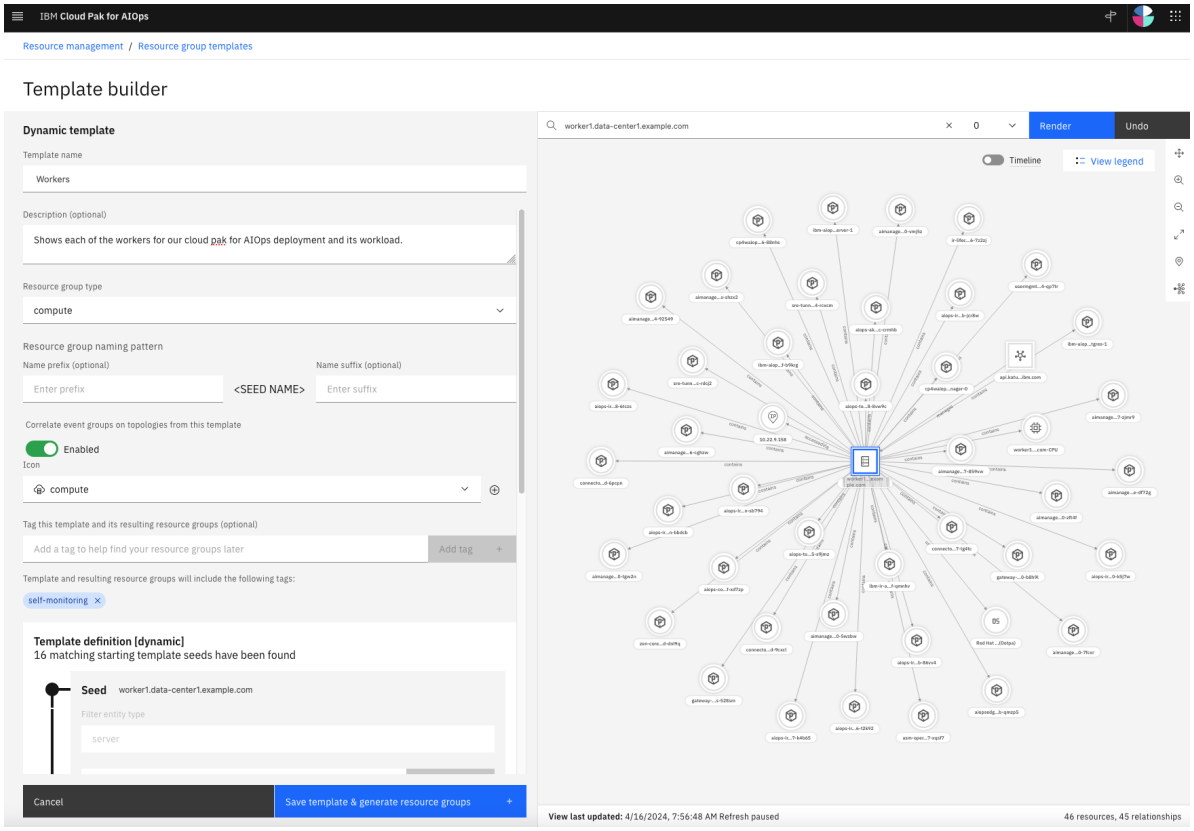


Figure. Workload of node

8. Provide a Template name and description.
9. Select the resource group type. For example, **compute**.
10. Optionally, add a tag name. For example, **self-monitoring**. Adding a tag helps with the filtering of results that are listed under the resource groups.
11. Click **Save template & generate resource groups**.

Cloud Pak for AIOps creates the template and automatically creates the resource groups for all the nodes, each showing the workload as seen from the following example.

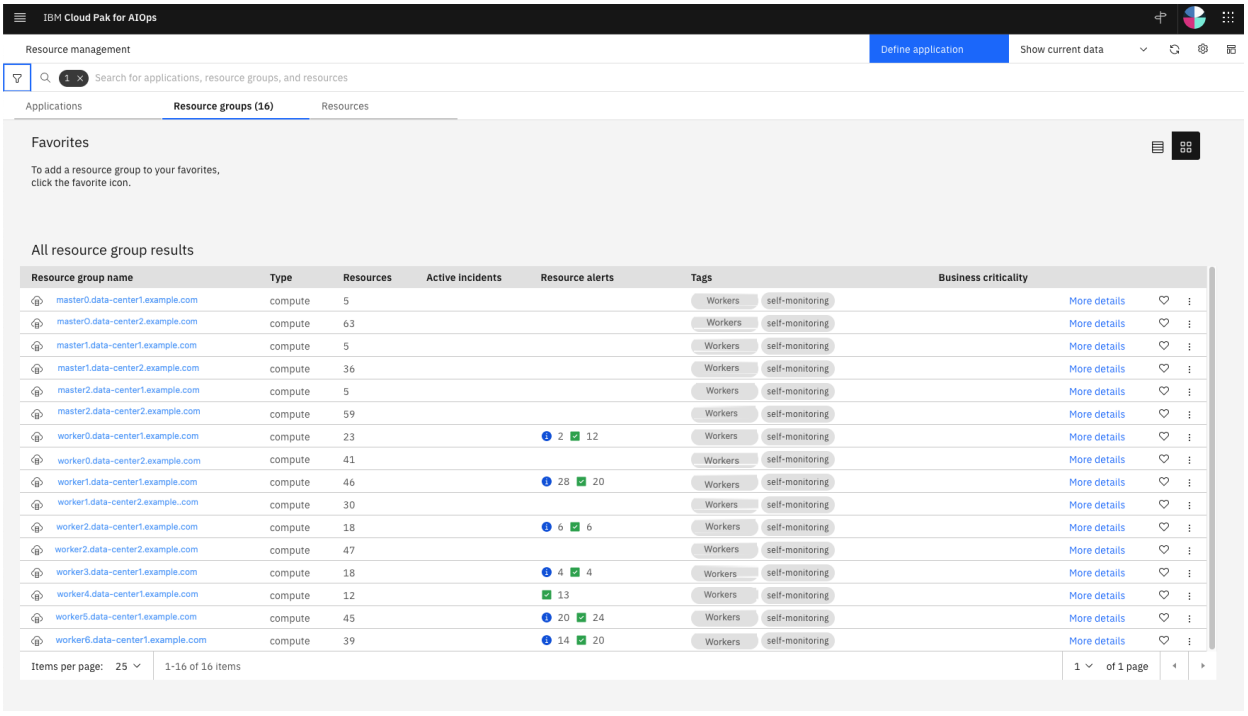


Figure. Resource group results

Monitoring with Prometheus and Grafana

You can use Prometheus as a monitoring system and Grafana as a visualization tool to monitor your existing installation of IBM Cloud Pak for AIOps. Prometheus and Grafana can be installed and configured on a Red Hat OpenShift Container Platform cluster.

Prometheus is an open-source toolkit that can be used with Grafana to create real-time dashboards for monitoring Cloud Pak for AIOps stability and performance.

Note: The following instructions are updated to support the Grafana v5 operator, which is compatible with OpenShift version 4.11.x or higher. Previously, these instructions used the Grafana v4 operator, which is not supported by OpenShift version 4.16.x or higher. If you followed the instructions for the Grafana v4 operator, there might be compatibility issues if you upgrade your OpenShift version to 4.16.x or higher. To check your version of the Grafana operator, go to your OpenShift console, **Operators** > **Installed Operators**. To upgrade your Grafana operator from v4 to v5, run the cleanup script, and follow the instructions detailed in the Viewing metrics in Grafana section.

Warning: The clean up script deletes any existing dashboards in Grafana. Copy the JSON of any existing dashboards that you want to keep.

Clean up script (optional):

```
oc project monitoring grafana
# delete CRs
oc delete grafana grafana
oc delete grafanadatasource prometheus

# delete subscription & CSV
CSV=$(oc get subscription monitoring-grafana -o json | jq -r '.status.installedCSV')
oc delete subscription monitoring-grafana
oc delete csv $CSV

# delete CRDs
# IMPORTANT: if you created any additional CRD instances, make sure to delete them before deleting the definition
oc delete crd grafanadashboards.integreatly.org
oc delete crd grafanadatasources.integreatly.org
oc delete crd grafanafolders.integreatly.org
oc delete crd grafananotificationchannels.integreatly.org
oc delete crd grafanas.integreatly.org

# delete the rest of v4 associated resources
oc delete clusterrole grafana-proxy
oc delete rolebinding grafana-proxy
oc delete secret grafana-k8s-proxy
```

Notes:

- The monitoring stack that is described in the following sections uses 3rd party components that are not owned by IBM, use with caution.
- The monitoring stack that is described in the following sections currently works with Cloud Pak for AIOps that is installed in OpenShift Container Platform cluster. Cloud Pak for AIOps installed in an air-gapped environment (offline) is not supported by the described monitoring stack.

Prerequisites

- You need network connectivity to the OpenShift Container Platform cluster where Cloud Pak for AIOps is installed.
- OpenShift command-line interface is required. For more information, see Getting started with the OpenShift CLI.
- You need to install jq for your operating system to run the commands for JSON processing. For more information, see Download jq.

Use the following sections to set up the monitoring for your Cloud Pak for AIOps environment:

- Setting up user workload monitoring
- Enabling additional metrics
- Viewing metrics in Grafana
- Importing dashboards

Setting up user workload monitoring

OpenShift Container Platform comes with a prometheus-based monitoring stack to track metrics about the cluster. You can extend the monitoring to user workloads by enabling user workload monitoring. User workloads include anything that is not a core OpenShift Container Platform service. For more information about user workloads monitoring and advanced configuration, see User workload monitoring.

To set up the user workload monitoring capability, you need to enable it.

Note: Once you run the following commands, the associated ConfigMaps (configuration settings) are applied cluster-wide. Do not run the commands on a shared cluster that includes applications other than Cloud Pak for AIOps.

Log in to your OpenShift Container Platform cluster by using the command line. Run the following commands to enable the user workload monitoring capability:

```
oc create -n openshift-monitoring -f - <<EOF || true
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/created-by: IBM
  name: cluster-monitoring-config
  namespace: openshift-monitoring
data:
  config.yaml: |
```

```
    enableUserWorkload: true
EOF

oc create -n openshift-user-workload-monitoring -f - <<EOF || true
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/created-by: IBM
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
EOF
```

After you run the preceding commands, you can view metrics in OpenShift Container Platform web console.

Viewing metrics in OpenShift Container Platform cluster

To view the metrics, go to **Observe** > **Metrics** in OpenShift Container Platform web console. You can enter a valid Prometheus query and graph the result. The following are some example queries that you can use to view the metrics for your Cloud Pak for AIOps environment:

1. The total number of pods that are in a running state across all namespaces:

   ```
   sum(kube_pod_status_phase{namespace=~".*", phase="Running"})
   ```

2. The total amount of memory that is allocated across all the nodes:

   ```
   sum (machine_memory_bytes{node=~"^.*$"})
   ```

3. Deployment replicas across all the namespaces:

   ```
   kube_deployment_status_replicas{namespace=~".*"}
   ```

Enabling additional metrics

In some cases you might need configuration to enable additional metrics for your Cloud Pak for AIOps environment.

Kafka

Since a large portion of communication within Cloud Pak for AIOps takes place in Kafka, it is highly beneficial to acquire metrics from Kafka.

Notes:

- Kafka metrics, such as those displayed in the usage dashboard, are only available from the moment the metrics are enabled. Viewing time windows that precede the enabling of the metrics results in nothing being displayed.
- The Kafka custom resource (CR) must be in a ready state for metric scraping to be enabled successfully. To view the status of the Kafka CR, navigate to **Home** > **Search**. Search for the **Kafka resource** and click on **Kafka instance**. The **Conditions** section at the end of the page shows the `Status`.

Use the following steps to enable Kafka metrics:

1. Switch to the project (namespace) where Cloud Pak for AIOps is installed:

   ```
   oc project <namespace>
   ```

   Where `<namespace>` is the project (namespace) where Cloud Pak for AIOps is installed

2. Define the Kafka `ConfigMap` and `PodMonitor` resources to configure the Kafka Prometheus exporter:

   1. Define the Kafka `ConfigMap` resource:

      ```
      KAFKA_METRICS_CONFIGMAP='
      kind: ConfigMap
      apiVersion: v1
      metadata:
        name: kafka-metrics
        labels:
          app: strimzi
      data:
        kafka-metrics-config.yml: |
          # See https://github.com/prometheus/jmx_exporter for more info about JMX Prometheus Exporter metrics
          lowercaseOutputName: true
          rules:
          # Special cases and very specific rules
          - pattern: kafka.server<type=(.+), name=(.+), clientId=(.+), topic=(.+), partition=(.*)><>Value
            name: kafka_server_$1_$2
            type: GAUGE
            labels:
             clientId: "$3"
             topic: "$4"
             partition: "$5"
          - pattern: kafka.server<type=(.+), name=(.+), clientId=(.+), brokerHost=(.+), brokerPort=(.+)><>Value
            name: kafka_server_$1_$2
            type: GAUGE
            labels:
             clientId: "$3"
             broker: "$4:$5"
          - pattern: kafka.server<type=(.+), cipher=(.+), protocol=(.+), listener=(.+), networkProcessor=(.+)>
      ```

```
<>connections
      name: kafka_server_$1_connections_tls_info
      type: GAUGE
      labels:
        cipher: "$2"
        protocol: "$3"
        listener: "$4"
        networkProcessor: "$5"
    - pattern: kafka.server<type=(.+), clientSoftwareName=(.+), clientSoftwareVersion=(.+), listener=(.+),
networkProcessor=(.+)><>connections
      name: kafka_server_$1_connections_software
      type: GAUGE
      labels:
        clientSoftwareName: "$2"
        clientSoftwareVersion: "$3"
        listener: "$4"
        networkProcessor: "$5"
    - pattern: "kafka.server<type=(.+), listener=(.+), networkProcessor=(.+)><>(.+):"
      name: kafka_server_$1_$4
      type: GAUGE
      labels:
       listener: "$2"
       networkProcessor: "$3"
    - pattern: kafka.server<type=(.+), listener=(.+), networkProcessor=(.+)><>(.+)
      name: kafka_server_$1_$4
      type: GAUGE
      labels:
       listener: "$2"
       networkProcessor: "$3"
    # Some percent metrics use MeanRate attribute
    # Ex) kafka.server<type=(KafkaRequestHandlerPool), name=(RequestHandlerAvgIdlePercent)><>MeanRate
    - pattern: kafka.(\w+)<type=(.+), name=(.+)Percent\w*><>MeanRate
      name: kafka_$1_$2_$3_percent
      type: GAUGE
    # Generic gauges for percents
    - pattern: kafka.(\w+)<type=(.+), name=(.+)Percent\w*><>Value
      name: kafka_$1_$2_$3_percent
      type: GAUGE
    - pattern: kafka.(\w+)<type=(.+), name=(.+)Percent\w*, (.+)=(.+)><>Value
      name: kafka_$1_$2_$3_percent
      type: GAUGE
      labels:
        "$4": "$5"
    # Generic per-second counters with 0-2 key/value pairs
    - pattern: kafka.(\w+)<type=(.+), name=(.+)PerSec\w*, (.+)=(.+), (.+)=(.+)><>Count
      name: kafka_$1_$2_$3_total
      type: COUNTER
      labels:
        "$4": "$5"
        "$6": "$7"
    - pattern: kafka.(\w+)<type=(.+), name=(.+)PerSec\w*, (.+)=(.+)><>Count
      name: kafka_$1_$2_$3_total
      type: COUNTER
      labels:
        "$4": "$5"
    - pattern: kafka.(\w+)<type=(.+), name=(.+)PerSec\w*><>Count
      name: kafka_$1_$2_$3_total
      type: COUNTER
    # Generic gauges with 0-2 key/value pairs
    - pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.+), (.+)=(.+)><>Value
      name: kafka_$1_$2_$3
      type: GAUGE
      labels:
        "$4": "$5"
        "$6": "$7"
    - pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.+)><>Value
      name: kafka_$1_$2_$3
      type: GAUGE
      labels:
        "$4": "$5"
    - pattern: kafka.(\w+)<type=(.+), name=(.+)><>Value
      name: kafka_$1_$2_$3
      type: GAUGE
    # Emulate Prometheus 'Summary' metrics for the exported 'Histogram's.
    # Note that these are missing the '_sum' metric!
    - pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.+), (.+)=(.+)><>Count
      name: kafka_$1_$2_$3_count
      type: COUNTER
      labels:
        "$4": "$5"
        "$6": "$7"
    - pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.*), (.+)=(.+)><>(\d+)thPercentile
      name: kafka_$1_$2_$3
      type: GAUGE
      labels:
        "$4": "$5"
        "$6": "$7"
        quantile: "0.$8"
    - pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.+)><>Count
      name: kafka_$1_$2_$3_count
      type: COUNTER
      labels:
```

```
        "$4": "$5"
    - pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.*)><>(\d+)thPercentile
      name: kafka_$1_$2_$3
      type: GAUGE
      labels:
        "$4": "$5"
        quantile: "0.$6"
    - pattern: kafka.(\w+)<type=(.+), name=(.+)><>Count
      name: kafka_$1_$2_$3_count
      type: COUNTER
    - pattern: kafka.(\w+)<type=(.+), name=(.+)><>(\d+)thPercentile
      name: kafka_$1_$2_$3
      type: GAUGE
      labels:
        quantile: "0.$4"
  zookeeper-metrics-config.yml: |
    # See https://github.com/prometheus/jmx_exporter for more info about JMX Prometheus Exporter metrics
    lowercaseOutputName: true
    rules:
    # replicated Zookeeper
    - pattern: "org.apache.ZooKeeperService<name0=ReplicatedServer_id(\\d+)><>(\\w+)"
      name: "zookeeper_$2"
      type: GAUGE
    - pattern: "org.apache.ZooKeeperService<name0=ReplicatedServer_id(\\d+), name1=replica.(\\d+)><>(\\w+)"
      name: "zookeeper_$3"
      type: GAUGE
      labels:
        replicaId: "$2"
    - pattern: "org.apache.ZooKeeperService<name0=ReplicatedServer_id(\\d+), name1=replica.(\\d+), name2=
(\\w+)><>(Packets\\w+)"
      name: "zookeeper_$4"
      type: COUNTER
      labels:
        replicaId: "$2"
        memberType: "$3"
    - pattern: "org.apache.ZooKeeperService<name0=ReplicatedServer_id(\\d+), name1=replica.(\\d+), name2=
(\\w+)><>(\\w+)"
      name: "zookeeper_$4"
      type: GAUGE
      labels:
        replicaId: "$2"
        memberType: "$3"
    - pattern: "org.apache.ZooKeeperService<name0=ReplicatedServer_id(\\d+), name1=replica.(\\d+), name2=
(\\w+), name3=(\\w+)><>(\\w+)"
      name: "zookeeper_$4_$5"
      type: GAUGE
      labels:
        replicaId: "$2"
        memberType: "$3"
'
```

2. Define the **PodMonitor** resource:

```
KAFKA_PODMONITOR='
apiVersion: monitoring.coreos.com/v1
kind: PodMonitor
metadata:
  name: kafka-resources-metrics
  labels:
    app: strimzi
spec:
  selector:
    matchExpressions:
      - key: "ibmevents.ibm.com/kind"
        operator: In
        values: ["Kafka", "KafkaConnect", "KafkaMirrorMaker", "KafkaMirrorMaker2"]
  namespaceSelector:
    matchNames:
      - myproject
  podMetricsEndpoints:
  - path: /metrics
    port: tcp-prometheus
    relabelings:
    - separator: ;
      regex: __meta_kubernetes_pod_label_(ibmevents_ibm_com_.+)
      replacement: $1
      action: labelmap
    - sourceLabels: [__meta_kubernetes_namespace]
      separator: ;
      regex: (.*)
      targetLabel: namespace
      replacement: $1
      action: replace
    - sourceLabels: [__meta_kubernetes_pod_name]
      separator: ;
      regex: (.*)
      targetLabel: kubernetes_pod_name
      replacement: $1
      action: replace
    - sourceLabels: [__meta_kubernetes_pod_node_name]
      separator: ;
      regex: (.*)
```

```
      targetLabel: node_name
      replacement: $1
      action: replace
    - sourceLabels: [__meta_kubernetes_pod_host_ip]
      separator: ;
      regex: (.*)
      targetLabel: node_ip
      replacement: $1
      action: replace
'
```

3. Create the **ConfigMap** and **PodMonitor** resources by using the apply command:

```
oc apply -f - <<EOF
${KAFKA_METRICS_CONFIGMAP}
---
${KAFKA_PODMONITOR}
EOF
```

4. Enable the Prometheus JMX exporter by patching the Kafka instance with the **ConfigMap** that you created in the preceding step.

   Note: When you run the following commands, the Kafka brokers perform a rolling restart to enable the metric exporter.

```
oc patch kafka iaf-system --type=merge -p '{
  "spec": {
    "kafka": {
      "metricsConfig": {
        "type": "jmxPrometheusExporter",
        "valueFrom": {
          "configMapKeyRef": {
            "key": "kafka-metrics-config.yml",
            "name": "kafka-metrics"
          }
        }
      }
    }
  }
}'
```

You can now explore the Kafka broker metrics in your cluster.

For more information about Kafka Prometheus Exporter, see Kafka Prometheus Exporter.

To view the metrics, go to **Observe** > **Metrics** in OpenShift Container Platform web console. You can enter a valid Prometheus query and graph the result. For example, you can the use the following Prometheus query to get the active count of all the Kafka topic partitions including the replicas.

```
sum(kafka_server_replicamanager_partitioncount)
```

Viewing metrics in Grafana

You can use Grafana to visualize metrics and logs that come from multiple sources, which include Prometheus and other monitoring tools. Grafana provides a web-based interface for creating and customizing dashboards, which can be used to display various metrics and logs from different sources. You need to install Grafana in your OpenShift Container Platform cluster and configure it to connect to Thanos. For more information about Thanos, see Thanos.

Grafana gives access to all metrics across all of the Prometheus instances (both cluster and user workload metrics). You can use the following commands to install the Grafana v5 operator, set up a Grafana instance, and connect Prometheus as a Grafana data source.

1. Run the following script.

```
oc apply -f - <<EOF
apiVersion: v1
kind: Namespace
metadata:
  name: monitoring-grafana
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: operatorgroup
  namespace: monitoring-grafana
spec:
  targetNamespaces:
  - monitoring-grafana
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: monitoring-grafana
  namespace: monitoring-grafana
spec:
  channel: v5
  installPlanApproval: Automatic
  name: grafana-operator
  source: community-operators
  sourceNamespace: openshift-marketplace
EOF

oc project monitoring-grafana

# wait for the Grafana operator to finish installing
```

```
installReady=false
while [ "$installReady" != true ]
do
    installPlan=`oc get subscription.operators.coreos.com monitoring-grafana -n monitoring-grafana -o json | jq -r
.status.installplan.name`
    if [ -z "$installPlan" ]
    then
        installReady=false
    else
        installReady=`oc get installplan -n monitoring-grafana "$installPlan" -o json | jq -r '.status|.phase ==
"Complete"'`
    fi

    if [ "$installReady" != true ]
    then
        sleep 5
    fi
done
installReady=false
while [ "$installReady" != true ]
do
    csv=`oc get subscription.operators.coreos.com monitoring-grafana -n monitoring-grafana -o json | jq -r
.status.currentCSV`
    if [ -z "$csv" ]
    then
        installReady=false
    else
        installReady=`oc get csv -n monitoring-grafana "$csv" -o json | jq -r '.status.phase == "Succeeded"'`
    fi

    if [ "$installReady" != true ]
    then
        sleep 5
    fi
done

# set up PVC for persisting dashboards (optional)
# if not using PVC, remove
oc apply -n monitoring-grafana -f - <<EOF
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: grafana-data
  namespace: monitoring-grafana
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 2Gi
EOF

# generate random secret for the proxy
oc create secret generic -n monitoring-grafana grafana-k8s-proxy --from-literal=session_secret=$(openssl rand --hex
32) || true

# create a serviceaccount for the grafana datasource instance
oc apply -n monitoring-grafana -f - <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: grafana-cluster-monitoring
  namespace: monitoring-grafana
EOF
oc adm policy add-cluster-role-to-user cluster-monitoring-view -z grafana-cluster-monitoring -n monitoring-grafana
oc apply -n monitoring-grafana -f - <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: grafana-auth-secret
  namespace: monitoring-grafana
  annotations:
    kubernetes.io/service-account.name: grafana-cluster-monitoring
type: kubernetes.io/service-account-token
EOF
# create the Grafana instance along with a cluster role and role binding
oc apply -n monitoring-grafana -f - <<EOF
apiVersion: grafana.integreatly.org/v1beta1
kind: Grafana
metadata:
  labels:
    dashboards: "grafana"
  name: grafana
  namespace: monitoring-grafana
spec:
  client:
    preferIngress: false
  config:
    auth:
      disable_login_form: "false"
      disable_signout_menu: "true"
```

```
    auth.anonymous:
      enabled: "true"
      org_role: Admin
    auth.basic:
      enabled: "true"
    log:
      level: warn
      mode: console
  deployment:
    spec:
      template:
        spec:
          containers:
            - args:
                - -provider=openshift
                - -pass-basic-auth=false
                - -https-address=:9091
                - -http-address=
                - -email-domain=*
                - -upstream=http://localhost:3000
                - "\$(SAR)"
                - '-openshift-delegate-urls={"/": {"resource": "namespaces", "verb": "get"}}'
                - -tls-cert=/etc/tls/private/tls.crt
                - -tls-key=/etc/tls/private/tls.key
                - -client-secret-file=/var/run/secrets/kubernetes.io/serviceaccount/token
                - -cookie-secret-file=/etc/proxy/secrets/session_secret
                - -openshift-service-account=grafana-sa
                - -openshift-ca=/etc/pki/tls/cert.pem
                - -openshift-ca=/var/run/secrets/kubernetes.io/serviceaccount/ca.crt
                - -skip-auth-regex=^/metrics
              env:
                - name: SAR
                  value: '-openshift-sar={"resource": "namespaces", "verb": "get"}'
              image: registry.redhat.io/openshift4/ose-oauth-proxy:v4.10
              imagePullPolicy: Always
              name: grafana-proxy
              ports:
                - containerPort: 9091
                  name: grafana-proxy
                  protocol: TCP
              resources: {}
              volumeMounts:
                - mountPath: /etc/tls/private
                  name: secret-grafana-k8s-tls
                  readOnly: false
                - mountPath: /etc/proxy/secrets
                  name: secret-grafana-k8s-proxy
                  readOnly: false
          volumes:
            - name: secret-grafana-k8s-tls
              secret:
                secretName: grafana-k8s-tls
            - name: secret-grafana-k8s-proxy
              secret:
                secretName: grafana-k8s-proxy
            - name: grafana-data
              persistentVolumeClaim:
                claimName: grafana-pvc
  persistentVolumeClaim:
    spec:
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 2Gi
  route:
    spec:
      port:
        targetPort: grafana-proxy
      tls:
        termination: reencrypt
      to:
        kind: Service
        name: grafana-service
        weight: 100
      wildcardPolicy: None
  service:
    metadata:
      annotations:
        service.alpha.openshift.io/serving-cert-secret-name: grafana-k8s-tls
    spec:
      ports:
        - name: grafana-proxy
          port: 9091
          protocol: TCP
          targetPort: grafana-proxy
  serviceAccount:
    metadata:
      annotations:
        serviceaccounts.openshift.io/oauth-redirectreference.primary:
'{"kind":"OAuthRedirectReference","apiVersion":"v1","reference":{"kind":"Route","name":"grafana-route"}}'
---
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: grafana-proxy
  namespace: monitoring-grafana
rules:
  - apiGroups:
      - authentication.k8s.io
    resources:
      - tokenreviews
    verbs:
      - create
  - apiGroups:
      - authorization.k8s.io
    resources:
      - subjectaccessreviews
    verbs:
      - create
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: grafana-proxy
  namespace: monitoring-grafana
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: grafana-proxy
subjects:
  - kind: ServiceAccount
    name: grafana-sa
EOF

# Create the grafanadatasource instance
oc apply -f - <<EOF
apiVersion: grafana.integreatly.org/v1beta1
kind: GrafanaDatasource
metadata:
  name: prometheus
  namespace: monitoring-grafana
spec:
  datasource:
    access: proxy
    isDefault: true
    jsonData:
      httpHeaderName1: 'Authorization'
      timeInterval: 5s
      tlsSkipVerify: true
    name: prometheus
    secureJsonData:
      httpHeaderValue1: 'Bearer \${token}'
    type: prometheus
    url: 'https://thanos-querier.openshift-monitoring.svc.cluster.local:9091'
  instanceSelector:
    matchLabels:
      dashboards: grafana
  valuesFrom:
    - targetPath: secureJsonData.httpHeaderValue1
      valueFrom:
        secretKeyRef:
          key: token
          name: grafana-auth-secret
EOF

echo "Monitoring will be available at https://$(oc get route -n monitoring-grafana grafana-route -o
jsonpath='{.status.ingress[0].host}')"
```

You might see the following error while the Grafana operator is in the process of deploying:

```
Error from server (NotFound): installplans.operators.coreos.com "null" not found
```

Importing dashboards

Dashboards can be imported into Grafana either by an ID or by importing the dashboard JSON itself.

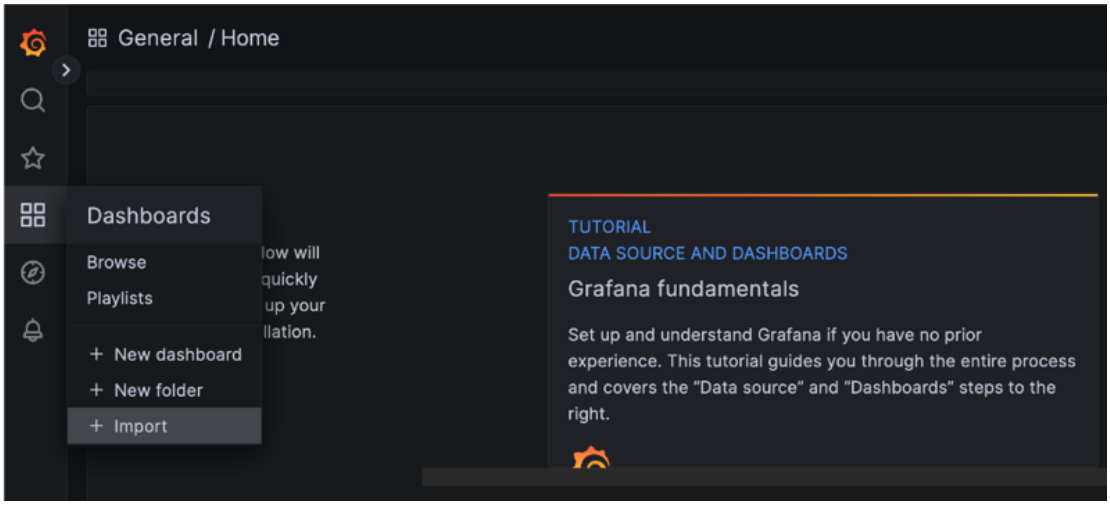


Figure. Grafana monitoring homepage

Importing Cloud Pak for AIOps dashboards to Grafana

Several dashboards are provided to assist in the operational monitoring of Cloud Pak for AIOps.

For more information about how to import a dashboard in Grafana, see Import dashboards.

Cloud Pak for AIOps offers the following custom dashboards available as JSON, which can be uploaded or copy pasted into Grafana. For more information, see the following sections:

- Cloud Pak for AIOps health dashboard
- Cloud Pak for AIOps usage dashboard

Importing other dashboards to Grafana

You can import the following dashboards in Grafana. These dashboards are developed by Grafana community members and can be imported with their ID field.

- **`K8 Cluster Detail Dashboard`** ID 10856
- **`kubernetes-networking-cluster`** ID 16371

Viewing available metrics

The preceding dashboards show a fraction of the metrics that are made available by Prometheus. To view the list of available metrics in your OpenShift Container Platform cluster, run the following commands as described in the Authentication with Bearer token.

1. Obtain the API URL for Thanos querier:

   ```
   oc get routes -n openshift-monitoring thanos-querier -o jsonpath='{.status.ingress[0].host}'
   ```

2. Retrieve the list of available Metric APIs and store the result in **`metrics.json`**:

   Note: The following command might not work on Federal Information Processing Standards (FIPS) enabled clusters due to networking reasons. To get around this issue, log in to OpenShift Container Platform online console and go to any pod. Exec into the pod and run the following command. Alternatively you can log in to the OpenShift Container Platform cluster by using the command line, exec into the pod, and then run the following command.

   ```
   curl -k -H "Authorization: Bearer $(oc whoami -t)" https://<thanos_querier_route>/api/v1/metadata > metrics.json
   ```

   Where **`<thanos_querier_route>`** is the route that you obtained in the preceding command.

All of the Metric APIs listed under the output of the preceding command can be copy and pasted in the cluster under **Observe** > **Metrics**. These queries can be used in Grafana to create new panels. The new panels can be saved in a new dashboard or in one of the existing custom (community) dashboards.

Cloud Pak for AIOps health dashboard

This dashboard focuses on the high level health of Cloud Pak for AIOps, its storage and the cluster that it is running on. This dashboard answers the following questions:

1. Does the cluster have enough resources?
2. How much storage is used?
3. Is the Cloud Pak for AIOps instance healthy?

To import this JSON dashboard, see Importing dashboard by using JSON.

You can import the following JSON into Grafana.

```
{
  "__inputs": [
    {
      "name": "DS_PROMETHEUS",
```

```
      "label": "prometheus",
      "description": "",
      "type": "datasource",
      "pluginId": "prometheus",
      "pluginName": "Prometheus"
    }
  ],
  "__elements": {},
  "__requires": [
    {
      "type": "panel",
      "id": "gauge",
      "name": "Gauge",
      "version": ""
    },
    {
      "type": "grafana",
      "id": "grafana",
      "name": "Grafana",
      "version": "10.4.3"
    },
    {
      "type": "panel",
      "id": "graph",
      "name": "Graph (old)",
      "version": ""
    },
    {
      "type": "datasource",
      "id": "prometheus",
      "name": "Prometheus",
      "version": "1.0.0"
    },
    {
      "type": "panel",
      "id": "stat",
      "name": "Stat",
      "version": ""
    },
    {
      "type": "panel",
      "id": "table",
      "name": "Table",
      "version": ""
    }
  ],
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": {
          "type": "grafana",
          "uid": "-- Grafana --"
        },
        "enable": true,
        "hide": true,
        "iconColor": "rgba(0, 211, 255, 1)",
        "name": "Annotations & Alerts",
        "target": {
          "limit": 100,
          "matchAny": false,
          "tags": [],
          "type": "dashboard"
        },
        "type": "dashboard"
      }
    ]
  },
  "editable": true,
  "fiscalYearStartMonth": 0,
  "graphTooltip": 0,
  "id": null,
  "links": [],
  "liveNow": false,
  "panels": [
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 0
      },
      "id": 8,
      "panels": [],
      "title": "Cluster Health",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
```

```
      },
      "description": "Percentage of CPU usage across all nodes taken at 5 minute averages",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "decimals": 2,
          "mappings": [
            {
              "options": {
                "match": "null",
                "result": {
                  "color": "rgba(50, 172, 45, 0.97)",
                  "text": "0"
                }
              },
              "type": "special"
            }
          ],
          "max": 100,
          "min": 0,
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "rgba(50, 172, 45, 0.97)",
                "value": null
              },
              {
                "color": "rgba(237, 129, 40, 0.89)",
                "value": 65
              },
              {
                "color": "rgba(245, 54, 54, 0.9)",
                "value": 90
              }
            ]
          },
          "unit": "percent"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 8,
        "x": 0,
        "y": 1
      },
      "id": 6,
      "maxDataPoints": 100,
      "options": {
        "minVizHeight": 75,
        "minVizWidth": 75,
        "orientation": "horizontal",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "showThresholdLabels": false,
        "showThresholdMarkers": true,
        "sizing": "auto"
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum (rate (container_cpu_usage_seconds_total{id=\"/\",node=~\"^.*$\"}[5m])) / sum (machine_cpu_cores{
node=~\"^.*$\"}) * 100",
          "format": "time_series",
          "interval": "10s",
          "intervalFactor": 1,
          "refId": "A",
          "step": 10
        }
      ],
      "title": "CPU usage (5m avg)",
      "type": "gauge"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Percentage of memory usage across all nodes",
      "fieldConfig": {
```

```
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [
            {
              "options": {
                "match": "null",
                "result": {
                  "color": "rgba(50, 172, 45, 0.97)",
                  "text": "0"
                }
              },
              "type": "special"
            }
          ],
          "max": 100,
          "min": 0,
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "rgba(50, 172, 45, 0.97)",
                "value": null
              },
              {
                "color": "rgba(237, 129, 40, 0.89)",
                "value": 65
              },
              {
                "color": "rgba(245, 54, 54, 0.9)",
                "value": 90
              }
            ]
          },
          "unit": "percent"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 8,
        "x": 8,
        "y": 1
      },
      "id": 4,
      "maxDataPoints": 100,
      "options": {
        "minVizHeight": 75,
        "minVizWidth": 75,
        "orientation": "horizontal",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "showThresholdLabels": false,
        "showThresholdMarkers": true,
        "sizing": "auto"
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum (container_memory_working_set_bytes{id=\"/\",node=~\"^.*$\"}) / sum
(machine_memory_bytes{node=~\"^.*$\"}) * 100",
          "format": "time_series",
          "interval": "10s",
          "intervalFactor": 1,
          "refId": "A",
          "step": 10
        }
      ],
      "title": "Memory usage",
      "type": "gauge"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Percentage of filesystem usage across all nodes",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
```

```
          "decimals": 2,
          "mappings": [
            {
              "options": {
                "match": "null",
                "result": {
                  "color": "rgba(50, 172, 45, 0.97)",
                  "text": "0"
                }
              },
              "type": "special"
            }
          ],
          "max": 100,
          "min": 0,
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "rgba(50, 172, 45, 0.97)",
                "value": null
              },
              {
                "color": "rgba(237, 129, 40, 0.89)",
                "value": 65
              },
              {
                "color": "rgba(245, 54, 54, 0.9)",
                "value": 90
              }
            ]
          },
          "unit": "percent"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 8,
        "x": 16,
        "y": 1
      },
      "id": 7,
      "maxDataPoints": 100,
      "options": {
        "minVizHeight": 75,
        "minVizWidth": 75,
        "orientation": "horizontal",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "showThresholdLabels": false,
        "showThresholdMarkers": true,
        "sizing": "auto"
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum (container_fs_usage_bytes{device=~\"^/dev/.*$\",id=\"/\",node=~\"^.*$\"}) / sum
(container_fs_limit_bytes{device=~\"^/dev/.*$\",id=\"/\",node=~\"^.*$\"}) * 100",
          "format": "time_series",
          "interval": "10s",
          "intervalFactor": 1,
          "legendFormat": "",
          "metric": "",
          "refId": "A",
          "step": 10
        }
      ],
      "title": "Filesystem usage",
      "type": "gauge"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Sum of the 5 min avg of the cumulative cpu time consumed in seconds for all nodes",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "decimals": 2,
```

```
        "mappings": [
          {
            "options": {
              "match": "null",
              "result": {
                "text": "0"
              }
            },
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green",
              "value": null
            },
            {
              "color": "red",
              "value": 80
            }
          ]
        },
        "unit": "none"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 3,
      "w": 4,
      "x": 0,
      "y": 6
    },
    "id": 11,
    "maxDataPoints": 100,
    "options": {
      "colorMode": "none",
      "graphMode": "none",
      "justifyMode": "auto",
      "orientation": "horizontal",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "showPercentChange": false,
      "textMode": "auto",
      "wideLayout": true
    },
    "pluginVersion": "10.4.3",
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "editorMode": "code",
        "expr": "sum (rate (container_cpu_usage_seconds_total{id=\"/\",node=~\"^.*$\"}[5m]))",
        "format": "time_series",
        "interval": "10s",
        "intervalFactor": 1,
        "range": true,
        "refId": "A",
        "step": 10
      }
    ],
    "title": "Used",
    "type": "stat"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "description": "Total number of logical cores across all nodes",
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "decimals": 2,
        "mappings": [
          {
            "options": {
              "match": "null",
              "result": {
                "text": "0"
              }
            },
```

```
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green",
              "value": null
            },
            {
              "color": "red",
              "value": 80
            }
          ]
        },
        "unit": "none"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 3,
      "w": 4,
      "x": 4,
      "y": 6
    },
    "id": 12,
    "maxDataPoints": 100,
    "options": {
      "colorMode": "none",
      "graphMode": "none",
      "justifyMode": "auto",
      "orientation": "horizontal",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "showPercentChange": false,
      "textMode": "auto",
      "wideLayout": true
    },
    "pluginVersion": "10.4.3",
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "editorMode": "code",
        "expr": "sum (machine_cpu_cores{node=~\"^.*$\"})",
        "interval": "10s",
        "intervalFactor": 1,
        "range": true,
        "refId": "A",
        "step": 10
      }
    ],
    "title": "Total",
    "type": "stat"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "decimals": 2,
        "mappings": [
          {
            "options": {
              "match": "null",
              "result": {
                "text": "0"
              }
            },
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green",
              "value": null
            },
```

```
            {
              "color": "red",
              "value": 80
            }
          ]
        },
        "unit": "bytes"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 3,
      "w": 4,
      "x": 8,
      "y": 6
    },
    "id": 9,
    "maxDataPoints": 100,
    "options": {
      "colorMode": "none",
      "graphMode": "none",
      "justifyMode": "auto",
      "orientation": "horizontal",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "showPercentChange": false,
      "textMode": "auto",
      "wideLayout": true
    },
    "pluginVersion": "10.4.3",
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "expr": "sum (container_memory_working_set_bytes{id=\"/\",node=~\"^.*$\"})",
        "interval": "10s",
        "intervalFactor": 1,
        "refId": "A",
        "step": 10
      }
    ],
    "title": "Used",
    "type": "stat"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "decimals": 2,
        "mappings": [
          {
            "options": {
              "match": "null",
              "result": {
                "text": "0"
              }
            },
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green",
              "value": null
            },
            {
              "color": "red",
              "value": 80
            }
          ]
        },
        "unit": "bytes"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 3,
```

```
        "w": 4,
        "x": 12,
        "y": 6
      },
      "id": 10,
      "maxDataPoints": 100,
      "options": {
        "colorMode": "none",
        "graphMode": "none",
        "justifyMode": "auto",
        "orientation": "horizontal",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum (machine_memory_bytes{node=~\"^.*$\"})",
          "interval": "10s",
          "intervalFactor": 1,
          "refId": "A",
          "step": 10
        }
      ],
      "title": "Total",
      "type": "stat"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "decimals": 2,
          "mappings": [
            {
              "options": {
                "match": "null",
                "result": {
                  "text": "0"
                }
              },
              "type": "special"
            }
          ],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              },
              {
                "color": "red",
                "value": 80
              }
            ]
          },
          "unit": "bytes"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 3,
        "w": 4,
        "x": 16,
        "y": 6
      },
      "id": 13,
      "maxDataPoints": 100,
      "options": {
        "colorMode": "none",
        "graphMode": "none",
        "justifyMode": "auto",
        "orientation": "horizontal",
        "reduceOptions": {
```

```
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "showPercentChange": false,
      "textMode": "auto",
      "wideLayout": true
    },
    "pluginVersion": "10.4.3",
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "expr": "sum (container_fs_usage_bytes{device=~\"^/dev/.*$\",id=\"/\",node=~\"^.*$\"})",
        "interval": "10s",
        "intervalFactor": 1,
        "refId": "A",
        "step": 10
      }
    ],
    "title": "Used",
    "type": "stat"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "decimals": 2,
        "mappings": [
          {
            "options": {
              "match": "null",
              "result": {
                "text": "0"
              }
            },
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green",
              "value": null
            },
            {
              "color": "red",
              "value": 80
            }
          ]
        },
        "unit": "bytes"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 3,
      "w": 4,
      "x": 20,
      "y": 6
    },
    "id": 14,
    "maxDataPoints": 100,
    "options": {
      "colorMode": "none",
      "graphMode": "none",
      "justifyMode": "auto",
      "orientation": "horizontal",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "showPercentChange": false,
      "textMode": "auto",
      "wideLayout": true
    },
    "pluginVersion": "10.4.3",
    "targets": [
```

```
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum (container_fs_limit_bytes{device=~\"^/dev/.*$\",id=\"/\",node=~\"^.*$\"})",
          "interval": "10s",
          "intervalFactor": 1,
          "refId": "A",
          "step": 10
        }
      ],
      "title": "Total",
      "type": "stat"
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 9
      },
      "id": 16,
      "panels": [],
      "title": "Node Health",
      "type": "row"
    },
    {
      "aliasColors": {},
      "bars": false,
      "dashLength": 10,
      "dashes": false,
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Sum of nodes in a \"ready\" state across a period of time",
      "fill": 0,
      "fillGradient": 0,
      "gridPos": {
        "h": 6,
        "w": 12,
        "x": 0,
        "y": 10
      },
      "hiddenSeries": false,
      "id": 2058,
      "legend": {
        "avg": false,
        "current": false,
        "max": false,
        "min": false,
        "show": true,
        "total": false,
        "values": false
      },
      "lines": true,
      "linewidth": 1,
      "nullPointMode": "null",
      "options": {
        "alertThreshold": true
      },
      "percentage": false,
      "pluginVersion": "10.4.3",
      "pointradius": 5,
      "points": false,
      "renderer": "flot",
      "seriesOverrides": [
        {
          "alias": "Total",
          "fillBelowTo": "Unavailable"
        }
      ],
      "spaceLength": 10,
      "stack": false,
      "steppedLine": false,
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum(kube_node_info{node=~\".*\"})",
          "format": "time_series",
          "intervalFactor": 2,
          "legendFormat": "Total",
          "refId": "A"
        },
        {
          "datasource": {
            "type": "prometheus",
```

```
          "uid": "${DS_PROMETHEUS}"
        },
        "expr": "sum(kube_node_status_condition{condition=\"Ready\", node=~\".*\", status!=\"true\"})",
        "format": "time_series",
        "instant": false,
        "intervalFactor": 2,
        "legendFormat": "Unavailable",
        "refId": "B"
      }
    ],
    "thresholds": [],
    "timeRegions": [],
    "title": "Available Nodes",
    "tooltip": {
      "shared": true,
      "sort": 0,
      "value_type": "individual"
    },
    "type": "graph",
    "xaxis": {
      "mode": "time",
      "show": true,
      "values": []
    },
    "yaxes": [
      {
        "$$hashKey": "object:95",
        "decimals": 0,
        "format": "short",
        "logBase": 1,
        "show": true
      },
      {
        "$$hashKey": "object:96",
        "format": "short",
        "logBase": 1,
        "show": true
      }
    ],
    "yaxis": {
      "align": false
    }
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "description": "Number of nodes experiencing low disk space (default value of 85%)",
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "mappings": [
          {
            "options": {
              "match": "null",
              "result": {
                "color": "#299c46",
                "text": "0"
              }
            },
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "#299c46",
              "value": null
            },
            {
              "color": "rgba(237, 129, 40, 0.89)",
              "value": 1
            },
            {
              "color": "#d44a3a"
            }
          ]
        },
        "unit": "none"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 3,
      "w": 4,
      "x": 12,
      "y": 10
    },
```

```
    "id": 2056,
    "maxDataPoints": 100,
    "options": {
      "colorMode": "background",
      "graphMode": "none",
      "justifyMode": "auto",
      "orientation": "horizontal",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "showPercentChange": false,
      "textMode": "auto",
      "wideLayout": true
    },
    "pluginVersion": "10.4.3",
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "expr": "sum(kube_node_status_condition{condition=\"DiskPressure\", node=~\".*\", status!=\"false\"})",
        "format": "time_series",
        "intervalFactor": 1,
        "refId": "A"
      }
    ],
    "title": "Disk Pressure",
    "type": "stat"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "description": "Number of nodes experiencing memory pressure",
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "mappings": [
          {
            "options": {
              "match": "null",
              "result": {
                "color": "#299c46",
                "text": "0"
              }
            },
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "#299c46",
              "value": null
            },
            {
              "color": "rgba(237, 129, 40, 0.89)",
              "value": 1
            },
            {
              "color": "#d44a3a"
            }
          ]
        },
        "unit": "none"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 3,
      "w": 4,
      "x": 16,
      "y": 10
    },
    "id": 2059,
    "maxDataPoints": 100,
    "options": {
      "colorMode": "background",
      "graphMode": "none",
      "justifyMode": "auto",
      "orientation": "horizontal",
      "reduceOptions": {
        "calcs": [
```

```
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum(kube_node_status_condition{condition=\"MemoryPressure\", node=~\".*\", status!=\"false\"})",
          "format": "time_series",
          "intervalFactor": 1,
          "refId": "A"
        }
      ],
      "title": "Memory Pressure",
      "type": "stat"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Number of nodes experiencing PID pressure",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [
            {
              "options": {
                "match": "null",
                "result": {
                  "color": "#299c46",
                  "text": "0"
                }
              },
              "type": "special"
            }
          ],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "#299c46",
                "value": null
              },
              {
                "color": "rgba(237, 129, 40, 0.89)",
                "value": 1
              },
              {
                "color": "#d44a3a"
              }
            ]
          },
          "unit": "none"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 3,
        "w": 4,
        "x": 20,
        "y": 10
      },
      "id": 2063,
      "maxDataPoints": 100,
      "options": {
        "colorMode": "background",
        "graphMode": "none",
        "justifyMode": "auto",
        "orientation": "horizontal",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
```

```
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum(kube_node_status_condition{condition=\"PIDPressure\", node=~\".*\", status!=\"false\"})",
          "format": "time_series",
          "intervalFactor": 1,
          "refId": "A"
        }
      ],
      "title": "PID Pressure",
      "type": "stat"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Number of nodes that are unschedulable. New pods cannot be scheduled on nodes in this state.",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [
            {
              "options": {
                "match": "null",
                "result": {
                  "color": "#299c46",
                  "text": "0"
                }
              },
              "type": "special"
            }
          ],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "#299c46",
                "value": null
              },
              {
                "color": "rgba(237, 129, 40, 0.89)",
                "value": 1
              },
              {
                "color": "#d44a3a"
              }
            ]
          },
          "unit": "none"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 3,
        "w": 4,
        "x": 12,
        "y": 13
      },
      "id": 2026,
      "maxDataPoints": 100,
      "options": {
        "colorMode": "background",
        "graphMode": "none",
        "justifyMode": "auto",
        "orientation": "horizontal",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum(kube_node_spec_unschedulable{node=~\".*\"})",
          "format": "time_series",
```

```
        "intervalFactor": 1,
        "refId": "A"
      }
    ],
    "title": "Unschedulable",
    "type": "stat"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "description": "Number of nodes that are out of disk storage",
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "mappings": [
          {
            "options": {
              "match": "null",
              "result": {
                "color": "#299c46",
                "text": "0"
              }
            },
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "#299c46",
              "value": null
            }
          ]
        },
        "unit": "none"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 3,
      "w": 4,
      "x": 16,
      "y": 13
    },
    "id": 2025,
    "maxDataPoints": 100,
    "options": {
      "colorMode": "background",
      "graphMode": "none",
      "justifyMode": "auto",
      "orientation": "horizontal",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "showPercentChange": false,
      "textMode": "auto",
      "wideLayout": true
    },
    "pluginVersion": "10.4.3",
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "expr": "sum(kube_node_status_condition{condition=\"OutOfDisk\", node=~\".*\", status=\"true\"})",
        "format": "time_series",
        "intervalFactor": 1,
        "refId": "A"
      }
    ],
    "title": "Out of Disk",
    "type": "stat"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "description": "Number of nodes with a \"NetworkUnavailable\" status",
    "fieldConfig": {
      "defaults": {
        "color": {
```

```
          "mode": "thresholds"
        },
        "mappings": [
          {
            "options": {
              "match": "null",
              "result": {
                "color": "#299c46",
                "text": "0"
              }
            },
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "#299c46",
              "value": null
            },
            {
              "color": "rgba(237, 129, 40, 0.89)",
              "value": 1
            },
            {
              "color": "#d44a3a"
            }
          ]
        },
        "unit": "none"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 3,
      "w": 4,
      "x": 20,
      "y": 13
    },
    "id": 2060,
    "maxDataPoints": 100,
    "options": {
      "colorMode": "background",
      "graphMode": "area",
      "justifyMode": "auto",
      "orientation": "horizontal",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "showPercentChange": false,
      "textMode": "auto",
      "wideLayout": true
    },
    "pluginVersion": "10.4.3",
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "expr": "sum(kube_node_status_condition{condition=\"NetworkUnavailable\", node=~\".*\", status!=\"false\"})",
        "format": "time_series",
        "intervalFactor": 1,
        "refId": "A"
      }
    ],
    "title": "Network Unavailable",
    "type": "stat"
  },
  {
    "collapsed": false,
    "gridPos": {
      "h": 1,
      "w": 24,
      "x": 0,
      "y": 16
    },
    "id": 2065,
    "panels": [],
    "title": "Storage Health",
    "type": "row"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
```

```
    "description": "Count of PVCs in a \"bound\" state",
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "mappings": [],
        "thresholds": {
          "mode": "percentage",
          "steps": [
            {
              "color": "green",
              "value": null
            }
          ]
        }
      },
      "overrides": []
    },
    "gridPos": {
      "h": 6,
      "w": 12,
      "x": 0,
      "y": 17
    },
    "id": 2067,
    "options": {
      "minVizHeight": 75,
      "minVizWidth": 75,
      "orientation": "auto",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "showThresholdLabels": false,
      "showThresholdMarkers": true,
      "sizing": "auto"
    },
    "pluginVersion": "10.4.3",
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "editorMode": "code",
        "expr": "count(kube_persistentvolumeclaim_status_phase{namespace=\"$namespace\",phase=\"Bound\"})",
        "legendFormat": "__auto",
        "range": true,
        "refId": "A"
      }
    ],
    "title": "Bound PVC",
    "type": "gauge"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "mappings": [],
        "thresholds": {
          "mode": "percentage",
          "steps": [
            {
              "color": "red",
              "value": null
            }
          ]
        }
      },
      "overrides": []
    },
    "gridPos": {
      "h": 6,
      "w": 12,
      "x": 12,
      "y": 17
    },
    "id": 2070,
    "options": {
      "minVizHeight": 75,
      "minVizWidth": 75,
      "orientation": "auto",
```

```
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "showThresholdLabels": false,
        "showThresholdMarkers": true,
        "sizing": "auto"
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by (phase)(kube_persistentvolumeclaim_status_phase{namespace=\"$namespace\",phase!~\"Bound\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Unbound PVC",
      "type": "gauge"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "PVC usage percentage for each PVC in the namespace, thresholds of 85% & 95% usage",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "decimals": 2,
          "displayName": "${__field.labels.persistentvolumeclaim}",
          "mappings": [],
          "max": 100,
          "thresholds": {
            "mode": "percentage",
            "steps": [
              {
                "color": "green"
              },
              {
                "color": "#EAB839",
                "value": 85
              },
              {
                "color": "red",
                "value": 95
              }
            ]
          },
          "unit": "percent"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 14,
        "w": 24,
        "x": 0,
        "y": 23
      },
      "id": 2071,
      "options": {
        "minVizHeight": 75,
        "minVizWidth": 75,
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "showThresholdLabels": false,
        "showThresholdMarkers": true,
        "sizing": "auto",
        "text": {
          "titleSize": 14
        }
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
```

```
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr":
"sort_desc(kubelet_volume_stats_used_bytes{namespace=\"$namespace\"}/kubelet_volume_stats_capacity_bytes{namespace=\"$nam
espace\"} * 100)",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "PVC Usage",
      "type": "gauge"
    },
    {
      "aliasColors": {},
      "bars": false,
      "dashLength": 10,
      "dashes": false,
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Max and current PVC usage",
      "fill": 1,
      "fillGradient": 0,
      "gridPos": {
        "h": 13,
        "w": 24,
        "x": 0,
        "y": 37
      },
      "hiddenSeries": false,
      "id": 56,
      "legend": {
        "alignAsTable": true,
        "avg": false,
        "current": true,
        "max": true,
        "min": false,
        "rightSide": true,
        "show": true,
        "sort": "current",
        "sortDesc": true,
        "total": false,
        "values": true
      },
      "lines": true,
      "linewidth": 1,
      "nullPointMode": "null",
      "options": {
        "alertThreshold": true
      },
      "percentage": false,
      "pluginVersion": "9.4.7",
      "pointradius": 2,
      "points": false,
      "renderer": "flot",
      "seriesOverrides": [],
      "spaceLength": 10,
      "stack": false,
      "steppedLine": false,
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum by (persistentvolumeclaim)
(kubelet_volume_stats_used_bytes{namespace=\"$namespace\"}/kubelet_volume_stats_capacity_bytes{namespace=\"$namespace\"}
* 100)",
          "interval": "",
          "legendFormat": "{{ persistentvolumeclaim }}",
          "refId": "A"
        }
      ],
      "thresholds": [],
      "timeRegions": [],
      "title": "PVC Usage (max & current)",
      "tooltip": {
        "shared": true,
        "sort": 2,
        "value_type": "individual"
      },
      "type": "graph",
      "xaxis": {
        "mode": "time",
        "show": true,
        "values": []
      },
      "yaxes": [
```

```
        {
          "$$hashKey": "object:539",
          "format": "percent",
          "logBase": 1,
          "show": true
        },
        {
          "$$hashKey": "object:540",
          "format": "short",
          "logBase": 1,
          "show": true
        }
      ],
      "yaxis": {
        "align": false
      }
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 50
      },
      "id": 2069,
      "panels": [],
      "title": "Pod Health",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Number of pods in a \"running\" state in $namespace namespace",
      "fieldConfig": {
        "defaults": {
          "color": {
            "fixedColor": "#629e51",
            "mode": "fixed"
          },
          "mappings": [
            {
              "options": {
                "match": "null",
                "result": {
                  "text": "0"
                }
              },
              "type": "special"
            }
          ],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green"
              },
              {
                "color": "red",
                "value": 80
              }
            ]
          },
          "unit": "none"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 12,
        "x": 0,
        "y": 51
      },
      "id": 2030,
      "maxDataPoints": 100,
      "options": {
        "colorMode": "none",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "horizontal",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "textMode": "auto"
      },
      "pluginVersion": "9.4.7",
```

```
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum(kube_pod_status_phase{namespace=\"$namespace\", phase=\"Running\"})",
          "format": "time_series",
          "interval": "",
          "intervalFactor": 1,
          "refId": "A"
        }
      ],
      "title": "Pods Running",
      "type": "stat"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Number of pods in a \"succeeded\" state in $namespace namespace",
      "fieldConfig": {
        "defaults": {
          "color": {
            "fixedColor": "#629e51",
            "mode": "fixed"
          },
          "mappings": [
            {
              "options": {
                "match": "null",
                "result": {
                  "text": "0"
                }
              },
              "type": "special"
            }
          ],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green"
              },
              {
                "color": "red",
                "value": 80
              }
            ]
          },
          "unit": "none"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 12,
        "x": 12,
        "y": 51
      },
      "id": 2033,
      "maxDataPoints": 100,
      "options": {
        "colorMode": "none",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "horizontal",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "textMode": "auto"
      },
      "pluginVersion": "9.4.7",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum(kube_pod_status_phase{namespace=\"$namespace\", phase=\"Succeeded\"})",
          "format": "time_series",
          "interval": "",
          "intervalFactor": 1,
          "refId": "A"
        }
      ],
      "title": "Pods Succeeded",
```

```
      "type": "stat"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Number of pods in a \"failed\" state in $namespace namespace",
      "fieldConfig": {
        "defaults": {
          "mappings": [
            {
              "options": {
                "match": "null",
                "result": {
                  "text": "0"
                }
              },
              "type": "special"
            }
          ],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green"
              },
              {
                "color": "red",
                "value": 1
              }
            ]
          },
          "unit": "none"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 4,
        "w": 12,
        "x": 0,
        "y": 56
      },
      "id": 2032,
      "maxDataPoints": 100,
      "options": {
        "colorMode": "none",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "horizontal",
        "reduceOptions": {
          "calcs": [
            "lastNotNull"
          ],
          "fields": "",
          "values": false
        },
        "textMode": "auto"
      },
      "pluginVersion": "9.4.7",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "expr": "sum(kube_pod_status_phase{namespace=\"$namespace\", phase=\"Failed\"})",
          "format": "time_series",
          "interval": "",
          "intervalFactor": 1,
          "refId": "A"
        }
      ],
      "title": "Pods Failed",
      "type": "stat"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Number of pods in an \"unknown\" state in $namespace namespace",
      "fieldConfig": {
        "defaults": {
          "color": {
            "fixedColor": "#629e51",
            "mode": "fixed"
          },
          "mappings": [
            {
              "options": {
                "match": "null",
```

```
              "result": {
                "text": "0"
              }
            },
            "type": "special"
          }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green"
            },
            {
              "color": "red",
              "value": 80
            }
          ]
        },
        "unit": "none"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 4,
      "w": 12,
      "x": 12,
      "y": 56
    },
    "id": 2034,
    "maxDataPoints": 100,
    "options": {
      "colorMode": "none",
      "graphMode": "area",
      "justifyMode": "auto",
      "orientation": "horizontal",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "textMode": "auto"
    },
    "pluginVersion": "9.4.7",
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "expr": "sum(kube_pod_status_phase{namespace=\"$namespace\", phase=\"Unknown\"})",
        "format": "time_series",
        "interval": "",
        "intervalFactor": 1,
        "refId": "A"
      }
    ],
    "title": "Pods Unknown",
    "type": "stat"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "description": "Number of pods in a \"pending\" state in $namespace namespace",
    "fieldConfig": {
      "defaults": {
        "color": {
          "fixedColor": "red",
          "mode": "fixed"
        },
        "custom": {
          "align": "auto",
          "cellOptions": {
            "type": "auto"
          },
          "inspect": false
        },
        "mappings": [],
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green"
            },
            {
              "color": "red",
              "value": 1
            }
```

```
            ]
          }
        },
        "overrides": []
      },
      "gridPos": {
        "h": 8,
        "w": 24,
        "x": 0,
        "y": 60
      },
      "id": 2073,
      "options": {
        "footer": {
          "countRows": false,
          "fields": "",
          "reducer": [
            "sum"
          ],
          "show": false
        },
        "frameIndex": 0,
        "showHeader": true,
        "sortBy": [
          {
            "desc": true,
            "displayName": "{pod=\"aimanager-aio-chatops-slack-integrator-944797d46-79dv6\",
reason=\"InvalidImageName\"}"
          }
        ]
      },
      "pluginVersion": "9.4.7",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "builder",
          "expr": "sum(kube_pod_container_status_waiting_reason{namespace=\"$namespace\"}) by (pod, reason)",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Pods pending",
      "transformations": [
        {
          "filter": {
            "id": "byRefId",
            "options": ""
          },
          "id": "limit",
          "options": {
            "limitField": 1
          }
        },
        {
          "id": "joinByLabels",
          "options": {
            "value": "reason"
          }
        }
      ],
      "type": "table"
    }
  ],
  "refresh": "",
  "revision": 1,
  "schemaVersion": 39,
  "tags": [],
  "templating": {
    "list": [
      {
        "current": {},
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "definition": "label_values(kube_namespace_status_phase{job=\"kube-state-metrics\", cluster=\"$cluster\"},
namespace)",
        "hide": 0,
        "includeAll": false,
        "multi": false,
        "name": "namespace",
        "options": [],
        "query": {
          "query": "label_values(kube_namespace_status_phase{job=\"kube-state-metrics\", cluster=\"$cluster\"},
namespace)",
          "refId": "StandardVariableQuery"
        },
        "refresh": 2,
```

```
        "regex": "",
        "skipUrlSync": false,
        "sort": 1,
        "type": "query"
      },
      {
        "current": {},
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "definition": "label_values(up{job=\"kube-state-metrics\"}, cluster)",
        "hide": 2,
        "includeAll": false,
        "multi": false,
        "name": "cluster",
        "options": [],
        "query": {
          "query": "label_values(up{job=\"kube-state-metrics\"}, cluster)",
          "refId": "StandardVariableQuery"
        },
        "refresh": 1,
        "regex": "",
        "skipUrlSync": false,
        "sort": 0,
        "type": "query"
      }
    ]
  },
  "time": {
    "from": "now-30m",
    "to": "now"
  },
  "timepicker": {},
  "timezone": "",
  "title": "CP4AIOps Health",
  "uid": "80acb6b1-edee-48e7-b9f6-92daf1dc4deb",
  "version": 2,
  "weekStart": ""
}
```

Cloud Pak for AIOps usage dashboard

This dashboard focuses on providing visibility into the traffic that flows through your environment and validates that Cloud Pak for AIOps is correctly configured to handle the traffic as described in Custom sizing.

To import this JSON dashboard, see Importing dashboard by using JSON.

You can import the following JSON into Grafana.

```
{
  "__inputs": [
    {
      "name": "DS_PROMETHEUS",
      "label": "prometheus",
      "description": "",
      "type": "datasource",
      "pluginId": "prometheus",
      "pluginName": "Prometheus"
    }
  ],
  "__elements": {},
  "__requires": [
    {
      "type": "grafana",
      "id": "grafana",
      "name": "Grafana",
      "version": "10.4.3"
    },
    {
      "type": "datasource",
      "id": "prometheus",
      "name": "Prometheus",
      "version": "1.0.0"
    },
    {
      "type": "panel",
      "id": "stat",
      "name": "Stat",
      "version": ""
    },
    {
      "type": "panel",
      "id": "text",
      "name": "Text",
      "version": ""
    },
```

```
    {
      "type": "panel",
      "id": "timeseries",
      "name": "Time series",
      "version": ""
    }
  ],
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": {
          "type": "grafana",
          "uid": "-- Grafana --"
        },
        "enable": true,
        "hide": true,
        "iconColor": "rgba(0, 211, 255, 1)",
        "name": "Annotations & Alerts",
        "target": {
          "limit": 100,
          "matchAny": false,
          "tags": [],
          "type": "dashboard"
        },
        "type": "dashboard"
      }
    ]
  },
  "editable": true,
  "fiscalYearStartMonth": 0,
  "graphTooltip": 0,
  "id": null,
  "links": [],
  "liveNow": false,
  "panels": [
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 0
      },
      "id": 328,
      "panels": [],
      "title": "Objective",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "gridPos": {
        "h": 4,
        "w": 9,
        "x": 0,
        "y": 1
      },
      "id": 330,
      "options": {
        "code": {
          "language": "plaintext",
          "showLineNumbers": false,
          "showMiniMap": false
        },
        "content": "# Objective\nThe goal of this dashboard is to provide visibility\n on the level of traffic flowing through the \n environment and make sure that Cloud Pak for\n  AIOps is properly configured to handle it\n  as described in the [sizing documentation](https://www.ibm.com/docs/en/cloud-paks/cloud-pak-aiops/4.6.1?topic=planning-hardware-requirements#aimgr_custom). ",
        "mode": "markdown"
      },
      "pluginVersion": "10.4.3",
      "type": "text"
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 5
      },
      "id": 20,
      "panels": [],
      "title": "Events",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
```

```
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Shows the per-second rate of events ingested into the system over a 1m average",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "events/s"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 17,
        "x": 0,
        "y": 6
      },
      "id": 2,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "mode": "single",
          "sort": "none"
        }
      },
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=\"cp4waiops-
cartridge.lifecycle.input.events\"}[1m]))",
          "legendFormat": "New events",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Event ingestion rate",
      "type": "timeseries"
    },
    {
      "datasource": {
```

```
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 7,
        "x": 17,
        "y": 6
      },
      "id": 288,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "delta"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=\"cp4waiops-
cartridge.lifecycle.input.events\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Events ingested in time range",
      "type": "stat"
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 11
      },
      "id": 304,
      "panels": [],
      "title": "Alerts",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Shows the per-second rate of alert create/update requests over a 1m average",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
```

```
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "reqps"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 17,
        "x": 0,
        "y": 12
      },
      "id": 5,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "mode": "single",
          "sort": "none"
        }
      },
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=\"cp4waiops-
cartridge.irdatalayer.changes.alerts\"}[1m]))",
          "hide": false,
          "legendFormat": "Alert creates & updates",
          "range": true,
          "refId": "C"
        }
      ],
      "title": "Alert request rate",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Count that tracks alert create/update requests in time range. Updates can come from events, manual
user changes, and Cloud Pak for AIOps enrichments.",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [],
```

```
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 7,
        "x": 17,
        "y": 12
      },
      "id": 289,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "delta"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=\"cp4waiops-
cartridge.irdatalayer.changes.alerts\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Alerts created/updated in time range",
      "type": "stat"
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 17
      },
      "id": 306,
      "panels": [],
      "title": "Incidents",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Shows the per-second rates of incidents being created/updated over a 1m average",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
```

```
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "reqps"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 10,
        "w": 17,
        "x": 0,
        "y": 18
      },
      "id": 6,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "mode": "single",
          "sort": "none"
        }
      },
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=\"cp4waiops-
cartridge.lifecycle.input.stories\"}[1m]))",
          "legendFormat": "Incident creates",
          "range": true,
          "refId": "A"
        },
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=\"cp4waiops-
cartridge.irdatalayer.changes.stories\"}[1m]))",
          "hide": false,
          "legendFormat": "Incident updates",
          "range": true,
          "refId": "C"
        }
      ],
      "title": "Incident request rates",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Count that tracks the creation of new incidents in time range. Policies that determine when to
create incidents can be configured in the Cloud Pak for AIOps Automations page",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
```

```
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 7,
        "x": 17,
        "y": 18
      },
      "id": 290,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "delta"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=\"cp4waiops-
cartridge.lifecycle.input.stories\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Incidents created in time range",
      "type": "stat"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Count that tracks the number of updates incidents have received in time range.",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [],
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 7,
        "x": 17,
        "y": 23
      },
      "id": 331,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
```

```
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "delta"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=\"cp4waiops-
cartridge.irdatalayer.changes.stories\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Incidents updated in time range",
      "type": "stat"
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 28
      },
      "id": 292,
      "panels": [],
      "title": "Logs",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Shows the per-second rate of logs ingested into the system over a 1m average. Log integrations can
be created in the Cloud Pak for AIOps integrations page.",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "noValue": "No logs ingested in time range",
          "thresholds": {
            "mode": "absolute",
            "steps": [
```

```
            {
              "color": "green",
              "value": null
            }
          ]
        },
        "unit": "logs/s"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 5,
      "w": 17,
      "x": 0,
      "y": 29
    },
    "id": 293,
    "options": {
      "legend": {
        "calcs": [],
        "displayMode": "list",
        "placement": "bottom",
        "showLegend": true
      },
      "tooltip": {
        "mode": "single",
        "sort": "none"
      }
    },
    "targets": [
      {
        "datasource": {
          "type": "prometheus",
          "uid": "${DS_PROMETHEUS}"
        },
        "editorMode": "code",
        "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-cartridge-logs-.*-raw\"}
[1m]))",
        "legendFormat": "Incoming logs",
        "range": true,
        "refId": "A"
      }
    ],
    "title": "Log rate",
    "type": "timeseries"
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "${DS_PROMETHEUS}"
    },
    "description": "",
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "mappings": [],
        "noValue": "0",
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green",
              "value": null
            }
          ]
        },
        "unit": "short"
      },
      "overrides": []
    },
    "gridPos": {
      "h": 5,
      "w": 7,
      "x": 17,
      "y": 29
    },
    "id": 294,
    "options": {
      "colorMode": "value",
      "graphMode": "area",
      "justifyMode": "auto",
      "orientation": "auto",
      "reduceOptions": {
        "calcs": [
          "delta"
        ],
        "fields": "",
        "values": false
      },
      "showPercentChange": false,
```

```
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-cartridge-logs-.*-
raw\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Logs ingested in time range",
      "type": "stat"
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 34
      },
      "id": 296,
      "panels": [],
      "title": "Metrics",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Shows the per-second rate of metrics ingested into the system over a 1m average. Metric
integrations can be created in the Cloud Pak for AIOps integrations page.",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "noValue": "No metrics ingested in time range",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "metrics/s"
        },
        "overrides": []
```

```
      },
      "gridPos": {
        "h": 5,
        "w": 17,
        "x": 0,
        "y": 35
      },
      "id": 297,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "mode": "single",
          "sort": "none"
        }
      },
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-
cartridge.analyticsorchestrator.metrics.itsm.raw\"}[1m]))",
          "legendFormat": "Incoming metrics",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Metrics rate",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [],
          "noValue": "0",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 7,
        "x": 17,
        "y": 35
      },
      "id": 298,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "delta"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
```

```
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-
cartridge.analyticsorchestrator.metrics.itsm.raw\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Metrics ingested in time range",
      "type": "stat"
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 40
      },
      "id": 300,
      "panels": [],
      "title": "Runbooks",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Shows the per-second rate of runbooks executed in the system over a 1m average. Runbooks can be
created and configured under the Cloud Pak for AIOps Automations page.",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "noValue": "No runbooks executed in time range",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "executions/s"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 17,
        "x": 0,
        "y": 41
      },
      "id": 301,
      "options": {
```

```
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "mode": "single",
          "sort": "none"
        }
      },
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-
cartridge.lifecycle.output.rba-exec-requests\"}[1m]))",
          "legendFormat": "Runbook executions",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Runbook execution rate",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [],
          "noValue": "0",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 7,
        "x": 17,
        "y": 41
      },
      "id": 302,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "delta"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-
cartridge.lifecycle.output.rba-exec-requests\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
```

```
      "title": "Runbooks executed in time range",
      "type": "stat"
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 46
      },
      "id": 308,
      "panels": [],
      "title": "ServiceNow Integration",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Shows the per-second rate of requests flowing from ServiceNow to Cloud Pak for AIOps* over a 1m
average. This includes: historic incident/change request records, alert updates, incident updates, and new change
requests. ServiceNow integrations can be created in the Cloud Pak for AIOps integrations page.\n\n\\* All API calls
originate from Cloud Pak for AIOps, but 2-way traffic is simulated via polling.",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "noValue": "No inbound ServiceNow traffic ingested in time range.",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "reqps"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 17,
        "x": 0,
        "y": 47
      },
      "id": 309,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
```

```
          "mode": "single",
          "sort": "none"
        }
      },
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-cartridge.incident\"}
[1m]))",
          "legendFormat": "Inbound ServiceNow traffic",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Inbound ServiceNow traffic rate",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Count that tracks the number of requests flowing from ServiceNow to Cloud Pak for AIOps* in time
range. This includes: historic incident/change request records, alert updates, incident updates, and new change requests.
ServiceNow integrations can be created in the Cloud Pak for AIOps integrations page.\n\n\\* All API calls originate from
Cloud Pak for AIOps, but 2-way traffic is simulated via polling.",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [],
          "noValue": "0",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 7,
        "x": 17,
        "y": 47
      },
      "id": 310,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "delta"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-
cartridge.incident\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Inbound ServiceNow traffic",
      "type": "stat"
    },
    {
```

```
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Shows the per-second rate of requests flowing from Cloud Pak for AIOps to ServiceNow over a 1m
average. This includes: incident create/update, alert update, & change risk assessment requests",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "noValue": "No outbound ServiceNow traffic produced in time range.",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "reqps"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 17,
        "x": 0,
        "y": 52
      },
      "id": 311,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "mode": "single",
          "sort": "none"
        }
      },
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-cartridge.snow-
handlers\"}[1m]))",
          "legendFormat": "Outbound ServiceNow traffic",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Outbound ServiceNow traffic rate",
```

```
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Count that tracks the number of requests flowing from Cloud Pak for AIOps to ServiceNow in time
range. This includes: incident create/update, alert update, & change risk assessment messages",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [],
          "noValue": "0",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 7,
        "x": 17,
        "y": 52
      },
      "id": 312,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "delta"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-cartridge.snow-
handlers\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Outbound ServiceNow traffic",
      "type": "stat"
    },
    {
      "collapsed": false,
      "gridPos": {
        "h": 1,
        "w": 24,
        "x": 0,
        "y": 57
      },
      "id": 326,
      "panels": [],
      "title": "Tickets (Jira and Github only)",
      "type": "row"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
      "description": "Shows the per-second rate of Jira/Github issues created by Cloud Pak for AIOps. over a 1m
average\n\nWARNING: This metric will not be accurate if a ServiceNow integration is also in use. This issue will be
corrected in a future release of the product. (ServiceNow metrics are not affected.)",
      "fieldConfig": {
```

```
        "defaults": {
          "color": {
            "mode": "palette-classic"
          },
          "custom": {
            "axisBorderShow": false,
            "axisCenteredZero": false,
            "axisColorMode": "text",
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 0,
            "gradientMode": "none",
            "hideFrom": {
              "legend": false,
              "tooltip": false,
              "viz": false
            },
            "insertNulls": false,
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
              "type": "linear"
            },
            "showPoints": "auto",
            "spanNulls": false,
            "stacking": {
              "group": "A",
              "mode": "none"
            },
            "thresholdsStyle": {
              "mode": "off"
            }
          },
          "mappings": [],
          "noValue": "No Github/Jira issues created in time range.",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "tickets/s"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 17,
        "x": 0,
        "y": 58
      },
      "id": 323,
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },
        "tooltip": {
          "mode": "single",
          "sort": "none"
        }
      },
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum (rate(kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-
cartridge.itsmincidentresponse\"}[1m]))",
          "legendFormat": "Ticket creation",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Ticket creation rate",
      "type": "timeseries"
    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "${DS_PROMETHEUS}"
      },
```

```
      "description": "Count that tracks the number of Jira/Github issues created by Cloud Pak for AIOps in time
range.\n\nWARNING: This metric will not be accurate if a ServiceNow integration is also in use. This issue will be
corrected in a future release of the product. (ServiceNow metrics are not affected.)",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "mappings": [],
          "noValue": "0",
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              }
            ]
          },
          "unit": "short"
        },
        "overrides": []
      },
      "gridPos": {
        "h": 5,
        "w": 7,
        "x": 17,
        "y": 58
      },
      "id": 324,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {
          "calcs": [
            "delta"
          ],
          "fields": "",
          "values": false
        },
        "showPercentChange": false,
        "textMode": "auto",
        "wideLayout": true
      },
      "pluginVersion": "10.4.3",
      "targets": [
        {
          "datasource": {
            "type": "prometheus",
            "uid": "${DS_PROMETHEUS}"
          },
          "editorMode": "code",
          "expr": "sum by(topic) (kafka_server_brokertopicmetrics_messagesin_total{topic=~\"cp4waiops-
cartridge.itsmincidentresponse\"})",
          "legendFormat": "__auto",
          "range": true,
          "refId": "A"
        }
      ],
      "title": "Tickets created in time range",
      "type": "stat"
    }
  ],
  "refresh": "",
  "revision": 1,
  "schemaVersion": 39,
  "tags": [],
  "templating": {
    "list": []
  },
  "time": {
    "from": "now-15m",
    "to": "now"
  },
  "timepicker": {},
  "timezone": "",
  "title": "Cloud Pak for AIOps Usage",
  "uid": "26d422b9-3167-43e2-915c-6dd2393fe14b",
  "version": 3,
  "weekStart": ""
}
```


Monitoring data processed

View the volume of data that is processed by Cloud Pak for AIOps to see whether the OpenShift Container Platform cluster is sized properly.

Use the following topics to view the volume of data that is processed by Cloud Pak for AIOps.

Events

For a rough estimate of the number of events, go to the AIOps Insights dashboard. The AIOps Insights dashboard can be accessed from the administration page of IBM Cloud Pak for AIOps, under **Overview** > **Quick navigation** > **AIOps insights**. Alternatively, click the navigation icon (four horizontal bars) to go to the main navigation menu. Then, click **Operate** > **AIOps insights**.

The noise reduction graph shows the number of events over the selected time period, such as 24 hours, 7 days, or 30 days. You can use the average of the events when you use the Cloud Pak for AIOps deployment sizing.



Figure. AIOps Insights dashboard

For more information about precise distribution of events, see Enabling monitoring for user-defined projects.

When the OpenShift user workload monitoring is enabled, you can view the metrics in the OpenShift dashboard.

The following Prometheus query can be used to view the number of events, which can be added in the Cloud Pak for AIOps console console under **Observe** > **Metrics**:

```
sum(irate((flink_taskmanager_job_task_operator_KafkaSourceReader_topic_partition_currentOffset{topic="cp4waiops_cartridge_lifecycle_input_events"}>0)[1m:30s]))
```

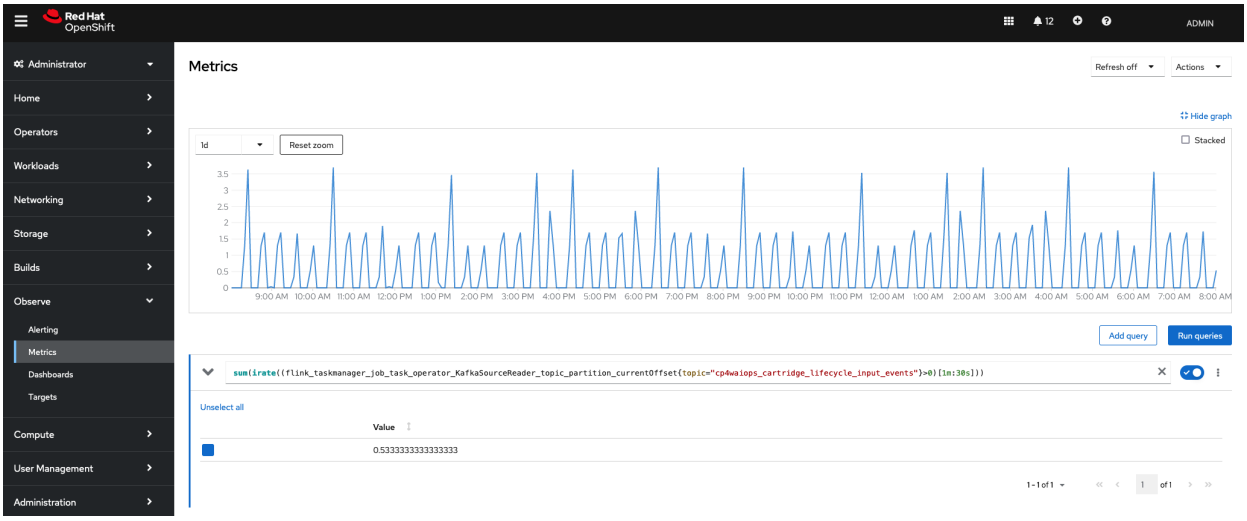


Figure. OpenShift Metrics

Topology

To view the number of resources, click **Resource Management**, and view the resource count. For example, you can see 864 resources in the following example.

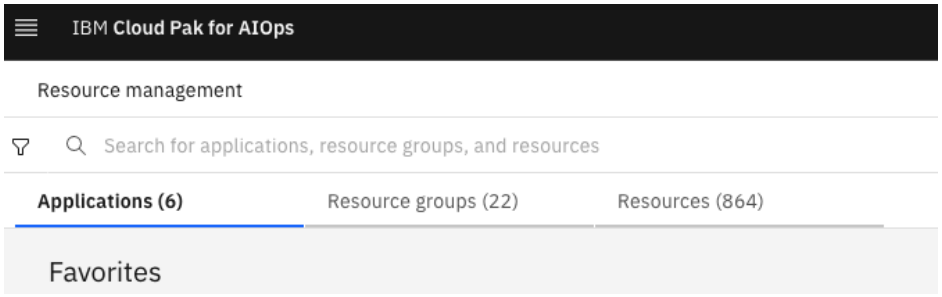


Figure. Resource management page

For more information about the Resource management, see The Resource management page.

Logs

To view the number of logs, extract the count from the Elastic backend:

1. Log in to your Red Hat® OpenShift® cluster, or SSH into a control plane node or the loadbalancer VM on your Linux® deployment. Then swtich to the namespace where IBM Cloud Pak for AIOps is installed.

   ```
   oc project <project>
   ```

   Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.

2. Run the following commands to determine the names of the log data indices:

   ```
   export ELASTIC_SET="$(oc get statefulset -l app.kubernetes.io/managed-by=ibm-elasticsearch -o
   jsonpath='{.items[0].metadata.name}')"
   oc exec "statefulset/${ELASTIC_SET}" -- curl -XGET -k --fail "http://127.0.0.1:19200/_cat/indices?v" | grep logtrain
   ```

   Example output, where the index name is **1000-1000-20220302-logtrain**:

   ```
   green open 1000-1000-20220302-logtrain        eEvaBufQ06yG074hmeHKG  8 1 244 6 72.7kb 72.7kb
   ```

3. For each log train index, get the document count using the following command:

   ```
   export INDEX_NAME=<index_name>
   export ELASTIC_SET="$(oc get statefulset -l app.kubernetes.io/managed-by=ibm-elasticsearch -o
   jsonpath='{.items[0].metadata.name}')"
   oc exec "statefulset/${ELASTIC_SET}" -- curl -XGET -k --fail "http://127.0.0.1:19200/${INDEX_NAME}/_count"
   ```

   Where **<index_name>** is an index name that you found in the previous step.

The outputs of the preceding curl commands provide the daily count of the logs processed. To align with the Cloud Pak for AIOps deployment sizing, you can average the daily count of logs, or pick the maximum value from both of the curl commands.

Metrics

To view the number of Metric KPIs being tracked, click **AI Model Management** > **Metric anomaly detection**.

The number of KPIs is noted under the **Metrics analyzed** field. For example, 146 metrics were analyzed in the following example.

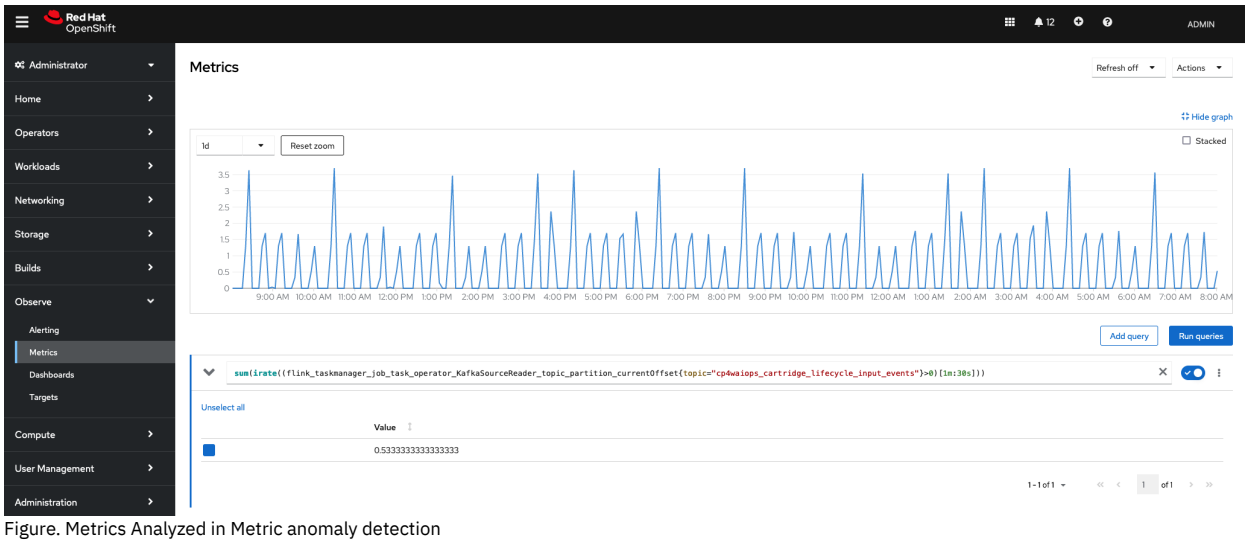


Figure. Metrics Analyzed in Metric anomaly detection

For more information about Metric anomaly detection, see the following links:

- About Metric anomaly detection.

- Setting up training for Metric anomaly detection.

Circuit breaker for Kafka robustness and stability

Enable Cloud Pak for AIOps to protect itself and remain up and running if an event storm occurs. You can implement a circuit breaker pattern to avoid Cloud Pak for AIOps failing and ensure that events, alerts, and incidents continue to be processed.

The following sections show the end-to-end flow of configuring a circuit breaker for Kafka robustness:

- Configuring prometheus.
- Creating pod monitors.
- Creating prometheus rules for generating alerts specific to Cloud Pak for AIOps Kafka storage and health.
- Push alerts generated in OpenShift to Cloud Pak for AIOps by using a webhook integration.
- Create Cloud Pak for AIOps policies to generate incidents.
- Integrate incidents with Cloud Pak for AIOps automations.

Note: There is some overlap with the initial two steps of these instructions and the procedure for Monitoring with Prometheus and Grafana.

Use cases for circuit breaker

- Monitoring the health of Kafka for restarts, failures, and so on. Sending a high priority alert that different components can take action on.
- Monitor and react to the amount of storage being used by the Kafka PVC.
- Too much data to Kafka and the databases it feeds into runs out of storage.

View the following sub-topics for details on implementing a circuit breaker pattern for Kafka:

- Creating a cluster monitoring configmap object
- Creating a user-defined workload monitoring configmap
- Configuring core OpenShift Container Platform monitoring components
- Configuring components that monitor user-defined projects
- Creating Prometheus rule to generate OpenShift alerts
- Push alerts that are generated in OpenShift into Cloud Pak for AIOps by using a webhook integration
- Creating a policy to promote critical Prometheus alerts to an incident

Note: There is some overlap with the initial two steps of these instructions and the procedure for Monitoring with Prometheus and Grafana.

Creating a cluster monitoring configmap object

You can configure the core OpenShift Container Platform monitoring components by creating the **`cluster-monitoring-config`** configmap object in the **`openshift-monitoring`** project. The Cluster Monitoring Operator (CMO) then configures the core components of the monitoring stack.

Prerequisites

- You must have access to the cluster as a user with the **`cluster-admin`** role.
- OpenShift command-line interface is required. For more information, see Getting started with the OpenShift CLI.

Procedure

- Check if the cluster-monitoring-config configmap object exists:

  ```
  oc -n openshift-monitoring get configmap cluster-monitoring-config
  ```

- If the configmap object does not exist:

  Create the following YAML manifest. In this example, the file is called cluster-monitoring-config.yaml:

  ```
  apiVersion: v1
  kind: ConfigMap
  metadata:
    name: cluster-monitoring-config
    namespace: openshift-monitoring
  data:
    config.yaml: |
  ```

  - Apply the configuration to create the configmap object:

    ```
    oc apply -f cluster-monitoring-config.yaml
    ```

Creating a user-defined workload monitoring configmap

You can configure the user workload monitoring components with the **user-workload-monitoring-config** configmap object in the **openshift-user-workload-monitoring** project. The Cluster Monitoring Operator (CMO) then configures the components that monitor user-defined projects.

Prerequisites

- You must have access to the cluster as a user with the **cluster-admin** role.
- OpenShift command-line interface is required. For more information, see Getting started with the OpenShift CLI.

Procedure

- Check if the **user-workload-monitoring-config** configmap object exists:

  ```
  oc -n openshift-user-workload-monitoring get configmap user-workload-monitoring-config
  ```

- If the **user-workload-monitoring-config** configmap object does not exist:

  Create the following YAML manifest. In this example, the file is called **user-workload-monitoring-config.yaml**:

  ```
  apiVersion: v1
  kind: ConfigMap
  metadata:
    name: user-workload-monitoring-config
    namespace: openshift-user-workload-monitoring
  data:
    config.yaml: |
  ```

  - Apply the configuration to create the configmap object:

    ```
    oc apply -f user-workload-monitoring-config.yaml
    ```

Configuring core OpenShift Container Platform monitoring components

- Edit the **cluster-monitoring-config** configmap object in the **openshift-monitoring** project.
- Add your configuration under data/config.yaml as a key-value pair <component_name>: <component_configuration>.

  For more information about configurations options, see https://docs.redhat.com/documentation/openshift_container_platform/4.17/html/monitoring/config-map-reference-for-the-cluster-monitoring-operator#cluster-monitoring-operator-configuration-reference.

- **Configmap updates**: enables userworkload and object configures a persistent volume claim (PVC) for Prometheus. This relates to the Prometheus instance that monitors core OpenShift Container Platform components only. Get the default OpenShift Container Platform storage class from the script below and then run the update command:

  ```
  if [[ -z "${PROJECT_CP4AIOPS}" ]]; then
    echo "ERROR: PROJECT_CP4AIOPS is not set"
    exit 1
  fi

  if [[ -z "${OCP_MONITOR_STORAGE_CLASS}" ]]; then
    OCP_MONITOR_STORAGE_CLASS=$(oc get sc -o jsonpath='{.items[?
  (@.metadata.annotations.storageclass\.kubernetes\.io/is-default-class=="true")].metadata.name}')
    if [[ -z "${OCP_MONITOR_STORAGE_CLASS}" ]]; then
      OCP_MONITOR_STORAGE_CLASS=$(oc get sc -o jsonpath='{range .items[*]}{.metadata.name}{"\n"}{end}' | grep 'rbd$')
      if [[ -z "${OCP_MONITOR_STORAGE_CLASS}" ]]; then
        echo "ERROR: Could not find a suitable storage class for OCP Monitoring"
        exit $LINENO
      fi
    fi
  fi
  ```

- Run the following command before running the script, or replace it with the Cloud Pak for AIOps namespace in the script.

  ```
  export PROJECT_CP4AIOPS=<aiops-namespace>
  ```

- Execute the command to apply the changes:

  ```
  cat << EOF | oc apply --validate -f -
  apiVersion: v1
  kind: ConfigMap
  metadata:
    name: cluster-monitoring-config
    namespace: openshift-monitoring
  data:
    config.yaml: |
      enableUserWorkload: true
      prometheusK8s:
        volumeClaimTemplate:
          spec:
            storageClassName: ${OCP_MONITOR_STORAGE_CLASS}
            volumeMode: Filesystem
            resources:
  ```

```
          requests:
            storage: 40Gi
```

Configuring components that monitor user-defined projects

- Edit the **cluster-monitoring-config** configmap object in the **openshift-monitoring project**:

```
cat << EOF | oc apply --validate -f -
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    prometheus:
      retention: 24h
      resources:
        requests:
          cpu: 200m
          memory: 2Gi
EOF
```

  For more information about **UserWorkloadConfiguration** options, see https://docs.redhat.com/documentation/openshift_container_platform/4.17/html/monitoring/config-map-reference-for-the-cluster-monitoring-operator#userworkloadconfiguration.

  - Defines the Prometheus component and the subsequent lines define its configuration.
  - Configures a twenty-four-hour data retention period for the Prometheus instance that monitors user-defined projects.
  - Defines a minimum resource request of 200 millicores for the Prometheus container.
  - Defines a minimum pod resource request of 2 GiB of memory for the Prometheus container.

Creating Prometheus rule to generate OpenShift alerts

Wait for user monitoring to be running and create a Prometheus rule to generate OpenShift alerts.

- Run the following command before running the script, or replace it with the Cloud Pak for AIOps namespace in the script.

```
export PROJECT_CP4AIOPS=<aiops-namespace>
```

- **Example**: alerts that inform you when the targets are down and when the enforced sample limit is approaching.

```
cat << EOF | oc apply --validate -f -
apiVersion: monitoring.coreos.com/v1
kind: PodMonitor
metadata:
  name: cluster-operator-metrics
  namespace: ${PROJECT_CP4AIOPS}
  labels:
    app: ibmevents.ibm.com
spec:
  selector:
    matchLabels:
      ibmevents.ibm.com/kind: cluster-operator
  namespaceSelector:
    matchNames:
      - ${PROJECT_CP4AIOPS}
  podMetricsEndpoints:
  - path: /metrics
    port: http
---
apiVersion: monitoring.coreos.com/v1
kind: PodMonitor
metadata:
  name: entity-operator-metrics
  namespace: ${PROJECT_CP4AIOPS}
  labels:
    app: ibmevents.ibm.com
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: entity-operator
  namespaceSelector:
    matchNames:
      - ${PROJECT_CP4AIOPS}
  podMetricsEndpoints:
  - path: /metrics
    port: healthcheck
---
apiVersion: monitoring.coreos.com/v1
kind: PodMonitor
metadata:
  name: kafka-resources-metrics
```

```
  namespace: ${PROJECT_CP4AIOPS}
  labels:
    app: ibmevents.ibm.com
spec:
  selector:
    matchExpressions:
      - key: ibmevents.ibm.com/kind
        operator: In
        values: [ Kafka ]
  namespaceSelector:
    matchNames:
      - ${PROJECT_CP4AIOPS}
  podMetricsEndpoints:
  - path: /metrics
    port: tcp-prometheus
    relabelings:
    - separator: ;
      regex: __meta_kubernetes_pod_label_(ibmevents_ibm_com_.+)
      replacement: \$1
      action: labelmap
    - sourceLabels: [__meta_kubernetes_namespace]
      separator: ;
      regex: (.*)
      targetLabel: namespace
      replacement: \$1
      action: replace
    - sourceLabels: [__meta_kubernetes_pod_name]
      separator: ;
      regex: (.*)
      targetLabel: kubernetes_pod_name
      replacement: \$1
      action: replace
    - sourceLabels: [__meta_kubernetes_pod_node_name]
      separator: ;
      regex: (.*)
      targetLabel: node_name
      replacement: \$1
      action: replace
    - sourceLabels: [__meta_kubernetes_pod_host_ip]
      separator: ;
      regex: (.*)
      targetLabel: node_ip
      replacement: \$1
      action: replace
---
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  labels:
    role: alert-rules
    app: events.ibm.com
  name: prometheus-k8s-rules
  namespace: ${PROJECT_CP4AIOPS}
spec:
  groups:
  - name: kafka
    rules:
    - alert: AIOpsKafkaRunningOutOfSpace
      expr: round(kubelet_volume_stats_available_bytes{persistentvolumeclaim=~"data(-[0-9]+)?-(.+)-kafka-[0-9]+"} *
100 / kubelet_volume_stats_capacity_bytes{persistentvolumeclaim=~"data(-[0-9]+)?-(.+)-kafka-[0-9]+"} < 15, 0.01)
      for: 10s
      labels:
        severity: critical
      annotations:
        summary: 'Kafka is running out of free disk space'
        description: |-
          There are only {{ \$value }} percent available at {{ \$labels.persistentvolumeclaim }} PVC
          See https://www.ibm.com/docs/en/cloud-paks/cloud-pak-aiops/4.6.1?topic=pak-adjusting-pvcs
          for how to add additional disk space
    - alert: AIOpsKafkaUnderReplicatedPartitions
      expr: kafka_server_replicamanager_underreplicatedpartitions > 0
      for: 10s
      labels:
        severity: warning
      annotations:
        summary: 'Kafka under replicated partitions'
        description: 'There are {{ \$value }} under replicated partitions on {{ \$labels.kubernetes_pod_name }}'
    - alert: AIOpsKafkaAbnormalControllerState
      expr: sum(kafka_controller_kafkacontroller_activecontrollercount) by (ibmevents_ibm_com_name) != 1
      for: 10s
      labels:
        severity: warning
      annotations:
        summary: 'Kafka abnormal controller state'
        description: 'There are {{ \$value }} active controllers in the cluster'
    - alert: AIOpsKafkaOfflinePartitions
      expr: sum(kafka_controller_kafkacontroller_offlinepartitionscount) > 0
      for: 10s
      labels:
        severity: warning
      annotations:
        summary: 'Kafka offline partitions'
```

```
        description: 'One or more partitions have no leader'
    - alert: AIOpsKafkaUnderMinIsrPartitionCount
      expr: kafka_server_replicamanager_underminisrpartitioncount > 0
      for: 10s
      labels:
        severity: warning
      annotations:
        summary: 'Kafka under min ISR partitions'
        description: 'There are {{ \$value }} partitions under the min ISR on {{ \$labels.kubernetes_pod_name }}'
    - alert: AIOpsKafkaOfflineLogDirectoryCount
      expr: kafka_log_logmanager_offlinelogdirectorycount > 0
      for: 10s
      labels:
        severity: warning
      annotations:
        summary: 'Kafka offline log directories'
        description: 'There are {{ \$value }} offline log directories on {{ \$labels.kubernetes_pod_name }}'
    - alert: AIOpsKafkaScrapeProblem
      expr: up{kubernetes_namespace!~"openshift-.+",kubernetes_pod_name=~".+-kafka-[0-9]+"} == 0
      for: 3m
      labels:
        severity: major
      annotations:
        summary: 'OpenShift monitoring is unable to gather metrics from {{ \$labels.kubernetes_pod_name }}/{{
\$labels.instance }}'
        description: 'OpenShift monitoring was unable to scrape metrics from {{ \$labels.kubernetes_pod_name }}/{{
\$labels.instance }} for more than 3 minutes'
    - alert: AIOpsKafkaClusterOperatorContainerDown
      expr: count((container_last_seen{container="ibm-events-operator"} > (time() - 90))) < 1 or
absent(container_last_seen{container="ibm-events-operator"})
      for: 1m
      labels:
        severity: major
      annotations:
        summary: 'The AIOps Kafka cluster operator down'
        description: 'The AIOps Kafka cluster operator/controller has been down for longer than 90 seconds'
    - alert: AIOpsKafkaBrokerContainersDown
      expr: absent(container_last_seen{container="kafka",pod=~".+-kafka-[0-9]+"})
      for: 3m
      labels:
        severity: major
      annotations:
        summary: 'All kafka containers down or in CrashLookBackOff status'
        description: 'All kafka containers have been down or in CrashLookBackOff status for 3 minutes'
    - alert: AIOpsKafkaContainerRestartedInTheLast5Minutes
      expr: count(count_over_time(container_last_seen{container="kafka"}[5m])) > 2 *
count(container_last_seen{container="kafka",pod=~".+-kafka-[0-9]+"})
      for: 5m
      labels:
        severity: warning
      annotations:
        summary: 'One or more Kafka containers restarted too often'
        description: 'One or more Kafka containers were restarted too often within the last 5 minutes'
  - name: zookeeper
    rules:
    - alert: AIOpsZookeeperAvgRequestLatency
      expr: zookeeper_avgrequestlatency > 10
      for: 10s
      labels:
        severity: warning
      annotations:
        summary: 'Zookeeper average request latency'
        description: 'The average request latency is {{ \$value }} on {{ \$labels.kubernetes_pod_name }}'
    - alert: AIOpsZookeeperOutstandingRequests
      expr: zookeeper_outstandingrequests > 10
      for: 10s
      labels:
        severity: warning
      annotations:
        summary: 'Zookeeper outstanding requests'
        description: 'There are {{ \$value }} outstanding requests on {{ \$labels.kubernetes_pod_name }}'
    - alert: AIOpsZookeeperRunningOutOfSpace
      expr: kubelet_volume_stats_available_bytes{persistentvolumeclaim=~"data-(.+)-zookeeper-[0-9]+"} < 5368709120
      for: 10s
      labels:
        severity: warning
      annotations:
        summary: 'Zookeeper is running out of free disk space'
        description: 'There are only {{ \$value }} bytes available at {{ \$labels.persistentvolumeclaim }} PVC'
    - alert: AIOpsZookeeperContainerRestartedInTheLast5Minutes
      expr: count(count_over_time(container_last_seen{container="zookeeper"}[5m])) > 2 *
count(container_last_seen{container="zookeeper",pod=~".+-zookeeper-[0-9]+"})
      for: 5m
      labels:
        severity: warning
      annotations:
        summary: 'One or more Zookeeper containers were restarted too often'
        description: 'One or more Zookeeper containers were restarted too often within the last 5 minutes. This
alert can be ignored when the Zookeeper cluster is scaling up'
    - alert: AIOpsZookeeperContainersDown
      expr: absent(container_last_seen{container="zookeeper",pod=~".+-zookeeper-[0-9]+"})
      for: 3m
```

```
      labels:
        severity: major
      annotations:
        summary: 'All zookeeper containers in the Zookeeper pods down or in CrashLookBackOff status'
        description: 'All zookeeper containers in the Zookeeper pods have been down or in CrashLookBackOff status
for 3 minutes'
  - name: entityOperator
    rules:
    - alert: AIOpsKafkaTopicOperatorContainerDown
      expr: absent(container_last_seen{container="topic-operator",pod=~".+-entity-operator-.+"})
      for: 3m
      labels:
        severity: major
      annotations:
        summary: 'Container topic-operator in Entity Operator pod down or in CrashLookBackOff status'
        description: 'Container topic-operator in Entity Operator pod has been or in CrashLookBackOff status for 3
minutes'
    - alert: AIOpsKafkaUserOperatorContainerDown
      expr: absent(container_last_seen{container="user-operator",pod=~".+-entity-operator-.+"})
      for: 3m
      labels:
        severity: major
      annotations:
        summary: 'Container user-operator in Entity Operator pod down or in CrashLookBackOff status'
        description: 'Container user-operator in Entity Operator pod have been down or in CrashLookBackOff status
for 3 minutes'
EOF
```

- **Example**: Red Hat OpenShift Container Platform alert when Kafka is running out of space.

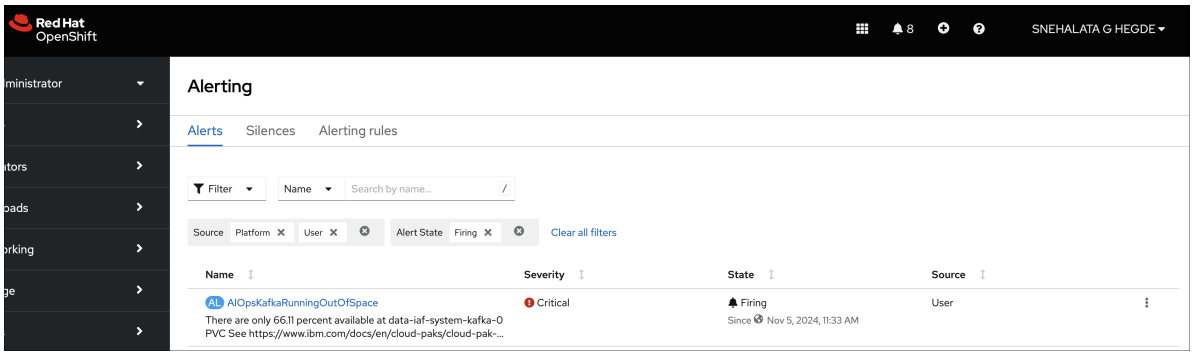


Figure. Red Hat OpenShift Container Platform alert when Kafka is running out of space

Push alerts that are generated in OpenShift into Cloud Pak for AIOps by using a webhook integration

- Follow the steps that are documented in Self-monitoring with alerts.

  Note: You can configure a webhook to only send the **`AIOpsKafka`** related alerts and severity levels.

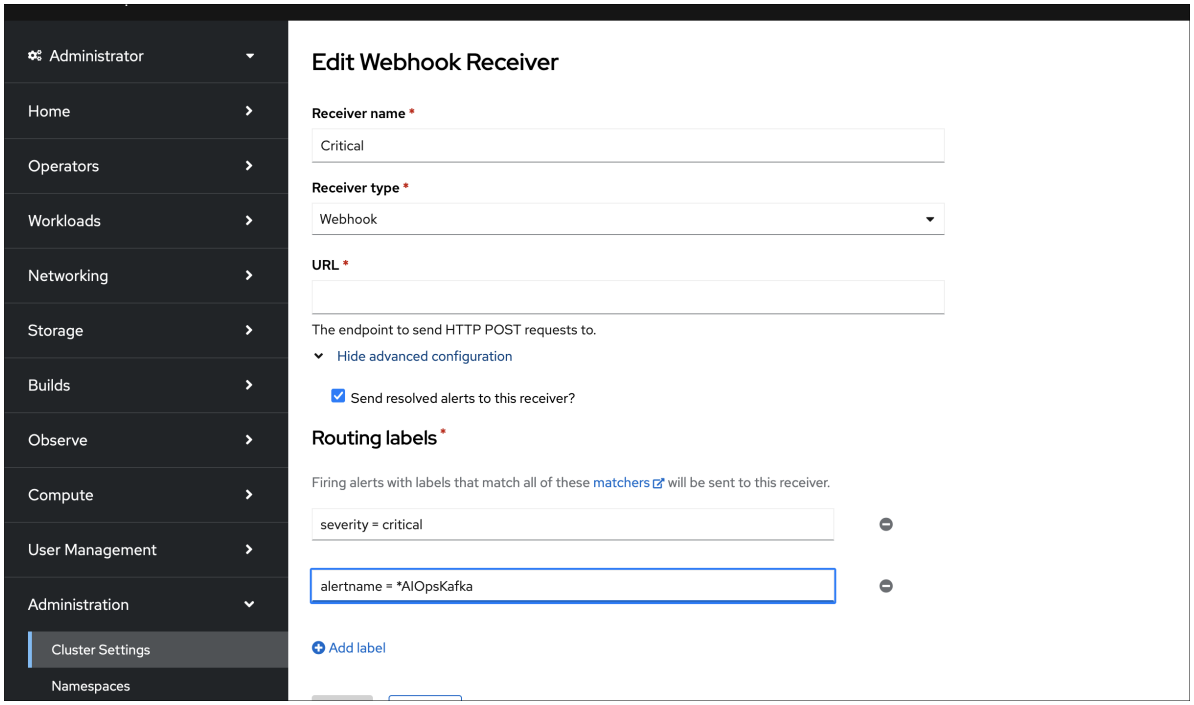


Figure. AIOpsKafka alerts and severity levels

- For the certificate manager to trust the webhook url without any certificate issues, add the Cloud Pak for AIOps CA signing cert to the OpenShift CA bundle.

  For use in insecure mode:

```
receivers:
- name: Default
  webhook_configs:
    - url: >-
        https://whconn-6d6f3a29-ddaf-4d5f-938b-6935791b5144-cp4waiops.apps.groot.cp.fyre.ibm.com/webhook-
connector/6oon1rwn4vn
      http_config:
        tls_config:
          insecure_skip_verify: true
- name: Watchdog
- name: Critical
  webhook_configs:
    - url: >-
        https://whconn-6d6f3a29-ddaf-4d5f-938b-6935791b5144-cp4waiops.apps.groot.cp.fyre.ibm.com/webhook-
connector/6oon1rwn4vn
      http_config:
        tls_config:
          insecure_skip_verify: true
```

- Alert surfacing in Cloud Pak for AIOps:

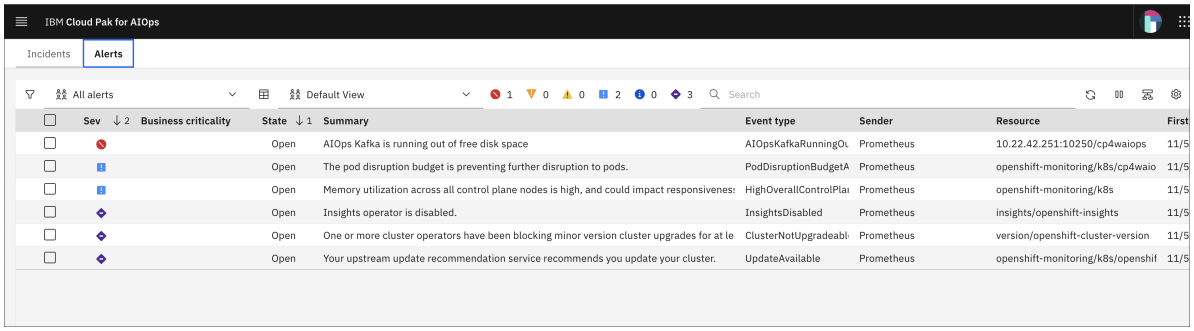


Figure. Alert surfacing in Cloud Pak for AIOps

- Alert details in Cloud Pak for AIOps:

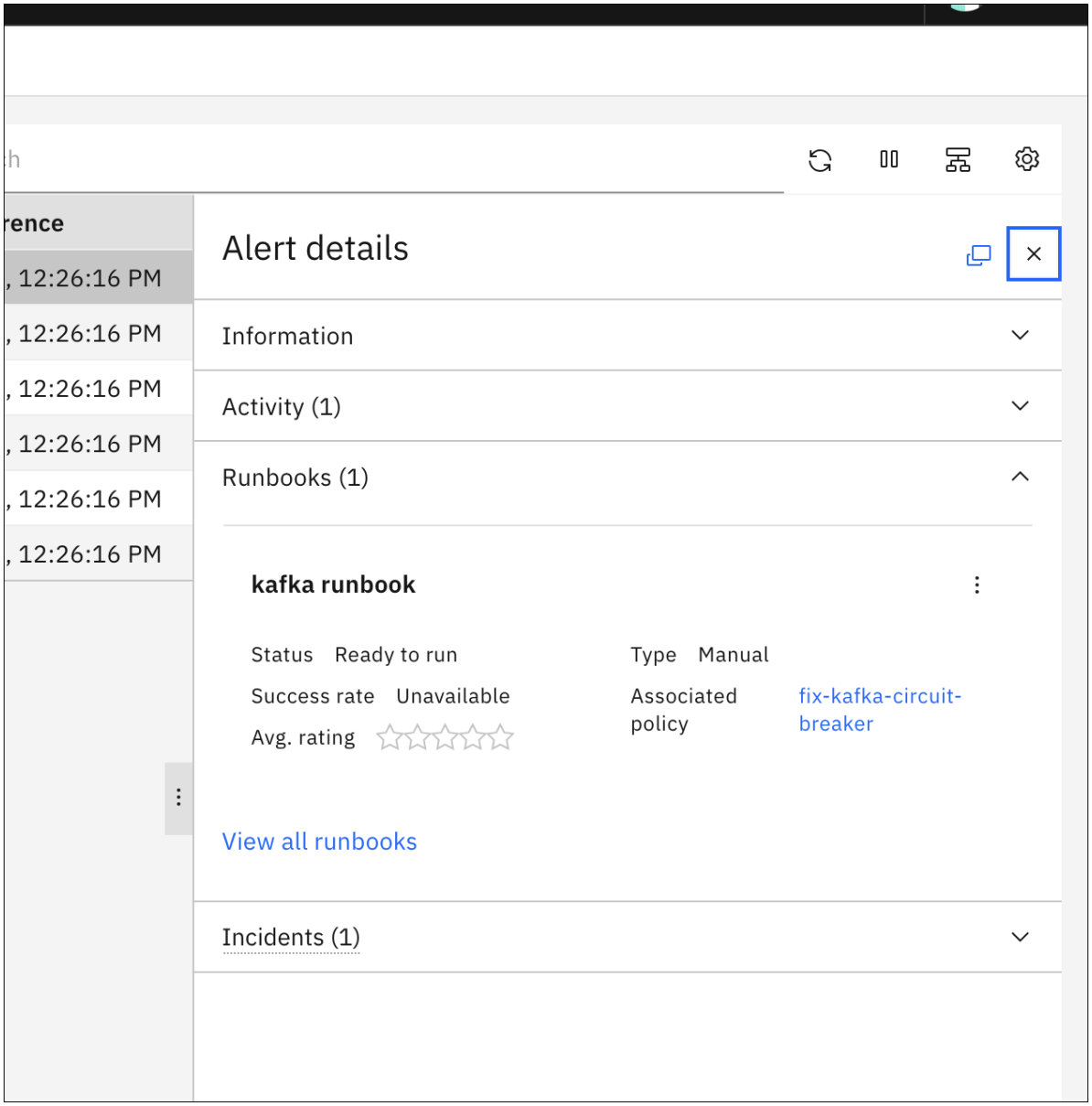


Figure. Cloud Pak for AIOps alert details

Creating a policy to promote critical Prometheus alerts to an incident

Create a policy to group the critical alerts from Prometheus that are related to Kafka and generate an incident in Cloud Pak for AIOps.

- Create a policy to Promote alerts to an incident.

  Example:

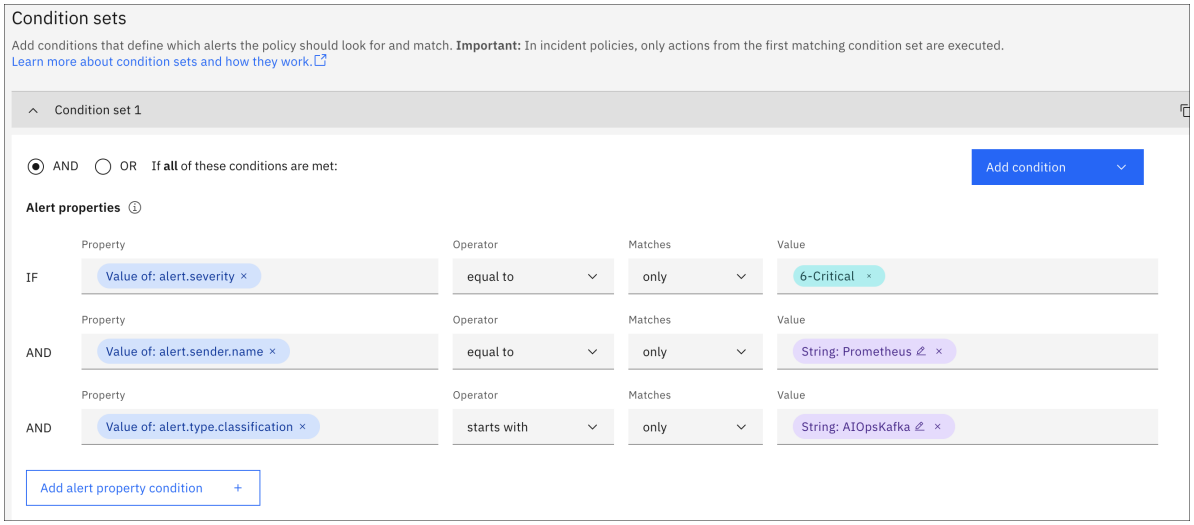


Figure. Policy to promote critical Prometheus alerts to an incident

Integrate the generated incidents with Cloud Pak for AIOps automations

- Create a policy to Assign a runbook to alerts to assign a runbook with the same set of conditions.
- An Incident is created when the alerts are generated. Details of runbooks are included in the alert details.
- Actions to stabilize examples: disable data flow for integrations, disable an existing integrations, and so on.

Certificate management (IBM Cloud Pak for AIOps on OpenShift)

With IBM Cloud Pak for AIOps, you can use IBM Cert Manager to manage your certificates or a certificate manager of your choosing. IBM Cert Manager is a service provided with IBM Cloud Pak® foundational services.

The procedure for installing a certificate for IBM Cloud Pak for AIOps depends on the certificate or endpoint that you want to secure.

- Red Hat OpenShift Container Platform certificates
- IBM Cloud Pak foundational services IBM Cert Manager certificates

Red Hat OpenShift Container Platform certificates

Red Hat OpenShift Container Platform cluster certificate

Once you have a certificate manager, you can then use the Red Hat OpenShift Container Platform cluster certificate or your own custom certificate for IBM Cloud Pak for AIOps (or a standalone installation of Infrastructure Automation) instead of the Red Hat OpenShift Container Platform. For more information, see Using a custom certificate.

Red Hat OpenShift Container Platform console certificate

This certificate is for the Red Hat OpenShift Container Platform console and is typically for use of only the cluster administrator, rather than other IBM Cloud Pak for AIOps users. It is critical that the certificate does not include the **`cp-console`** or the **`cpd-aiops`** routes in the certificate.

The default ingress certificate for Red Hat OpenShift Container Platform can have only a SAN of:

```
DNS.1 = *.apps.mycluster.example.com
```

For more information, see Red Hat OpenShift Container Platform certificate.

IBM Cloud Pak foundational services - IBM Cert Manager certificates

To add and manage certificates for IBM Cloud Pak for AIOps, you can use IBM Cert Manager, which is included with IBM Cloud Pak foundational services. With IBM Cert Manager installed, you can begin to use the available capabilities to add and manage your certificates. The IBM Cloud Pak for AIOps installation procedures include steps to optionally install IBM Cloud Pak foundational services Cert Manager.

For more information about the IBM Cloud Pak foundational services Cert Manager, see Using IBM Certificate manager (cert-manager).

Using a custom certificate

Learn how to add a custom certificate to a deployment of IBM Cloud Pak for AIOps or Infrastructure Automation on Red Hat OpenShift.

If you want to use your own custom certificate for IBM Cloud Pak® for AIOps or Infrastructure Automation instead of the Red Hat® OpenShift® Container Platform cluster certificate, then complete the following steps after you have installed IBM Cloud Pak® for AIOps or Infrastructure Automation.

1. Export an environment variable containing your project's name.

   ```
   export PROJECT=<project>
   ```

   Where **<project>** is the project (namespace) where IBM Cloud Pak for AIOps or Infrastructure Automation is deployed.

2. Ensure that you have created the following four PEM-encoded certificate files.

   - **cachain.pem**: A certificate chain containing the root and intermediate certificates.
   - **carootcert.pem**: The root certificate of your Certificate Authority (CA), with no intermediate or application certificates.
   - **aiops.key.pem**: A key file for the signed certificate in **aiops.pem**.
   - **aiops.pem**: A IBM Cloud Pak for AIOps or Infrastructure Automation certificate, which includes the two fully qualified domain names (FQDNs) for **cpd-${PROJECT}** and **cp-console-${PROJECT}** in the Subject Alternative Name (SAN) list.

   You can find the FQDNs for **cp-console-${PROJECT}** and **cpd-${PROJECT}** with the following commands.

   ```
   oc get route cp-console -n ${PROJECT} -o jsonpath="{.spec.host}"
   oc get route cpd -n ${PROJECT} -o jsonpath="{.spec.host}"
   ```

3. Add your root CA to the CA bundle, to ensure that it is trusted across the Red Hat OpenShift cluster.

   For more information, see the instructions in Replacing the CA Bundle certificate.

4. Create a secret called **custom-tls-secret** with your custom certificates.

   ```
   oc project ${PROJECT}
   oc create secret generic custom-tls-secret --from-file=ca.crt=./cachain.pem --from-file=tls.crt=./aiops.pem --from-
   file=tls.key=./aiops.key.pem
   ```

5. Create a file named **cs-onprem-tenant-config.yaml** with the following contents:

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: cs-onprem-tenant-config
     labels:
       cs_onprem_tenant_config: "true"
   data:
     custom_host_certificate_secret: custom-tls-secret
   ```

6. Run the following command to create the ConfigMap:

   ```
   oc apply -f cs-onprem-tenant-config.yaml -n $PROJECT
   ```

7. Restart the **nginx** pods to pickup the new custom certificates.

   ```
   oc delete pod -l component=ibm-nginx
   ```

8. Patch **ZenService** with the new **custom-tls-secret**.

   ```
   oc patch ZenService/iaf-zen-cpdservice --type merge -p '{"spec":{"zenCustomRoute":
   {"route_reencrypt":true,"route_secret": "custom-tls-secret"}}}'
   ```

9. Patch the **cp-console** route with the new **custom-tls-secret**.

   ```
   oc patch route cp-console -p '{"spec":{"tls":{"certificate":"'"$(oc get secret custom-tls-secret -o
   jsonpath='{.data.tls\.crt}' | base64 -d | tr '\n' ':' | sed 's/:/\\n/g')"'", "key":"'"$(oc get secret custom-tls-
   secret -o jsonpath='{.data.tls\.key}'| base64 -d | tr '\n' ':' | sed 's/:/\\n/g')"'", "caCertificate":"'"$(oc get
   secret custom-tls-secret -o jsonpath='{.data.ca\.crt}' | base64 -d | tr '\n' ':' | sed 's/:/\\n/g')"'"}}}'
   ```

10. Run the following commands:

    ```
    oc -n ${PROJECT} delete pod -l app=platform-auth-service
    oc -n ${PROJECT} delete pod -l app=platform-identity-provider
    oc -n ${PROJECT} delete pod -l name=ibm-zen-operator
    oc -n ${PROJECT} delete pod -l component=usermgmt
    oc -n ${PROJECT} delete pod -l name=operand-deployment-lifecycle-manager
    ```

11. Wait for around 15 minutes for these updates to be reconciled, and for the new configuration to be in place.

    You can check the progress of the reconciliation with the following commands:

    ```
    oc get zenservice iaf-zen-cpdservice -o jsonpath='{.status.Progress}'
    oc get zenservice iaf-zen-cpdservice -o jsonpath='{.status.ProgressMessage}'
    ```

    When **Progress** is shown as *100%*, then reconciliation is complete and you can log in to the console.

Rebalancing workloads on a multi-zone architecture (IBM Cloud Pak for AIOps on OpenShift)

BETA The use of the multi-zone high availability (HA) deployment architecture for IBM Cloud Pak for AIOps is available only for technology preview, and is not for production usage.

Use these steps to manually redistribute IBM Cloud Pak® for AIOps workloads across a multi-zone Red Hat® OpenShift® Container Platform cluster.

Overview

For more information about deploying IBM Cloud Pak for AIOps on a multi-zone cluster, see Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HA).

The distribution of IBM Cloud Pak for AIOps workloads across the zones in your multi-zone cluster is important to help ensure high availability. You might need to rebalance your workloads manually to achieve an optimal distribution.

If a zone outage occurs in a multi-zone cluster, Red Hat OpenShift automatically reschedules IBM Cloud Pak for AIOps workloads on the affected zone to alternative available zones. This helps IBM Cloud Pak for AIOps to continue functioning during a zone outage, but it means that workloads are not optimally distributed when the affected zone is restored. Rebalancing is then needed to help ensure that all available zones have sufficient replicas to maintain high availability if another zone outage occurs. Red Hat OpenShift does not automatically rebalance workloads because it can be disruptive and cause further problems. Rebalancing must be done manually when convenient, for example during a scheduled maintenance window.

Procedure

Use the following steps to correct the distribution of workloads, for example after the restoration of a failed zone.

1. Check whether all the recovered nodes are healthy.

   Run the following command to validate the readiness of all nodes:

   ```
   oc get nodes
   ```

   Example output:

   ```
   $ oc get nodes
   NAME                 STATUS   ROLES                  AGE    VERSION
   master0.acme.com     Ready    control-plane,master   4d3h   v1.27.13+048520e
   master1.acme.com     Ready    control-plane,master   4d3h   v1.27.13+048520e
   master2.acme.com     Ready    control-plane,master   4d3h   v1.27.13+048520e
   worker0.acme.com     Ready    infra,worker           4d3h   v1.27.13+048520e
   worker1.acme.com     Ready    infra,worker           4d3h   v1.27.13+048520e
   worker10.acme.com    Ready    worker                 4d3h   v1.27.13+048520e
   worker11.acme.com    Ready    worker                 4d3h   v1.27.13+048520e
   worker12.acme.com    Ready    worker                 4d3h   v1.27.13+048520e
   worker13.acme.com    Ready    worker                 4d3h   v1.27.13+048520e
   worker14.acme.com    Ready    worker                 4d3h   v1.27.13+048520e
   worker2.acme.com     Ready    infra,worker           4d3h   v1.27.13+048520e
   worker3.acme.com     Ready    worker                 4d3h   v1.27.13+048520e
   worker4.acme.com     Ready    worker                 4d3h   v1.27.13+048520e
   worker5.acme.com     Ready    worker                 4d3h   v1.27.13+048520e
   worker6.acme.com     Ready    worker                 4d3h   v1.27.13+048520e
   worker7.acme.com     Ready    worker                 4d3h   v1.27.13+048520e
   worker8.acme.com     Ready    worker                 4d3h   v1.27.13+048520e
   worker9.acme.com     Ready    worker                 4d3h   v1.27.13+048520e
   ```

   Do not proceed until all nodes have a **STATUS** of **Ready**.

2. Assess whether workloads are distributed well.

   Download the status checker plug-in from github.com/IBM. Run the following command to assess the distribution of IBM Cloud Pak for AIOps workloads across your cluster zones:

   ```
   oc waiops multizone status
   ```

   No further action is needed if the script reports **INFO: All pods are distributed appropriately for multizone.**

   Example output where pods are not optimally distributed across zones:

   ```
   $ oc waiops multizone status

   Regions : CentralNC
   Zones   : ChapelHill Durham Raleigh

   NAME                 ZONE          REGION
   worker11.acme.com    ChapelHill    CentralNC
   worker14.acme.com    ChapelHill    CentralNC
   worker2.acme.com     ChapelHill    CentralNC
   worker5.acme.com     ChapelHill    CentralNC
   worker8.acme.com     ChapelHill    CentralNC
   worker1.acme.com     Durham        CentralNC
   worker10.acme.com    Durham        CentralNC
   worker13.acme.com    Durham        CentralNC
   worker4.acme.com     Durham        CentralNC
   worker7.acme.com     Durham        CentralNC
   worker0.acme.com     Raleigh       CentralNC
   worker12.acme.com    Raleigh       CentralNC
   worker3.acme.com     Raleigh       CentralNC
   worker6.acme.com     Raleigh       CentralNC
   worker9.acme.com     Raleigh       CentralNC

   INFO: Checking the namespace aiops.................................done.

   WARNING: Not all pods are distributed optimally.
   ```

```
NAMESPACE   NAME                                                  ZONE         NODE                KIND
aiops       aimanager-aio-log-anomaly-detector-7554c44d69-4ztcb   Durham       worker7.acme.com    Deployment
aiops       aimanager-aio-log-anomaly-detector-7554c44d69-tsq87   Durham       worker13.acme.com   Deployment
aiops       aimanager-aio-log-anomaly-detector-7554c44d69-27w6h   Raleigh      worker0.acme.com    Deployment
aiops       aimanager-aio-log-anomaly-detector-7554c44d69-47hf9   Raleigh      worker12.acme.com   Deployment
aiops       aiops-base-ui-59d8b785d8-g7v58                        Durham       worker7.acme.com    Deployment
aiops       aiops-base-ui-59d8b785d8-f2mwb                        Raleigh      worker6.acme.com    Deployment
aiops       aiops-base-ui-59d8b785d8-hbv7s                        Raleigh      worker0.acme.com    Deployment
aiops       aiops-ir-lifecycle-datarouting-66b7fdd5df-4ml7r       Raleigh      worker9.acme.com    Deployment
aiops       aiops-ir-lifecycle-datarouting-66b7fdd5df-l5618       Raleigh      worker12.acme.com   Deployment
aiops       aiops-ir-lifecycle-policy-registry-svc-7cd8dcbdff-h4zcv   Durham   worker10.acme.com   Deployment
aiops       aiops-ir-lifecycle-policy-registry-svc-7cd8dcbdff-q584z   Durham   worker4.acme.com    Deployment
aiops       platform-auth-service-69847b59dd-dfzgm                Durham       worker10.acme.com   Deployment
aiops       platform-auth-service-69847b59dd-nkzwf                Durham       worker4.acme.com    Deployment
aiops       platform-auth-service-69847b59dd-nh22p                Raleigh      worker12.acme.com   Deployment
aiops       platform-identity-management-6c6fd4dcc8-wknsp         Durham       worker4.acme.com    Deployment
aiops       platform-identity-management-6c6fd4dcc8-ps565         Raleigh      worker12.acme.com   Deployment
aiops       platform-identity-management-6c6fd4dcc8-zktfc         Raleigh      worker6.acme.com    Deployment
aiops       platform-identity-provider-649c6dd6cd-45v7c           Durham       worker7.acme.com    Deployment
aiops       platform-identity-provider-649c6dd6cd-jfhmc           Durham       worker4.acme.com    Deployment
aiops       platform-identity-provider-649c6dd6cd-hc9v5           Raleigh      worker12.acme.com   Deployment
```

In the preceding example output, pods that are not optimally distributed are listed after **`WARNING: Not all pods are distributed optimally`**. The distribution is suboptimal because pod replicas are on the same zone, instead of being across distinct zones.

3. Manually redistribute the identified pods on the multi-zone cluster to achieve an optimal distribution for high availability.

   For each microservice identified in the previous step, delete pods that have the same zone as another pod within that microservice - an example is given at the end of this step. The order of deletion is significant. If the pods identified by the status checker plug-in contain any of the following, then delete them in the following order before you delete any other pods:

   - **`iaf-system-zookeeper`**
   - **`iaf-system-kafka`**
   - **`aiops-topology-cassandra`**
   - **`aiops-ibm-elasticsearch-es-server-zone0`**
   - **`aiops-ibm-elasticsearch-es-server-zone1`**
   - **`aiops-ibm-elasticsearch-es-server-zone2`**
   - **`aiops-ir-analytics-spark-worker`**
   - **`aiops-lad-flink-taskmanager`**
   - **`aiops-lad-flink`**
   - **`aiops-ir-lifecycle-flink-taskmanager`**
   - **`aiops-ir-lifecycle-flink`**
   - **`<instance>-edb-postgres`**, where **`<instance>`** is the name of your IBM Cloud Pak for AIOps instance, usually **`ibm-cp-aiops`**.
   - **`aiops-ir-lifecycle-datarouting`**
   - **`aimanager-aio-similar-incidents-service`**
   - **`aimanager-aio-log-anomaly-golden-signals`**
   - **`aiops-topology-topology`**
   - **`aimanager-aio-change-risk`**
   - **`aiops-ir-core-ncoprimary`**
   - **`aiops-ir-core-ncobackup`**
   - **`<instance>-redis-server`**, where **`<instance>`** is the name of your IBM Cloud Pak for AIOps instance, usually **`ibm-cp-aiops`**.
   - **`aiops-topology-layout`**

   Note: To achieve an optimal distribution, delete pods individually. You can delete multiple pods together if you need to reduce disruption time.

   Run the following command to delete a pod:

   ```
   oc delete pod -n <project> <pod_to_delete>
   ```

   Where

   - **`<project>`** is the project that your IBM Cloud Pak for AIOps installation is deployed in.
   - **`<pod_to_delete>`** is a pod with multiple replicas in the same zone.

   For example, the **`platform-identity-provider`** microservice in the example output for step 2 has two pods in Durham and one pod in Raleigh. Delete a pod in Durham so that a new pod can be scheduled on a node in the Chapel Hill zone, achieving distribution across all zones, as in the following example:

   Example command:

   ```
   oc delete pod -n aiops platform-identity-provider-649c6dd6cd-45v7c
   ```

4. Validate whether the distribution is now optimal.

   Re-run the status checker plug-in to check that all pods are now optimally distributed across the cluster zones.

   ```
   oc waiops multizone status
   ```

   Example output where pods are optimally distributed across zones:

   ```
   $ oc waiops multizone status

   Regions : CentralNC
   Zones   : ChapelHill Durham Raleigh

   NAME                ZONE         REGION
   ```

```
worker11.acme.com   ChapelHill   CentralNC
worker14.acme.com   ChapelHill   CentralNC
worker2.acme.com    ChapelHill   CentralNC
worker5.acme.com    ChapelHill   CentralNC
worker8.acme.com    ChapelHill   CentralNC
worker1.acme.com    Durham       CentralNC
worker10.acme.com   Durham       CentralNC
worker13.acme.com   Durham       CentralNC
worker4.acme.com    Durham       CentralNC
worker7.acme.com    Durham       CentralNC
worker0.acme.com    Raleigh      CentralNC
worker12.acme.com   Raleigh      CentralNC
worker3.acme.com    Raleigh      CentralNC
worker6.acme.com    Raleigh      CentralNC
worker9.acme.com    Raleigh      CentralNC

INFO: Checking the namespace aiops.................................done.

INFO: All pods are distributed appropriately for multizone.
```

If the status checker plug-in still reports a nonoptimal distribution, then repeat step 3. If the service remains incorrectly distributed, then you can run the following command to cordon nodes within a zone so that deleted pods cannot be rescheduled on them:

```
oc adm cordon $(oc get no -l topology.kubernetes.io/zone=<zone_name> -o jsonpath='{.items[*].metadata.name}')
```

Where **<zone_name>** is the name of the zone that you do not want pods to be scheduled in.

Example command to cordon nodes in the Durham zone to help ensure that a pod is not scheduled in the Durham zone:

```
oc adm cordon $(oc get no -l topology.kubernetes.io/zone=Durham -o jsonpath='{.items[*].metadata.name}')
```

When the service has an optimal distribution, uncordon the nodes:

```
oc adm uncordon $(oc get no -l topology.kubernetes.io/zone=<zone_name>> -o jsonpath='{.items[*].metadata.name}')
```

Where **<zone_name>** is the name of the zone that you do not want pods to be scheduled in.

Example command:

```
oc adm uncordon $(oc get no -l topology.kubernetes.io/zone=Durham -o jsonpath='{.items[*].metadata.name}')
```

Resizing storage (IBM Cloud Pak for AIOps on OpenShift)

Learn about increasing the persistent volume claim (PVC) sizes for IBM Cloud Pak for AIOps.

The original PVC sizes in Persistent storage sizing provide adequate space for a user to begin working with IBM Cloud Pak for AIOps. Based on your actual usage and needs, the PVCs might need to be adjusted up to handle the workload when IBM Cloud Pak for AIOps is installed and running. In particular, CouchDB, Elasticsearch, and Kafka might need significantly more disk space. Expand the following sections to review details about custom sizings for these PVCs.

Cassandra

Cassandra is used to primarily store Topology Management resources, as well as incoming metric and alert data. By default, 50 GB is allocated for each of the 3 Cassandra instances (PVCs). This default configuration is targeted for environments with less than 500,000 topology resources to provide ample storage for topology resources, metric data (120K metrics per 5 minute KPI window), and Cassandra's internal data management activities. For environments with larger topology resource requirements, an additional 70 GB is recommended for each additional 1M topology resources, and an additional 15 GB for each additional 100K metrics per 5 minute KPI window for Metric storage.

In comparison to a default production environment, a larger environment requires additional storage. For example:

Example 1: An environment with 1.5M topology resources, and 120,000 metrics per 5 minute KPI window would require an additional 70 GB of storage for topology resources, no additional storage for metric processing, for a total of 120 GB required per Cassandra storage PVC.

Example 2: An environment with 3M topology resources, and 1.2M metrics per 5 minute KPI window would require an additional 210 GB for Topology storage and an additional 135 GB for Metric storage should be allocated to raise the total Cassandra allocation (per PVC) from the default of 50 GB to 395 GB of storage per Cassandra storage PVC.

Increasing the size of the Cassandra PVCs

You can increase the configured Cassandra storage size if you need to support more than the default configuration supports. You can also incrementally add disk storage as resource counts or metric rates increase over time. To add disk storage, use the Red Hat OpenShift (oc) CLI to scale your PVC storage:

1. Export environment variables and switch to your project.

```
export PROJECT=<project>
export INSTALLATION_NAME="$(oc get installation -n ${PROJECT} -o jsonpath="{.items[0].metadata.name}")"
export SIZE=<size>Gi

oc project ${PROJECT}
```

Where

- `<project>` is the project that your IBM Cloud Pak for AIOps installation is deployed in.
- `<size>` is the storage size that you require for the Cassandra PVCs in Gi. For example, if you require 110Gi use `export SIZE=110Gi`.

2. Run the following command to increase Cassandra's PVC claims.

```
oc patch installation ${INSTALLATION_NAME} --type merge -p '{"spec":{"storage":{"data-aiops-topology-cassandra":
{"size":"'${SIZE}'"}}}}'
```

CouchDB

CouchDB is used to store the results of manually run and fully automated runbook activities. CouchDB storage is uncapped to allow you access to historical activities. Each CouchDB PVC is set with a default value of 20 GB of storage.

As an example, in a small environment with an average of 60 runbook executions per hour (1 per minute, 1440 per day), you budget about 20 MB of storage for CouchDB per day. In a Large (high availability) deployment, with the same rate of runbook executions, you budget 60 MB of storage space for CouchDB per day. By using the default CouchDB storage sizes with this example scenario, the default storage for IBM Cloud Pak for AIOps supports about 1000 days of retention.

A sample runbook is provided to reduce the historical storage footprint when needed. For more information about sample runbooks, see Load and reset example runbooks.

EDB Postgres

Increasing the size of the Postgres PVCs

1. Export environment variables and switch to your project.

```
export PROJECT=<project>
export INSTALLATION_NAME="$(oc get installation -n ${PROJECT} -o jsonpath="{.items[0].metadata.name}")"
export SIZE=<size>Gi

oc project ${PROJECT}
```

Where

- `<project>` is the project that your IBM Cloud Pak for AIOps installation is deployed in.
- `<size>` is the storage size that you require for the Postgres PVCs in Gi. For example, if you require 110Gi use `export SIZE=110Gi`.

2. Run the following command to increase EDB Postgres's PVC claim.

```
oc patch installation ${INSTALLATION_NAME} --type merge -p '{"spec":{"storage":{"ibm-cp-aiops-edb-postgres":
{"size":"'${SIZE}'"}}}}'
```

Important: You must run the preceding command as it is, even if the name of your Postgres PVC does not start with `ibm-cp-aiops`.

Elasticsearch

Elasticsearch is used to store various data sets in IBM Cloud Pak for AIOps, the largest of which is log data for training and analysis. If there are large volumes of log data, then you might need to increase the provision for Elasticsearch. Use the instructions for Storage class requirements under the IBM Cloud Pak for AIOps only section of hardware requirements. The exact amount of Elasticsearch data, and therefore the amount of storage that is needed, varies greatly from case to case.

For estimation purposes, plan for around 10Gi of disk space usage per replica per day, running at 1000 log messages per second.

One replica of the data is stored in Elasticsearch on a starter deployment of IBM Cloud Pak for AIOps, and three copies of the data are stored in Elasticsearch for a production IBM Cloud Pak for AIOps deployment.

The default retention period for Elasticsearch data is 14 days, so plan for 15 days of storage.

The total estimated space must be available across all of the Elasticsearch instances in the cluster combined. A starter deployment of IBM Cloud Pak for AIOps has one Elasticsearch node, and a production IBM Cloud Pak for AIOps deployment has five Elasticsearch nodes.

Plan for a minimum of a 20% buffer to allow for variations in log data size.

Example 1: Running at 4000 logs per second on a starter deployment for 14+ days.

- This would be roughly 40Gi x 15 days x 1 replica = 600Gi of Elasticsearch PVC usage.
- Multiply this by a buffer of 1.2.
- This results in a recommendation of 720Gi in total for the one Elasticsearch instance deployed by default in a starter deployment.
- Set the storage size for the Elasticsearch PVC to 720Gi.

Example 2: Running at 8000 logs per second on a production deployment for 14+ days.

- This would be roughly 80Gi x 15 days x 3 replicas = 3600Gi of Elasticsearch PVC usage.
- Multiply this by a buffer of 1.2.
- This results in a recommendation of 4,320Gi total across all five Elasticsearch instances, or roughly 864Gi per instance.
- Set the storage size for the Elasticsearch PVC to 864Gi.

Increasing the size of the Elasticsearch PVCs

1. Export environment variables and switch to your project.

```
export PROJECT=<project>
export INSTALLATION_NAME="$(oc get installation -n ${PROJECT} -o jsonpath="{.items[0].metadata.name}")"
```

```
export SIZE=<size>Gi

oc project ${PROJECT}
```

Where

- **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.
- **<size>** is the storage size that you require for the Elasticsearch PVCs in Gi. For example, if you require 110Gi use **export SIZE=110Gi**.

2. Run the following command to increase Elasticsearch's PVC claims, where **<size>** is the storage size that you require for Elasticsearch's PVCs.

```
oc patch installation ${INSTALLATION_NAME} --type merge -p '{"spec":{"storage":{"data-aiops-ibm-elasticsearch-es-server-all":{"size":"'${SIZE}'"}}}}'
```

Important: If you have a multizone environment, the name of your PVC might end with the zone name instead of **all**, but it is important that the preceding codeblock is run as it is, with **data-aiops-ibm-elasticsearch-es-server-all**.

Kafka

Kafka's storage requirements are primarily driven by Kafka log integrations, and also by other integrations such as Topology observers and Metric integrations, including custom Metric integrations.

The amount of storage that Kafka needs varies depending on your usage, and you might need to increase the storage provision for Kafka. Data collection from a metric data source commonly starts with historical data collection. For this reason, it is good practice to increase the PVC size for Kafka.

Increasing the size of the Kafka PVCs

You can increase Kafka's PVC size with the following commands. The Kafka pods will then restart with the resized PVCs.

1. Export environment variables and switch to your project.

```
export PROJECT=<project>
export INSTALLATION_NAME="$(oc get installation -n ${PROJECT} -o jsonpath="{.items[0].metadata.name}")"
export SIZE=<size>Gi

oc project ${PROJECT}
```

Where

- **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.
- **<size>** is the storage size that you require for the Kafka PVCs in Gi. For example, if you require 110Gi use **export SIZE=110Gi**.

2. Run the following command to increase Kafka's PVC claim, where **<size>** is the storage size that you require for Kafka's PVCs.

```
oc patch installation ${INSTALLATION_NAME} --type merge -p '{"spec":{"storage":{"data-iaf-system-kafka":{"size":"'${SIZE}'"}}}}'
```

Notes:

- For Metric Anomaly Detection, it is recommended to increase the persistent storage for Kafka by 2Gi for every 10,000 KPIs being processed.
- For Log Anomaly Detection, it is recommended to increase the Kafka PVC space based on the size of the logs, plus roughly an additional 20% buffer due to overheads, such as the key strings that are added to the messages.
- For example, if you plan on uploading 40Gi of log data then add around 48Gi (40Gi plus a 20% buffer) to the existing 30Gi for a total of 78Gi for the Kafka PVC.

MinIO

Increasing the size of the MinIO PVCs

1. Switch to your IBM Cloud Pak for AIOps project.

```
oc project <project>
```

Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.

2. Run the following command to increase MinIO's PVC claims, where **<size>** is the storage size that you require for the MinIO PVCs.

```
for p in $(oc get pvc -l app.kubernetes.io/name=ibm-minio -o jsonpath='{.items[*].metadata.name}'); do oc patch pvc $p --type merge -p '{"spec":{"resources":{"requests":{"storage":"<size>Gi"}}}}' ; done
```

For example, to change the MinIO PVC size to 80Gi:

```
for p in $(oc get pvc -l app.kubernetes.io/name=ibm-minio -o jsonpath='{.items[*].metadata.name}'); do oc patch pvc $p --type merge -p '{"spec":{"resources":{"requests":{"storage":"80Gi"}}}}' ; done
```

3. Delete the **aimanager-ibm-minio** StatefulSet and leave its pods.

```
oc delete sts --cascade=orphan aimanager-ibm-minio
```

4. Run the following command to update the **aimanager** custom resource with the new MinIO storage size, where **<size>** is the storage size that you set for the MinIO PVCs.

```
oc patch aimanager aimanager --type='json' -p="[{'op': 'replace', 'path': '/spec/minio/storage/storageSize', 'value': '<size>Gi'}]"
```

For example, if MinIO PVC size is updated to 80Gi:

```
oc patch aimanager aimanager --type='json' -p="[{'op': 'replace', 'path': '/spec/minio/storage/storageSize',
'value': '80Gi'}]"
```

5. Verify that the **aimanager-ibm-minio** statefulset has the correct storage size.

```
oc get statefulset  aimanager-ibm-minio -o
jsonpath='{.spec.volumeClaimTemplates[0].spec.resources.requests.storage}'
```

Restarting the environment (IBM Cloud Pak for AIOps on OpenShift)

Learn how to shutdown and restart the Red Hat OpenShift cluster where IBM Cloud Pak for AIOps is deployed.

Overview

Use this procedure before a known maintenance window or outage to shut down the Red Hat OpenShift cluster where IBM Cloud Pak for AIOps is installed, and to restart the cluster and workloads afterward.

Warning: If you need to shut down the cluster where IBM Cloud Pak for AIOps is installed, then you must use the following procedure. Failure to do so can result in data loss or corruption.

Procedure

1. Validate the installation
2. Check the certificates
3. Prepare to scale down
4. Scale down the workloads and drain the nodes
5. Shut down the cluster
6. Restart the cluster
7. Scale up the workloads
8. Validate the installation

1. Validate the installation

1. Set environment variables.

   Make a note of these environment variables or save them to a file, as you will need to export them again after you restart your cluster.

   ```
   export AIOPS_NAMESPACE=<project>
   export AIOPS_INSTANCE=$(oc get installation -o jsonpath='{.items[0].metadata.name}' -n ${AIOPS_NAMESPACE})
   ```

   Where **<project>** is the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in.

2. Run the describe command:

   ```
   oc describe installations.orchestrator.aiops.ibm.com -n "${AIOPS_NAMESPACE}"
   ```

   Review the **ComponentStatus** fields to confirm that all components are marked as **Ready** and the phase is **Running**.

   Example output:

   ```
   Name:         ibm-cp-aiops
   Namespace:    aiops
   API Version:  orchestrator.aiops.ibm.com/v1alpha1
   Kind:         Installation
   Spec:
   ...
   Status:
   Componentstatus:
      Aimanager:                        Ready
      Aiopsanalyticsorchestrator:       Ready
      Aiopsedge:                        Ready
      Aiopsui:                          Ready
      Asm:                              Ready
      Baseui:                           Ready
      Cluster:                          Ready
      Commonservice:                    Ready
      Elasticsearchcluster:             Ready
      Flinkdeployment:                  Ready
      Issueresolutioncore:              Ready
      Kafka:                            Ready
      Lifecycleservice:                 Ready
      Lifecycletrigger:                 Ready
      Rediscp:                          Ready
      Tunnel:                           Ready
      Zenservice:                       Ready
   Phase:                    Running
   ```

2. Check the certificates

Ensure that none of the certificates have problems or are expired.

Run the following command:

```
while read l; do echo "$l" | grep '^NAME' || (n=$(echo $l | sed 's/ .*//'); s=$(echo $l | sed 's/^[^ ]* *\([^ ]*\).*/\1/'); x=$(oc get secret -n $n $s -o jsonpath='{.data.tls\.crt}' | base64 -d | openssl x509 -noout -enddate 2>/dev/null | sed 's!notAfter=!!'); echo "$l" | sed 's![^ ][^ ]*$!'"$x"'!'); done< <(oc get secret -A --field-selector=type==kubernetes.io/tls -o custom-columns=NAMESPACE:.metadata.namespace,NAME:.metadata.name,EXPIRY:.metadata.name)
```

Example output excerpt:

```
ibm-licensing   ibm-license-service-cert                                  Jan  8  13:32:07  2025  GMT
ibm-licensing   ibm-license-service-cert-internal                         Jan  7  13:31:12  2026  GMT
ibm-licensing   ibm-licensing-service-prometheus-cert                     Jan  7  13:31:25  2026  GMT
cp4aiops        aimanager-aio-log-anomaly-feedback-learning-cert          Jan  7  14:01:43  2026  GMT
cp4aiops        aimanager-aio-log-anomaly-golden-signals-cert             Jan  7  14:01:43  2026  GMT
cp4aiops        aimanager-aio-oob-recommended-actions-cert                Jan  7  14:01:43  2026  GMT
<...>
```

Renew or re-create any certificates that have problems, are expired, or will expire before the cluster is restarted. For more information about certificate management, see Renew or re-create certificates in Openshift 4.x.

3. Prepare to scale down

1. Cordon the worker nodes.

   Run the following command for each of the worker nodes:

   ```
   oc adm cordon <node>
   ```

   Where **<node>** is the name of the node to cordon.

2. Make a note of the number of replicas.

   1. Make a note of the number of replicas for each StatefulSet.

      ```
      oc get statefulsets -n ${AIOPS_NAMESPACE}
      ```

      Example output:

      ```
      NAME                                        READY   AGE
      aimanager-ibm-minio                         1/1     18m
      aiops-ir-analytics-spark-worker             2/2     33m
      aiops-ir-core-ncobackup                     1/1     37m
      aiops-ir-core-ncoprimary                    1/1     39m
      aiops-topology-cassandra                    1/1     43m
      c-example-couchdbcluster-m                  1/1     40m
      aiops-ibm-elasticsearch-es-server-all       1/1     49m
      ibm-cp-aiops-redis-server                   3/3     45m
      zen-minio                                   3/3     40m
      ```

      Note:

      - If you do not have a IBM® Netcool® Operations Insight® probe integration, then `aiops-ir-core-ncobackup` and `aiops-ir-core-ncoprimary` has zero replicas.
      - If you upgraded from an earlier version of IBM Cloud Pak for AIOps, you also have an `icp-mongodb` StatefulSet.

   2. Make a note of the number of replicas for each StrimziPodSet.

      ```
      oc get strimzipodset -n ${AIOPS_NAMESPACE}
      ```

      Example output:

      ```
      NAME                          PODS   READY PODS   CURRENT PODS   AGE
      iaf-system-kafka              3      3            3              13d
      iaf-system-zookeeper          3      3            3              13d
      ```

   3. Make a note of the number of replicas for each Flink deployment by using the following command:

      ```
      oc get deployment | grep flink | grep -v "operator"
      ```

      Example output:

      ```
      NAME                                    PODS   READY PODS   CURRENT PODS   AGE
      aiops-ir-lifecycle-flink                1/1     1            1             137m
      aiops-ir-lifecycle-flink-taskmanager    1/1     1            1             137m
      aiops-lad-flink                         1/1     1            1             139m
      aiops-lad-flink-taskmanager             2/2     2            2             139m
      ```

      Notes:

      - If you have a base deployment, then `aiops-lad-flink` and `aiops-lad-flink-taskmanager` do not show in the preceding output.
      - If you upgraded from an earlier version of IBM Cloud Pak for AIOps, you also have an `icp-mongodb` StatefulSet.

4. Scale down the workloads and drain the nodes

1. Scale down the operator deployments in the IBM Cloud Pak for AIOps namespace.

   ```
   oc scale deployment -l olm.owner.kind=ClusterServiceVersion -n ${AIOPS_NAMESPACE} --replicas=0
   ```

   Run the following command to check that the number of replicas for each of the operator deployments is now 0.

   ```
   oc get deployment -n ${AIOPS_NAMESPACE} -l olm.owner.kind=ClusterServiceVersion
   ```

   Example output:

   ```
   NAME                                                READY   UP-TO-DATE   AVAILABLE   AGE
   aimanager-operator-controller-manager               0/0     0            0           47m
   aiopsedge-operator-controller-manager               0/0     0            0           47m
   asm-operator                                        0/0     0            0           47m
   flink-kubernetes-operator                           0/0     0            0           54m
   ibm-aiops-orchestrator-controller-manager           0/0     0            0           58m
   ibm-common-service-operator                         0/0     0            0           56m
   ibm-commonui-operator                               0/0     0            0           53m
   ibm-elasticsearch-operator-ibm-es-controller-manager 0/0     0            0           54m
   ibm-events-operator-v5.0.1                          0/0     0            0           54m
   ibm-iam-operator                                    0/0     0            0           54m
   ibm-ir-ai-operator-controller-manager               0/0     0            0           47m
   ibm-redis-cp-operator                               0/0     0            0           49m
   ibm-secure-tunnel-operator                          0/0     0            0           48m
   ibm-watson-aiops-ui-operator-controller-manager     0/0     0            0           48m
   ibm-zen-operator                                    0/0     0            0           54m
   ir-core-operator-controller-manager                 0/0     0            0           47m
   ir-lifecycle-operator-controller-manager            0/0     0            0           47m
   operand-deployment-lifecycle-manager                0/0     0            0           55m
   postgresql-operator-controller-manager-1-18-12      0/0     0            0           54m
   ```

   Note: If you upgraded from an earlier version of IBM Cloud Pak for AIOps, you also have an `icp-mongodb-operator` deployment.

2. Scale down the StatefulSets that you noted in step 3.2.

   You can use the Cloud Pak for AIOps console, or create a shell script to do this.

   If you have a base deployment, then remove the following lines from the example shell script:

   ```
   oc scale deployment aiops-lad-flink --replicas=0 -n ${AIOPS_NAMESPACE}
   oc scale deployment aiops-lad-flink-taskmanager --replicas=0 -n ${AIOPS_NAMESPACE}
   ```

   Note:

   - If you have a base deployment, then `aiops-lad-flink` and `aiops-lad-flink-taskmanager` do not show in the preceding output.
   - If you upgraded from an earlier version of IBM Cloud Pak for AIOps, you also have an `icp-mongodb` StatefulSet.

   If you upgraded from an earlier version of IBM Cloud Pak for AIOps, then add the following line to the example shell script:

   ```
   oc scale statefulsets icp-mongodb --replicas=0 -n ${AIOPS_NAMESPACE}
   ```

   Example shell script:

   ```
   #!/bin/bash

   oc scale statefulsets aimanager-ibm-minio --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale statefulsets ${AIOPS_INSTANCE}-redis-server --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale statefulsets aiops-ir-analytics-spark-worker --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale statefulsets aiops-ir-core-ncobackup --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale statefulsets aiops-ir-core-ncoprimary --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale deployment aiops-ir-lifecycle-flink --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale deployment aiops-ir-lifecycle-flink-taskmanager --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale statefulsets aiops-topology-cassandra --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale statefulsets c-example-couchdbcluster-m --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale deployment aiops-lad-flink --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale deployment aiops-lad-flink-taskmanager --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale statefulsets -l app.kubernetes.io/managed-by=ibm-elasticsearch --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   oc scale statefulsets zen-minio --replicas=0 -n ${AIOPS_NAMESPACE}
   sleep 2
   ```

   Run the following command to check that the number of replicas for each of the StatefulSets is now 0.

   ```
   oc get statefulsets -n ${AIOPS_NAMESPACE}
   ```

   Example output:

```
NAME                                          READY   AGE
aimanager-ibm-minio                           0/0     112m
aiops-ir-analytics-spark-worker               0/0     128m
aiops-ir-core-ncobackup                       0/0     131m
aiops-ir-core-ncoprimary                      0/0     133m
aiops-topology-cassandra                      0/0     138m
c-example-couchdbcluster-m                    0/0     134m
aiops-ibm-elasticsearch-es-server-all         0/0     143m
ibm-cp-aiops-redis-server                     0/0     140m
zen-minio                                     0/0     134m
```

3. Run the following command to check that the number of replicas for each of the Flink Deployments is now 0.

```
oc get deployments -n ${AIOPS_NAMESPACE} | grep flink | grep -v "operator"
```

Example output:

```
NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
aiops-ir-lifecycle-flink              0/0     0            0           137m
aiops-ir-lifecycle-flink-taskmanager  0/0     0            0           137m
aiops-lad-flink                       0/0     0            0           139m
aiops-lad-flink-taskmanager           0/0     0            0           139m
```

4. Shutdown the **Kafka** and **ZooKeeper** pods.

```
oc delete pod -l ibmevents.ibm.com/name=iaf-system-kafka -n ${AIOPS_NAMESPACE}
oc delete pod -l ibmevents.ibm.com/name=iaf-system-zookeeper -n ${AIOPS_NAMESPACE}
```

Run the following command to check that the Kafka and ZooKeeper pods have successfully shutdown. If the shutdown is complete, no pods are returned.

```
oc get pod -l ibmevents.ibm.com/controller=strimzipodset -n ${AIOPS_NAMESPACE}
```

5. Scale down the PostgreSQL pods.

When shutting down a PostgreSQL cluster, it is best to remove the primary replica last. The following script removes each database replica in the cluster with the primary removed last.

Before running the script, replace **<project>** with the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in.

```
#!/bin/bash

AIOPS_NAMESPACE=<project>

# Get array of Postgres clusters
CLUSTERS=($(oc get clusters.postgresql.k8s.enterprisedb.io -n "${AIOPS_NAMESPACE}" -o go-template='{{range .items}}
{{.metadata.name}}{{" "}}{{end}}'))

# For each Postgres cluster, shutdown primary last
for cluster_name in "${CLUSTERS[@]}"; do
    primary=$(oc get clusters.postgresql.k8s.enterprisedb.io -n "${AIOPS_NAMESPACE}" "${cluster_name}" -o go-
template='{{.status.currentPrimary}}')
    instances=($(oc get clusters.postgresql.k8s.enterprisedb.io -n "${AIOPS_NAMESPACE}" "${cluster_name}" -o go-
template='{{range .status.instanceNames}}{{print . " "}}{{end}}'))
    for instance_name in "${instances[@]}"; do
        # Shutdown non-primary replicas
        if [ "${instance_name}" != "${primary}" ]; then
            oc delete pod -n "${AIOPS_NAMESPACE}" "${instance_name}" --ignore-not-found
        fi
    done

    # Shutdown the primary once all other replicas are down
    oc delete pod -n "${AIOPS_NAMESPACE}" "${primary}" --ignore-not-found
done
```

Wait for all the Postgres pods to be deleted. All pods are deleted when the following command returns no pods:

```
oc get pod -l k8s.enterprisedb.io/podRole=instance -n ${AIOPS_NAMESPACE}
```

6. (Optional) After the StatefulSets and StrimziPodSets are scaled down, drain the worker nodes.

Run the following command for each of the worker nodes:

```
oc adm drain <node>
```

Where **<node>** is the name of the node to drain.

Note: Some pods, such as storage pods, do not stop because this would violate the disruption budget. If this problem occurs, run the commands in each node until only the storage pods are left, and then stop the command and drain the next node.

5. Shut down the cluster

1. Shut down all the worker nodes on the cluster.
2. Shut down all the master nodes on the cluster.
3. Shut down the API node on the cluster.

For more information about shutting down your cluster nodes, see step 4 in the Red Hat OpenShift documentation Shutting down a cluster gracefully.

6. Restart the cluster

1. Re-export the environment variables that you saved in step 1.1.
2. Restart the cluster nodes in the following order:
    1. Restart the API node.
    2. Restart the master nodes. Check whether all master nodes are in **ready** status by running the following command:

        ```
        oc get nodes
        ```

    3. Restart the worker nodes. Check whether all worker nodes are in **ready** status by running the following command:

        ```
        oc get nodes
        ```

3. After all the nodes are up, uncordon the worker nodes.

    Run the following command for each of the worker nodes:

    ```
    oc adm uncordon <node>
    ```

    Where **<node>** is the name of the node to uncordon.

7. Scale up the workloads

Scaling up the workloads in the following order helps to minimize startup time and resource contention issues.

1. Scale the events operator back up.

    ```
    oc scale deployment --replicas=1 $(oc get deployment -o custom-columns=NAME:.metadata.name --no-headers -n
    ${AIOPS_NAMESPACE}  | grep '^ibm-events-operator-') -n ${AIOPS_NAMESPACE}
    ```

2. Check whether the Kafka and Zookeeper pods are running again. This can take a few minutes.

    ```
    oc get pod -l ibmevents.ibm.com/controller=strimzipodset -n ${AIOPS_NAMESPACE}
    ```

    Example output when the Kafka and Zookeeper pods are running:

    ```
    NAME                                  READY   STATUS    RESTARTS   AGE
    iaf-system-kafka-0                    1/1     Running   0          13d
    iaf-system-kafka-1                    1/1     Running   0          13d
    iaf-system-kafka-2                    1/1     Running   0          13d
    iaf-system-zookeeper-0                1/1     Running   0          13d
    iaf-system-zookeeper-1                1/1     Running   0          13d
    iaf-system-zookeeper-2                1/1     Running   0          13d
    ```

3. You need to scale up each of the StatefulSets to the number of replicas as noted in step 3.2.
    1. Scale up Cassandra, Elasticsearch, and Spark StatefulSets in the following order:
        - aiops-topology-cassandra
        - aiops-ibm-elasticsearch-es-server-all
        - aiops-ir-analytics-spark-worker

        Run the following command to scale up each StatefulSet:

        ```
        oc scale statefulsets <statefulset> --replicas=<number of replicas> -n ${AIOPS_NAMESPACE}
        ```

        Where:

        - **<statefulset>** is the StatefulSet to be scaled up
        - **<number_of_replicas>** is the number of replicas the StatefulSet it to be scaled up to

        For example,

        ```
        oc scale statefulsets aiops-topology-cassandra --replicas=1 -n cp4aiops
        ```

        Note: If IBM Cloud Pak for AIOps is deployed on a multi-zone architecture, then there are multiple **Elasticsearch** StatefulSets with numbered zone names. Each **Elasticsearch** StatefulSet must be scaled up during this step.

    2. Scale the Flink deployments in the following order:
        - aiops-ir-lifecycle-flink
        - aiops-ir-lifecycle-flink-taskmanager
        - aiops-lad-flink
        - aiops-lad-flink-taskmanager

        Note: If you have a base deployment, then do not scale up **aiops-lad-flink** and **aiops-lad-flink-taskmanager**.

        Run the following command to scale up the Flink deployments:

        ```
        oc scale deployment <flink_deployment> --replicas=<number of replicas> -n ${AIOPS_NAMESPACE}
        ```

        Where **<flink_deployment>** is the name of the Flink deployment.

    3. Scale the following StatefulSets in the specified order:

- aiops-ir-core-ncoprimary
- aiops-ir-core-ncobackup
- c-example-couchdbcluster-m
- ${AIOPS_INSTANCE}-redis-server
- aimanager-ibm-minio
- zen-minio

Run the following command to scale up each StatefulSet:

```
oc scale statefulsets <statefulset> --replicas=<number of replicas> -n ${AIOPS_NAMESPACE}
```

Where:

- `<statefulset>` is the StatefulSet to be scaled up
- `<number_of_replicas>` is the number of replicas the StatefulSet it to be scaled up to

4. Scale up the operator deployments.

```
oc scale deployment -l olm.owner.kind=ClusterServiceVersion -n ${AIOPS_NAMESPACE} --replicas=1
```

8. Validate the installation

Note: After a complete cluster restart, it might take approximately an hour for the installation to start running again.

Run the describe command:

```
oc describe installations.orchestrator.aiops.ibm.com -n "${AIOPS_NAMESPACE}"
```

Review the `ComponentStatus` fields to confirm that all components are marked as `Ready` and the phase is `Running`.

Secure Tunnel (IBM Cloud Pak for AIOps on OpenShift)

A Secure Tunnel provides endpoint-to-endpoint connections across a hybrid network without opening firewall rules in an enterprise network. It allows access between different networks through TCP over HTTPS technology.

It can be applied in many use scenarios, such as MCMP, Runbooks, Infrastructure Automation, Slack integrations for IBM Cloud Pak for AIOps, Instana, and Turbonomic.

The traffic through these connections is encrypted with HTTPS. A Secure Tunnel can control access to resources between different networks, with more granular control, and collect all the operations and traffic logins in audit records.

It is not necessary to modify any access rules and firewall configuration between existing infrastructure if you use a Secure Tunnel. It is like a traditional VPN, bridging two networks between the **Secure Tunnel server** and the **Secure Tunnel Connector**.

To learn more about Secure Tunnel, see the following sections:

- Terminology
- Architecture
- Security
- Certificate expiration and renewal
- How to use Secure Tunnel

Terminology

To manage a tunnel connection, you need to understand the following concepts:

- **Secure Tunnel** a group of microservices that include the following services:
  - **UI Server**
    - Used to provide the static resources of the **Secure Tunnel console** and to handle some UI logic.
  - **API Server**
    - Verifies and processes the user's operation request that comes from the **Secure Tunnel console**. The server operates the **Secure Tunnel Custom Resources(CRs)** according to the request.
  - **Controller**
    - Used to monitor updates to the **Secure Tunnel Custom Resources(CRs)** and to take actions according to the CR updates.
  - **Tunnel Connection**
    - The configuration data for bridging two networks.
  - **Application mapping**
    - The configuration data used to control:
      - which applications on the network of the **Secure Tunnel** side can be accessed by the **Secure Tunnel Connector** side and vice versa.
    - The **Tunnel Connection worker** and **Secure Tunnel Connector** use this data to control the access to applications.
  - **Tunnel Connection worker**
    - The instance of the **Tunnel Connection** is created on one or more **Tunnel Connection worker** pods. This worker is also the server side of a TCP-over-HTTPS tunnel. Based on the OSS project.
  - **Secure Tunnel Connector**
    - The other side of a TCP over HTTP tunnel is installed on the peer network of the **Secure Tunnel** to bridge the network where the **Secure Tunnel** is installed. It is based on the OSS project.

Architecture

You can use the following examples of Secure Tunnel to act as a bridge in your enterprise environments.

- **Cloud hosted cluster to On-premise resource**: The cluster is hosted in the cloud and connects to a host in your on-premises system. An example of this is OpenStack connection. For more information, see OpenStack connection.

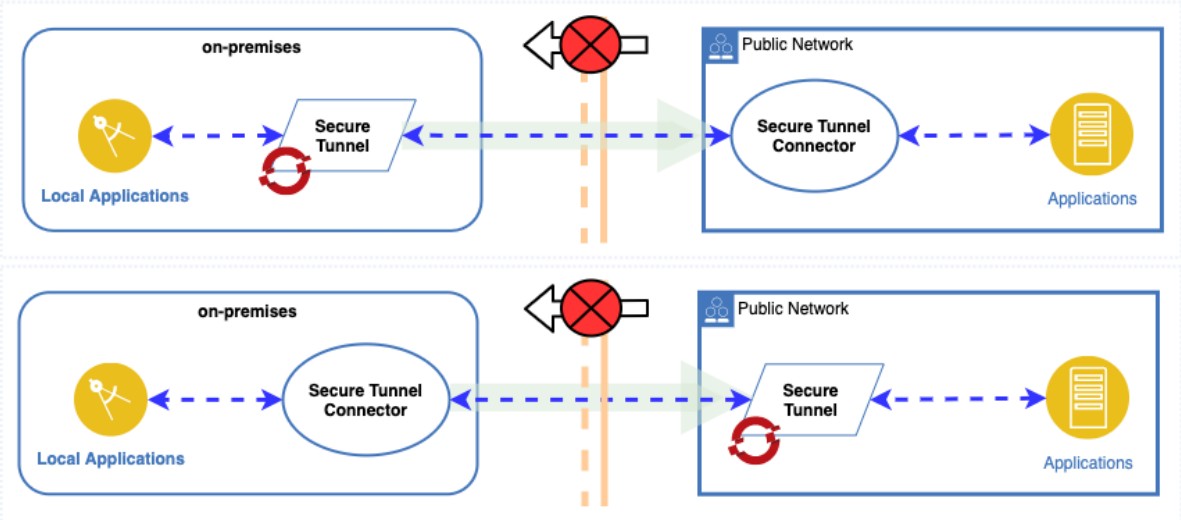


Figure. High-level design

- **Cloud hosted resource to On-premise cluster**: If your Cloud Pak for AIOps instance is installed in on-premises (enterprise network), then you can open the Cloud Pak for AIOps instance to the public network by creating a Secure Tunnel. This is the case where you want to connect to a public resource, such as Slack or Microsoft Teams. An example of this is ChatOps connection. For more information, see ChatOps connection.
- **Bridge two private networks through a third**: You can use Secure tunnel to act as a bridge from one network to another network through a third network. An example of this is the Turbonomic connection. For more information, see Turbonomic connection.

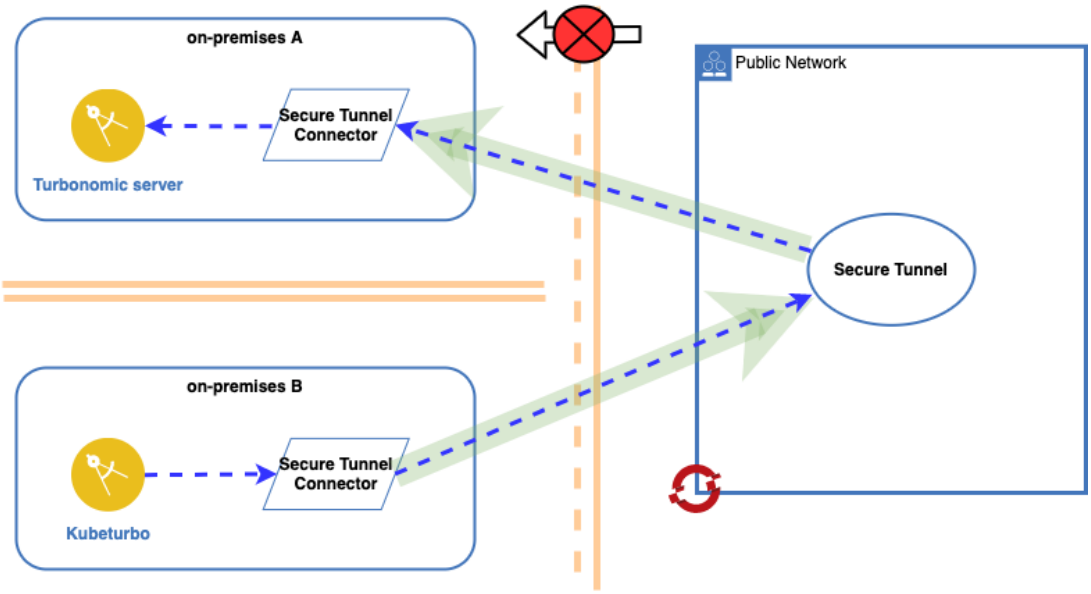


Figure. High-level design for Turbonomic

Features

- Like a traditional VPN, Secure Tunnel can bridge two or more networks. Unlike VPN, it provides more fine-grained resource control.
- **Authentication and authorization** with the IBM Cloud Pak for AIOps console or Red Hat OpenShift Container Platform OAuth and RBAC system.
- Provides two user interfaces
  - **Secure Tunnel console**: A customer can configure a **Secure Tunnel Connection** and control which applications can be accessed by the peer network from the console UI.
  - **Command line**: An automation script can configure a **Secure Tunnel Connection** and control which applications can be accessed by the peer network by using the `kubectl` or `oc` command line tools to operate the CRs of the **Secure Tunnel**.
- The connection is encrypted with Mutual Transport Layer Security(mTLS).
- Based on OSS project.

High-Level Design

- **Secure Tunnel** can be installed from the **Operator Hub** of the Red Hat OpenShift Container Platform console.
- **Secure Tunnel Connector** can be installed on a Red Hat OpenShift Container Platform cluster, Kubernetes (such as IBM Kubernetes Service) or a host machine (VM or physical machine).
- It can bridge the network on the **Secure Tunnel** side and the network on the **Secure Tunnel Connector** side through a WebSocket connection between the **Secure Tunnel** and **Secure Tunnel Connector** with mTLS authentication.

You can use the **Secure Tunnel console** or **command line tools** to control:

- which applications on the network of the **Secure Tunnel** side can be accessed by the **Secure Tunnel Connector** side.
- which applications on the network of the **Secure Tunnel Connector** side can be accessed by the **Secure Tunnel** side.

Security

The following security measures make Secure Tunnel a safe tool to use.

- **Authentication**
  - IBM Cloud Pak for AIOps console accesses the Tunnel Connection worker with JWT authentication token.
  - The Connector is authenticated with Mutual Transport Layer Security (mTLS).
- **Access control**
  - Only IBM Cloud Pak for AIOps console users with the `Automation Adminstrator` or `Administrator` role can use Secure Tunnel.
  - For more information about users and roles, see Managing user access control.
- **Data encryption in communication**
  - All traffic from out-cluster (Connector or console) is transported by HTTPS.
  - All application mapping traffic are encrypted by TCP over HTTPS.
- **Auditing**
  - All configuration changes (tunnel connection or application mapping) are stored in the log system for auditing. For more information, see Configuring tunnel audit logs.
  - All connections are stored in the log system for usage metrics and auditing.

Certificate expiration and renewal

If you renew the Secure Tunnel TLS certificate or it expires and the IBM Certificate Manager renews it, then the Secure Tunnel worker server automatically reloads the renewed certificate. However, the Secure Tunnel connector does not automatically reload the renewed certificate. You must download a new Secure Tunnel connector installation script, and then uninstall and reinstall the Secure Tunnel connector. For more information, see the instructions in Install the Secure Tunnel Connector.

How to use Secure Tunnel

- To create a Secure Tunnel for an integration, see Creating Secure Tunnel Connections.
- To copy the application mapping URL that can be accessed by an integration, or view the configuration details and status of all tunnel connections and application mappings, see Viewing Secure Tunnel.
- To modify configurations for a Secure Tunnel, see Modifying Secure Tunnel.
- To audit Secure Tunnel operations and connections, see Auditing Secure Tunnel.

Installing Secure Tunnel

Use the following instructions to help you install Secure Tunnel for use with IBM Cloud Pak for AIOps.

Two methods are avilable to activate Secure Tunnel for use with IBM Cloud Pak for AIOps:

- Install Secure Tunnel from IBM Cloud Pak for AIOps
- Install Secure Tunnel without IBM Cloud Pak for AIOps (as a stand-alone component) to your Red Hat OpenShift cluster

Install Secure Tunnel from IBM Cloud Pak for AIOps

1. Ensure that IBM Cloud Pak for AIOps is installed.
2. An IBM Cloud Pak for AIOps console user with the `Automation Administrator` or `Administrator role` is needed to use Secure Tunnel. For more information about users and roles, see Managing user access control.
3. If you want to use Secure Tunnel to integrate with Slack or Microsoft Teams, you need to create a Slack integration or a Microsoft Teams integration on the IBM Cloud Pak for AIOps console. For more information, see Creating Slack integrations and Microsoft Teams integrations.
4. The Secure Tunnel component is not enabled by default at IBM Cloud Pak for AIOps installation time. Before you use Secure Tunnel, you need to enable the Secure Tunnel component.

   Two methods are available for enabling Secure Tunnel.

   - Console method:

- Access Red Hat® OpenShift® Container Platform console.
- Click **Operators** > **Installed Operators**.
- Select **IBM Cloud Pak for AIOps** operator, and locate the **IBM Cloud Pak for AIOps** instance.
- Set **Enable Secure Tunnel** to `true` to install Secure

- YAML method:
  - Access Red Hat® OpenShift® Container Platform console.
  - From your Red Hat OpenShift console, click **Operators** > **Installed Operators**.
  - From the **Project** menu, select the project where you created the **IBM Cloud Pak for AIOps** instance.
  - Select **IBM Cloud Pak for AIOps** operator, and locate the **IBM Cloud Pak for AIOps** instance.
  - Click **Installation Item**. Switch to the **YAML** view, and update the YAML file by changing the field of `enableConnectionModule` from `false` to `true`.

    ```
    spec:
      enableConnectionModule: true
    ```

    Note: If you cannot find the connection and enabled fields in the YAML file, you need to add them to the YAML file manually.
  - Ensure that all Secure Tunnel pods are in running state by running the following command:

    ```
    oc -n <namespace> get pod | grep tunnel
    ```

    Where `<namespace>` is the IBM Cloud Pak for AIOps namespace.

    The command result can resemble the following example:

    ```
    NAME                                              READY   STATUS    RESTARTS   AGE
    ibm-secure-tunnel-operator-677c7b4b7-gqjbg        1/1     Running   0          150m
    sre-tunnel-controller-7b9d59957b-rkxp7            1/1     Running   0          147m
    sre-tunnel-tunnel-network-api-985df588c-z6m4z     1/1     Running   0          147m
    sre-tunnel-tunnel-ui-mcmtunnelui-56c744598-qhlfw  1/1     Running   0          147m
    ```

5. **(Optional) Enable EgressNetworkPolicy**: There is no egress firewall policy defined when you install Secure Tunnel, so outgoing traffic from workload pods to the internal and external network is unrestricted. If you require a more secure environment, complete the following steps:

   1. Go to your **OpenShift console** > **Operators** > **Installed Operators** > **IBM Secure Tunnel** > **Tunnel** > **sre-tunnel** > **YAML**.
   2. Add the following configuration to the YAML of the Tunnel CR to enable the egress NetworkPolicy:

      ```
      spec:
        enableEgressNetworkPolicy: true
        ...
      ```

      Note: This feature only supports Red Hat OpenShift Container Platform version 4.10 and above, and ensures there are not any restrictive EgressNetworkPolicies in place.

6. The Secure Tunnel Connector can be installed to both a cluster (OpenShift or IBM Kubernetes Service cluster) or a host machine (virtual machine or physical machine). Depending on which method you use, make the following preparations.

   - **If you want to install the Connector in a cluster (OpenShift or IBM Kubernetes Service cluster)**:
     - Ensure that the following minimum resource requirements are met:
       - 1 vCPU
       - 1 GB random access memory (RAM)
     - Ensure that you can access the cluster from a public network. The cluster domain name needs to be resolved by a public DNS.
     - Ensure that the cluster has a *CA-signed TLS certificate* so that the verification handshake of the Slack server or the Microsoft Teams server can be successful. A self-signed TLS certificate does not work for a Slack connection.
   - **If you want to install the Connector in a host machine (virtual machine or physical machine)**:
     - Ensure that the following minimum resource requirements are met:
       - 1 CPU
       - 1 GB random access memory (RAM)
     - Ensure that Podman is installed, and running on the virtual machine or physical machine.
     - If this is used to integrate with the Slack or MS Teams. You need to meet the following extra conditions:
       - Ensure that you can access the host machine from a public network. The hostname needs to be resolved by a public DNS.
       - Ensure that a CA-signed TLS certificate is available. This is used when you create the ChatOps connection.

         Note: If you have an IBM Cloud account, see Getting started with IBM Cloud Internet Services to apply for a **DNS name**, and see Preparing to order public certificates to apply for a CA-signed certificate. Extract the CA-signed certificate file, `hostname.pem`, for the certificate and `hostname.key` for the private key. If an intermediate file is in your CA-signed certificate file – such as hostname-intermediate.pem – concatenate this file with a cert file, such as hostname.pem, by running the following command:

         ```
         cat hostname-intermediate.pem >> hostname.pem
         ```

Install Secure Tunnel without IBM Cloud Pak for AIOps (as a stand-alone component) to your OpenShift cluster

1. Ensure that your cluster meets the following requirements:

- Supported Red Hat OpenShift Container Platform versions
- Hardware requirements: Ensure that the following minimum resource requirements are met:
  - 2 vCPU
  - 2 GB random access memory (RAM)

2. Create the catalog source with the OpenShift console

   1. Log in to your Red Hat OpenShift cluster's console.

   2. Add the IBM Operators CatalogSource. Click the **plus icon** in the upper right corner to open the *Import YAML* dialog box. Paste in the following content, and then click **Create**:

      ```
      apiVersion: operators.coreos.com/v1alpha1
      kind: CatalogSource
      metadata:
        name: ibm-aiops-catalog
        namespace: openshift-marketplace
      spec:
        displayName: ibm-aiops-catalog
        publisher: IBM Content
        sourceType: grpc
        image: icr.io/cpopen/ibm-aiops-
      catalog@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753
      ```

   3. Go to **Administration** > **Cluster Settings**. Under **Global Configuration** > **OperatorHub** > **Sources**, verify that the `ibm-aiops-catalog` CatalogSource object is present.

      Note: For more information, see Catalog Sources

3. Install the Secure Tunnel operator with the Red Hat OpenShift console

   1. Log in to your Red Hat OpenShift cluster's console.
   2. Click **Operators** > **OperatorHub**. The *OperatorHub* page is displayed.
   3. In the *All Items* field, enter **IBM Secure Tunnel**. The Secure Tunnel operator is displayed.
   4. Click the **IBM Secure Tunnel** tile. The **IBM Secure Tunnel** window is displayed.
   5. Click **Install**. The *Install Operator* page is displayed.
   6. Enter the following values:
      1. Set the **Installation mode** to **All namespaces on the cluster(default)**.
      2. Set **Update Channel** to **v4.8**.
      3. Set **Approval Strategy** to **Automatic**.
   7. Click **Install** and wait for the Secure Tunnel operator to install.
   8. Verify that the IBM Cloud Pak for AIOps operator is successfully installed.
      1. Navigate to **Operators** > **Installed Operators**, and select your project from the *Projects* dropdown. Secure Tunnel operator in the project is listed with a status of *Succeeded*.

4. Install Secure Tunnel

   1. To create a Secure Tunnel project (namespace) with the Red Hat OpenShift console From your Red Hat OpenShift console, click **Home** > **Projects**. Select **Create Project**, specify the name of the project that you want to create (for example `tunnel-stand-alone`) and click **Create**.

      Note: For more information, see Namespace.

   2. Create the entitlement key secret with the Red Hat OpenShift console. Complete the following steps to create a docker-registry secret to enable your deployment to pull the Secure Tunnel images from the IBM® Entitled Registry:

      1. Obtain the entitlement key that is assigned to your IBMid. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.

      2. In the *Entitlement key* section, select **Copy key** to copy the entitlement key to the clipboard.

      3. From your Red Hat OpenShift console, click **Workloads** > **Secrets**.

      4. From the **Project** menu, select the **project** that you created earlier in **Create a Secure Tunnel project (namespace)**.

      5. Click the **Create** button, and select Image Pull Secret from the menu. The Create Image Pull Secret form is displayed. Enter the following values and then click **Create**.

         - **Secret Name:** `secure-tunnel-pull-secret`
         - **Authentication Type:** Image Registry Credentials
         - **Registry Server Address:** `cp.icr.io`
         - **Username:** `cp`
         - **Password:** Use the entitlement key that you copied earlier.

      Note: For more information, see Entitlement Keys

   3. Create an instance of the Secure Tunnel with the Red Hat OpenShift console. Secure Tunnel is supported by namespace isolation, so you can create multiple Secure Tunnel instances in different namespaces, for different users. Secure Tunnel also uses the Red Hat OpenShift OAuth&RBAC for authentication and authorization. You can configure the permissions for each Secure Tunnel instance user.

      1. From your Red Hat OpenShift console, click **Operators** > **Installed Operators**.

      2. From the **Project** menu, select the project that you want to create the Secure Tunnel instance in. Use the project that you created earlier in **Create a Secure Tunnel project (namespace)**.

      3. Select **IBM Secure Tunnel** operator.

      4. Click **Create Installation** on the **Tunnel** box. The default **Form View** is displayed. Enter the following values:

- **Name:** Specify the name that you want your Secure Tunnel instance to be called.
- **License:** Expand the *License* section and read the agreement. Toggle the *License Acceptance* switch to *True* to accept the license.
- **Image Pull Secret:** Select the `secure-tunnel-pull-secret` secret that you created in the step **Create the entitlement key secret**.
- **runAsStandAlone**: If the **IBM Certificate manager** is not installed in your environment. Toggle this switch to **True** and it will create the TLS Certificate that is used by the Secure Tunnel without the **IBM Certificate manager**.
- **Size:** Select the size that you require for your Secure Tunnel installation.
- **Audit Receiver:** Refer to the audit log guide
- (Optional) **Enable EgressNetworkPolicy:** There is no egress firewall policy defined when you install Secure Tunnel, so outgoing traffic from workload pods to the internal and external network is unrestricted. If you require a more secure environment, toggle this switch to `True` to enable the egress NetworkPolicy. ***Note:** This feature only supports Red Hat OpenShift Container Platform version 4.10 and above, and ensures there are not any restrictive EgressNetworkPolicies in place.

Refer below to the tunnel parameters with their description, variable type and default values.

Tunnel Settings

Parameter	Description	Default value	Type
size	The sizing of the Secure Tunnel. Supported values: `small`, `medium`, `large`. `small`: replica is 1 for all operands, no HA; `medium`: replica is 3 with HA enabled; `large`: replica is 3 with HA enabled and with extra capacity for CPU/memory.	small	String
license.accept	Expand the License section and read the agreement. Toggle the License Acceptance switch to True to accept the license.	false	Boolean
uiReplicas	The replicas of the UI server pods.	Depends on size	int
apiReplicas	The replicas of the API server pods.	Depends on size	int
controllerReplicas	The replicas of the controller server pods.	Depends on size	int
workerReplicas	The replicas of the worker server pods per tunnel connection.	Depends on size	int
pullSecret	The pull secret of the Secure Tunnel images.	secure-tunnel-pull-secret	String
runAsStandAlone	If you want to install the Secure tunnel without IBM Cloud Pak for AIOps, toggle this to True	false	Boolean
showInContainerLog	Show audit logs in the container log.	true	Boolean
auditReceiver.mongoDB.mongoAddress.name	The name of the audit log MongoDB receiver.	null	String
auditReceiver.mongoDB.mongoAddress.mongoAddress	The audit log MongoDB receiver address. Example: `<IP>:27017`	null	String
auditReceiver.mongoDB.mongoAddress.passwordSecretName	The Kubernetes secret that contains the audit log MongoDB receiver login username and password. Create it in the Secure Tunnel namespace. It should contain "user" and "password" data items.	null	String
auditReceiver.mongoDB.mongoAddress.mongoDatabaseName	The database name of the audit log MongoDB receiver	null	String
auditReceiver.mongoDB.mongoAddress.tlsSecret The tls Kubernetes secret name of the audit log MongoDB receiver. Create it in the Secure Tunnel namespace. It should contain "ca.crt", "tls.crt" and "tls.key".	null	String	
auditReceiver.syslog.name	The name of the audit log Syslog receiver.	null	String
auditReceiver.syslog.syslogAddress	The server address of the audit log Syslog receiver.	null	String
defaultTunnelWorkerResources.limits.cpu	The default Tunnel Worker cpu limit.	3000m	String
defaultTunnelWorkerResources.limits.memory	The default Tunnel Worker memory limit.	3072Mi	String
defaultTunnelWorkerResources.requests.cpu	The default Tunnel Worker cpu requests.	100m	String
defaultTunnelWorkerResources.requests.memory	The default Tunnel Worker memory requests.	128Mi String	
applicationMappingPortRang	Application Mapping listen port range. For example: 30000-65000	30000-65000	String
enableEgressNetworkPolicy	If you require a more secure Secure Tunnel environment you can enable this item. Then it will create the egress network policy automatically. Note: This only works on Red Hat OpenShift 4.10 or later.	false	Boolean
connectionTLSCertDurationTime	By default, the TLS certificate that is used by the Tunnel connection. Duration time is 87600h0m0s(10 years). Update this item if you want to change the duration time.	87600h0m0s(10 years)	String

5. Click **Create**. This creates the Secure Tunnel instance.

6. After a few minutes, select the project that you specified for your Secure Tunnel instance from the **Project** menu. Navigate to **Operators** > **Installed Operators** > **IBM Secure Tunnel** and then click the **Tunnel** tab. Under **Installations**, look for the entry with the name that you specified for your Secure Tunnel instance, and verify that it has a *Status* of **Phase: Running**.

7. Refer to: Production installation

Note: One project(namespace) can only create one Secure Tunnel instance.

5. Access the Secure Tunnel console:
    1. From your Red Hat OpenShift console, click **Networking** > **Routes**.
    2. From the **Project** menu, select the project where you created the **Secure Tunnel** instance.
    3. Select the **sre-tunnel-tunnel-ui**.
    4. Click the url of the **Location**. The Secure Tunnel console opens. Log in with your Red Hat OpenShift Username and password.

Uninstalling Secure Tunnel

- **If the Secure Tunnel was installed from IBM Cloud Pak for AIOps**, complete the following steps to disable it.
    1. From your Red Hat OpenShift console, click **Operators > Installed Operators**.
    2. From the **Project** menu, select the project where you created the **IBM Cloud Pak for AIOps** instance.
    3. Select the **IBM Cloud Pak for AIOps operator, and locate the **IBM Cloud Pak for AIOps** instance.
    4. Click **Installation Item**. Switch to the **YAML** view, and update the YAML file by changing the `enableConnectionModule` value for the connection from `true` to `false`.

        ```
        spec:
          enableConnectionModule: false
        ```

- **If the Secure Tunnel was installed from the Operator hub as stand-alone**, complete the following steps to uninstall it.
    1. From your Red Hat OpenShift console, click **Operators > Installed Operators**.
    2. From the **Project** menu, select the project where you installed the **IBM Secure Tunnel**.
    3. Select **IBM Secure Tunnel** operator, and locate the Tunnel instance.
    4. Delete the *sre-tunnel* (by default) instance of the Tunnels.
    5. Uninstall the IBM Secure Tunnel operator from **Operators > Installed Operators**.

Note: There can be some resources, such as the CRDs and TLS cert secret of the secure tunnel that will not be deleted. If you want to completely clear them, you can download the `uninstall-secure-tunnel.sh` and run command `./uninstall-secure-tunnel.sh -n <the namespace of the secure tunnel>` to do that.

Install the Secure Tunnel Connector

Install the Secure Tunnel Connector for the network being connected to a cluster, Red Hat® OpenShift® on IBM Cloud® service (ROKS) cluster, or a host machine (VM or physical). With the Secure Tunnel Connector installed you can establish a secure tunnel between the network of the cluster where you installed the **Secure Tunnel** (server), such as your IBM Cloud Pak for AIOps cluster, and the cluster where you installed the **Secure Tunnel Connector**.

Important Depending on how you plan to create your Secure Tunnel connection, the Secure Tunnel Connector can be installed automatically. For instance, if you are connecting IBM Cloud Pak for AIOps to a Red Hat OpenShift Container Platform cluster for a ChatOps application, the Secure Tunnel Connector can be installed automatically. For more information, see Creating a Secure Tunnel connection.

Note: If your Secure Tunnel server is installed on an OpenShift cluster with a domain name that is not resolvable from DNS, then Secure Tunnel might not be able access it through the OpenShift URL. So you might find that the Tunnel Connector fails to connect to the Secure Tunnel server. You need to log in to the host machine, on which Tunnel Connector is installed, and add the IP and domain (for example, 9.30.111.xxx console-openshift-console.apps.test.cp4waiops.com) to `/etc/hosts` to resolve all the domain names. Follow these steps:

1. Login to the host machine on which Tunnel Connector is installed.
2. Find out all the domain names to be added, run this command in the `tunnel-connector-install-scripts` directory

    ```
    cat install-connector.sh | grep "-e SERVER=" | head -n 1
    ```

    You find outputs like this:

    ```
    SERVER='https://r8b74ff6d23d84fd50-cp4waiops.apps.svt-airgap-1.svt.cp4waiops.com/sre-tunnel-
    tunnel/ws;https://r8b74ff6d23d84fd51-cp4waiops.apps.svt-airgap-1.svt.cp4waiops.com/sre-tunnel-
    tunnel/ws;https://r8b74ff6d23d84fd52-cp4waiops.apps.svt-airgap-1.svt.cp4waiops.com/sre-tunnel-tunnel/ws;' \,
    ```

3. Add each domain in the output to the `/etc/hosts` with the IP address of your OpenShift cluster(for example `9.30.111.xxx r8b74ff6d23d84fd50-cp4waiops.apps.svt-airgap-1.svt.cp4waiops.com`)
4. If you already installed Tunnel Connector before doing these steps, then you need to restart Tunnel Connector by using the command `podman restart <the container id>`.
5. If you completed the above steps and the tunnel connection console UI displays the **Ready** status, but your application, which is using the Secure tunnel, still does not work, check whether you added all domains to `/etc/hosts`.
6. If you selected `Cloud` when creating the Tunnel Connector, then you need to re-create the Tunnel Connector, select `Host machine`, then download the Tunnel Connector installation package and install it to a host machine.

Prerequisites

Download the Secure Tunnel Connector installation scripts:

- From the console, go to **Manage connections**, locate the tunnel connection name, and click **Action** > **Install Connector**.
- Obtain the installation package **`tunnel-connector-install-scripts.tar.gz`** by clicking **Download**. Or you can click **Copy**, and then run the copied commands on a Linux system.

  Note: The installation script is for only a specific **Secure Tunnel Connection**, and cannot be used for other Secure Tunnel Connections.

Installing the Secure Tunnel connector to a cluster

1. Transfer the installation package **`tunnel-connector-install-scripts.tar.gz`** to a virtual machine where you can log in and access the cluster where you want to install the Connector.

   Note: Ensure that the OpenShift **`oc`** or **`kubectl`** tool is installed in the virtual machine.

2. Log in to to the cluster on the virtual machine with the **`oc login`** command, and unpack the installation package.
3. Create the project (namespace) where you want to install the Connector.
4. Unpack the **`tunnel-connector-install-scripts.tar.gz`** package.
5. Change to the **`tunnel-connector-install-scripts`** directory.
6. Run the following command to install the Connector:

   ```
   ./install-connector.sh --namespace <namespace> --accept-license true
   ```

7. Verify whether the Connector is installed by running the following command:

   ```
   oc get po -n <namespace>
   ```

Installing the Secure Tunnel connector to a host machine (virtual machine or physical machine)

1. Transfer the installation package **`tunnel-connector-install-scripts.tar.gz`** to the virtual machine where you want to install the **`Connector`**.
2. Log in to the virtual machine and unpack the installation package.
3. Install Podman on the virtual machine.

   Note: Docker is not shipped or supported for Red Hat Enterprise Linux (RHEL) 8. The Podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 systems. For more information, see Running containers without Docker in the Red Hat documentation.

4. Unpack **`tunnel-connector-install-scripts.tar.gz`**.
5. Change to the **`tunnel-connector-install-scripts`** directory.
6. Run the **`./install-connector.sh`** script to install the Connector on the virtual machine. Accept the license agreement when prompted to start the installation. Alternatively, you can accept the license by adding the **`--accept-license true`** parameter on the command line.
7. When the script finishes, verify that the Connector is installed by running the following command:

   ```
   podman ps
   ```

Installing more than one Secure Tunnel Connector to a host machine

If you want to install more than one Secure Tunnel Connector:

- Ensure they are from a different Secure Tunnel server or different IBM Cloud Pak for AIOps instance. If they are from the same Secure Tunnel server or IBM Cloud Pak for AIOps instance, you just need to add another ApplicationMapping to one **Tunnel Connection** instead of creating another **Tunnel Connection** and installing another **Tunnel Connector**.
- Configure each Secure Tunnel operator to add the **`applicationMappingPortRang: 30000-65000`** item to the **`YAML`**. The **`applicationMappingPortRang`** should be different for each Secure Tunnel. For example, the first one should be **`30000-40000`** and the second should be **`40001-50000`**. Follow these steps to configure the **`applicationMappingPortRang`**:

  1. From the Red Hat OpenShift Container Platform console, click **Operators** > **Installed Operators**.
  2. From the **Project** dropdown menu, select the project in which you created the Secure Tunnel instance.
  3. Click the **IBM Secure Tunnel** > **Tunnel** tab > click the existing **sre-tunnel** instance > change to **YAML** tab.
  4. Add **`applicationMappingPortRang: 30000-65000`** to the YAML and save it. The final YAML should look like:

```
spec:
  license:
    accept: true
  version: 1.0.0
  pullSecret: secure-tunnel-pull-secret
  showInContainerLog: true
  runType: production
  size: small
```

```
controllerReplicas: 1
applicationMappingPortRang: 30000-65000
```

The Secure Tunnel Connector container runs in the host network on host machines. Avoid port conflicts if you install multiple Secure Tunnel Connectors in the same host machine. For the following reasons, do not use port numbers **9000**, **50443** or **12443**

- Secure Tunnel Connector container listens on port **9000** for health check by default.
- The Secure Tunnel Connector container in a *private network to public network connector* setup connects to the Secure Tunnel server on port **50443** by default.
- Secure Tunnel Connector container for **Slack** and **Microsoft Teams** listens on port **12443** on **Exposed mapping port** by default.
- When you create a **Secure Tunnel Connection** from a **Custom** template and select **Secure Tunnel in private network initializes connection to connector in public network** in the **Direction** page, and select **Host machine** in the **Connector settings** page you must input a different port number than **50443** for your connection. And this port number must be different than the port number for all the connectors that run on the same host machine.

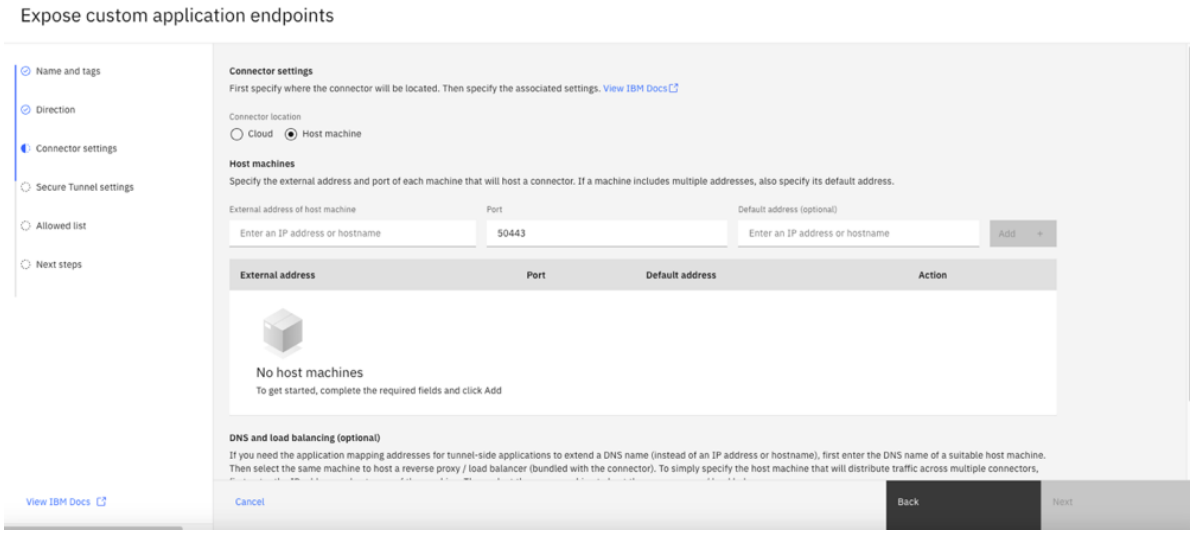


Figure. Port configuration

- When you create a **Secure Tunnel Connection** from the **ChatOps** template, and select **Host machine** in the **Connector settings** page you must input a different port number than **50443** in the **Port** field and a different port number to **12443** in **Exposed mapping port** field. These port numbers must be different than the port numbers for all the connectors you are running on the same host machine.

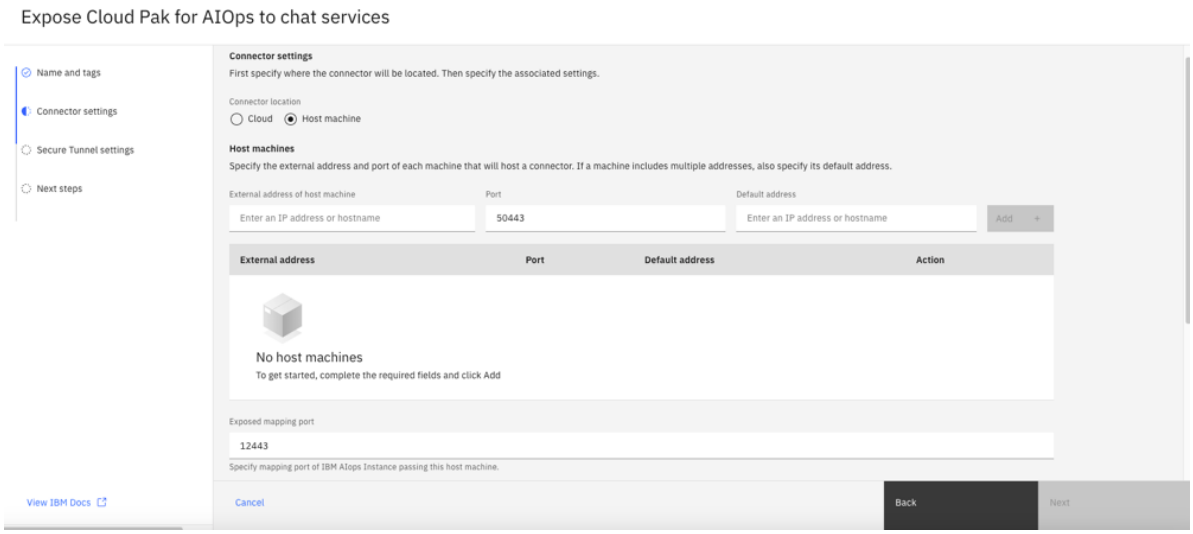


Figure. Port configuration

- When you complete creating the connection, you can download and transfer the installation package to the host machine on which a Secure Tunnel connector is already installed.
- When you install the connector you must specify a different health check port to **9000** and the other health check ports used by other secure tunnel connectors that run on this host machine. Use this command to assign a specific **<port number>**,

```
./install-connector.sh --accept-license true --port <port number>
```

Note: For Port conflicts **install-connector.sh** uses **fuser** to check port conflicts during installation. If you do not have **fuser** installed on the host machine, **install-connector.sh** might miss some conflicts. If it does, your connector container might fail to start.

You can check the Secure Tunnel Connector container log by running the following command.

`podman logs <container id>`

where **<container id>** is the id of the container.

- If you find a port conflict on the health check port in the container log:
    1. Uninstall the container by using the command

       `./uninstall-connector.sh`

    2. Reinstall the connector with a different health check port. For example, `./install-connector.sh --accept-license true --port 9002`
- If you find a port conflict on the port that is used to connect the Secure Tunnel server and/or on the **Exposed mapping port**, then
    1. Uninstall the container by using the command:

       `./uninstall-connector.sh`

    2. Delete the Secure Tunnel connection from the web UI.
    3. Create a new Secure Tunnel connection with different ports and try again.

For more information, see Troubleshooting Secure Tunnel.

Optional: Uninstalling the Secure Tunnel Connector

Use the script to uninstall the **Secure Tunnel Connector**:

1. Download the Connector installation package `tunnel-connector-install-scripts.tar.gz`
    1. From the console, go to **Manage connections**, locate the tunnel name.
    2. Under the **Action** column for the tunnel, expand the menu (three dots icon) and click **Install**.
    3. Obtain the installation package `tunnel-connector-install-scripts.tar.gz` by clicking **Download**. Or you can click **Copy**, and then run the copied commands on a Linux system.

       Note: The installation script is for only a specific **Secure Tunnel Connection**, and cannot be used for other Secure Tunnel Connections.
2. Uninstall the Connector from the **Cloud** with the installation package:
    1. Transfer the installation package `tunnel-connector-install-scripts.tar.gz` to a virtual machine on which the OpenShift CLI (`oc`) is installed. Log in and access the **Cloud** where you want to uninstall the Connector.
    2. Log in to the virtual machine and unpack the installation package.
    3. Change to `tunnel-connector-install-scripts` directory
    4. Run `./uninstall-connector.sh` script with parameters like below to uninstall the Connector from the Cloud.

       `./uninstall-connector.sh --namespace <namespace>`
3. Uninstall the Connector from the **host machine** with installation package:
    1. Transfer the installation package `tunnel-connector-install-scripts.tar.gz` to the host machine on which the Connector is installed.
    2. Log in to the virtual machine and unpack the installation package.
    3. Change to `tunnel-connector-install-scripts` directory.
    4. Run `./uninstall-connector.sh` script with the following parameters to uninstall the Connector on the host machine.

       `./uninstall-connector.sh --namespace <namespace>`

Optional: Delete the Secure Tunnel Connector resources after a tunnel connection is deleted

If you deleted a tunnel connection by clicking **Action** > **Delete** in the *Manage connections* page on the IBM Cloud Pak for AIOps console, and you did not uninstall the connector, you can delete the connector resources in the cluster virtual machine or the host machine manually. Otherwise, a new connector cannot be installed on the Cluster virtual machine or the host machine.

- If the connector is running in an Red Hat OpenShift Container Platform cluster or Red Hat OpenShift on IBM Cloud, follow the steps to delete connector resources:
    1. Log in to the Red Hat OpenShift Container Platform cluster or Red Hat OpenShift on IBM Cloud cluster from command line, and then find all connector resources by running the following command:

       ```
       oc get sts,secret,service,role,route -n <your tunnel connector namespace> \
             -l app.kubernetes.io/group=<your release name>-tunnel-connector.app
       ```

    2. You can find the `release name` by checking the tunnel server tunnel prefix of `tunnel-network-api`, like this, which gives the `release name` as `sre-tunnel`:

       ```
       oc get deploy -n cp4aiops | grep tunnel-network-api
           sre-tunnel-tunnel-network-api                          3/3     3            3           142m
       ```

    3. Remove these resources by running this command:

```
for res in $(oc get sts,secret,service,role,route -n <your tunnel connector namespace> \
      -l app.kubernetes.io/group=<your release name>-tunnel-connector.app -o name )
do
oc delete -n <your tunnel connector namespace> $res
done
```

4. If there is no other Tunnel connector installed in the namespace, delete the secret by running the command:

   ```
   oc get secret -n <your tunnel connector namespace> | grep tunnel-connector
   ```

   Note, Do not attempt to delete the secret if the tunnel connectors are still present as the secret might be used by one of the connectors.

Upgrading Secure Tunnel

Use the following instructions to help you upgrade Secure Tunnel either with IBM Cloud Pak for AIOps or as a standalone component.

Two methods are avilable to upgrade Secure Tunnel for use with IBM Cloud Pak for AIOps:

- Upgrading Secure Tunnel from IBM Cloud Pak for AIOps
- Upgrading Secure Tunnel as a stand-alone component in your Red Hat OpenShift cluster

Upgrading Secure Tunnel from IBM Cloud Pak for AIOps

Secure Tunnel is upgraded when you upgrade IBM Cloud Pak for AIOps. For details about upgrading IBM Cloud Pak for AIOps, see Upgrading.

Upgrading Secure Tunnel as a stand-alone component in your Red Hat OpenShift cluster

1. **Configure automatic catalog polling**

   1. Log in to the console for your OpenShift cluster.
   2. Go to **Administration** > **Cluster Settings**.
   3. Click **Configuration**, then scroll down and click **OperatorHub**.
   4. Select **Sources**, then scroll down and click **`ibm-aiops-catalog`**.
   5. Click **YAML** to switch to the YAML view, and if **`spec.image`** is not set to **`icr.io/cpopen/ibm-aiops-catalog@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753`**, then update the YAML to have the following entry in the **`spec`** block, and save it.

      ```
      spec:
        displayName: ibm-aiops-catalog
        publisher: IBM Content
        sourceType: grpc
        image: icr.io/cpopen/ibm-aiops-
      catalog@sha256:58c0082ad5e9de6bc2869b528e847702f6c183a06ac49a82d64ad8efd339f753
      ```

2. **Update the operator subscription**

   Use the following steps to update the *spec.channel* value of the Secure Tunnel subscription to the release that you want to upgrade to.

   1. Log in to the console for your OpenShift cluster.
   2. Select **Home** > **Search**.
   3. From the **Project** list, select the project (namespace) that your Secure Tunnel subscription is deployed in. This is your Secure Tunnel project if your deployment is namespace scoped, or openshift-operators if your deployment has a cluster wide scope.
   4. In the **Resources** list, select **SUB Subscription**. A list of subscriptions is displayed.
   5. Click the subscription that has the name *ibm-secure-tunnel-operator*. A new window with the subscription details for *ibm-secure-tunnel-operator* is displayed.
   6. Click the value for the **Update channel** box. A new window called *Change Subscription update channel* is displayed.
   7. Change the channel to the version that you want and click **Save**.

      If the subscription has been configured with **Manual** for the **Update approval**, then you need to click **Upgrade available** on the **Upgrade status** > **Preview InstallPlan** > **Approve** to approve this upgrade.

      Note: You need to upgrade the channel versions one by one. For example, if your current channel version is 3.7.x, you cannot skip the 4.1.x to change the channel to 4.2.x. You must upgrade it to 4.1.x and then upgrade it to 4.2.x.

3. **Verify the deployment**

   Use the following procedure to check the status of your upgraded Secure Tunnel deployment. It takes around 2-10 minutes for the upgrade to complete (subject to the speed with which images can be pulled).

   1. Log in to the console for your OpenShift cluster.

2. Click **Operators** > **Installed Operators**.
3. From the **Project** list, select the project (namespace) that Secure Tunnel is deployed in if your deployment is namespace scoped, or openshift-operators if your deployment has a cluster-wide scope.
4. Locate **IBM Secure Tunnel** in the list and verify that the annotation shows the version that is required.

Creating Secure Tunnel connections

A Secure Tunnel connection links the network that contains the Secure Tunnel to a remote network. Your applications can be on either side of the connection.

Secure Tunnel comes with connection templates for common application scenarios, including ChatOps, VMware, and Turbonomic. You can also create custom connections to support more scenarios.

- Prerequisites
- Secure Tunnel Web Console
- Creating a secure tunnel
  - ChatOps connection
    - ChatOps connection (Microsoft Azure as host machine)
  - VMware connection
  - Turbonomic connection
  - OpenStack connection
  - Custom connection
    - Creating a custom Instana connection
    - Creating a Secure Tunnel generic webhook connection

Prerequisites

- Ensure that you have Red Hat OpenShift Container Platform installed on one of your networks (referred to as **network-A** in the following steps). You use this instance of OpenShift to bridge to the other network.
- Ensure that another system (Red Hat OpenShift Container Platform, Kubernetes, or other system) or a host machine (virtual machine or physical machine) is available on a different network (referred to as **network-B** in the following steps). You use this network as the location for installing the Secure Tunnel Connector.
- Ensure that the network access is available on **network-A** and **network-B**. Either **network-A** or **network-B** needs to have a public IP address.
- Ensure that you install Secure Tunnel. For more information, see Installing Secure Tunnel.
- Verify that you can access the Secure Tunnel web console.

Secure Tunnel UI

Use either of these methods to access the Secure Tunnel UI:

1. If you installed Secure Tunnel in IBM Cloud Pak for AIOps, log in to the IBM Cloud Pak for AIOps console. From the navigation menu (four horizontal bars), click **Administration** > **Secure Tunnel**.

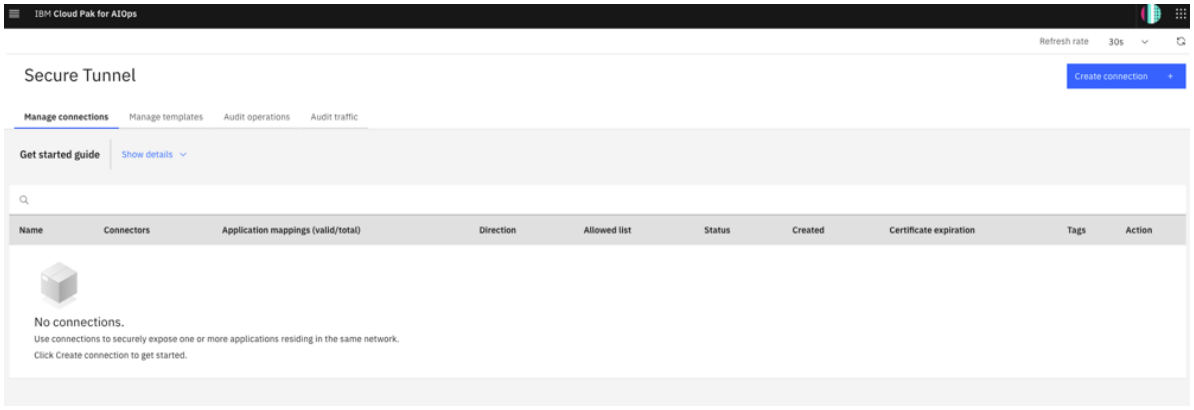


Figure. Secure Tunnel page

2. If Secure Tunnel is not installed in IBM Cloud Pak for AIOps, use the Red Hat OpenShift Container Platform route to access the stand-alone Secure Tunnel. To get the Secure Tunnel URL, enter the following command:

   ```
   oc get route -n <namespace> sre-tunnel-tunnel-ui -o jsonpath='{.spec.host}'
   ```

   Where **<namespace>** is the project (namespace) where Secure Tunnel is installed.

With the Secure Tunnel UI, you can create the following types of connections:

- ChatOps connection
- VMware connection
- Turbonomic connection
- OpenStack connection

- Custom connection

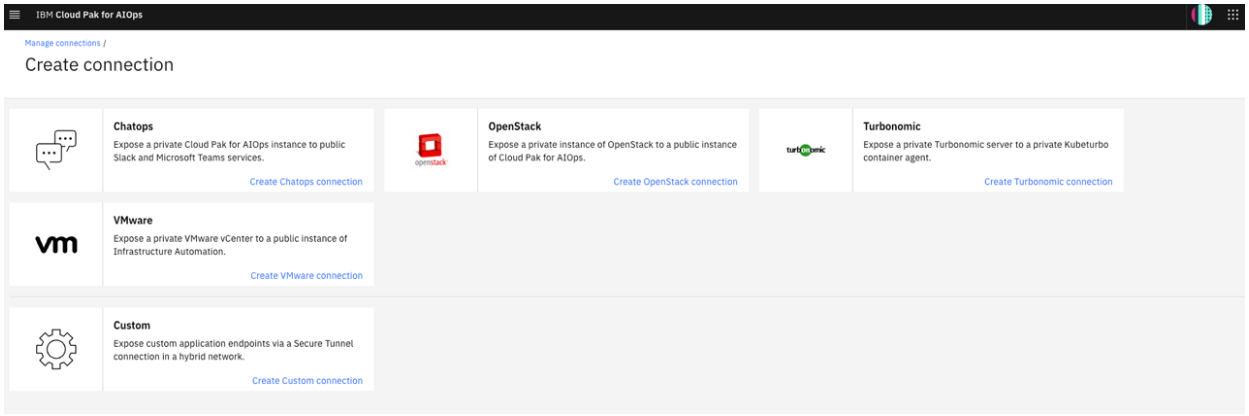


Figure. Secure Tunnel create connection page

Creating a Secure Tunnel

To create a Secure Tunnel connection, click **Create connection** on the Secure Tunnel UI and then follow the steps for the specific connection that you want to create.

ChatOps connection

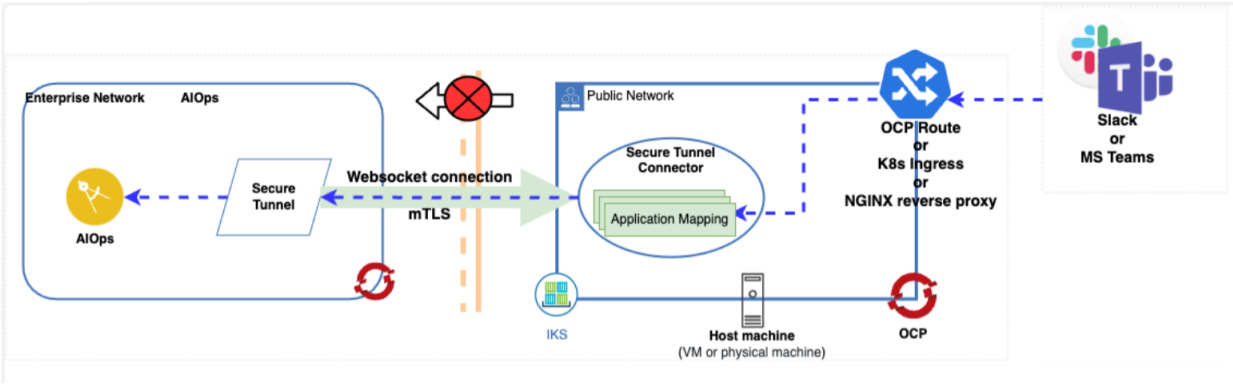


Figure. High-level design for VMware

To integrate IBM Cloud Pak for AIOps with a service like Slack or Microsoft Teams, the IBM Cloud Pak for AIOps instance needs to be accessible from the public network. If your IBM Cloud Pak for AIOps is not accessible from the public network, you can open the IBM Cloud Pak for AIOps instance to the public network by creating a Secure Tunnel. You can do it by exposing the IBM Cloud Pak for AIOps instance to either a cluster (Red Hat OpenShift Container Platform or IBM Kubernetes Service cluster) or a host machine (virtual machine or physical machine), which is accessible from the public network. Simplify the creation of a secure tunnel by using the ChatOps template from the Secure Tunnel Console on the **Manage connections** page. Take the following steps to create a ChatOps connection:

1. Click **Create ChatOps connection** and a panel with an introduction to the ChatOps template opens. Click **Get started**.

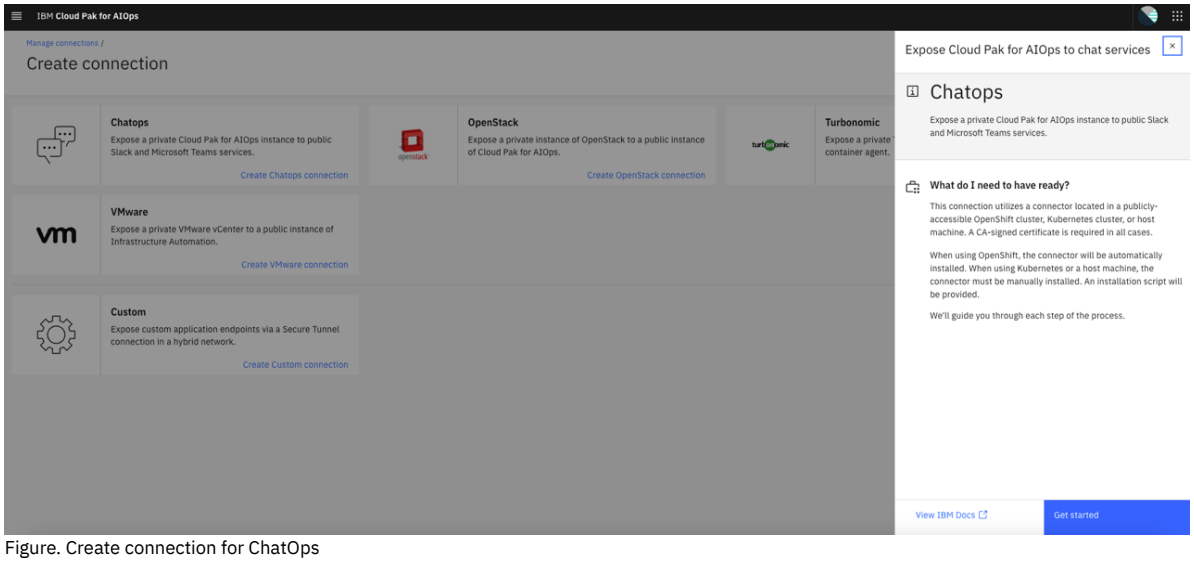


Figure. Create connection for ChatOps

2. In the *Name and Tags* section, provide a name and tag for this connection.

1. *Connection name*: required. The name of the connection. The name can include only lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
2. *New tag*: optional. The tag of the Connection.

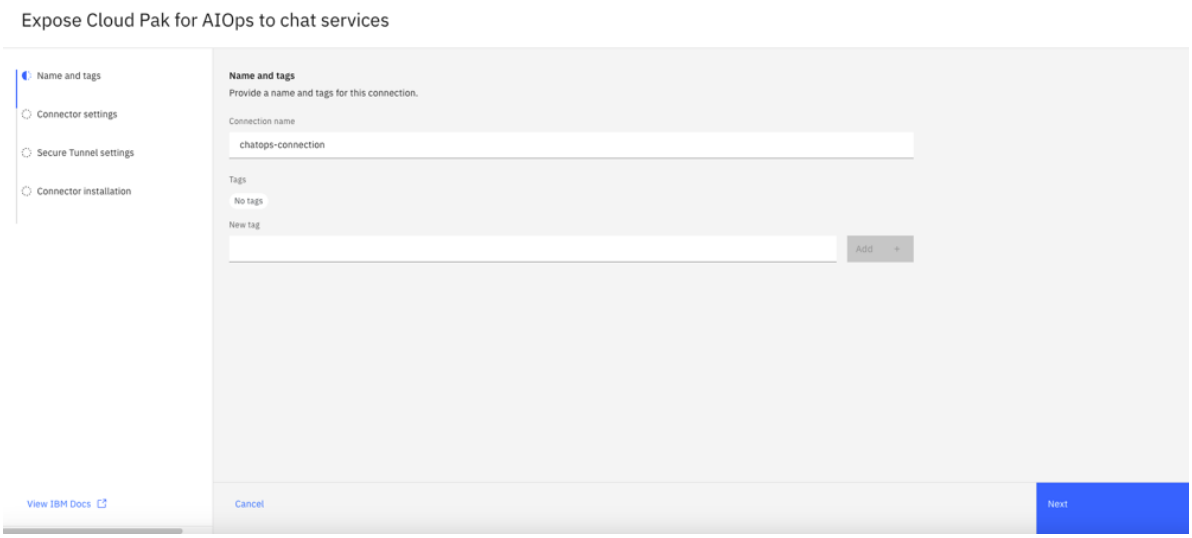


Figure. Name and tags for ChatOps

3. In the *Connector settings* section:

Specify the Connector location, such as *Cloud* or *Host machine*.

- If you choose to install the connector in the Red Hat OpenShift cluster, the connector can be installed automatically. Complete the following actions:
  - *Login command for target Red Hat OpenShift cluster*: required. Specify a publicly accessible cluster with a CA-signed domain certificate. Do not install the connector in the cluster that contains the Tunnel Connection worker.
  - *Target namespace to install the connector*: required. The project (namespace) where you want to install the connector.
  - *Connector replicas*: configure the tunnel connector replicas to set High Availability
- If you choose to install the connector on a Red Hat OpenShift Container Platform cluster manually, you can use the Custom method
- If you choose to install the connector in a Kubernetes cluster, complete the following actions:
  - *Cloud domain*: Run the command to get IBM Kubernetes Service cluster domain name:

```
ibmcloud ks cluster get --cluster <Cluster ID> --output json | jq '.ingress.hostname'
```

  - *Port*: required. The port of the Kubernetes cluster.
  - *Connector replicas*: you will config the tunnel connector replicas to set High Availability.

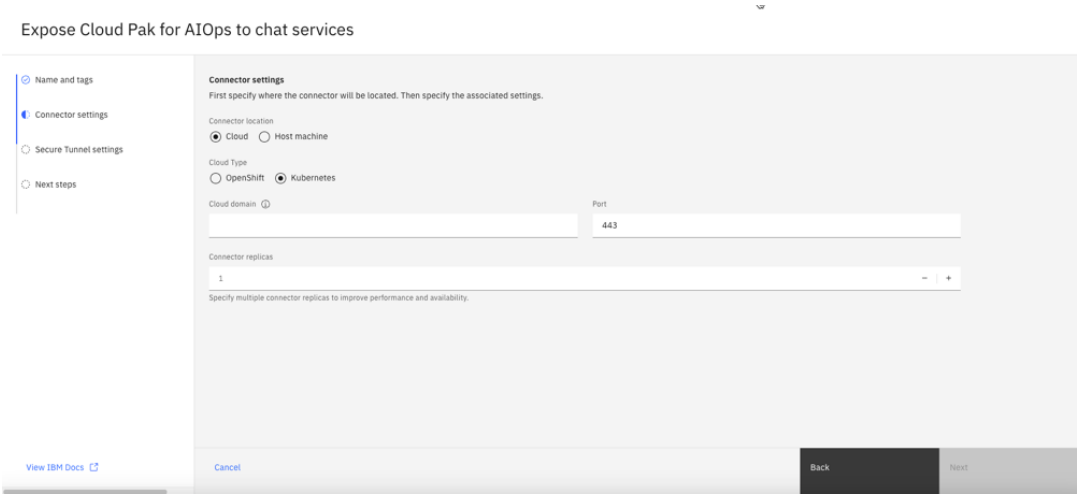


Figure. Connector settings for Kubernetes

- If you choose to install the connector in the *Host machine*, complete the following actions:
  - *Host machine addresses*: required. Includes the host machine external address, Port, host machine default address. When a host machine includes multiple addresses, a default IP address is required. If you want to use one host machine to expose more than one IBM Cloud Pak for AIOps, you need to set a different port for other ChatOps connections.
    - *Expose mapping port*: required. Default is **12443**. You can change this port to allow several IBM Cloud Pak for AIOps instances to have different exposed ports. Then, you can use one host machine to create more than one ChatOps connection.

- *DNS and load balancing*: required. First, enter the DNS name of a suitable host machine. Next, select the same machine to host a reverse proxy / load balancer (bundled with the connector). Finally, specify the corresponding certificate and private key.

  Note: If you have an IBM Cloud account, see Getting started with IBM Cloud Internet Services to apply for a **DNS name**, and see Preparing to order public certificates to apply for a CA-signed certificate. Extract the CA-signed certificate file, **`hostname.pem`**, for the certificate and **`hostname.key`** for the private key. If an intermediate file is in your CA-signed certificate file – such as hostname-intermediate.pem – concatenate this file with a cert file, such as hostname.pem, by running the following command:

  ```
  cat hostname-intermediate.pem >> hostname.pem
  ```

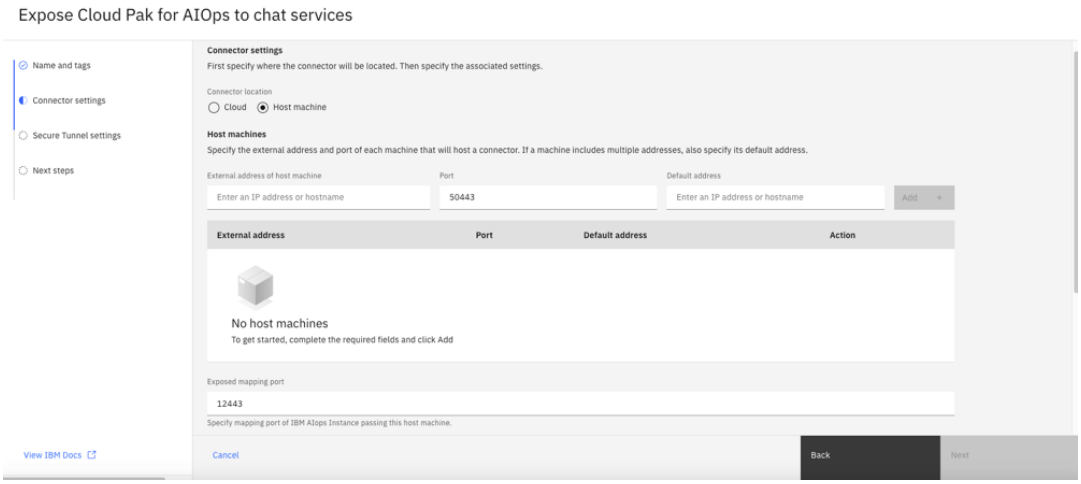


Figure. Connector settings for Host machine

4. In the *Secure Tunnel settings*, configure the tunnel worker replicas to set High Availability.

5. *HTTPS proxy*: optional. Specify a forward proxy to enable the Secure Tunnel to reach the connector in the other network, such as username:password@hostname:port.

   Note: It is necessary to enter the HTTPS proxy information when Secure Tunnel is in an air-gapped environment. If your Secure Tunnel is in an air-gapped environment without internet connectivity, you need to prepare a host machine (virtual machine or physical machine). Ensure that the host machine can be accessed from the IBM Cloud Pak for AIOps cluster, and also that it can connect to the internet. Then, install an HTTPS proxy server in the host machine, and set the HTTPS proxy server in the HTTPS proxy field.

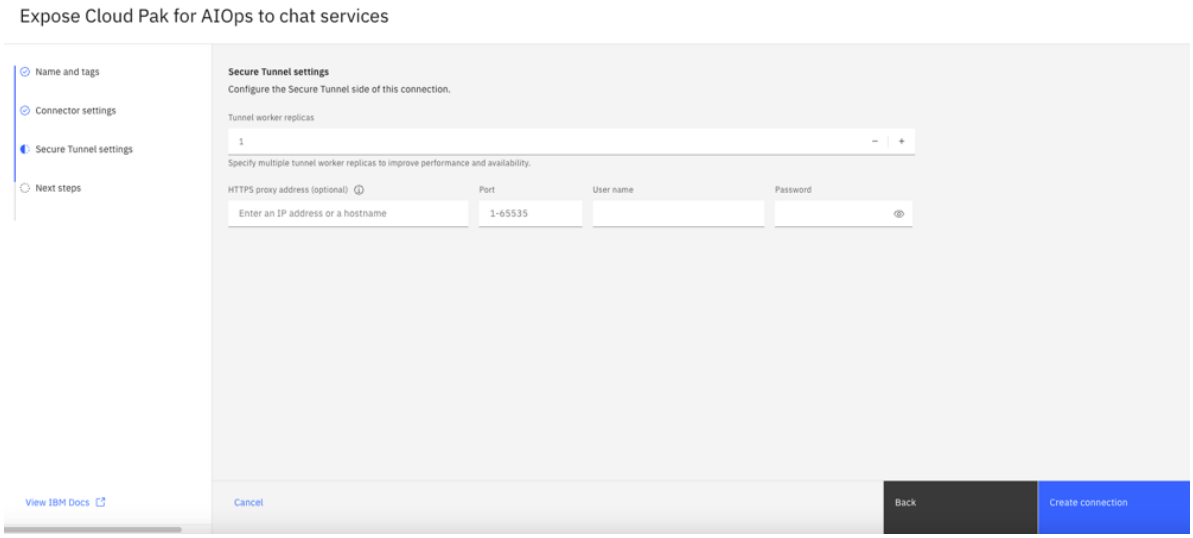


Figure. Secure Tunnel settings for ChatOps

If you install a connector in Red Hat OpenShift Container Platform, and the Secure Tunnel in an air-gapped environment and you configured the HTTPS proxy for it, choose **Kubernetes** for the **Cloud Type** in the above step. Then, to get the domain for the **Cloud domain** field, run

```
HOSTNAME=$(oc get route -n openshift-console console -o=jsonpath={.spec.host}) && HOSTNAME=${HOSTNAME#console-openshift-console.} && echo $HOSTNAME
```

For example, squid is a simple HTTPS proxy server. See the following example of using squid to set up an HTTPS proxy server (Do not use this example in your production environment). For a production environment, contact your network administrator to help you configure the HTTPS proxy.

Save the following configurations in a file **`squid.conf`**:

```
acl all src 0.0.0.0/0.0.0.0
acl SSL_ports port 443
acl CONNECT method CONNECT

http_access deny CONNECT !SSL_ports
```

```
http_access allow all

# Squid normally listens to port 3128
http_port 3128

# Uncomment and adjust the following to add a disk cache directory.
cache_dir ufs /var/spool/squid 100 16 256
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:         1440    20%    10080
refresh_pattern ^gopher:    1440    0%    1440
refresh_pattern -i (/cgi-bin/|\?) 0    0%    0
refresh_pattern (Release|Packages(.gz)*)$      0       20%     2880
# example lin deb packages
#refresh_pattern (\.deb|\.udeb)$   129600 100% 129600
refresh_pattern .         0    20%    4320
```

Start the squid server by running the following command:

```
podman run --name squid -d -p 3128:3128 -v `pwd`/squid.conf:/etc/squid/squid.conf datadog/squid
```

Set the `<host machine IP>:3128` in the **HTTPS proxy** field.

6. Click **Create connection** to create a ChatOps connection.

   - If you choose to install the connector on a Red Hat OpenShift cluster, the connector is automatically installed. You can see the installation status in the connector installation section. When the installation is finished, you can get the application mapping URLs for Slack and Microsoft Teams connections.
   - If you choose to install the connector on Kubernetes or a host machine, the connector must be manually installed. You can get the installation script in the *Next steps* section. Install the connector following this installation guide. When the connector is installed, you can use the appropriate application mapping address to integrate with Slack or Microsoft Teams.

ChatOps connection (Microsoft Azure as host machine)

To set up a ChatOps connection with Microsoft Azure as the host machine, you need to install IBM Cloud Pak for AIOps in *network-A* (private cloud) and set up ChatOps in *network-B* (Slack or Microsoft Teams in public cloud). This process involves configuring ChatOps and a Secure Tunnel out of *network-A* to communicate with ChatOps in the public cloud, *network-B*.

1. Install IBM Cloud Pak for AIOps.
2. User deploys a VM to Azure, gives it a public internet-facing endpoint. **Note:** This must have a signed trusted certificate that ChatOps accepts as trusted (that is not self-signed).
3. On Azure, open the `50443` port on the VM that uses an NSG (Network Security Group).
4. Give the VM a public DNS name.
5. Check whether it is a private IP on the Azure UI. For example, the private IP in this case is `172.18.0.4`:

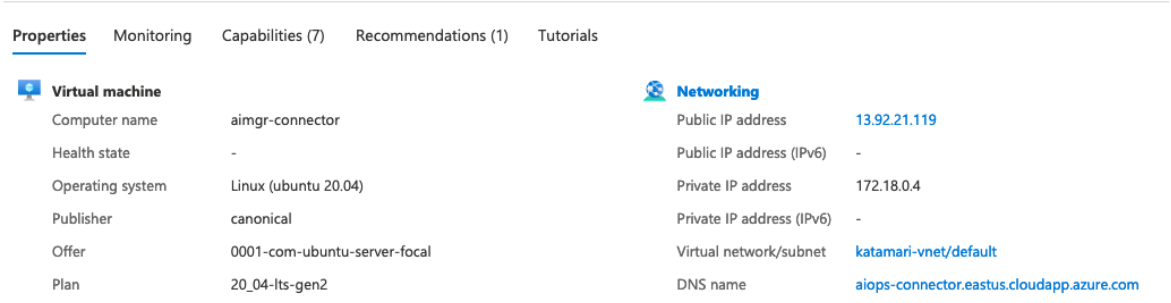


Figure. Azure UI

6. Get the signed trusted certs, including the root and any intermediate CAs.
7. To configure the tunnel connector on the Secure Tunnel UI, enter the following details:

   1. Select *Connector Location* as `Cloud`.
   2. Enter the *External address of host machines*. This is the IP address/addresses of host machines. For example, aiops-cocnnector.eastus.cloudapp.azure.com.
   3. Enter *Port*. This is the port number of the host.
   4. Enter *Exposed Mapping Port* . The default is `12443`. You can change this port to allow several IBM Cloud Pak for AIOps instances to have different exposed ports.
   5. For *DNS and load balancing*, enter the DNS name of a suitable host machine. Next, select the same machine to host a reverse proxy / load balancer (bundled with the connector).

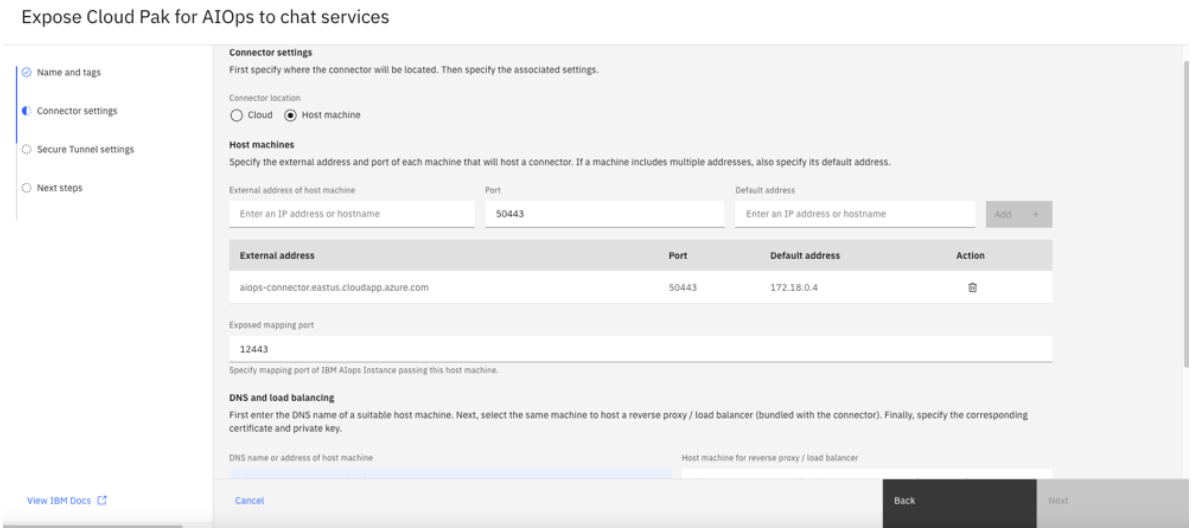


Figure. Connector settings for ChatOps

8. Note where the private IP is entered and note the values for DNS. You need to select the host machine in the dropdown menu.
9. Enter your cert and keys in the connector UI.

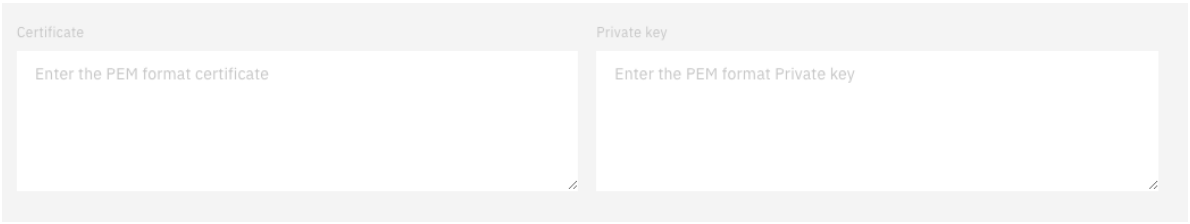


Figure. Enter certifaction keys

10. You then get a **`tar.gz`**. Download this file and transfer it to Azure VM.
11. On the Azure VM, where you unpack the download, ensure that nothing is connected to the **`443`** port (like an Apache server) and run the secure tunnel installer as root, as it needs that privilege to bind to the **`443`** port.
12. You then get a successful installation and see a *Ready* status back on the Secure Tunnel UI.

    Script output

```
Install Tunnel Connector successfully
{"ok":true}
{"ok":true}

Start Tunnel Connector successfully

{"description":""}
{"description":""}
A
Tunnel Connector is ready to use.
```

    IBM Cloud Pak for AIOps UI

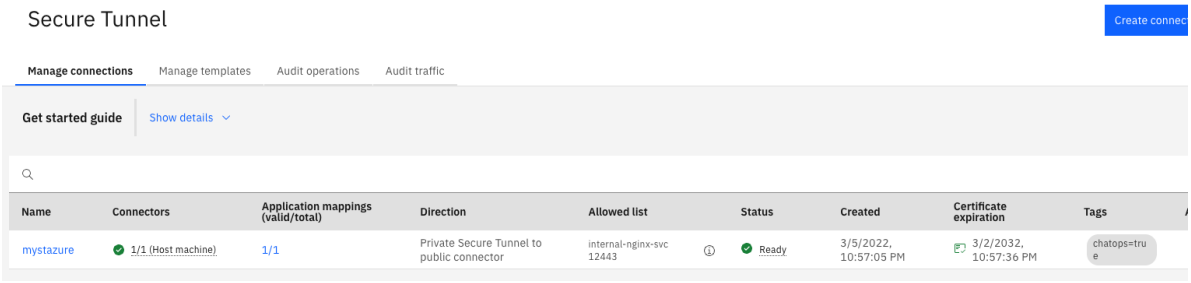


Figure. Additional Secure Tunnel UI

13. Click the application mappings entry, which shows you the URLs to use in your Slack or Microsoft Teams. You can renew the certificate and private key by editing the application mapping.

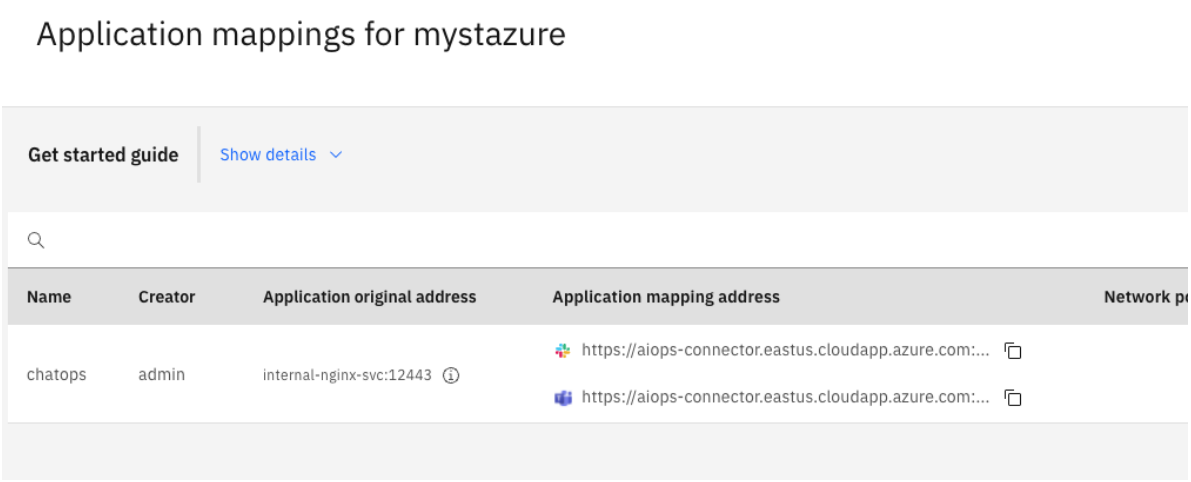


Figure. Application mappings

14. In the Microsoft Teams bot, enter the URL in the *Messaging endpoint* field.

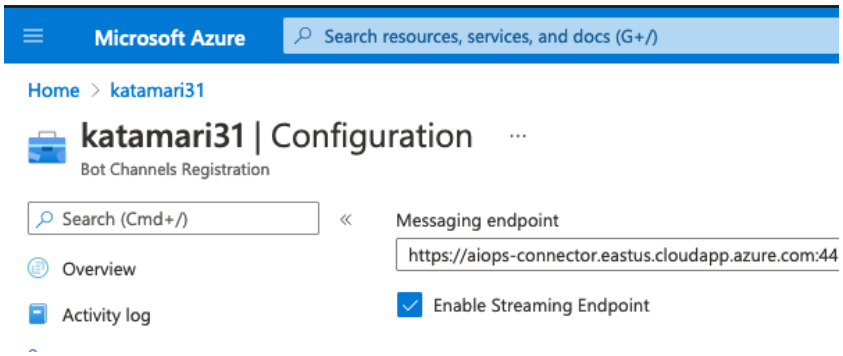


Figure. Azure team

VMware connection

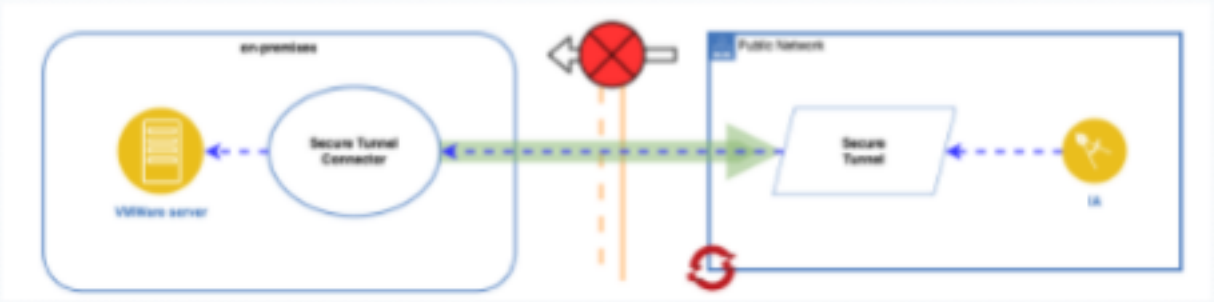


Figure. High-level design for VMware

Follow the installation instructions to install Secure Tunnel.

You can access services in a private VMware® vCenter from a public cloud **Infrastructure automation**. In this case, install a Secure Tunnel in the Infrastructure automation cluster, with IBM Cloud Pak for AIOps or a stand-alone version. Then, take the following steps to configure the VMware tunnel connection:

Note: VMware template supports the VMware scenario specifically. Other scenarios can be created by using a Custom template.

Simplifying the creation of a Secure Tunnel by using VMware template in IBM Cloud Pak for AIOps:

1. Click the **VMware** tile. The VMware side panel opens and provides an introductory explanation to the VMware template. Click **Get started**.
2. In the *Name and Tags* section, provide a name and tag for this connection:
    1. Connection name: required. The name of the connection. The name can include only lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
    2. New tag: optional. The tag of the Connection.
3. In the *Connector settings* section, install the tunnel connector to *Cloud* or *Host machine* by selecting the connector location and fill the section with appropriate details.

    If you choose to install the connector in the *Host machine*, complete the following actions:

    - *Host machine addresses*: optional. Includes the host machine external address, Port, host machine default address. When a host machine includes multiple addresses, a default IP address is required. If you want to use one host machine to expose more than one IBM Cloud Pak for AIOps, you need to set a different port for other ChatOps connections.

- *DNS and load balancing*: optional. First, enter the DNS name of a suitable host machine. Next, select the same machine to host a reverse proxy / load balancer (bundled with the connector).

If you choose to install the connector in the *Cloud*, complete the following actions:

- *Connector replicas*: configure the tunnel connector replicas to high availability.

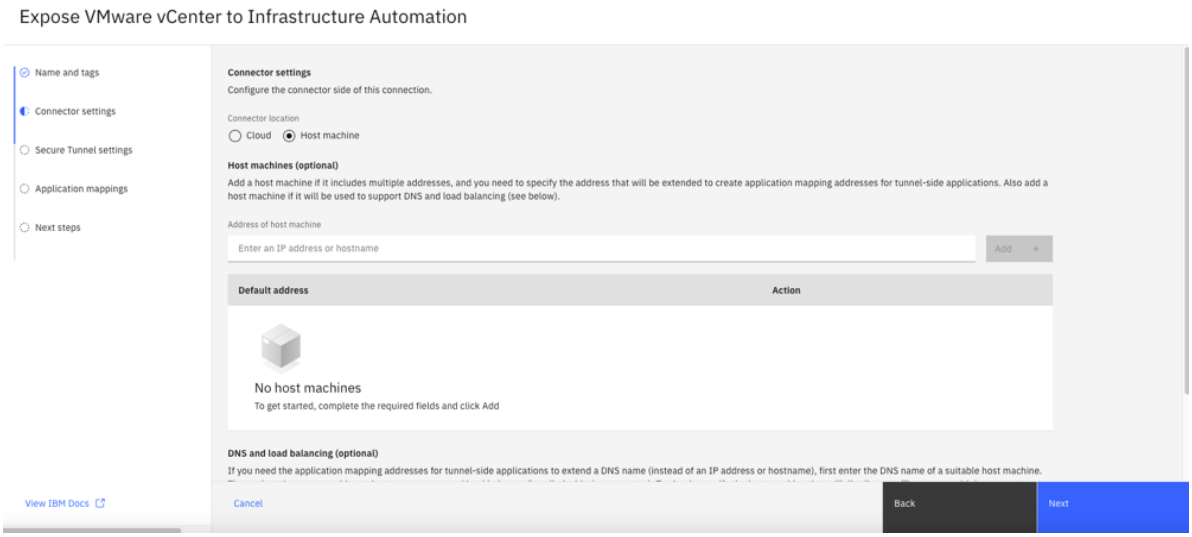


Figure. Connector settings for VMware

4. In the *Secure Tunnel setting*, configure the tunnel worker replicas to set High Availability.
5. *HTTPS proxy*: optional. Specify a forward proxy to enable the Secure Tunnel to reach the connector in the other network, such as username:password@hostname:port.

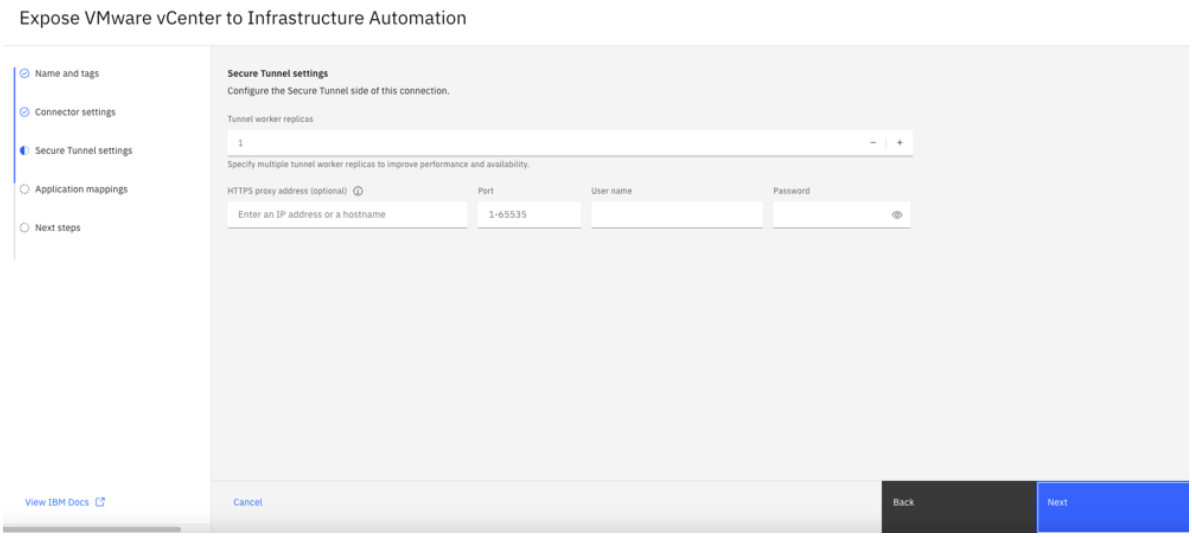


Figure. Tunnel settings for VMware

Note: It is necessary to enter the HTTPS proxy information when Secure Tunnel is in an air-gapped environment. If your Secure Tunnel is in an air-gapped environment without internet connectivity, you need to prepare a host machine (virtual machine or physical machine). Ensure that the host machine can be accessed from the IBM Cloud Pak for AIOps cluster, and can also connect to the internet. Then, install an HTTPS proxy server in the host machine, and set the HTTPS proxy server in the HTTPS proxy field.

If you install the connector in OCP, and Secure Tunnel in an air-gapped environment, you need use the Custom method to create a ChatOps connection

For example, squid is a simple HTTPS proxy server. See the following example of using squid to set up an HTTPS proxy server. (It is just an example. Don't use it in your production environment.) For the production environment, contact your network administrator to help you configure the HTTPS proxy.

1. Save the following configurations in a file squid.conf.

```
acl all src 0.0.0.0/0.0.0.0
acl SSL_ports port 443
acl CONNECT method CONNECT

http_access deny CONNECT !SSL_ports

http_access allow all

# Squid normally listens to port 3128
```

```
http_port 3128

# Uncomment and adjust the following to add a disk cache directory.
cache_dir ufs /var/spool/squid 100 16 256
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:         1440    20%    10080
refresh_pattern ^gopher:    1440    0%    1440
refresh_pattern -i (/cgi-bin/|\?) 0    0%    0
refresh_pattern (Release|Packages(.gz)*)$      0       20%     2880
# example lin deb packages
#refresh_pattern (\.deb|\.udeb)$   129600 100% 129600
refresh_pattern .         0    20%    4320
```

2. Start the squid server by running the following command:

```
podman run --name squid -d -p 3128:3128 -v `pwd`/squid.conf:/etc/squid/squid.conf datadog/squid
```

3. Set the `< host machine IP >:3128` in the *HTTPS proxy* field.

6. In the *Application mappings* section, edit the VMware addresses to match your environment. If it is the same as the address in the template, you don't need to change anything.

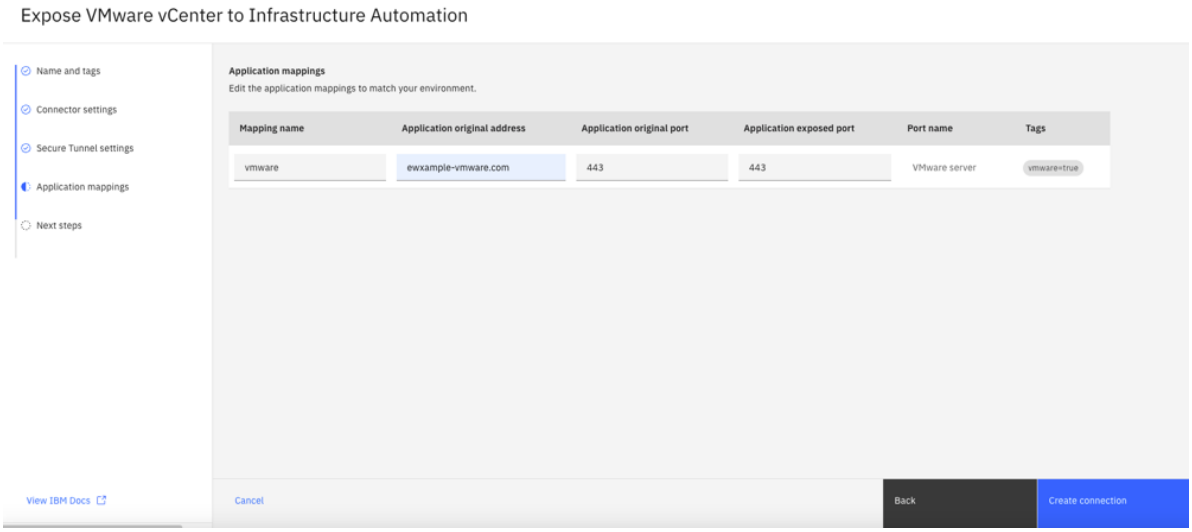


Figure. Application mappings for VMware

7. Click **Create connection** to create a VMware connection. You can get the installation script in the *Next steps* section. Install the Connector following the guide. When the connector is installed, the application mapping address is accessible. You can use this address as the VMware vCenter service address to configure the VMware connection from the Infrastructure Automation UI.

Turbonomic connection

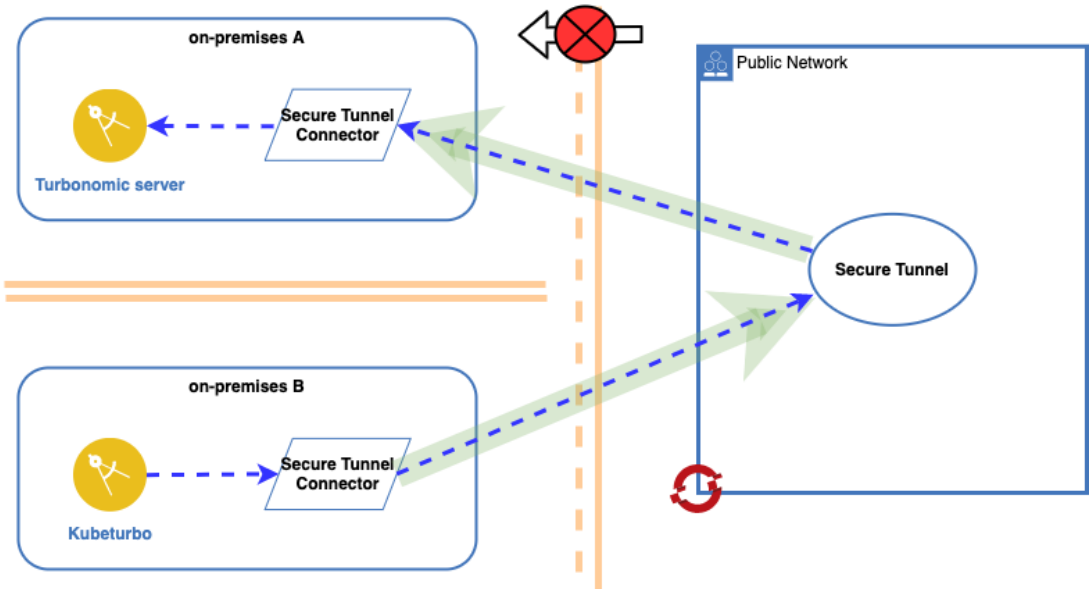


Figure. High-level design for Turbonomic

The Turbonomic server is in one private network cluster and you want to integrate it with a kubeturbo probe in another private network.

In this case, install a stand-alone Secure Tunnel on a public Cloud. Then, create one tunnel connection to export the **Turbonomic** server endpoint to the public cloud cluster. Next, create another tunnel connection to export the exposed **Turbonomic** server endpoint (a Kubernetes service in the public cloud cluster) to the **kubeturbo** private network cluster. The **kubeturbo** can now access the **Turbonomic** server through the Secure Tunnel.

Simplifying the creation of a secure tunnel by using the Turbonomic template:

1. Click the **Turbonomic** tile to open the Turbonomic side panel. Next, click **Get started** to create a connection. The first connection exposes Turbonomic to the Secure Tunnel server.
2. In the *Name and Tags* section, provide a name and tags for this connection:
   - *Connection name*: required. The name of the connection. The name can include only lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
   - *New tag*: optional. The tag of the Connection.
3. In the *Performance and availability* section:
   - *Connector replicas*: configure the tunnel connector replicas to High Availability
   - *Tunnel worker replicas*: configure the tunnel connector replicas to High Availability

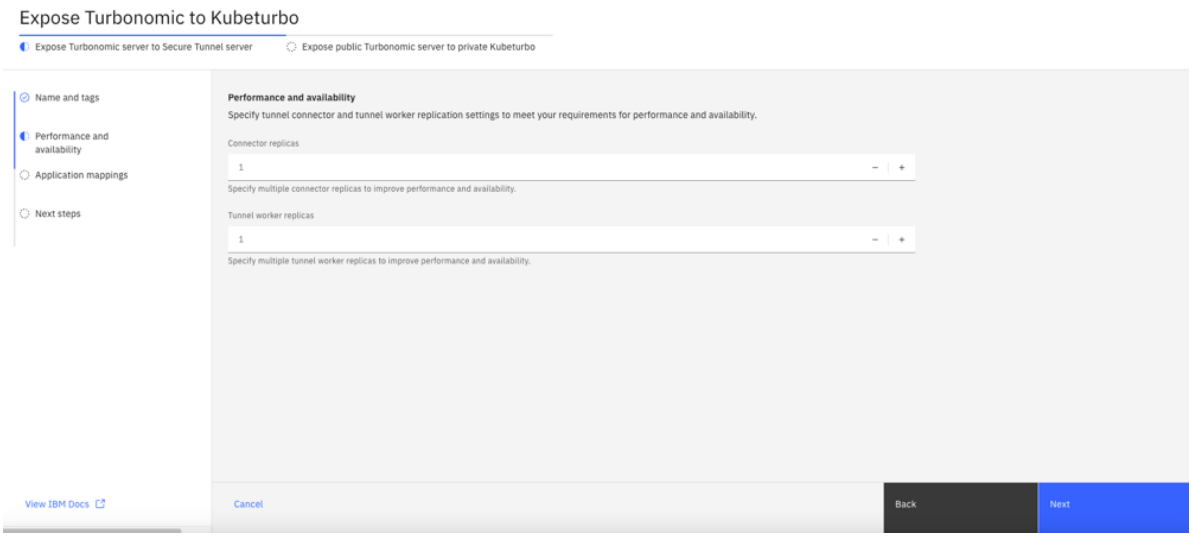


Figure. Performance and availability for Turbonomic

4. In the *Application mappings* section, edit the Turbonomic server addresses to match your environment.(**In particular, check the namespace part of the Turbonomic Kubernetes service address, and check whether it matches the Turbonomic Kubernetes service address**). If it is the same as the address in the template, you don't need to change anything.

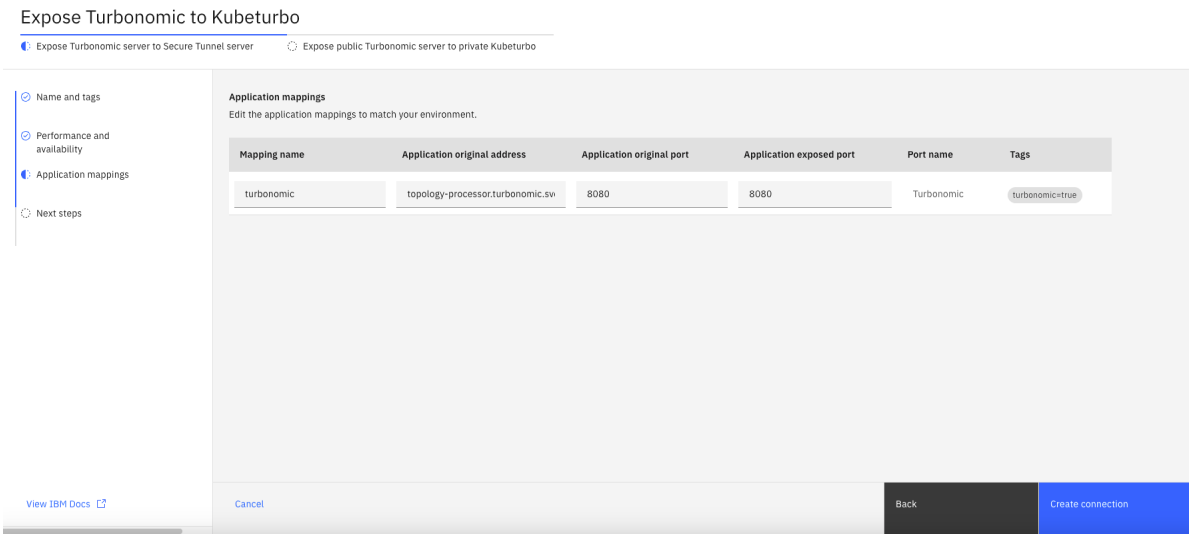


Figure. Application mappings for Turbonomic

5. Click **Create connection** to create a Turbonomic connection to expose the Turbonomic server to the public cloud cluster (as a Kubernetes network service), then you get the installation script in the *Next steps* section. You can install the connector by following this guide or install it later in the *Manage connections* page, by clicking **Action** > **installation connector**.

6. Click **Next**. Install the second connection. This step is the same as the first. The difference is that you cannot change the application address in the **Application mappings** section. Now create a tunnel connection to export the exposed **Turbonomic** server from the public cluster (a Kubernetes network service) to the **kubeturbo** private network cluster. Then, click **Create connection** to create a Turbonomic connection. You can get the installation script in the **Next steps** section, and install the connector.

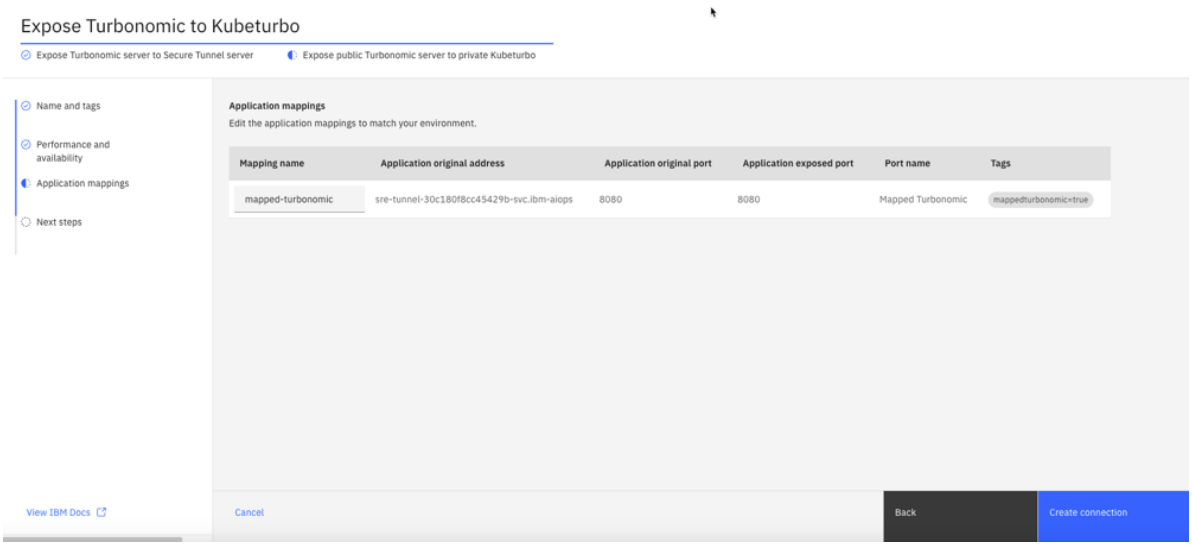


Figure. Application mappings for Turbonomic for the second connection

7. Finally, select the **<the name that you provided in the Name and Tags>-kubeturbo** tunnel connection and obtain the application mapping address. Then, you can use this application mapping address as the Turbonomic server address in the **kubeturbo**. The address is similar to this: **http://sre-tunnel-9a4f8fbb26374fa7.tunnel-connector-turbo:8080**.

8. Install the **kubeturbo**. Create a Project(namespace), like **kubeturbo**, to install the **kubeturbo** operator and instance from the Red Hat OpenShift console **OperatorHub**. Enter **kubeturbo** into the **All Items** field. Click the **Kubeturbo Operator** panel to install the **kubeturbo**. Navigate to **Installed Operators** > **Kubeturbo Operator**, click **Create instance** from the **Kubeturbo operator** panel. Change to the **YAML View** and edit the YAML like this:

```
spec:
  replicaCount: 1
  restAPIConfig:
    opsManagerPassword: <password of the Turbonomic server>
    opsManagerUserName: <username of the Turbonomic server>
  serverMeta:
    turboServer: '<the turbo server application mapping address, should be like: http://sre-tunnel-
9a4f8fbb26374fa7.tunnel-connector-turbo:8080>'
    version: '8'
  targetConfig:
    targetName: <give a name like: tunneltest>
```

Note: Check that the connectors have successfully installed for the two connections. One is in the private Turbonomic server cluster, the other is in the private kubeturbo agent container.

OpenStack connection

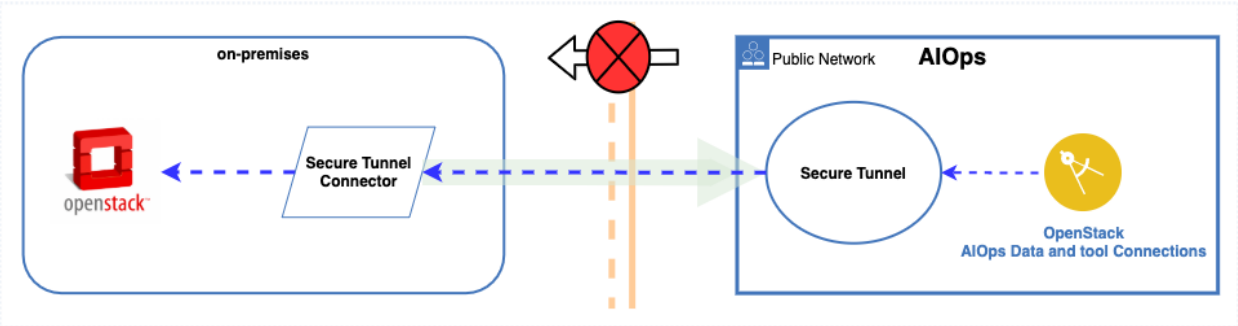


Figure. High-level design for Openstack

Scenario: The OpenStack server is in a private network cluster and IBM Cloud Pak for AIOps is in a public cluster. To integrate with the private OpenStack server, you can use a Secure Tunnel to create an OpenStack connection. The OpenStack template that is described in the following section supports this type of connection. You can create other templates through a custom connection. For more information, see Custom connection.

Simplifying the creation of a Secure Tunnel by using an OpenStack template:

From the **Secure Tunnel Console** on the **Manage connections** tab page, click *Create connection*. Then, take the following steps to create an OpenStack connection:

1. Click the **OpenStack** tile. The OpenStack introductory side panel opens. Click **Get started**.

2. In the *Name and Tags* section, provide a name and tags for this connection:

- *Connection name*: required. The name of the connection. The name can include lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
- *New tag*: optional. A tag for the connection.

3. In the *Connector settings* section, configure the tunnel connector replicas to set High Availability and choose to install the tunnel connector to *Cloud* or *Host machine* by selecting the *Connector location* section.
4. In the *Secure Tunnel setting*, configure the tunnel worker replicas to set High Availability.
5. In the *HTTPS proxy*, optionally specify a forward proxy to enable the Secure Tunnel to reach the connector in the other network, such as *username:password@hostname:port*.

   Note: It is necessary to enter the HTTPS proxy information when Secure Tunnel is in an air-gapped environment. If your Secure Tunnel is in an air-gapped environment without internet connectivity, you need to prepare a host machine (virtual machine or physical machine). Ensure that the host machine can be accessed from the IBM Cloud Pak for AIOps cluster, and can also connect to the internet. Then, install an HTTPS proxy server in the host machine and set the HTTPS proxy server in the HTTPS proxy field.

   For example, squid is a simple HTTPS proxy server. See the following example of using squid to set up a HTTPS proxy server. Do not use this example in your production environment. For the production environment, contact your network administrator to help you configure the HTTPS proxy.

   1. Save the following configurations in a file squid.conf.

```text {: codeblock}
acl all src 0.0.0.0/0.0.0.0
acl SSL_ports port 443
acl CONNECT method CONNECT

http_access deny CONNECT !SSL_ports

http_access allow all

# Squid normally listens to port 3128
http_port 3128

# Uncomment and adjust the following to add a disk cache directory.
cache_dir ufs /var/spool/squid 100 16 256
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:         1440    20%     10080
refresh_pattern ^gopher:      1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern (Release|Packages(.gz)*)$       0       20%     2880
# example lin deb packages
#refresh_pattern (\.deb|\.udeb)$   129600 100% 129600
refresh_pattern .         0     20%     4320
```

   2. Start the squid server by running the following command:

```sh {: codeblock}
podman run --name squid -d -p 3128:3128 -v `pwd`/squid.conf:/etc/squid/squid.conf datadog/squid
```

   3. Set the :3128 in the HTTPS proxy field.
6. In the *Application mappings* section, edit the OpenStack addresses to match your environment, and enable *Origin application uses HTTPS*.

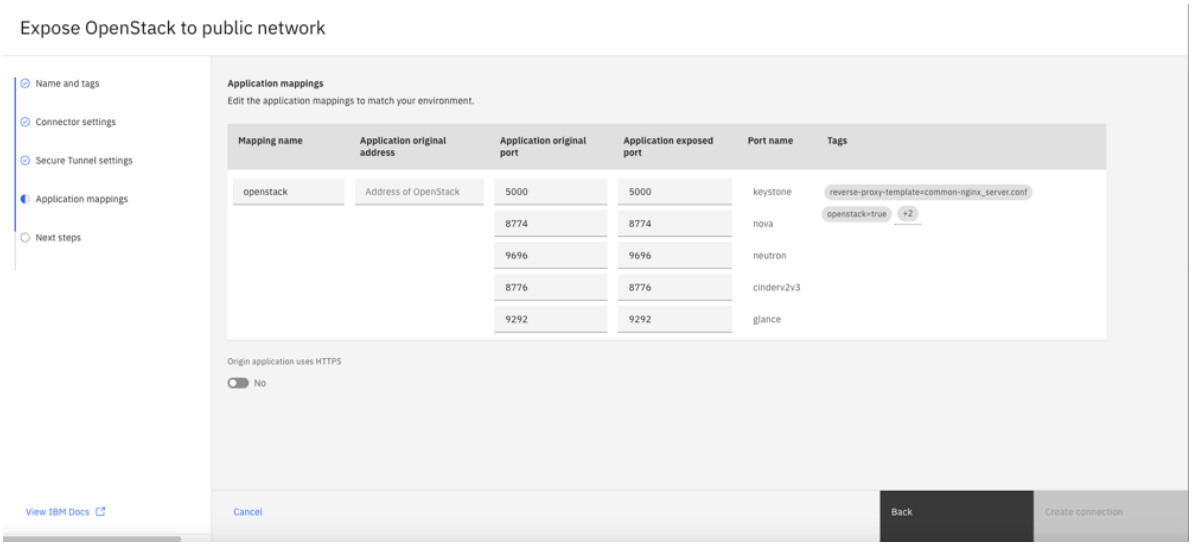


Figure. Application mappings for Openstack

7. Click **Create connection** to create an OpenStack connection. You can get the installation script in the **Next steps** section. Follow the installing connector steps. After installing the connector, the application mapping address is accessible. Use this address as the VMware vCenter service address to configure the VMware connection from the Infrastructure Automation UI.

Custom connection

Secure Tunnel can accommodate specific requirements with a Custom connection. For instance, you can use this type of connection for ChatOps to help you understand how to customize your own connection.

The installation connector in Red Hat OpenShift is automatically supported when you use the ChatOps template. If you do not prefer this approach, you can install the connector in Red Hat OpenShift manually with the Custom method.

Use case summary: IBM Cloud Pak for AIOps is installed. Slack (for ChatOps) is in a public network. To access IBM Cloud Pak for AIOps, expose the IBM Cloud Pak for AIOps instance to the public cluster through a connector that is installed in Red Hat OpenShift.

1. Create a **connection** to create a bridge configuration for the network where **Secure Tunnel** is installed and the network that has the connector.
    1. Click **Custom** tile. This opens the Custom side panel. Then, click **Get started**.
    2. In the *Name and Tags* section, Provide a name and tags for this connection:
        - Connection name: required. The name of the connection. The name can include lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
        - New tag: optional. The tag of the Connection. Add the tag **`chatops=true`** in our example.
    3. In *Direction*, specify whether the relationship is one of below:
        - *connector in private network initializes connection to Secure Tunnel in public network*
        - *Secure Tunnel in private network initializes connection to connector in public network*.

       Because Secure Tunnel is in the IBM Cloud Pak for AIOps cluster, choose option **Secure Tunnel in private network initializes connection to connector in public network**.

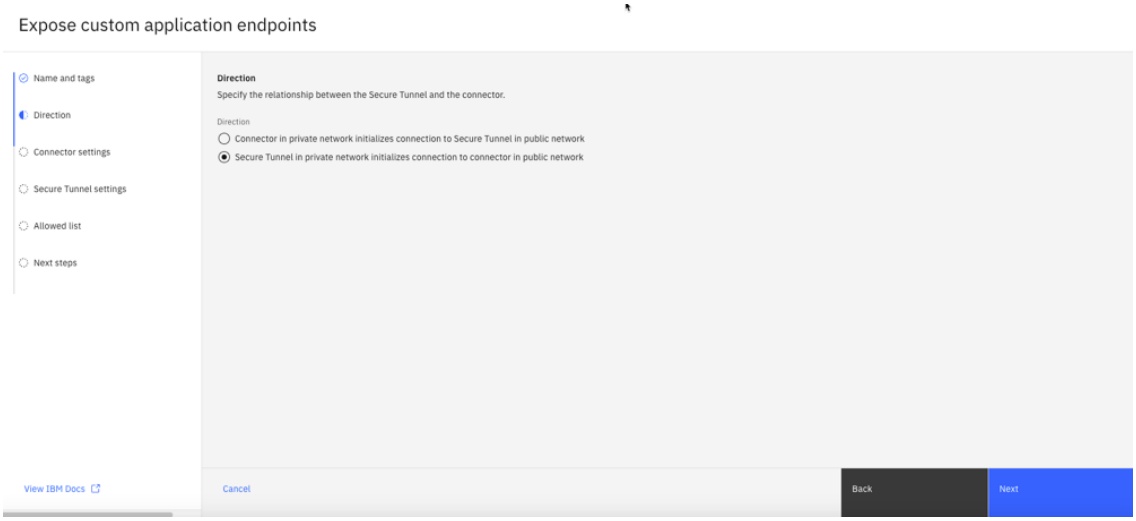


Figure. Direction for custom connection

    4. In the **Connector settings** section:
        1. *Connector location*: required. Select whether installation connector is in the *Cloud* or in a *Host machine*. Because the connector is installed on Red Hat OpenShift, choose **Cloud**.
        2. *Connector replicas*: Configure the tunnel connector replicas to set High Availability, 1 - 16, as you need.
        3. *Cloud domain*: If you install the connector in the Cloud direction, the Secure Tunnel in the private network initializes the connection to a connector in a public network. You need to input the cloud domain of the cluster where you installed the connection. Run this command to get the Red Hat OpenShift cluster domain name:

           ```
           oc get route -n openshift-console console -o jsonpath={.spec.host} | sed -e 's/[^\.]*\.//
           ```

           Run this command to get the IBM Kubernetes Service cluster domain name:

           ```
           ibmcloud ks cluster get --cluster <Cluster ID> --output json | jq '.ingress.hostname'`
           ```

        4. Host Machine: if you install the connector in a VM or Physical machine, you need:
            1. Host machine addresses
                1. If your Connection direction is *Secure Tunnel in private network initializes connection to connector in public network*, it must include the *External address of host machine* and *Port* and *Default address of host machine* .(Optionally, specify a default IP address when a host machine includes multiple addresses).
                2. If your connection direction is *Connector in private network initializes connection to Secure Tunnel in public network* it must include *Default address of host machine* (optionally, specify a default IP address when a host machine includes multiple addresses.)
                3. *Common DNS name for all inbound traffic (optional)*: required, in this case. Specify a DNS name for all inbound (connector to tunnel) traffic. When multiple host machines are present, you can enter a DNS name or the external IP address or hostname of

the machine hosting the reverse proxy/load balancer. **Note:** do not complete this field unless you have at least one host machine address.

4. *Host machine for reverse proxy / load balancer (optional)*: required, in this case. All inbound (connector to tunnel) traffic is routed to an NGINX-based reverse proxy/load balancer deployed to this host machine during installation of the connector.

5. Connector replicas: configure the tunnel connector replicas to set High Availability.

5. In the *Secure Tunnel setting*, configure the tunnel worker replicas to set High Availability, limit: 1-16.

6. *HTTPS proxy*: optional. Specify a forward proxy to enable the Secure Tunnel to reach the connector in the other network, such as username:password@hostname:port.

   Note: It is necessary to enter the HTTPS proxy information when Secure Tunnel is in an air-gapped environment. If your Secure Tunnel is in an air-gapped environment without internet connectivity, you need to prepare a host machine (virtual machine or physical machine), and ensure that the host machine can be accessed from the IBM Cloud Pak for AIOps cluster and also can connect to the internet. Then, install an HTTPS proxy server in the host machine, and set the HTTPS proxy server in the HTTPS proxy field.

7. *Allowed list*: Specify the IP addresses and ports that are exposed across this connection through application mappings. You can specify address ranges and port ranges. If you do not define an allowed list, the application mapping URL is inaccessible. To expose the IBM Cloud Pak for AIOps instance, input IBM Cloud Pak for AIOps instance address: **`internal-nginx-svc 12443`**

8. Click **Create connection** to initiate a custom connection. In the *Next steps* page, you install a connector and create application mappings.

2. Follow the Install Secure Tunnel Connector guide to enable the connection.

3. Create *Application Mapping* to expose the applications to another network (Secure Tunnel side to connector side, or opposite). Set application mappings to expose the IBM Cloud Pak for AIOps instance. Go to the **Secure Tunnel Console** on the **Manage connections** tab page, and click the **Application mappings** of the *tunnel connection* list. You are redirected to the *Application mappings* page. The *Application mappings* page can resemble the following screenshot:

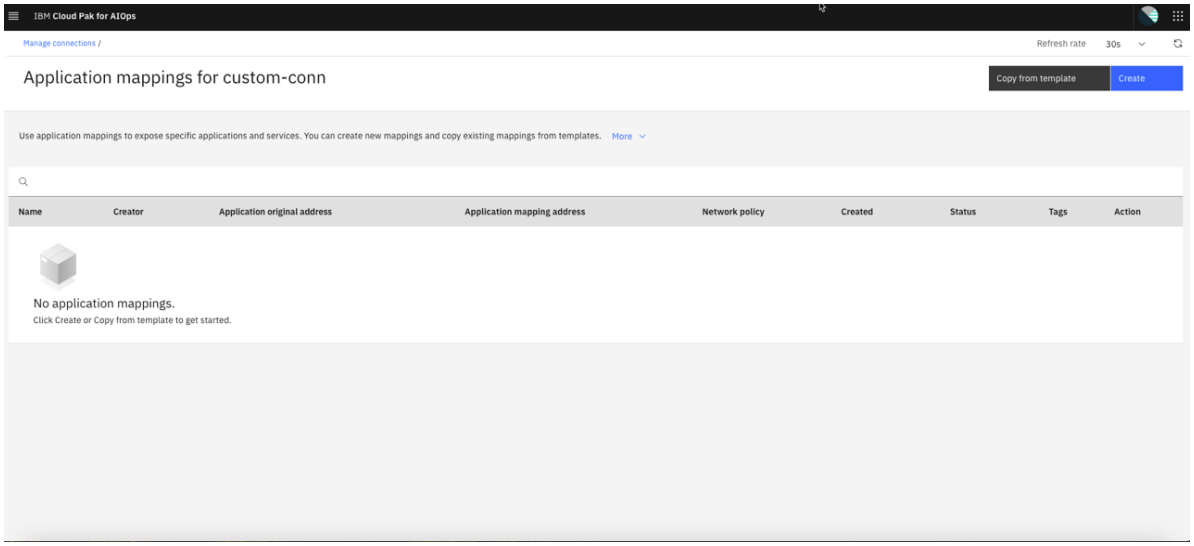


Figure. Create Application Mappings

You have two ways to create a connection:

- *Create a connection from scratch*
- *Create a connection from existing template*.

1. Create a connection from scratch:

   1. Click **Create** to open the panel to create an application mapping from scratch.
      1. In the *Name and Tags* section, provide a name and tags for this application mapping:

         1. Application mapping name: required. The name of the application mapping name can include lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
            1. New tag: optional. The tag of the application. Add a tag **`chatops=true`** in our example.
      2. In *Direction*, specify how you expose the application *from connector side to Secure Tunnel side* or *from Secure Tunnel side to connector side*. Because the IBM Cloud Pak for AIOps instance is in Secure Tunnel side, in this case, choose *from Secure Tunnel side to connector side*.

      3. In the *Application details* section,

         1. Specify the address of the application and the ports that you want to expose. The address and ports must be included in the allowed list for the connection. Configure DNS so that the application can be accessed by using the original address or the application mapping address (optional) . Input the IBM Cloud Pak for AIOps instance address: **`internal-nginx-svc`**, and port: **`12443`**, and the exposed mapping port (default is **`12443`**. Change as you need.). **Note:** Before you add an IP address or hostname, check the existing application mappings from the *Application mappings* page in the IBM Cloud Pak for AIOps console. Check for an existing application mapping. If one is available, check whether the original address is the same IP address or hostname as the new one that you want to specify. Edit the existing mapping to add the new port value instead of creating another application mapping.

         2. *DNS*: Configure DNS so that application can be accessed by using the original address or application mapping address.

3. If you choose to expose the application *from the connector side to the Secure Tunnel side* in the *Direction* section, you can optionally configure the **Reverse proxy**. Toggle each switch to *Yes* to enable the following:

    - *Origin application uses HTTP/HTTPS, and server will check hostname or respond to other URLs of application through HTTP response payload* (for example, Falcon LogScale)
    - *Origin application uses HTTPS* (for example, OpenStack)
    - *Origin application responds to other URLs of application through HTTP response payload* (for example, OpenStack)

4. In *Security*, you can create a Secure Tunnel generic webhook connection. If this is not required, do not switch it on.

5. In *Access rule*, you can use access rules to control the traffic flow at the IP address or port level for particular applications in your cluster. The default network policy allows all ingress and egress traffic for each application mapping. *Network policy*: Network policies help you specify how a pod is allowed to communicate with various network "entities" over the network. Network policies apply to a connection with a pod on one or both ends, and are not relevant to other connections.

2. Create a connection from existing template:

    1. Click **Copy from template** button, and the new page opens.

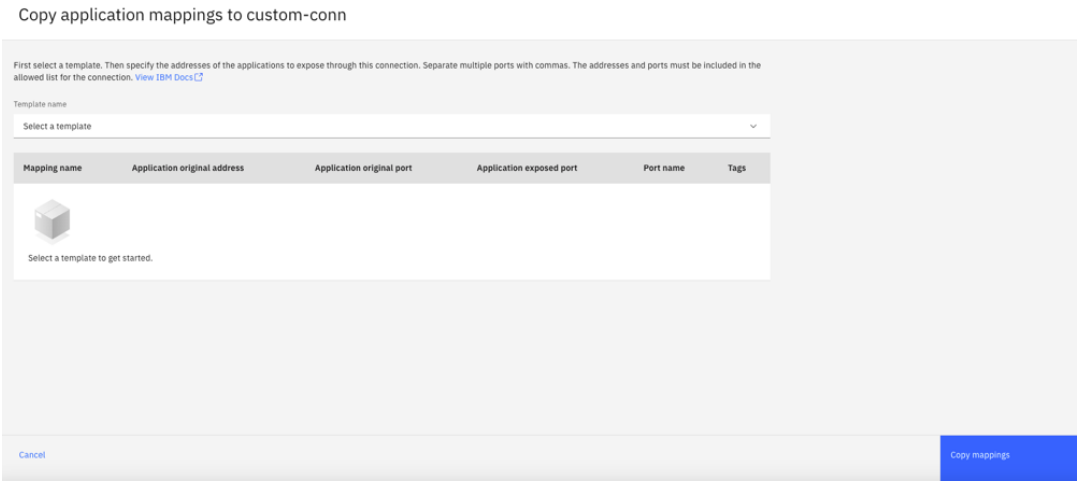


Figure. Copy Application Mappings

    2. Choose a **template**. You can see the application mappings in this template is displayed in the table.

    3. In the table, you can edit the *Application Mapping name*, *Application original address*, *Application original Port* and *Application exposed port*

    4. Click **Copy mappings** to create application mappings through this connection.

4. Now you have successfully exposed the application as you need. In our example, IBM Cloud Pak for AIOps is exposed to the public network, so you can use the application mapping address to access IBM Cloud Pak for AIOps.

Creating a Custom Instana connection

1. Ensure that your cluster has the following prerequisites installed and configured:

    - An Instana server is installed and runs in an Red Hat OpenShift Container Platform cluster.
    - An IBM Cloud Pak for AIOps cluster is installed and runs in a public cloud with Secure Tunnel enabled.

2. Create the secure tunnel connection for Instana

    1. From the IBM Cloud Pak for AIOps console main navigation menu, click **Administration** > **Secure Tunnel**.
    2. In the *Secure Tunnel* page, click **Create connection**.
    3. Click the **Create connection** link under *Custom* section and then click **Get started**.
    4. On the *Name and tags* page, input a name in *Connection name* field, for example `instana-via-tunnel`, click **Next**.
    5. Choose **Connector in private network initializes connection to Secure Tunnel in public network** in *Direction*, click **Next**.
    6. On *Connector settings* page, choose **Host machine**. Input the hostname or IP address of the host machine where you installed the Secure Tunnel connector. Click **Add** > **Next**.
    7. On *Secure Tunnel settings* page, click **Next**.
    8. On *Allowed list* page, input the IP address and port of Instana server, for example `9.46.108.75` and `443`, click **Add** button and click **Create connection**.
    9. Wait for the *Next steps* page display, click **Close**.
    10. Download the Secure Tunnel Connector Installation package by using one of the following options:
        - If you are still in the `Next steps` page, you can download the installation package by clicking download link under *Install a connector*
        - If you are not in the `Next steps` page yet, navigate to *Manage Connection* page, find the `to-instana-through-ocp` row, and click the three dots in the **Actions** column, then click **Install connector** link. You can now download the installation package by clicking the download link in the open panel.

3. Install Secure Tunnel Connector to a host machine

    1. refer to Install the Secure Tunnel Connector to a host machine (virtual machine or physical machine)
    2. On the *Secure Tunnel* page, the *Status* column of `instana-via-tunnel` connection is green.

4. Create Application Mapping to expose Instana:

    1. From the Cloud Pak for AIOps console navigation menu, click **Administration** > **Secure Tunnel**.

2. On the *Secure Tunnel* page, click the **Application mapping** column on the row `instana-via-tunnel`.
3. On the *Application mappings for instana-via-tunnel* page, click **Create** button.
4. On the *Create application mapping* page, click **Next**.
5. On the *Name and tags* page, input a name, for example `instana` then click **Next**.
6. On *Direction* page, choose **From connector side to Secure Tunnel side** and click **Next**.
7. On *Application details* page, input Instana server IP address and port, for example `9.46.108.75` and `443` and click **Add** button.
8. On *Access rules (optional)* page click **Create**.
9. On the *Application mappings for instana-via-tunnel* page, wait for a few minutes, then the `Status` column of `instana` row become `Ready`.
10. In the *Application mapping address* column, click the **information button(a circle with an `i` in it)**, and copy the address in the open message. For example, `sre-tunnel-c60f3456bad94a10-svc.cp4waiops:443`. Save it to a note pad.

5. Create an Instana connection

1. From the Cloud Pak for AIOps console navigation menu, click **Define** > **Integrations**.
2. On the Integrations, click **Add integration** button.

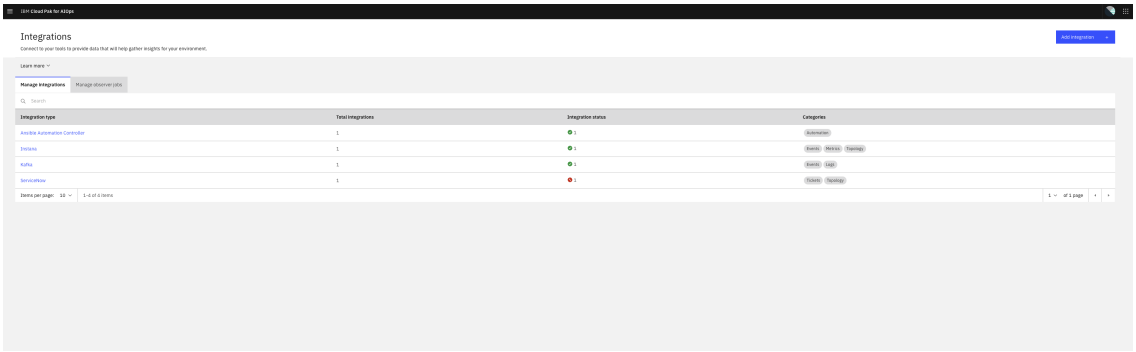


Figure. Add integration

3. On the *Add connections* page, search for *Instana*, click the *Instana* tile, and then click **Connect** button.
4. On the Instana *Add integration* page, add the following:
    1. *Name*, for example `tunnel-2-instana-public`
    2. *Description*, for example `Connect to an Instant in public cloud through the Secure Tunnel`
    3. *Endpoint*, by using the `Application mapping address` copied from Secure Tunnel Application Mapping, for example `https://sre-tunnel-c60f3456bad94a10-svc.cp4waiops:443`.
    4. *API Token*, by using the API Token find from your Instana server, for example `W6p9WimUT5aYQ_skNu0W9A`
    5. *Deployment options*, Local deployment or remote deployment.
        - Local: Deploy the integration in the same cluster and namespace where Cloud Pak for AIOps is installed.
        - Remote: Deploy the integration anywhere you choose, for example, a different network region, on SaaS, or remote on-premises (VM, cluster, container etc).
    6. Click **Next**
5. On the *Configure* proxy page, enable or disable the proxy.
6. On the *Collect topology data* page, click **Next**.
7. On the *Collect event data* page, click **Next**.
8. On the *Collect metric data* page, click **Done**. You are redirected to the *Instana* connection page.

    Your new connection is listed in the table on the page. The connection has an intial status of *Initializing*. After a few minutes, the status changes to *Running* if your connection is successful.

6. Verify the Instana Connection

1. From the Cloud Pak for AIOps console navigation menu, click **Operate** > **AI model management**.
2. On the *AI model management* page, click the **Data assets** tab.
3. The **Data flow status** column for the `tunnel-2-instana-public` connection should display a status of *Running*.

Creating a Secure Tunnel Generic Webhook connection

To integrate IBM Cloud Pak for AIOps **Generic Webhook** with a service like New Relic, the IBM Cloud Pak for AIOps instance must be accesible from the public network. If your IBM Cloud Pak for AIOps is not accessible from the public network, you can open the IBM Cloud Pak for AIOps **Generic Webhook** to the public network by creating a Secure Tunnel. You can do this by exposing the IBM Cloud Pak for AIOps **Generic Webhook** to either a cluster (Red Hat OpenShift Container Platform or IBM Kubernetes Service cluster) or a host machine (a virtual machine or a physical machine), which is accessible from the public network. You can create the **Generic Webhook** Secure Tunnel from the **Secure Tunnel Console** on the **Manage connections** page.

To create a Secure Tunnel generic Webhook connection, complete the following steps:

1. Create the generic webhook connection with the IBM Cloud Pak for AIOps console. For more information, see Creating Generic Webhook connections.

    As you are creating the connection, copy the *Webhook route*. The Webhook route can resemble the following example route:

    `https://whconn-f78d2a04-c881-40c1-a8cc-f329a297b7f6-cp4waiops.apps.katui.cp.fyre.ibm.com/webhook-connector/h35qpu67hjh`

    The part `f78d2a04-c881-40c1-a8cc-f329a297b7f6` is the *Webhook uuid* and the part `/webhook-connector/h35qpu67hjh` is the *Webhook route path*.

2. Log in to the IBM Cloud Pak for AIOps cluster and then execute below command to get the **certificate CA** of the *Generic Webhook* service.

   ```
   oc -n <the namespace of the IBM Cloud Pak for AIOps> get ConfigMap ibm-grpc-webhook-connector-ca-cert-<Webhook uuid> -o=jsonpath='{.data.service-ca\.crt}'
   ```

   For example execute *oc -n aiops get ConfigMap ibm-grpc-webhook-connector-ca-cert-f78d2a04-c881-40c1-a8cc-f329a297b7f6 -o=jsonpath='{.data.service-ca.crt}'*

3. Follow the Custom connection instructions to create a **`Secure Tunnel in private network initializes connection to connector in public network`** direction Tunnel Connection and install the Secure Tunnel Connector.

   Note:

   - In the *Direction* section, select the **`Secure Tunnel in private network initializes connection to connector in public network`** radio button.
   - In the *Allowed list* section, add **`webhook-connector-<Webhook uuid>.<the namespace of the IBM Cloud Pak for AIOps>.svc.cluster.local 443`** (for example, **`webhook-connector-f78d2a04-c881-40c1-a8cc-f329a297b7f6.aiops.svc.cluster.local 443`**) to the *Allowed list*.
   - If you want to install the connector in the Cloud, specify a publicly accessible cluster with a CA-signed domain certificate.
   - If you want to install the connector on the Host machine, ensure you have a signed trusted certificate for this Host machine.

4. Create an **Application Mapping** to expose the IBM Cloud Pak for AIOps **Generic Webhook** service to the public network.

   1. Go to the Secure Tunnel Console on the Manage connections tab page, and click the **Application mappings(valid/total)** of the tunnel connection list. You are redirected to the **Application mappings** page.
   2. Click **Create** to open the panel to create an application mapping from scratch.
      1. In the *Name and Tags* section, provide a name and tags for this application mapping:
         - *Application mapping name:* required. The name of the application mapping name can include lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
         - *New tag:* optional.
      2. In the *Direction* section, select **From Secure Tunnel side to connector side**.
      3. In the *Application details* section, add the endpoint that is in the allowed list, (for example **`webhook-connector-f78d2a04-c881-40c1-a8cc-f329a297b7f6.aiops.svc.cluster.local`**) to the **IP address or hostname** and **`443`** to the **Port**. You can update the **Exposed mapping port** to change any no-conflict valid port that you want.
   3. Click the **Add** button.
      1. In the *Security* section, choose **Yes** for the **Enable security for the exported application which is HTTP protocol.**.
      2. *Select termination type:* select *Reencrypt*.
         - If you want to install the connector in the Cloud, **Destination CA certificate** the **certificate CA** of the **Generic Webhook** service.
         - If you want to install the connector in the Host machine:
            - *Certificate:* the **`cert`** of the signed trusted certificate of the Host machine.
            - *Private key:* the **`key`** of the signed trusted certificate of the Host machine.
            - *Destination CA certificate:* the certificate CA of the Generic Webhook service.

5. Get the exposed **Webhook** URL to integrate with a service like **New Relic**. From the Secure Tunnel *Application Mapping* list UI page, get the **Application mapping address** for example **`https://secure-tunnel-connector.fyre.ibm.com:443`**.

   You can then integrate the **Generic Webhook** with a service like **New Relic** with the URL **`<Application mapping address><Webhook route path>`** (for example **`https://secure-tunnel-connector.fyre.ibm.com:443/webhook-connector/h35qpu67hjh`**).

Viewing Secure Tunnel

With the Secure Tunnel tool in the IBM Cloud Pak for AIOps console, you can get the application mapping URL that can be accessed by your integration. In addition, you can view the configuration details and status of all tunnel connections and application mappings. If you have not accessed the Secure Tunnel web console previously, read this.

- Secure Tunnel Components
- Viewing tunnel connection list
- Viewing the application mappings
- Certificate expiration

Secure Tunnel Components

There are four components on the Secure Tunnel page in the console that you can use to manage your connections. To access these, go to **Administration** > **Secure Tunnel**.

- **Manage connections**

  Use **Manage connections** to manage your connections to remote networks, A connection links the network containing the Secure Tunnel to a remote network. Your applications can reside on either side of the connection. Secure Tunnel includes connection templates for common scenarios(ChatOps, VMware, Turbonomic). You can also create custom connections to support additional scenarios.

- **Manage templates**

  Use **Manage templates** to manage your templates. This tool can help you create a new connection quickly. Secure Tunnel has pre-defined templates for common scenarios, but users can also create templates for their special requirements. Find out more about using templates

- **Audit operations**

Use **Audit operations** to check Secure Tunnel configuration changes, such as tunnel connection or application mapping configuration changes for auditing. Find out more about auditing Secure Tunnel

- **Audit traffic**

  Use **Audit traffic** to check open/close connection events traffic and see the total amount of data that is sent or recieved from the specific connection listed in the close event. Find out more about auditing Secure Tunnel

Viewing tunnel connection list

Go to **Administration** > **Secure Tunnel**. In the *Manage connections* page, you can see the tunnel connection list with the following columns:

- **Name**: Name of the tunnel connection. Click the name of the tunnel connection in the *Manage connections* page and you are redirected to the *Application mappings* page and the *Connection details* page.
- **Connectors**: The number of connections. This indicates how many Connectors connect to the tunnel worker.
- **Application mapping(valid/total)**: The number of valid application mappings and the total number of application mappings that are passing this connection. You can click the entry in this column to go to the application mappings page for more details. See Viewing the application mappings.
- **Direction**: The relationship between the Secure Tunnel and the Connector. A Connector in a private network initializes a connection to a Secure Tunnel in a public network or a Secure Tunnel in a private network initializes a connection to a Connector in a public network
- **Allowed list**: The IP addresses and ports that are exposed across this connection by using application mappings.
- **Created**: The date and time when the Connector was created.
- **Certificate expiration**: Expiration time of certificates.
- **Status**: Indicates whether the tunnel connection is accessible. Three kinds of status are provided:
  - `Pending`: This means that the tunnel connection is creating the tunnel worker. When the connection in in `Pending` status, you can't install the Connector for the connection.
  - `Waiting`: This means that tunnel worker is ready, but no Connector is connecting to it. You need to install a Connector and create application mapping for this connection.
  - `Ready`: This means that the connection is ready.
  - `Error`: This means that there are some errors with the tunnel connection. Hover over the status to see the detailed error message.
  - `Incomplete`: This means that the application mapping for the connection is invalid.
- **Tags**: Used to signify the tunnel connection.

Viewing the application mappings

Go to **Administration** > **Secure Tunnel**. This opens the *Manage connections* page. Click the number in the **Application mappings (valid/total)** column. You are redirected to the *Application mappings* page. This provides details of all the application mappings of a connection.

- **Name**: Name of the application mapping.
- **Creator**: User that creates the application mapping.
- **Application original address**: The private address of the application that you want to expose.
- **Application mapping address**: The application mapping URL. You can access the private application by this mapping address.
- **Network policy**: To control the traffic flow at the IP address or port level for this application.
- **Created**: Time when the application mapping was created.
- **Status**: Indicates whether the application mapping is accessible. Three kinds of status are provided:
  - `Error`: This means that there are some errors with the application mapping. You can move the mouse up to see the detailed error message.
  - `Incomplete`: If the application mapping status is `Incomplete`, hover your mouse pointer over the *Status* column to discover the problem. It could be that you did not install the Connector or that Connector is still waiting to connect. If a port conflict is indicated, uninstall the Tunnel connector, resolve the port conflict, and then re-install the Tunnel connector.
  - `Ready`: This means that the application mapping address is accessible.
- **Tags**: Tags of the application mapping.

Certificate expiration

When the certificate for the Secure Tunnel connection expires, the IBM Certificate Manager renews it. The Secure Tunnel worker server automatically reloads the certificate, but the Secure Tunnel connector does not. You must download a new Secure Tunnel connector installation script, and then uninstall and reinstall the Secure Tunnel connector. For more information, see the instructions in Install the Secure Tunnel Connector.

Modifying Secure Tunnel

In the Secure Tunnel web console you can modify the configurations for a tunnel connection and application mapping to meet your needs. (Read this if you have not used Secure Tunnel web console previously).

- Configuring tunnel connections
- Managing Template
- View Template
- Delete Template
- How to obtain the application mapping address

Configuring tunnel connections

- **Configuring tunnel connections**
  - **Edit connection**: To modify the connection, locate the connection, and click **Action** > **Edit** connection.
  - **Renew certification**: To renew the certificate of the tunnel connection, locate the connection, and click **Action > Renew** certificate. Then, the certificate of the tunnel connection is updated, and the certificate expiration time field is changed.
  - **Delete connection**: To delete a tunnel connection, locate the connection, and click **Action > Delete connection**.
- **Configuring application mapping**
  - **Edit application mapping**: To edit the application mapping, locate the application mapping, click **Action > Edit**.
  - **Delete application mapping**:To delete the application mapping, locate the application mapping, and click **Action > Delete**.
- **Installing Secure Tunnel connector**
  - Follow this guide

Managing Connection Templates

A template includes pre-defined connection settings and one or more application mappings. Examples of pre-defined templates available in IBM Cloud Pak for AIOps are ChatOps and Vmware, which you can access in **Manage connections** > **Create connection**.

You can use a template to quickly create a new connection, or to add application mappings to an existing connection. You can easily modify the application mappings to meet the unique requirements of the connection.

Create Template

Click **Create template** button to open the page:

1. Provide a name and a description for this template.
2. Select the connection upon which the template is based.
3. Select the application mappings for the template.

Once you create a template, you can use it to simplify the creation of future connections.

View Template

Click **Action** in the template column. Then, click **View details** to view the template details.

Delete Template

Click **Action** on the template column. Then, click **Delete** to delete the template.

How to obtain the Application URL mapping address

1. From the Secure Tunnel console, select **Manage connections** tab.
2. Click the **Application mappings** item of the tunnel connection.
3. Choose the **Application mapping** by name from the **Application mapping** list page.
4. Then, copy the address from the **Application mapping address** item.

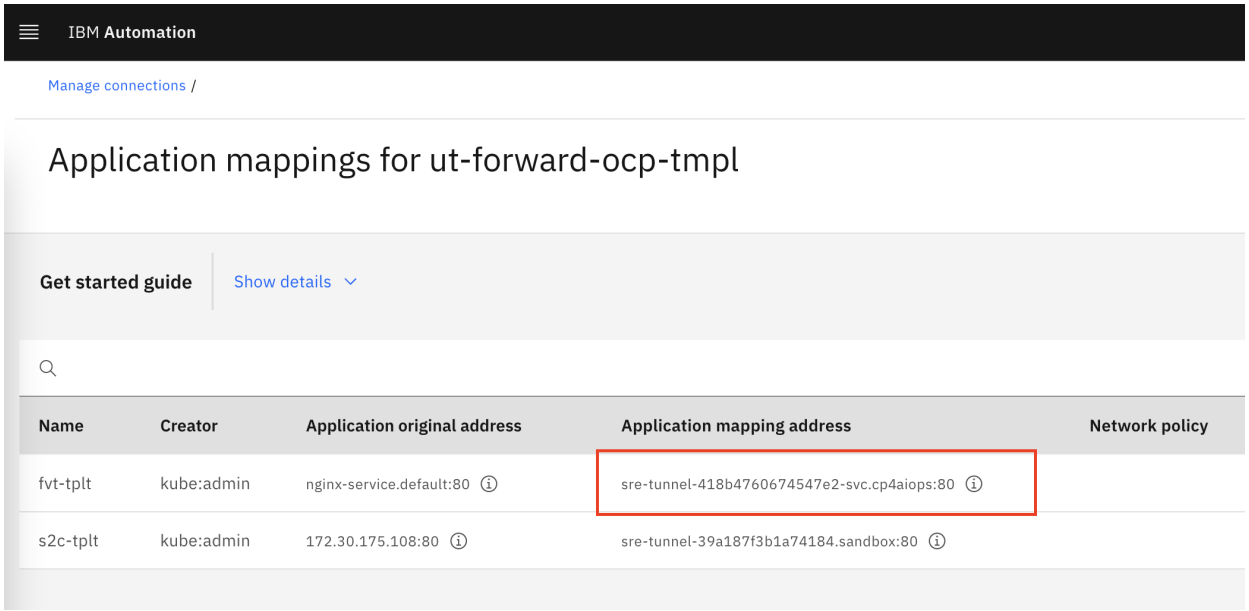


Figure. Application mapping

Auditing Secure Tunnel

You can check Secure Tunnel configuration changes, such as tunnel connection or application mapping configuration changes for auditing.

How to audit Secure Tunnel

Go to **Administration** > **Secure Tunnel** on IBM Cloud Pak for AIOps console. You see the **Audit operations** and **Audit traffic** tabs.

1. In the *Audit operations* tab, you can see the operation logs, such as deleting a tunnel connection or creating a tunnel connection. You can filter the logs by changing the start time and end time in the **Start time** and **End time** fields.

   Note: You can click the **Operation details** column to see more about the tunnel operation.

2. In the *Audit traffic* tab, the tunnel **`open/close`** connection events are listed, and a total **`send/receive`** data amount for the specific connection is listed in the **`close`** event. You can filter the logs by changing the start time and end time in the **Start time** and **End time** fields.

Configuring Secure Tunnel audit logs

Secure Tunnel can send audit logs to its container log, a MongoDB database, and a remote syslog server. By default, Secure Tunnel sends audit logs to its container log, and thus you can't see audit logs on the Secure Tunnel console. To check audit logs on the Secure Tunnel console, you need to send audit logs to MongoDB. To check audit logs on the query user interface of a specific syslog, you need to send audit logs to the remote syslog server.

Configuring with the operator UI when creating a Secure Tunnel installation

1. When you create the Secure Tunnel instance from your OpenShift console, click **Operators** > **Installed Operators**.
2. From the ***Project** menu, select the project that you want to create the Secure Tunnel instance in. Use the project that you created earlier in **Create a Secure Tunnel project (namespace)**.
3. Select **IBM Secure Tunnel** operator.
4. Click **Create Installation** on the *Tunnel* box. The default **Form View** is displayed. Edit the **Audit logs** to configure the audit log receivers of the Secure Tunnel:
5. *Show audit logs in the container log:* If the Secure Tunnel should send audit logs to its container log(the pod terminal).
6. *Audit Receiver:* to add a **MongoDB audit log receiver** or a **Syslog audit log receiver**

   1. Click the **MongoDB receiver** > **Add MongoDB receiver** button. To add a **MongoDB audit log receiver** > click **MongoDB Information**. Enter the following values:

      1. *Receiver name:* The name of the MongoDB receiver, it's the unique name of the audit log receiver. for example: **`mongodb-receiver-1`**.
      2. *TLS Secret:* if you want the MongoDB to support TLS, you should create a TLS Kubernetes secret for the MongoDB TLS certification, and enter the *k8s secret name* to here. the Kubernetes secret should be in the Secure Tunnel namespace and contain the "ca.crt", "tls.crt" and "tls.key" of the mongoDB TLS. TLS secret example:


```
kind: Secret
apiVersion: v1
metadata:
name: mongodb-tls-certificate
namespace: tunnel-stand-alone
data:
ca.crt:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURDVENDQWZHZ0F3SUJBZ0lRTjRtVDZOTFgxZjF1K0pSVFR4LzRoakFOQmdrcWhraU
c5dzBCQVFzRkFEQWcKTVI0d0hBWURWUVFERXhWMGRXNXVaV3d0Y21WaFpHbHVaWE56TFdObGNuUXdIaGNOTWpJd01qRXlNVFV4TURReQpX
aGNOTXpJd01qRXdNVFV4TURReVdqQWdNUjR3SEFZRFZRUURFeFYwZFc1dVpXd3RjbVZoWkdsdVpYTnpMV05sCmNuUXdnZ0VpTUEwR0NTcU
dTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUtBb0lCQVFEUFA0MHQ4WG5aWmZNTUdzVWoKZ0FlNkpJUE90OXlWVVV1Y2ppM3dHZVJwaUszVDdB
Y2EvQzdTUHprenE4YkRyZnVwMUZDc0hYTFo0RmFSeS9BVQpTV0RtQ3NPZXpNYzFjNkYyUkJ0Y1djOGF4VDRlVEI2VE9DeTQ3cFdaOVJNVj
JnQUNJQktDc1JWZHIzZ1JQY1F6CmtCbWptSERFRTVtOUlrNVFSYktFMWl2SDV0NjhXMHFaNEZWUDNuQS9WSXhBaXlHNDczYVFFbHg5NFc1
OFV4a24KRkNOSzdEZjkycE1GUS95QmdlMEkvb1VUWUpicHdQMDBMTE1XTTgyalJ1OFJ6Q0F6SUhaNXVTUXp0UGhYQWZQVQpwS3NtRVRNVm
s1a2MvQnN1VW9QWDhGeFl1di85RHNiOEw1K21sSFJZcFRNUVhhWWs0cG1nODdYSFF5RDdnb1V1CnJmejdBZ01CQUFHalB6QTlNQTRHQTFV
ZER3RUIvd1FFQXdJQ3BEQVBCZ05WSFJNQkFmOEVCVEFEQVFIL01Cb0cKQTFVZEVRUVRNQkdDQ1d4dlkyRnNhRzl6ZEljRWZ3QUFBVEFOQm
drcWhraUc5dzBCQVFzRkFBT0NBUUVBQlBxbwpuL01NODV6QnBBZ2poUTVIYkdOU0l2MVBOWkw0MmJ3blhkbVRFQVR3WHFQbzlUWmIvd0ht
S3hUUFJ4TjhDLzRvCmFxaEdkMmRPb21zTjJ1SStNTE5JMjZVak1VNWFxVS8zajdMYzV1QmJQVHRXVURnZXE2aTFUdXVDNncrd1hTRjYKRl
V4YTFVYjF2dWZkZzl3OHREWlF2dWlnQlkwaCtYWnNjckFQVVdzNGZQMnpTQytYSElSSCtENXltR1NlWk4zeQpoam9LdGVCQTFVNlpZemtW
VDVHV1QwK1p0MVBXUS8wQmQ4RUxzSHRjOHkrVGZVckZvaUkxNlRWYm8vUDlJN1hVCnBhaXU2SFJFUFh6MFdISnppSUZyRUE2dFZIdmJXMz
dFMnJmR0ZSUWxCSnM1T0RyWTJHQndKM0RCd3MzbGN4cWUKZURyQTNzaWRZWE9TSThvMFRRPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0t
LQo=
tls.crt:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURDVENDQWZHZ0F3SUJBZ0lRTjRtVDZOTFgxZjF1K0pSVFR4LzRoakFOQmdrcWhraU
c5dzBCQVFzRkFEQWcKTVI0d0hBWURWUVFERXhWMGRXNXVaV3d0Y21WaFpHbHVaWE56TFdObGNuUXdIaGNOTWpJd01qRXlNVFV4TURReQpX
aGNOTXpJd01qRXdNVFV4TURReVdqQWdNUjR3SEFZRFZRUURFeFYwZFc1dVpXd3RjbVZoWkdsdVpYTnpMV05sCmNuUXdnZ0VpTUEwR0NTcU
dTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUtBb0lCQVFEUFA0MHQ4WG5aWmZNTUdzVWoKZ0FlNkpJUE90OXlWVVV1Y2ppM3dHZVJwaUszVDdB
Y2EvQzdTUHprenE4YkRyZnVwMUZDc0hYTFo0RmFSeS9BVQpTV0RtQ3NPZXpNYzFjNkYyUkJ0Y1djOGF4VDRlVEI2VE9DeTQ3cFdaOVJNVj
JnQUNJQktDc1JWZHIzZ1JQY1F6CmtCbWptSERFRTVtOUlrNVFSYktFMWl2SDV0NjhXMHFaNEZWUDNuQS9WSXhBaXlHNDczYVFFbHg5NFc1
OFV4a24KRkNOSzdEZjkycE1GUS95QmdlMEkvb1VUWUpicHdQMDBMTE1XTTgyalJ1OFJ6Q0F6SUhaNXVTUXp0UGhYQWZQVQpwS3NtRVRNVm
s1a2MvQnN1VW9QWDhGeFl1di85RHNiOEw1K21sSFJZcFRNUVhhWWs0cG1nODdYSFF5RDdnb1V1CnJmejdBZ01CQUFHalB6QTlNQTRHQTFV
ZER3RUIvd1FFQXdJQ3BEQVBCZ05WSFJNQkFmOEVCVEFEQVFIL01Cb0cKQTFVZEVRUVRNQkdDQ1d4dlkyRnNhRzl6ZEljRWZ3QUFBVEFOQm
drcWhraUc5dzBCQVFzRkFBT0NBUUVBQlBxbwpuL01NODV6QnBBZ2poUTVIYkdOU0l2MVBOWkw0MmJ3blhkbVRFQVR3WHFQbzlUWmIvd0ht
S3hUUFJ4TjhDLzRvCmFxaEdkMmRPb21zTjJ1SStNTE5JMjZVak1VNWFxVS8zajdMYzV1QmJQVHRXVURnZXE2aTFUdXVDNncrd1hTRjYKRl
V4YTFVYjF2dWZkZzl3OHREWlF2dWlnQlkwaCtYWnNjckFQVVdzNGZQMnpTQytYSElSSCtENXltR1NlWk4zeQpoam9LdGVCQTFVNlpZemtW
VDVHV1QwK1p0MVBXUS8wQmQ4RUxzSHRjOHkrVGZVckZvaUkxNlRWYm8vUDlJN1hVCnBhaXU2SFJFUFh6MFdISnppSUZyRUE2dFZIdmJXMz
dFMnJmR0ZSUWxCSnM1T0RyWTJHQndKM0RCd3MzbGN4cWUKZURyQTNzaWRZWE9TSThvMFRRPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0t
LQo=
tls.key:
LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb3dJQkFBS0NBUUVBenorTkxmRjUyV1h6REJyRkk0QUh1aVNEenJmY2xWRk
```

```
xuSTR0OEJua2FZaXQwK3dICkd2d3Uwajg1TTZ2R3c2MzdxZFJRckIxeTJlQldrY3Z3RkVsZzVnckRuc3pITlhPaGRrUWJYRm5QR3NVK0hr
d2UKa3pnc3VPNlZtZlVURmRvQUFpQVNnckVWWGE5NEVUM0VNNUFabzVod3hCT1p2U0pPVUVXeWhOWXJ4K2JldkZ0SwptZUJWVDk1d1AxU0
1RSXNodU85MmtCSmNmZUZ1ZkZNWkp4UWpTdXczL2RxVEJVUDhnWUh0Q1A2RkUyQ1c2Y0Q5Ck5DeXpGalBObzBidkVjd2dNeUIyZWJra003
VDRWd0h6MUtTckpoRXpGWk9aSFB3YkxsS0QxL0JjV0xyLy9RN0cKL0MrZnBwUjBXS1V6RUYybUpPS1pvUE8xeDBNZys0S0ZMcTM4K3dJRE
FRQUJBb0lCQVFDYmZ6OWFQVmdET1c4RwpvZjlRQjN2RXVpajdMUkNnS0FMblQvWS9zKzJoRVIvN1paaGZNdCtlZURsZS8zOSt5a2VNd1BS
V2J6SUJMMFZFCjF4TW01WGVDb29qcjJSR09sVmVMK01YZW9MTzNqazV4UlFPYnlnWWtLejZyNlNtTXQ2aFI3K1p6a0pKTFhLQmwKcjBaS3
VjL0M0MW1ZaEd4UmNDLzZJcEV6QWM2UGlOcVZaVmJEdEl1SzdMNzBsbzRlQ09VSzl6WUNJcG9IcUc0MgprSjIyczIva2ppVkRiTmwyRVk0
bVkrRm1RemgxMVEyaE5VZk5FS2YwZzk0b1pMdWdsWWVRLzhzV1h5UjExbWQ3CjgwbEF0eUYzWWJEaVNWOWFWMHNsUUhuZjYvSmFFcWxySW
xTa2M5OVdRN1ZpQUFyaEpSVlZjR29xQ0d0a3EyOFIKd3ZqenJrbWhBb0dCQVB0a09VblhST093VkRWdWp0aUl6RDFnZG9iSVptclNjZm5C
ZXlRUlFmRExJRXlZcVI3OQpzZ3VsMTVER1lBQXJySnR5NXRiUGdUZ0JLclUxeS93aDExd2I2NW5wT3FaVmp6MDZzMXA5eFpIb3o0RVBYNm
ZZCkEranN3ZmRVMDk1b3BtMnpxSy9DME01WlhvTEVEcTRsV0xVL21kb203WWo2T21FdzlXUm1zTVBuQW9HQkFOTU0KS2dVTGIzSGdPSERV
Q0s1M2daRkwrZVpoaGNWdENBSllHU05CN0JHdDVHWnJuZmVMVURJVThhL0hEK1YyNkdGago0dWwzNDRrM1ZjSTJITnkxMlViQTVBb3poVF
FQTWlLNUNwUW5hTjFRSmk4ejNDMlBFa2c1TjBHbEM5eWpoZHgvClJpV0VLNncrT01jV3ZHVWlGSmFncUR0a2hQaGJ0aDRVZUFvRzBsdk5B
b0dBRXhWZG4vUmVmMlpRVGlvUlgvQ3EKaUhJOUxTZ2pWaktaZ2xza3JPMCtjemdJUUtoS0ZsY3QzK0o4QmJwNWFTbm5sWGxZZXJBUDRCOS
tqWGtDL2lYWApnbCtBT0JQVFp2TVJ4NVVnTGx0ajF4cTBQZGpiNnhMajRSZXY5Y1dUdTh0Mzd0WDNsVGdWdjhSYS8xUkF4UW1DCnZXZVJs
MVRqTFVYSmpyMDNzNkowQXJzQ2dZQmkybHJBUENmNk5lVVo3Q2NZM1Q4bU9Ha1lJT3ZRbW81YTByc0wKaWJBeVk3b013ekdkcG5KczB1U1
Z5ZTdITTFnTGJnYjJCa0xjbmIrY2xVNFZqMVB6VVJ3L0x2eWdjdEd2OCs1aAprTWdNaWFNcER1aDlxbnNNdTBnSGhuQTA2NThpdGJhOGtk
a0c5Nm9PanJhMXhDM1NrL1BZeGUrR2lTYVhsa3pYCm5YUmdyUUtCZ0VmSkZEM0p1N1lHT3doUkRER2dOMUdRWkd0TFFzbFdUUnBYWlRtWD
FiT016VWxSeUIzaUdvdHoKalJHbWdYVE9lQzJIK3dBczQzdWJHbklrbThrVk1OaWtXSlI0a3VOeDViOEFQUkc1R2llMysvKzJHU0NuZ1NG
UAp6TVU1dWpEN2lNQklBL0VGMDhMSzVJY2I5bTJJYVo0ZmJqNmdYZU5oeTdNdHh2VDR0eGdDCi0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS
0tLS0tCg==
type: kubernetes.io/tls
```

3. *Database name:* The database name of the MongoDB receiver, for example: **mongodb-audit-log-receiver-1**
4. *Address:* The MongoDB receiver address. For example: **<IP or hostname>:27017**
5. *MongoDB log in Username, Password Secret:*
    1. By default, MongoDB does not require a username and password to access, but it can be enabled by configuring the authentication. If the MongoDB has a username and password, you should create a Kubernetes secret that contains **user: <MongoDB username>** and **password: <MongoDB password>** in the Secure Tunnel namespace, then enter the secret name here.

       Kubernetes secret example:

       ```
       kind: Secret
       apiVersion: v1
       metadata:
         name: mongodb-credential
         namespace: tunnel-stand-alone
       data:
         password: YmJiYg==
         user: YWFhYQ==
       type: Opaque
       ```

2. Click **Syslog receiver** > **Add Syslog receiver** to add a **Syslog audit log receiver**, Enter the following values:

- *Receiver name:* the name of the Syslog receiver. It's a unique name of the audit log receiver. for example: *syslog-receiver-1*.
- *Address:* The syslog server address. for example: **<IP or hostname>:5514**

Configuring YAML to update the existing Tunnel instance

1. From the Openshift console, click **Operators** > **Installed Operators**.
    - From the **Project** dropdown menu, select the project that you created the Secure Tunnel instance in.
    - Click the **IBM Secure Tunnel** > **Tunnel** tab > click the existing **sre-tunnel** instance > change to **YAML** tab.
2. For Secure Tunnel to send audit logs to its container log, configure the **YAML** file:

```
spec:
  ...
  showInContainerLog: true
  ...
```

3. For Secure Tunnel to send audit logs to MongoDB, configure the **YAML**:

```
spec:
  ...
  auditReceiver:
    mongoDB:
      - mongoAddress:
          mongoAddress: '<IP or hostname that the MongoDB running in>:27017'
          mongoDatabaseName: secure-tunnel-001
          name: mongodb-audit-log
          passwordSecretName: <The name of the Kubernetes secret that saved the MongoDB username and password>
          tlsSecret: <The name of the Kubernetes secret that saved the SSL certificate of the MongoDB service>
  ...
```

4. For Secure Tunnel send audit logs to a remote syslog server, configure the **YAML**:

```
spec:
  ...
  auditReceiver:
    syslog:
      - name: syslog
        syslogAddress: '<IP or hostname that the Syslog service running in>:5514'
  ...
```

Tip: It's easier to configure the audit log with the Configuring with the operator UI when create Secure Tunnel Installation method, so you can configure it from there. Switch to the **YAML view**, obtain the configuration yaml and update the existing Secure Tunnel instance yaml to configure the audit log.

Updating the deployment type (IBM Cloud Pak for AIOps on OpenShift)

Learn how to switch between the base deployment and extended deployment types.

Overview

After IBM Cloud Pak® for AIOps is installed, you can change your deployment type from a base deployment to an extended deployment, or from an extended deployment to a base deployment. For more information about the base deployment and extended deployment types, see Incremental adoption.

If you have log or ticket integrations configured that require an extended deployment, you are prevented from changing to a base deployment until these integrations are removed.

Important: If you are changing from a base deployment to an extended deployment, ensure that you meet the additional hardware requirements. For more information, see Hardware requirements.

You can change your deployment type with the Red Hat® OpenShift® Container Platform console or CLI.

Switching deployment type with the Red Hat OpenShift console

1. From your Red Hat OpenShift console, click **Operators** > **Installed Operators**.
2. From the **Project** dropdown menu, select the project that IBM Cloud Pak for AIOps is deployed in.
3. Select the **IBM Cloud Pak for AIOps** operator, then click the **IBM Cloud Pak for AIOps** tab.
4. Click the IBM Cloud Pak for AIOps deployment that is displayed in the **Installations** section, and then switch to the **YAML** view.
5. If the section **spec** > **pakModules** > **logAnomalyDetection** does not exist, then add it. Otherwise, find and edit the `logAnomalyDetection` section under **spec** > **pakModules**.

   Set the value of **enabled** to one of the following options and then click **Save**:

   - `false` for a base deployment without log anomaly detection and ticket analysis capabilities.
   - `true` for an extended deployment with log anomaly detection and ticket analysis capabilities.

   Example `pakModules` excerpt for a base deployment:

```
pakModules:
 - enabled: true
   name: aiManager
 - enabled: true
   name: aiopsFoundation
 - enabled: true
   name: applicationManager
 - enabled: false
   name: connection
 - enabled: false
   name: logAnomalyDetection
```

   Example `pakModules` excerpt for an extended deployment:

```
pakModules:
 - enabled: true
   name: aiManager
 - enabled: true
   name: aiopsFoundation
 - enabled: true
   name: applicationManager
 - enabled: false
   name: connection
 - enabled: true
   name: logAnomalyDetection
```

   Warning: The pakModules `aiManager`, `aiopsFoundation`, and `applicationManager` must be enabled as in the preceding YAML. Do not change these values to false.

Switching deployment type with the Red Hat OpenShift CLI

1. Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login`.
2. Set environment variables.

```
export AIOPS_NAMESPACE=<project>
export AIOPS_INSTANCE=$(oc get installation -o jsonpath='{.items[0].metadata.name}' -n ${AIOPS_NAMESPACE})
```

   Where `<project>` is the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in.

3. Edit your IBM Cloud Pak for AIOps `installation` instance by running the following command:

   ```
   oc edit installation ${AIOPS_INSTANCE} -n ${AIOPS_NAMESPACE}
   ```

4. If the section **spec** > **pakModules** > **logAnomalyDetection** does not exist, then add it. Otherwise, find and edit the `logAnomalyDetection` section under **spec** > **pakModules**.

   Set the value of **enabled** to one of the following options and then click **Save**:

   - `false` for a base deployment without log anomaly detection and ticket analysis capabilities.
   - `true` for an extended deployment with log anomaly detection and ticket analysis capabilities.

   Example `pakModules` excerpt for a base deployment:

   ```
   pakModules:
    - enabled: true
      name: aiManager
    - enabled: true
      name: aiopsFoundation
    - enabled: true
      name: applicationManager
    - enabled: false
      name: connection
    - enabled: false
      name: logAnomalyDetection
   ```

   Example `pakModules` excerpt for an extended deployment:

   ```
   pakModules:
    - enabled: true
      name: aiManager
    - enabled: true
      name: aiopsFoundation
    - enabled: true
      name: applicationManager
    - enabled: false
      name: connection
    - enabled: true
      name: logAnomalyDetection
   ```

   Warning: The pakModules `aiManager`, `aiopsFoundation`, and `applicationManager` must be enabled as in the preceding YAML. Do not change these values to false.

Uninstalling IBM Cloud Pak for AIOps on OpenShift

These instructions can be used to uninstall an online or offline (airgap) deployment of IBM Cloud Pak® for AIOps, and to clean up the resources that it created.

1. Run the uninstall script
2. (Optional) Delete the IBM Cloud Pak for AIOps catalog
3. (Optional) Delete the Cert Manager
4. (Optional) Delete the License Service

Before you begin

The Red Hat® OpenShift® Container Platform command line interface (`oc`) must be installed on your Red Hat OpenShift cluster, and you must be logged in with `oc login`.

1. Run the uninstall script

Download the uninstall and cleanup script from github.com/IBM.

You can configure the script to uninstall and clean up your IBM Cloud Pak for AIOps deployment. Details on how to configure the script are provided in the same location.

You can configure the script to remove specific components and resources. Remove the following resources only if you are sure that they are not used by other CloudPaks on your Red Hat OpenShift cluster, and you do not want to persist the data.

- Persistent Volume Claims (PVCs)
- shared components that are used by other Cloud Paks

Important: If the script appears to be stuck and 1.5 hours has elapsed, then contact IBM Support for assistance in safely cleaning up your environment. Do not manually delete resources.

Example output:

```
# ./uninstall-cp4waiops.sh -s
[INFO] Uninstall v2.0 for AIOPs v4.8
[INFO]
[INFO] This script will uninstall IBM Cloud Pak for AIOps.
[INFO] ##### Properties in uninstall-cp4waiops.props #####
```

```
[INFO]
[INFO]  The script uninstall-cp4waiops.props has 'DELETE_ALL=true', hence the script will execute wih below values:
[INFO]
[INFO] CP4WAIOPS_PROJECT=cp4aiops
[INFO] INSTALLATION_NAME=ibm-cp-aiops
[INFO] DELETE_PVCS=true
[INFO] DELETE_CRDS=true
[INFO]
[INFO] ##### Properties in uninstall-cp4waiops.props #####
[INFO]
[INFO] Checking where operators are installed...
[INFO] Uninstalling AIOps from the namespace cp4aiops.
Using project "default" on server "https://api.o1-123456.cp.test.com:6443".

[INFO] Prereq checks passed. Starting uninstall of IBM Cloud Pak for AIOps ...

[INFO] Checking if user created instances are found...
error: the server doesn't have a resource type "kongs"
error: the server doesn't have a resource type "eventmanagergateways"
[INFO] No remaining user-created instances, proceeding ahead...
[INFO] Deleting all ConnectorConfigurations
[INFO] waiting for ConnectorComponent termination
No resources found
[INFO] Finished deleting ConnectorConfigurations
[INFO] Deleting the installation CR...
[INFO] Found installation CR ibm-cp-aiops to delete.
[INFO] Waiting for  instances to be deleted.  This will take a while....
installation.orchestrator.aiops.ibm.com "ibm-cp-aiops" deleted
[INFO] ibm-cp-aiops instance got deleted successfully!
[INFO] Checking if any additional installation resources found in the cluster.
No resources found
[INFO] No additional installation resources found in the cluster.
[INFO] Checking if operandrequests are all deleted
[INFO] Expected operandrequests got deleted successfully!
[INFO] Deleting the misc resources in cp4aiops
[INFO] Deleting resource  rolebinding.authorization.openshift.io/example-redis..
[INFO] Deleting resource  role.authorization.openshift.io/example-redis..
[INFO] Deleting resource  role/cp4waiops-eventprocessor-eve-29ee-ep-role..
[INFO] Deleting resource  rolebinding/cp4waiops-eventprocessor-eve-29ee-ep-role..
[INFO] Deleting resource  rolebinding.rbac.authorization.k8s.io/cp4waiops-eventprocessor-eve-29ee-ep-rb..
[INFO] Deleting resource  rolebinding.authorization.openshift.io/cp4waiops-eventprocessor-eve-29ee-ep-rb..
[INFO] Deleting resource  networkpolicy.networking.k8s.io/c-example-redis-default-deny..
[INFO] Deleting resource  secret/aiopsedge-client-cert..
secret "aiopsedge-client-cert" deleted
[INFO] Deleting resource  priorityclass.scheduling.k8s.io/high-priority..
Warning: deleting cluster-scoped resources, not scoped to the provided namespace
[INFO] Deleting resource  priorityclass.scheduling.k8s.io/aiops-topology-cassandra-priority..
Warning: deleting cluster-scoped resources, not scoped to the provided namespace
[INFO] Deleting resource  issuers.cert-manager.io/aiops-flink-issuer..
[INFO] Deleting resource  issuers.cert-manager.io/aiops-flink-ss-issuer..
[INFO] Deleting resource  issuers.cert-manager.io/automationbase-sample-automationbase-ab-ss-issuer..
[INFO] Deleting resource  priorityclass.scheduling.k8s.io/aimanager-aio-luigi-priority..
Warning: deleting cluster-scoped resources, not scoped to the provided namespace
[INFO] Deleting resource  rolebinding/flink-role-binding..
rolebinding.rbac.authorization.k8s.io "flink-role-binding" deleted
[INFO] Deleting resource  role/flink..
role.rbac.authorization.k8s.io "flink" deleted
[INFO] Deleting the Secure tunnel resources in cp4aiops
No resources found
No resources found
No resources found
[INFO] Finished deleting securetunnel resources
[INFO] Uninstalling the CP4AIOps operator...
[INFO] Deleting the subscription subscription.operators.coreos.com/ibm-aiops-orchestrator
subscription.operators.coreos.com "ibm-aiops-orchestrator" deleted
[INFO] Deleting the clusterserviceversion ibm-aiops-orchestrator.v0.1.0-202412021600
clusterserviceversion.operators.coreos.com "ibm-aiops-orchestrator.v0.1.0-202412021600" deleted
[INFO] Waiting for the deletion of all the ClusterServiceVersions ibm-aiops-orchestrator.v0.1.0-202412021600 for the
subscription of the operator subscription.operators.coreos.com/ibm-aiops-orchestrator
[INFO] Deletion of all the ClusterServiceVersions ibm-aiops-orchestrator.v0.1.0-202412021600 for the subscription of the
operator subscription.operators.coreos.com/ibm-aiops-orchestrator completed successfully.
[INFO] Deleting MISC certs in cp4aiops
[INFO] Deleting cert certificates.cert-manager.io/ibm-cp-aiops-edb-postgres-client-cert..
[INFO] Deleting cert certificates.cert-manager.io/ibm-cp-aiops-edb-postgres-server-cert..
[INFO] Deleting cert certificates.cert-manager.io/ibm-cp-aiops-edb-postgres-ss-ca..
[INFO] Deleting kafkatopics in cp4aiops
[INFO] Deleting kafkatopic with label ibmevents.ibm.com/cluster=iaf-system...
No resources found
[INFO] Deleting KafkaUser in cp4aiops
kafkauser.ibmevents.ibm.com "cp4waiops-cartridge-kafka-auth-0" deleted
[INFO] Deleting lease in cp4aiops
[INFO] Deleting lease lease.coordination.k8s.io/9efbc0e4.aiops.ibm.com..
lease.coordination.k8s.io "9efbc0e4.aiops.ibm.com" deleted
[INFO] Deleting lease lease.coordination.k8s.io/5ba58da1.ai.ir.aiops.ibm.com..
lease.coordination.k8s.io "5ba58da1.ai.ir.aiops.ibm.com" deleted
[INFO] Deleting lease lease.coordination.k8s.io/e1d7d41e.ibm.com..
lease.coordination.k8s.io "e1d7d41e.ibm.com" deleted
[INFO] Deleting lease lease.coordination.k8s.io/fc7f2af9.ibm.com..
lease.coordination.k8s.io "fc7f2af9.ibm.com" deleted
[INFO] Deleting lease lease.coordination.k8s.io/ff20eae3.connectors.aiops.ibm.com..
lease.coordination.k8s.io "ff20eae3.connectors.aiops.ibm.com" deleted
[INFO] Deleting lease lease.coordination.k8s.io/aiopsedge-connector-manager..
```

```
lease.coordination.k8s.io "aiopsedge-connector-manager" deleted
[INFO] Deleting internal configmaps in cp4aiops
[INFO] Deleting configmap configmap/ff20eae3.connectors.aiops.ibm.com..
[INFO] Deleting configmap configmap/aiopsedge-maven-settings..
[INFO] Deleting configmap configmap/aiops-topology-cassandra-bootstrap-config..
configmap "aiops-topology-cassandra-bootstrap-config" deleted
[INFO] Deleting configmap configmap/cloud-native-postgresql-image-list..
configmap "cloud-native-postgresql-image-list" deleted
Error from server (NotFound): configmaps "aiops-custom-size-profile" not found
[INFO] Delete EDB Connection Secret
Error from server (NotFound): secrets "ibm-cp-aiops-edb-secret" not found
[INFO] Delete EDB Cert related resources
secret "ibm-cp-aiops-edb-postgres-client-cert" deleted
secret "ibm-cp-aiops-edb-postgres-server-cert" deleted
secret "ibm-cp-aiops-edb-postgres-ss-cacert" deleted
Error from server (NotFound): certificaterequests.cert-manager.io "ibm-cp-aiops-edb-postgres-client-cert-1" not found
Error from server (NotFound): certificaterequests.cert-manager.io "ibm-cp-aiops-edb-postgres-server-cert-1" not found
Error from server (NotFound): certificaterequests.cert-manager.io "ibm-cp-aiops-edb-postgres-ss-ca-1" not found
Error from server (NotFound): certificates.cert-manager.io "ibm-cp-aiops-edb-postgres-client-cert" not found
Error from server (NotFound): certificates.cert-manager.io "ibm-cp-aiops-edb-postgres-server-cert" not found
Error from server (NotFound): certificates.cert-manager.io "ibm-cp-aiops-edb-postgres-ss-ca" not found
[INFO] Delete EDB secrets
secret "postgresql-operator-controller-manager-config" deleted
[INFO] Delete Redis resources in  Namespace
serviceaccount "ibm-redis-cp-operator-serviceaccount" deleted
lease.coordination.k8s.io "ibm-redis-cp-operator" deleted
Error from server (NotFound): configmaps "ibm-redis-cp-operator" not found
[INFO] Delete Redis Secrets
secret "ibm-cp-aiops-redis-client-cert" deleted
secret "ibm-cp-aiops-redis-server-cert" deleted
secret "ibm-cp-aiops-redis-ss-cacert" deleted
[INFO] Delete Leftover Redis Roles and Role Bindings
role.rbac.authorization.k8s.io "ibm-redis-cp.v1.2.2-ibm--1xwwhWRXYx3Q8jcfUvhXuvyC9A7m8nzINg0QTA" deleted
rolebinding.rbac.authorization.k8s.io "ibm-redis-cp.v1.2.2-ibm--1xwwhWRXYx3Q8jcfUvhXuvyC9A7m8nzINg0QTA" deleted
[INFO] Deleting Internal PVCs in cp4aiops
[INFO] Deleting PVCs with label app=cassandra..
persistentvolumeclaim "back-aiops-topology-cassandra-0" deleted
persistentvolumeclaim "data-aiops-topology-cassandra-0" deleted
[INFO] Deleting PVCs with label formation_type=redis..
No resources found
[INFO] Deleting PVCs with label app.kubernetes.io/component=ep..
No resources found
[INFO] Deleting PVCs in cp4aiops
[INFO] Deleting PVCs with label app.kubernetes.io/instance=aimanager..
persistentvolumeclaim "export-aimanager-ibm-minio-0" deleted
[INFO] Deleting PVCs with label app.kubernetes.io/instance=example-couchdbcluster..
No resources found
[INFO] Deleting PVCs with label app.kubernetes.io/managed-by=ir-core-operator..
persistentvolumeclaim "data-c-example-couchdbcluster-m-0" deleted
[INFO] Deleting PVCs with label app.kubernetes.io/managed-by=ibm-elastic-operator..
No resources found
[INFO] Deleting PVCs with label app.kubernetes.io/managed-by=ibm-elasticsearch..
persistentvolumeclaim "data-aiops-ibm-elasticsearch-es-server-all-0" deleted
[INFO] Deleting Linked secrets in cp4aiops
[INFO] Deleting Linked secrets to some PVC's with name secret/aimanager-ca-certificate-secret..
[INFO] Deleting Linked secrets to some PVC's with name secret/aimanager-certificate-secret..
secret "aimanager-certificate-secret" deleted
[INFO] Deleting Linked secrets to some PVC's with name secret/aiops-ir-core-model-secret..
secret "aiops-ir-core-model-secret" deleted
[INFO] Deleting Linked secrets to some PVC's with name secret/aiops-ir-core-omni-secret..
secret "aiops-ir-core-omni-secret" deleted
[INFO] Deleting Linked secrets to some PVC's with name secret/kafka-truststore..
[INFO] Deleting Linked secrets to some PVC's with name secret/aiops-topology-asm-iafelastic-ca..
[INFO] Deleting Linked secrets to some PVC's with name secret/aiops-topology-asm-iafkafka-ca..
[INFO] Deleting Linked secrets to some PVC's with name secret/aiops-metrics-processor..
[INFO] Deleting Linked secrets to some PVC's with name secret/iaf-system-cluster-ca..
secret "iaf-system-cluster-ca" deleted
[INFO] Deleting Linked secrets to some PVC's with name secret/iaf-system-cluster-ca-cert..
secret "iaf-system-cluster-ca-cert" deleted
[INFO] Deleting internal secret  secret/aiopsedgeca-cert-secret..
secret "aiopsedgeca-cert-secret" deleted
[INFO] Deleting internal secret  secret/aiops-topology-cassandra-cert-manager-ca..
secret "aiops-topology-cassandra-cert-manager-ca" deleted
[INFO] Deleting internal secret  secret/automationbase-sample-automationbase-ab-ss-ca..
secret "automationbase-sample-automationbase-ab-ss-ca" deleted
[INFO] Deleting internal secret  secret/iaf-system-elasticsearch-es-client-cert-kp..
[INFO] Deleting internal secret  secret/iaf-system-elasticsearch-es-ss-cacert-kp..
[INFO] Deleting internal secret  secret/connector-bridge-cert-secret..
secret "connector-bridge-cert-secret" deleted
[INFO] Deleting internal secret  secret/aiops-appconnect-ir-secret..
secret "aiops-appconnect-ir-secret" deleted
[INFO] Deleting internal secret  secret/cp4waiops-cartridge-es-auth..
secret "cp4waiops-cartridge-es-auth" deleted
[INFO] Deleting internal secret with label release=aiops-topology..
secret "aiops-topology-cassandra-auth-secret" deleted
[INFO] Deleting the serviceaccounts in cp4aiops
[INFO] Deleting serviceaccounts serviceaccount/cp4waiops-eventprocessor-eve-29ee-ep..
[INFO] Deleting serviceaccounts serviceaccount/flink..
serviceaccount "flink" deleted
[INFO] Deleting serviceaccounts serviceaccount/postgresql-operator-manager..
serviceaccount "postgresql-operator-manager" deleted
namespace/cp4aiops annotated
```

```
[INFO] Finding if IBM Cloud Pak Foundational Services is Namespaced or Global
[INFO] IBM Cloud Pak Foundational Services: Namespaced Scope

[INFO] Starting uninstall of IBM Cloud Pak Foundational Services components
[INFO] Delete ZenService related resources
[INFO] Checking for ZenClient
NAME                 AGE
zenclient-cp4aiops   78m
[INFO] Deleting ZenClient found
client.oidc.security.ibm.com/zenclient-cp4aiops patched
Error from server (NotFound): clients.oidc.security.ibm.com "zenclient-cp4aiops" not found
[INFO] Delete zen setup-job
Delete Zen Service Account
serviceaccount "ibm-zen-operator-serviceaccount" deleted
[INFO] Delete Zen secrets
secret "ibm-zen-objectstore-cert" deleted
secret "zen-metastore-edb-replica-client" deleted
secret "zen-metastore-edb-server" deleted
No resources found
commonservice.operator.ibm.com "common-service" deleted
commonservice.operator.ibm.com "im-common-service" deleted
Uninstall CPFS
[INFO] Deleting the subscription subscription.operators.coreos.com/aiops-ibm-common-services
subscription.operators.coreos.com "aiops-ibm-common-services" deleted
[INFO] Deleting the clusterserviceversion ibm-common-service-operator.v4.6.8
clusterserviceversion.operators.coreos.com "ibm-common-service-operator.v4.6.8" deleted
[INFO] Waiting for the deletion of all the ClusterServiceVersions ibm-common-service-operator.v4.6.8 for the subscription
of the operator subscription.operators.coreos.com/aiops-ibm-common-services
[INFO] Deletion of all the ClusterServiceVersions ibm-common-service-operator.v4.6.8 for the subscription of the operator
subscription.operators.coreos.com/aiops-ibm-common-services completed successfully.
[INFO] Checking if operandrequests are all deleted
[INFO] Expected operandrequests got deleted successfully!
Check for operandconfigs to delete
operandconfig.operator.ibm.com "common-service" deleted
Check for operandregistries
operandregistry.operator.ibm.com "common-service" deleted
Uninstall ODLM
[INFO] Deleting the subscription subscription.operators.coreos.com/operand-deployment-lifecycle-manager-app
subscription.operators.coreos.com "operand-deployment-lifecycle-manager-app" deleted
[INFO] Deleting the clusterserviceversion operand-deployment-lifecycle-manager.v4.3.7
clusterserviceversion.operators.coreos.com "operand-deployment-lifecycle-manager.v4.3.7" deleted
[INFO] Waiting for the deletion of all the ClusterServiceVersions operand-deployment-lifecycle-manager.v4.3.7 for the
subscription of the operator subscription.operators.coreos.com/operand-deployment-lifecycle-manager-app
[INFO] Deletion of all the ClusterServiceVersions operand-deployment-lifecycle-manager.v4.3.7 for the subscription of the
operator subscription.operators.coreos.com/operand-deployment-lifecycle-manager-app completed successfully.
service "meta-api-svc" deleted
Warning: deleting cluster-scoped resources, not scoped to the provided namespace
[INFO] Deleting ValidatingWebhookConfiguration
configmap "common-service-maps" deleted
[INFO] Checking is namespace cs-control exists
[INFO] Cleaning up IBM Cert Manager from cs-control namespace
[WARNING] The subscription for the operator  with label operators.coreos.com/ibm-cert-manager-operator.cs-control does
not exist in cs-control, skipping unsubscription.
[INFO] Cleaning up IBM Namespace Scope Operator from cs-control namespace
error: the server doesn't have a resource type "namespacescopes"
[WARNING] The subscription for the operator  with label operators.coreos.com/ibm-namespace-scope-operator.cs-control does
not exist in cs-control, skipping unsubscription.
Error from server (NotFound): namespaces "cs-control" not found
[INFO] Checking for CPFS custom resources
[INFO] Resource Check Attempt: 1
authentications.operator.ibm.com
No resources found
None Found

clients.oidc.security.ibm.com
No resources found
None Found

commonservices.operator.ibm.com
No resources found
None Found

commonwebuis.operators.ibm.com
No resources found
None Found

kafkamirrormaker2s.ibmevents.ibm.com
No resources found
None Found

kafkamirrormakers.ibmevents.ibm.com
No resources found
None Found

kafkas.ibmevents.ibm.com
NAMESPACE   NAME         DESIRED KAFKA REPLICAS   DESIRED ZK REPLICAS   READY   WARNINGS
cp4aiops    iaf-system   1                        1                     True

kafkatopics.ibmevents.ibm.com
No resources found
None Found
```

```
kafkausers.ibmevents.ibm.com
No resources found
None Found

navconfigurations.foundation.ibm.com
No resources found
None Found

operandbindinfos.operator.ibm.com
No resources found
None Found

operandconfigs.operator.ibm.com
No resources found
None Found

operandregistries.operator.ibm.com
No resources found
None Found

operandrequests.operator.ibm.com
No resources found
None Found

switcheritems.operators.ibm.com
No resources found
None Found

zenservices.zen.cpd.ibm.com
No resources found
None Found

strimzipodsets.core.ibmevents.ibm.com
NAMESPACE   NAME                   PODS   READY PODS   CURRENT PODS   AGE
cp4aiops    iaf-system-kafka       1      1            1              118m
cp4aiops    iaf-system-zookeeper   1      1            1              119m

zenextensions.zen.cpd.ibm.com
No resources found
None Found


CustomResources found... Trying again in 30 seconds.
[INFO] Resource Check Attempt: 2
authentications.operator.ibm.com
No resources found
None Found

clients.oidc.security.ibm.com
No resources found
None Found

commonservices.operator.ibm.com
No resources found
None Found

commonwebuis.operators.ibm.com
No resources found
None Found

kafkamirrormaker2s.ibmevents.ibm.com
No resources found
None Found

kafkamirrormakers.ibmevents.ibm.com
No resources found
None Found

kafkas.ibmevents.ibm.com
NAMESPACE   NAME         DESIRED KAFKA REPLICAS   DESIRED ZK REPLICAS   READY   WARNINGS
cp4aiops    iaf-system   1                        1                     True

kafkatopics.ibmevents.ibm.com
No resources found
None Found

kafkausers.ibmevents.ibm.com
No resources found
None Found

navconfigurations.foundation.ibm.com
No resources found
None Found

operandbindinfos.operator.ibm.com
No resources found
None Found

operandconfigs.operator.ibm.com
No resources found
None Found
```

```
operandregistries.operator.ibm.com
No resources found
None Found

operandrequests.operator.ibm.com
No resources found
None Found

switcheritems.operators.ibm.com
No resources found
None Found

zenservices.zen.cpd.ibm.com
No resources found
None Found

strimzipodsets.core.ibmevents.ibm.com
NAMESPACE   NAME                   PODS   READY PODS   CURRENT PODS   AGE
cp4aiops    iaf-system-kafka       1      1            1              119m
cp4aiops    iaf-system-zookeeper   1      1            1              119m

zenextensions.zen.cpd.ibm.com
No resources found
None Found


CustomResources found... Trying again in 30 seconds.
[INFO] Resource Check Attempt: 3
authentications.operator.ibm.com
No resources found
None Found

clients.oidc.security.ibm.com
No resources found
None Found

commonservices.operator.ibm.com
No resources found
None Found

commonwebuis.operators.ibm.com
No resources found
None Found

kafkamirrormaker2s.ibmevents.ibm.com
No resources found
None Found

kafkamirrormakers.ibmevents.ibm.com
No resources found
None Found

kafkas.ibmevents.ibm.com
NAMESPACE   NAME         DESIRED KAFKA REPLICAS   DESIRED ZK REPLICAS   READY   WARNINGS
cp4aiops    iaf-system   1                        1                     True

kafkatopics.ibmevents.ibm.com
No resources found
None Found

kafkausers.ibmevents.ibm.com
No resources found
None Found

navconfigurations.foundation.ibm.com
No resources found
None Found

operandbindinfos.operator.ibm.com
No resources found
None Found

operandconfigs.operator.ibm.com
No resources found
None Found

operandregistries.operator.ibm.com
No resources found
None Found

operandrequests.operator.ibm.com
No resources found
None Found

switcheritems.operators.ibm.com
No resources found
None Found

zenservices.zen.cpd.ibm.com
No resources found
None Found
```

```
strimzipodsets.core.ibmevents.ibm.com
NAMESPACE   NAME                    PODS   READY PODS   CURRENT PODS   AGE
cp4aiops    iaf-system-kafka        1      1             1              120m
cp4aiops    iaf-system-zookeeper    1      1             1              120m

zenextensions.zen.cpd.ibm.com
No resources found
None Found


CustomResources found... Trying again in 30 seconds.
[INFO] Resource Check Attempt: 4
authentications.operator.ibm.com
No resources found
None Found

clients.oidc.security.ibm.com
No resources found
None Found

commonservices.operator.ibm.com
No resources found
None Found

commonwebuis.operators.ibm.com
No resources found
None Found

kafkamirrormaker2s.ibmevents.ibm.com
No resources found
None Found

kafkamirrormakers.ibmevents.ibm.com
No resources found
None Found

kafkas.ibmevents.ibm.com
NAMESPACE   NAME          DESIRED KAFKA REPLICAS   DESIRED ZK REPLICAS   READY   WARNINGS
cp4aiops    iaf-system    1                        1                     True

kafkatopics.ibmevents.ibm.com
No resources found
None Found

kafkausers.ibmevents.ibm.com
No resources found
None Found

navconfigurations.foundation.ibm.com
No resources found
None Found

operandbindinfos.operator.ibm.com
No resources found
None Found

operandconfigs.operator.ibm.com
No resources found
None Found

operandregistries.operator.ibm.com
No resources found
None Found

operandrequests.operator.ibm.com
No resources found
None Found

switcheritems.operators.ibm.com
No resources found
None Found

zenservices.zen.cpd.ibm.com
No resources found
None Found

strimzipodsets.core.ibmevents.ibm.com
NAMESPACE   NAME                    PODS   READY PODS   CURRENT PODS   AGE
cp4aiops    iaf-system-kafka        1      1             1              121m
cp4aiops    iaf-system-zookeeper    1      1             1              121m

zenextensions.zen.cpd.ibm.com
No resources found
None Found


CustomResources found... Trying again in 30 seconds.
Deleting CPFS CRDS
[INFO] Deleting CRD authentications.operator.ibm.com..
customresourcedefinition.apiextensions.k8s.io "authentications.operator.ibm.com" deleted
[INFO] Deleting CRD clients.oidc.security.ibm.com..
customresourcedefinition.apiextensions.k8s.io "clients.oidc.security.ibm.com" deleted
```

```
[INFO] Deleting CRD commonservices.operator.ibm.com..
customresourcedefinition.apiextensions.k8s.io "commonservices.operator.ibm.com" deleted
[INFO] Deleting CRD commonwebuis.operators.ibm.com..
customresourcedefinition.apiextensions.k8s.io "commonwebuis.operators.ibm.com" deleted
[INFO] Deleting CRD kafkamirrormaker2s.ibmevents.ibm.com..
customresourcedefinition.apiextensions.k8s.io "kafkamirrormaker2s.ibmevents.ibm.com" deleted
[INFO] Deleting CRD kafkamirrormakers.ibmevents.ibm.com..
customresourcedefinition.apiextensions.k8s.io "kafkamirrormakers.ibmevents.ibm.com" deleted
[INFO] Deleting CRD kafkas.ibmevents.ibm.com..
customresourcedefinition.apiextensions.k8s.io "kafkas.ibmevents.ibm.com" deleted
[INFO] Deleting CRD kafkatopics.ibmevents.ibm.com..
customresourcedefinition.apiextensions.k8s.io "kafkatopics.ibmevents.ibm.com" deleted
[INFO] Deleting CRD kafkausers.ibmevents.ibm.com..
customresourcedefinition.apiextensions.k8s.io "kafkausers.ibmevents.ibm.com" deleted
[INFO] Deleting CRD navconfigurations.foundation.ibm.com..
customresourcedefinition.apiextensions.k8s.io "navconfigurations.foundation.ibm.com" deleted
[INFO] Deleting CRD operandbindinfos.operator.ibm.com..
customresourcedefinition.apiextensions.k8s.io "operandbindinfos.operator.ibm.com" deleted
[INFO] Deleting CRD operandconfigs.operator.ibm.com..
customresourcedefinition.apiextensions.k8s.io "operandconfigs.operator.ibm.com" deleted
[INFO] Deleting CRD operandregistries.operator.ibm.com..
customresourcedefinition.apiextensions.k8s.io "operandregistries.operator.ibm.com" deleted
[INFO] Deleting CRD operandrequests.operator.ibm.com..
customresourcedefinition.apiextensions.k8s.io "operandrequests.operator.ibm.com" deleted
[INFO] Deleting CRD switcheritems.operators.ibm.com..
customresourcedefinition.apiextensions.k8s.io "switcheritems.operators.ibm.com" deleted
[INFO] Deleting CRD zenservices.zen.cpd.ibm.com..
customresourcedefinition.apiextensions.k8s.io "zenservices.zen.cpd.ibm.com" deleted
[INFO] Deleting CRD strimzipodsets.core.ibmevents.ibm.com..
customresourcedefinition.apiextensions.k8s.io "strimzipodsets.core.ibmevents.ibm.com" deleted
[INFO] Deleting CRD zenextensions.zen.cpd.ibm.com..
customresourcedefinition.apiextensions.k8s.io "zenextensions.zen.cpd.ibm.com" deleted
[INFO] Checking for Bedrock resources
[INFO] Resource Check Attempt: 1
authentications.operator.ibm.com
error: the server doesn't have a resource type "authentications"
None Found

clients.oidc.security.ibm.com
error: the server doesn't have a resource type "clients"
None Found

commonservices.operator.ibm.com
error: the server doesn't have a resource type "commonservices"
None Found

commonwebuis.operators.ibm.com
error: the server doesn't have a resource type "commonwebuis"
None Found

kafkamirrormaker2s.ibmevents.ibm.com
error: the server doesn't have a resource type "kafkamirrormaker2s"
None Found

kafkamirrormakers.ibmevents.ibm.com
error: the server doesn't have a resource type "kafkamirrormakers"
None Found

kafkas.ibmevents.ibm.com
error: the server doesn't have a resource type "kafkas"
None Found

kafkatopics.ibmevents.ibm.com
error: the server doesn't have a resource type "kafkatopics"
None Found

kafkausers.ibmevents.ibm.com
error: the server doesn't have a resource type "kafkausers"
None Found

navconfigurations.foundation.ibm.com
error: the server doesn't have a resource type "navconfigurations"
None Found

operandbindinfos.operator.ibm.com
error: the server doesn't have a resource type "operandbindinfos"
None Found

operandconfigs.operator.ibm.com
error: the server doesn't have a resource type "operandconfigs"
None Found

operandregistries.operator.ibm.com
error: the server doesn't have a resource type "operandregistries"
None Found

operandrequests.operator.ibm.com
error: the server doesn't have a resource type "operandrequests"
None Found

switcheritems.operators.ibm.com
error: the server doesn't have a resource type "switcheritems"
```

```
None Found

zenservices.zen.cpd.ibm.com
error: the server doesn't have a resource type "zenservices"
None Found

strimzipodsets.core.ibmevents.ibm.com
error: the server doesn't have a resource type "strimzipodsets"
None Found

zenextensions.zen.cpd.ibm.com
error: the server doesn't have a resource type "zenextensions"
None Found


[INFO] Delete MISC IAM ConfigMap and Role/Rolebinding
configmap "cs-keycloak-theme" deleted
[INFO] Delete role and rolebinding
Error from server (NotFound): rolebindings.rbac.authorization.k8s.io "leader-election-rolebinding" not found
Error from server (NotFound): roles.rbac.authorization.k8s.io "leader-election-role" not found
[INFO] Deleting CPFS leases in cp4aiops
[INFO] Deleting lease lease.coordination.k8s.io/2133ec8b.automation.ibm.com..
[INFO] Deleting lease lease.coordination.k8s.io/ibmevents-cluster-operator..
lease.coordination.k8s.io "ibmevents-cluster-operator" deleted
[INFO] Deleting lease lease.coordination.k8s.io/db9c8771.k8s.enterprisedb.io..
lease.coordination.k8s.io "db9c8771.k8s.enterprisedb.io" deleted
[INFO] Deleting CPFS configmaps in cp4aiops
[INFO] Deleting configmap configmap/ibm-cpp-config..
configmap "ibm-cpp-config" deleted
[INFO] Deleting configmap configmap/namespace-scope..
configmap "namespace-scope" deleted
[INFO] Deleting CPFS PVCs in cp4aiops
[INFO] Deleting PVC app.kubernetes.io/instance=iaf-system..
persistentvolumeclaim "data-iaf-system-kafka-0" deleted
persistentvolumeclaim "data-iaf-system-zookeeper-0" deleted
[INFO] Deleting PVC app=icp-mongodb..
No resources found
[INFO] Deleting Namespaced Bedrock Secrets
secret "platform-identity-management" deleted
secret "internal-tls-pkcs12" deleted
secret "identity-provider-secret" deleted
[INFO] Deleting Certmanager Resources
issuer.cert-manager.io "cs-ss-issuer" deleted
issuer.cert-manager.io "cs-ca-issuer" deleted
[INFO] Deleting certificate cs-ca-certificate in Installation Namespace
certificate.cert-manager.io "cs-ca-certificate" deleted
secret "cs-ca-certificate-secret" deleted
[INFO] Deleting other cert manager secrets in AIOPS install namespace
No resources found
lease.coordination.k8s.io "2a3e2c5f.ibm.com" deleted
[INFO] Checking for CP4AIOPS custom resources
[INFO] Resource Check Attempt: 1
aimanagermainprods.ai-manager.watson-aiops.ibm.com
No resources found
None Found

aimanagers.ai-manager.watson-aiops.ibm.com
No resources found
None Found

aimodeluis.consoleui.aiops.ibm.com
No resources found
None Found

aiopskafkatopics.connectors.aiops.ibm.com
No resources found
None Found

algorithms.ai-manager.watson-aiops.ibm.com
No resources found
None Found

applicationmanageragents.aiops.ibm.com
error: the server doesn't have a resource type "applicationmanageragents"
None Found

applicationmanagers.aiops.ibm.com
error: the server doesn't have a resource type "applicationmanagers"
None Found

applicationuis.consoleui.aiops.ibm.com
error: the server doesn't have a resource type "applicationuis"
None Found

baseuis.consoleui.aiops.ibm.com
No resources found
None Found

connectoruis.consoleui.aiops.ibm.com
No resources found
None Found
```

```
eventmanagergateways.ai-manager.watson-aiops.ibm.com
error: the server doesn't have a resource type "eventmanagergateways"
None Found

installations.orchestrator.aiops.ibm.com
No resources found
None Found

aiopsanalyticsorchestrators.ai.ir.aiops.ibm.com
No resources found
None Found

aiopsedges.connectors.aiops.ibm.com
No resources found
None Found

aiopsuis.consoleui.aiops.ibm.com
No resources found
None Found

automationactions.connectors.aiops.ibm.com
No resources found
None Found

bundlemanifests.connectors.aiops.ibm.com
No resources found
None Found

bundlemanifests.cp4waiops.management.ibm.com
error: the server doesn't have a resource type "bundlemanifests"
None Found

connectorcomponents.connectors.aiops.ibm.com
No resources found
None Found

connectorconfigurations.connectors.aiops.ibm.com
No resources found
None Found

connectorschemas.connectors.aiops.ibm.com
No resources found
None Found

dashboarduis.consoleui.aiops.ibm.com
error: the server doesn't have a resource type "dashboarduis"
None Found

gitapps.connectors.aiops.ibm.com
No resources found
None Found

insightsuis.consoleui.aiops.ibm.com
No resources found
None Found

issueresolutioncores.core.ir.aiops.ibm.com
No resources found
None Found

lifecycleservices.lifecycle.ir.aiops.ibm.com
No resources found
None Found

lifecycletriggers.lifecycle.ir.aiops.ibm.com
No resources found
None Found

modeltrains.modeltrain.ibm.com
error: the server doesn't have a resource type "modeltrains"
None Found

microedgeconfigurations.connectors.aiops.ibm.com
No resources found
None Found

flinkclusters.flink.automation.ibm.com
error: the server doesn't have a resource type "flinkclusters"
None Found

flinkdeployments.flink.ibm.com
No resources found
None Found

flinksessionjobs.flink.ibm.com
No resources found
None Found


[INFO] CRDs from CP4AIOPS are safe to be deleted
[INFO] Deleting the CP4AIOps Internal CRDs...
[INFO] Deleting CRD aimanagermainprods.ai-manager.watson-aiops.ibm.com..
```

```
customresourcedefinition.apiextensions.k8s.io "aimanagermainprods.ai-manager.watson-aiops.ibm.com" deleted
[INFO] Deleting CRD aimanagers.ai-manager.watson-aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "aimanagers.ai-manager.watson-aiops.ibm.com" deleted
[INFO] Deleting CRD aimodeluis.consoleui.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "aimodeluis.consoleui.aiops.ibm.com" deleted
[INFO] Deleting CRD aiopskafkatopics.connectors.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "aiopskafkatopics.connectors.aiops.ibm.com" deleted
[INFO] Deleting CRD algorithms.ai-manager.watson-aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "algorithms.ai-manager.watson-aiops.ibm.com" deleted
[INFO] Deleting CRD applicationmanageragents.aiops.ibm.com..
[INFO] Deleting CRD applicationmanagers.aiops.ibm.com..
[INFO] Deleting CRD applicationuis.consoleui.aiops.ibm.com..
[INFO] Deleting CRD baseuis.consoleui.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "baseuis.consoleui.aiops.ibm.com" deleted
[INFO] Deleting CRD connectoruis.consoleui.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "connectoruis.consoleui.aiops.ibm.com" deleted
[INFO] Deleting CRD eventmanagergateways.ai-manager.watson-aiops.ibm.com..
[INFO] Deleting CRD installations.orchestrator.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "installations.orchestrator.aiops.ibm.com" deleted
[INFO] Deleting CRD aiopsanalyticsorchestrators.ai.ir.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "aiopsanalyticsorchestrators.ai.ir.aiops.ibm.com" deleted
[INFO] Deleting CRD aiopsedges.connectors.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "aiopsedges.connectors.aiops.ibm.com" deleted
[INFO] Deleting CRD aiopsuis.consoleui.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "aiopsuis.consoleui.aiops.ibm.com" deleted
[INFO] Deleting CRD automationactions.connectors.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "automationactions.connectors.aiops.ibm.com" deleted
[INFO] Deleting CRD bundlemanifests.connectors.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "bundlemanifests.connectors.aiops.ibm.com" deleted
[INFO] Deleting CRD bundlemanifests.cp4waiops.management.ibm.com..
[INFO] Deleting CRD connectorcomponents.connectors.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "connectorcomponents.connectors.aiops.ibm.com" deleted
[INFO] Deleting CRD connectorconfigurations.connectors.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "connectorconfigurations.connectors.aiops.ibm.com" deleted
[INFO] Deleting CRD connectorschemas.connectors.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "connectorschemas.connectors.aiops.ibm.com" deleted
[INFO] Deleting CRD dashboarduis.consoleui.aiops.ibm.com..
[INFO] Deleting CRD gitapps.connectors.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "gitapps.connectors.aiops.ibm.com" deleted
[INFO] Deleting CRD insightsuis.consoleui.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "insightsuis.consoleui.aiops.ibm.com" deleted
[INFO] Deleting CRD issueresolutioncores.core.ir.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "issueresolutioncores.core.ir.aiops.ibm.com" deleted
[INFO] Deleting CRD lifecycleservices.lifecycle.ir.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "lifecycleservices.lifecycle.ir.aiops.ibm.com" deleted
[INFO] Deleting CRD lifecycletriggers.lifecycle.ir.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "lifecycletriggers.lifecycle.ir.aiops.ibm.com" deleted
[INFO] Deleting CRD modeltrains.modeltrain.ibm.com..
[INFO] Deleting CRD microedgeconfigurations.connectors.aiops.ibm.com..
customresourcedefinition.apiextensions.k8s.io "microedgeconfigurations.connectors.aiops.ibm.com" deleted
[INFO] Deleting CRD flinkclusters.flink.automation.ibm.com..
[INFO] Deleting CRD flinkdeployments.flink.ibm.com..
customresourcedefinition.apiextensions.k8s.io "flinkdeployments.flink.ibm.com" deleted
[INFO] Deleting CRD flinksessionjobs.flink.ibm.com..
customresourcedefinition.apiextensions.k8s.io "flinksessionjobs.flink.ibm.com" deleted
[INFO] Checking for CP4AIOPS Dependent custom resources
[INFO] Resource Check Attempt: 1
elasticsearchclusters.elasticsearch.opencontent.ibm.com
No resources found
None Found

rediscps.redis.ibm.com
No resources found
None Found

applications.tunnel.management.ibm.com
error: the server doesn't have a resource type "applications"
None Found

tunnels.sretooling.management.ibm.com
No resources found
None Found


[INFO] CRDs from CP4AIOPS Dependent are safe to be deleted
[INFO] Deleting the CP4AIops Dependent CRDs...
[INFO] Deleting CRD elasticsearchclusters.elasticsearch.opencontent.ibm.com..
customresourcedefinition.apiextensions.k8s.io "elasticsearchclusters.elasticsearch.opencontent.ibm.com" deleted
[INFO] Deleting CRD rediscps.redis.ibm.com..
customresourcedefinition.apiextensions.k8s.io "rediscps.redis.ibm.com" deleted
[INFO] Deleting CRD applications.tunnel.management.ibm.com..
[INFO] Deleting CRD tunnels.sretooling.management.ibm.com..
customresourcedefinition.apiextensions.k8s.io "tunnels.sretooling.management.ibm.com" deleted
No resources found
No resources found
[INFO] Checking if any additional ASM resources are on the cluster.
No resources found
No resources found
[INFO] No ASM resource instances were found outside the cp4aiops namespace, so the ASM CRDs can be deleted.
[INFO] Checking for ASM custom resources
[INFO] Resource Check Attempt: 1
asmformations.asm.ibm.com
```

```
No resources found
None Found

asms.asm.ibm.com
No resources found
None Found


[INFO] CRDs from ASM are safe to be deleted
[INFO] Deleting ASM CRDs...
[INFO] Deleting CRD asmformations.asm.ibm.com..
customresourcedefinition.apiextensions.k8s.io "asmformations.asm.ibm.com" deleted
[INFO] Deleting CRD asms.asm.ibm.com..
customresourcedefinition.apiextensions.k8s.io "asms.asm.ibm.com" deleted
[SUCCESS] ----Congratulations! IBM Cloud Pak for AIOps has been uninstalled!----
```

2. (Optional) Delete the IBM Cloud Pak for AIOps catalog

If you do not require the IBM Cloud Pak for AIOps catalog **`ibm-aiops-catalog`**, then remove it with one of the following methods:

Online deployments:

- Option 1: Delete the IBM Cloud Pak for AIOps catalog with the Red Hat OpenShift console
- Option 2: Delete the IBM Cloud Pak for AIOps catalog with the Red Hat OpenShift CLI

Air-gapped (offline) deployments:

- Option 3: Delete the IBM Cloud Pak for AIOps catalog in an air-gapped deployment

Option 1: Delete the IBM Cloud Pak for AIOps catalog with the Red Hat OpenShift console

1. In the OpenShift console, select **Administration** > **Cluster Settings**.
2. Select the **Configuration** tab, and then select **OperatorHub** from the **Configuration Resource** column.
3. Select the **Sources** tab.
4. Select the CatalogSource that you want to remove: **`ibm-aiops-catalog`**.
5. Select **Delete CatalogSource** from the **Actions** list (the three vertical dots at the end of the row).
6. Verify that the CatalogSource that you removed is not shown in **Administration** > **Cluster Settings** > **Configuration** > **OperatorHub** > **Sources**.

Option 2: Delete the IBM Cloud Pak for AIOps catalog with the Red Hat OpenShift CLI

1. Run the following command for the catalog that you want to remove.

   ```
   oc delete catalogsource -n openshift-marketplace <catalogsource>
   ```

   Where **`<catalogsource>`** is the name of the CatalogSource that you want to delete: **`ibm-aiops-catalog`**.

2. Verify that the following command doesn't return the catalog that you removed:

   ```
   oc get catalogsource -n openshift-marketplace
   ```

Option 3: Delete the IBM Cloud Pak for AIOps catalog in an air-gapped deployment

Note: If you have an air-gapped deployment of IBM Cloud Pak for AIOps that was initially installed at a version lower than 3.5, then follow the instructions for **`Option 3: Delete the AI Manager catalog in an upgraded air-gapped deployment`** in the IBM Cloud Pak for AIOps 3.5 documentation: Uninstalling IBM Cloud Pak® for AIOps IBM Cloud Pak for AIOps.

If you have an air-gapped deployment of IBM Cloud Pak for AIOps, then remove the catalog by using the **`uninstall-catalog`** case action.

1. Export the following environment variables.

   The local registry that you specify must be the registry in your air-gapped environment that you created when you installed Cloud Pak for AIOps.

   ```
   export CASE_NAME=ibm-cp-waiops
   export CASE_VERSION=1.14.0
   export CASE_INVENTORY_SETUP=cpwaiopsSetup
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   ```

2. Run the **`uninstall-catalog`** CASE action.

   ```
   oc ibm-pak launch $CASE_NAME \
   --version $CASE_VERSION \
   --action uninstall-catalog \
   --inventory $CASE_INVENTORY_SETUP \
   --namespace openshift-marketplace \
   --args "--registry $TARGET_REGISTRY --recursive \
   --inputDir ~/.ibm-pak/data/cases/$CASE_NAME/$CASE_VERSION"
   ```

3. Verify that the catalog that you removed is not returned by the following command:

```
oc get catalogsource -n openshift-marketplace
```

If the uninstallation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade to help you identify any problems.

3. (Optional) Delete Cert Manager

If you installed the IBM Cloud Pak® foundational services Cert Manager, and no other IBM Cloud Paks® are using it, then you can remove it.

1. Follow the instructions in Manual steps for uninstalling singleton services - Cert Manager in the IBM Cloud Pak foundational services documentation.
2. If you have an air-gapped deployment, remove the IBM Cloud Pak foundational services Cert Manager catalog.

```
oc delete catsrc -n openshift-marketplace "ibm-cert-manager-catalog"  --ignore-not-found
```

4. (Optional) Delete the License Service

If you installed the IBM Cloud Pak foundational services License Service, and no other IBM Cloud Paks® are using it, then you can remove it.

1. Follow the instructions in Uninstalling License Service in the IBM Cloud Pak foundational services documentation.
2. If you have an air-gapped deployment, remove the IBM Cloud Pak foundational services License Service catalog.

```
oc delete catsrc -n openshift-marketplace "ibm-licensing-catalog" --ignore-not-found
```

Infrastructure Automation

To learn more about installing and using Infrastructure Automation, see the following topics in the documentation section for Infrastructure Automation.

- Infrastructure automation
  - Planning
  - Installing
  - Upgrading
  - Administering
  - Infrastructure Management
  - Uninstalling

Defining integrations

You can use the IBM Cloud Pak for AIOps console to define integrations and applications.

The integrations and applications that you create serve as the basis for:

- Incoming information that is used to model your IT infrastructure
- Observers that monitor your virtual environments
- Outgoing information that notifies you of any issues

IBM Cloud Pak for AIOps includes many available pre-built integrations that you can configure. If the available integrations do not meet directly meet your requirements, you can use the custom, webhook, and REST integrations to help you create your own integration, such as to integrate with other third-party systems (Blazemeter, Thousand eyes, Catchpoint). In addition you can create custom integrations using the integration SDK.

For more information about defining different types of available integrations in IBM Cloud Pak for AIOps, see the following topics:

- Planning data loading, checking and training
- Mapping data
- Configuring Integrations
- Defining observer jobs

For more informaiton about using the custom integration capabilities, see:

- Creating custom integrations (for log integrations)
- Creating Kafka integrations (for log, event integrations)
- Creating Generic Webhook integrations (for event integrations)
- Creating REST integrations (for topology, resource, application data integrations)
- Creating custom integrations using the integration SDK.

Available integrations

Use this information to see all the integrations that are available.

IBM Cloud Pak for AIOps provides many connectors, probes, and observers by default for setting up integrations with common IBM and third-party systems and services. In addition, if a default integration is not available for connecting to a system or service that you want to use with IBM Cloud Pak for AIOps, you can set up a custom integration to connect to that system or service. For more information about setting up a custom integration, see Creating custom integrations

Note: Each integration that you configure requires an additional 1 vCPU and 3GB memory. For example, if you configure five integrations then you will require an additional 5 vCPUs and 15GB memory.

All available integrations

Click the integration name to learn more about creating, editing, and deleting a particular integration, or about defining an observer job.

Available integration types

Integration / Link	Class	Category
Alcatel-Lucent 1300 XMC	Probe	Events
Alcatel-Lucent 5529 OAD v6	Probe	Events
Alcatel-Lucent 5620 Logfile	Probe	Events
Alcatel-Lucent 5620 SAM 3GGP v8	Probe	Events
Alcatel-Lucent 5620 SAM v13 JMS	Probe	Events
Alcatel-Lucent 5ESS	Probe	Events
Alcatel-Lucent 9353 WNMS	Probe	Events
Alcatel-Lucent DSC DEX	Probe	Events
Alcatel-Lucent ECP	Probe	Events
Alcatel-Lucent ITM-NM/OMS	Probe	Events
Alcatel-Lucent ITM-SC	Probe	Events
Alcatel-Lucent OMC-R	Probe	Events
Alcatel-Lucent OS-OS	Probe	Events
ALM	Observer	Application, Topology
Ansible AWX	Observer	Application, Topology
Ansible	Integration	Runbooks, Remote Access
Apache Pulsar	Probe	Events
Appdynamics	Integration	Metrics
AppDynamics	Observer	Application, Topology
Avaya Definity G3	Probe	Events
AWS	Observer	Application, Topology
AWS	Probe	Events
AWSCloudWatch	Integration	Metrics
Azure	Observer	Application, Topology
Big Cloud Fabric	Observer	Application, Topology
BigFix Inventory	Observer	Application, Topology
BMC Patrol V9	Probe	Events
BMC Patrol	Probe	Events
CA Spectrum V9 (CORBA)	Probe	Events
CA Spectrum V9.4 (CORBA)	Probe	Events
Ciena Blue Planet	Observer	Application, Topology
Ciena Blue Planet	Probe	Events
Cisco ACI	Observer	Application, Topology
Cisco APIC	Probe	Events
Cisco EPNM	Probe	Events
Cisco Transport Manager 9.0 (CORBA)	Probe	Events
Comverse TRILOGUE INfinity	Probe	Events
Contrail	Observer	Application, Topology
Custom	Integration	Logs
Dantel PointMaster	Probe	Events
Datadog	Observer	Application, Topology
DNS	Observer	Application, Topology
Docker	Observer	Application, Topology
Dynatrace	Integration	Metrics
Dynatrace	Observer	Application, Topology
ECI eNM	Probe	Events
ELK	Integration	Logs
Email	Probe	Events
Exec	Probe	Events
FalconLogScale	Integration	Logs
FIFO	Probe	Events
File	Observer	Application, Topology
Genband IEMS	Probe	Events
Generic 3GPP	Probe	Events
Generic Log File Java	Probe	Events
Generic Multi-Technology Operations Systems Interface (MTOSI)	Probe	Events
Generic TMF814	Probe	Events

Integration / Link	Class	Category
GitHub	Integration	Ticketing (Incident tickets)
Glenayre VMS	Probe	Events
GoogleCloud	Observer	Application, Topology
Heartbeat	Probe	Events
HP NFVD	Observer	Application, Topology
HP Operations Manager	Probe	Events
HPE Operations Manager i	Probe	Events
HTTP Server Error Log	Probe	Events
Huawei M2000 MML	Probe	Events
Huawei U2000 (CORBA)	Probe	Events
Huawei U2000 3GPP (CORBA)	Probe	Events
Huawei U2000	Probe	Events
IBM Cloud Platform Common Services Monitoring	Probe	Events
IBM Cloud	Observer	Application, Topology
IBM Event Streams for IBM Cloud	Probe	Events
IBM SevOne Network Performance Management (NPM)	Probe	Events
IBM Tivoli Common Event Infrastructure (CEI)	Probe	Events
IBM Tivoli EIF	Probe	Events
IBM Tivoli NetView	Probe	Events
IBM Turbonomic	Probe	Events
IBM WebSphere MQ	Probe	Events
iDirect Pulse	Probe	Events
IEC CIM Advanced Metering Infrastructure	Probe	Events
Instana	Integration	Events, Metrics, Topology
Itron OpenWay Collection Engine (OWCE) EMS	Probe	Events
Java Socket	Probe	Events
JDBC	Probe	Events
Jenkins	Observer	Application, Topology
Jira	Integration	Ticketing (Incident tickets)
Juniper Contrail Alerts	Probe	Events
Juniper Contrail	Probe	Events
Juniper CSO	Observer	Application, Topology
Kafka	Integration	Logs, Events
Kodiak EMS (CORBA)	Probe	Events
Kubernetes	Observer	Application, Topology
Message Bus	Probe	Events
Mezmo	Integration	Logs
Microfocus (HP) Network Node Manager-i	Probe	Events
Microsoft Azure	Probe	Events
Microsoft EWS	Probe	Events
Microsoft SCOM 2012	Probe	Events
Microsoft SCOM 2016	Probe	Events
Microsoft Windows Event Log	Probe	Events
MicrosoftTeams	Integration	ChatOps
Motorola MUNO	Probe	Events
Motorola OMC-R (iDEN)	Probe	Events
Motorola OMC-R GSR9 (CORBA)	Probe	Events
Netcool Impact	Integration	Events
Netcool Operations Insight ObjectServer	Integration	Events
Network Manager	Observer	Application, Topology
New Relic	Observer	Application, Topology
NewRelic	Integration	Metrics
Nokia Network Functions Manager for Packet	Probe	Events
Nokia Network Services Platform	Probe	Events
Nokia-Siemens EWSD (Server)	Probe	Events
Nokia-Siemens EWSD (UNIX)	Probe	Events
Nokia-Siemens NetAct 3GPP V6 (CORBA)	Probe	Events
Nokia-Siemens SMNOS	Probe	Events
Nokia-Siemens Switch/Radio/@vantage Commander (CORBA) 3GPP V6.3	Probe	Events
Nokia-Siemens TNMS-SNMP Interface	Probe	Events
Nortel DMS	Probe	Events
Nortel EIF	Probe	Events
Nortel Magellan NMS	Probe	Events
Nortel Meridian	Probe	Events

Integration / Link	Class	Category
Nortel Preside EAI	Probe	Events
OPC Unified Architecture (UA)	Probe	Events
Open NerveCenter	Probe	Events
OpenStack	Observer	Application, Topology
PagerDuty	Integration	Events
PDS Snyder	Probe	Events
Ping	Probe	Events
Rancher	Observer	Application, Topology
REST	Observer	Application, Topology
Samsung Base Station Manager	Probe	Events
SDC ONAP	Observer	Application, Topology
ServiceNow	Observer	Application, Topology
ServiceNow integration	Integration	Ticketing (Change tickets, Incident tickets)
SevOne	Observer	Application, Topology
Simnet	Probe	Events
Sitescope	Probe	Events
Slack	Integration	ChatOps
SNMP (Writer)	Probe	Events
Splunk	Integration	Logs
SSH	Integration	Runbooks, Remote Access
Standard Input	Probe	Events
Syslog	Probe	Events
Syslogd	Probe	Events
Tandem SCP	Probe	Events
Tandem	Probe	Events
Tekelec Eagle STP	Probe	Events
Tekelec LSMS	Probe	Events
Tellabs 8000 INM	Probe	Events
Tellabs FOCUS NM2100	Probe	Events
Tivoli Application Dependency Discovery Manager	Observer	Application, Topology
TL1	Probe	Events
Viptela	Observer	Application, Topology
VMware NSX	Observer	Application, Topology
VMware vCenter	Observer	Application, Topology
Zabbix	Integration	Metrics
Zabbix	Observer	Application, Topology
ZTE M31 NetNumen WCDMA and U31 Wireless (CORBA)	Probe	Events
ZTE NetNumen N31 (CORBA)	Probe	Events
ZTE ZXONM E300 and U31 Wired (CORBA)	Probe	Events

Integrations by category

- Application integrations
- ChatOps integrations
- Event integrations
- Log integrations
- Metric integrations
- Remote access integrations
- Runbook integrations
- Ticketing integrations
- Topology integrations

Application integrations

Click the integration name to learn more about defining an integration for collecting application data. Many of these integrations can also be used to collect topology data.

- ALM
- Ansible AWX
- AppDynamics
- AWS
- Azure
- Big Cloud Fabric
- BigFix Inventory
- Ciena Blue Planet
- Cisco ACI
- Contrail
- Datadog

- DNS
- Docker
- Dynatrace
- File
- GoogleCloud
- HP NFVD
- IBM Cloud
- Jenkins
- Juniper CSO
- Kubernetes
- Network Manager
- New Relic
- OpenStack
- Rancher
- REST
- SDC ONAP
- ServiceNow
- SevOne
- Tivoli Application Dependency Discovery Manager
- Viptela
- VMware NSX
- VMware vCenter
- Zabbix

ChatOps integrations

- MicrosoftTeams
- Slack

Event integrations

- Alcatel-Lucent 1300 XMC
- Alcatel-Lucent 5529 OAD v6
- Alcatel-Lucent 5620 Logfile
- Alcatel-Lucent 5620 SAM 3GGP v8
- Alcatel-Lucent 5620 SAM v13 JMS
- Alcatel-Lucent 5ESS
- Alcatel-Lucent 9353 WNMS
- Alcatel-Lucent DSC DEX
- Alcatel-Lucent ECP
- Alcatel-Lucent ITM-NM/OMS
- Alcatel-Lucent ITM-SC
- Alcatel-Lucent OMC-R
- Alcatel-Lucent OS-OS
- Apache Pulsar
- Avaya Definity G3
- AWS
- BMC Patrol V9
- BMC Patrol
- CA Spectrum V9 (CORBA)
- CA Spectrum V9.4 (CORBA)
- Ciena Blue Planet
- Cisco APIC
- Cisco EPNM
- Cisco Transport Manager 9.0 (CORBA)
- Comverse TRILOGUE INfinity
- Dantel PointMaster
- ECI eNM
- Email
- Exec
- FIFO
- Genband IEMS
- Generic 3GPP
- Generic Log File Java
- Generic Multi-Technology Operations Systems Interface (MTOSI)
- Generic TMF814
- Glenayre VMS
- Heartbeat
- HP Operations Manager
- HPE Operations Manager i
- HTTP Server Error Log
- Huawei M2000 MML
- Huawei U2000 (CORBA)
- Huawei U2000 3GPP (CORBA)
- Huawei U2000

- Kafka
- IBM Cloud Platform Common Services Monitoring
- IBM Event Streams for IBM Cloud
- IBM SevOne Network Performance Management (NPM)
- IBM Tivoli Common Event Infrastructure (CEI)
- IBM Tivoli EIF
- IBM Tivoli NetView
- IBM Turbonomic
- IBM WebSphere MQ
- iDirect Pulse
- Instana
- IEC CIM Advanced Metering Infrastructure
- Itron OpenWay Collection Engine (OWCE) EMS
- Java Socket
- JDBC
- Juniper Contrail Alerts
- Juniper Contrail
- Kodiak EMS (CORBA)
- Kubernetes
- Message Bus
- Microfocus (HP) Network Node Manager-i
- Microsoft Azure
- Microsoft EWS
- Microsoft SCOM 2012
- Microsoft SCOM 2016
- Microsoft Windows Event Log
- Motorola MUNO
- Motorola OMC-R (iDEN)
- Motorola OMC-R GSR9 (CORBA)
- Netcool Impact
- Netcool Operations Insight ObjectServer
- Nokia Network Functions Manager for Packet
- Nokia Network Services Platform
- Nokia-Siemens EWSD (Server)
- Nokia-Siemens EWSD (UNIX)
- Nokia-Siemens NetAct 3GPP V6 (CORBA)
- Nokia-Siemens SMNOS
- Nokia-Siemens Switch/Radio/@vantage Commander (CORBA) 3GPP V6.3
- Nokia-Siemens TNMS-SNMP Interface
- Nortel DMS
- Nortel EIF
- Nortel Magellan NMS
- Nortel Meridian
- Nortel Preside EAI
- OPC Unified Architecture (UA)
- Open NerveCenter
- PagerDuty
- PDS Snyder
- Ping
- Samsung Base Station Manager
- Simnet
- Sitescope
- SNMP (Writer)
- Standard Input
- Syslog
- Syslogd
- Tandem SCP
- Tandem
- Tekelec Eagle STP
- Tekelec LSMS
- Tellabs 8000 INM
- Tellabs FOCUS NM2100
- TL1
- ZTE M31 NetNumen WCDMA and U31 Wireless (CORBA)
- ZTE NetNumen N31 (CORBA)
- ZTE ZXONM E300 and U31 Wired (CORBA)

Log integrations

- Custom
- ELK
- FalconLogScale
- Kafka
- Mezmo
- Splunk

Metric integrations

- Appdynamics
- AWSCloudWatch
- Dynatrace
- Instana
- NewRelic
- Zabbix

Remote access integrations

- Ansible
- SSH

Runbook integrations

- Ansible
- SSH

Ticketing integrations

Click the integration name to learn more about creating, editing, and deleting a particular ticketing integration.

- GitHub (Incident tickets)
- Jira (Incident tickets)
- ServiceNow integration (Change tickets, Incident tickets)

Topology integrations

Click the integration name to learn more about defining an integration for collecting topology data. Many of these integrations can also be used to collect application data.

- ALM
- Ansible AWX
- AppDynamics
- AWS
- Azure
- Big Cloud Fabric
- BigFix Inventory
- Ciena Blue Planet
- Cisco ACI
- Contrail
- Datadog
- DNS
- Docker
- Dynatrace
- File
- GoogleCloud
- HP NFVD
- IBM Cloud
- Instana
- Jenkins
- Juniper CSO
- Kubernetes
- Network Manager
- New Relic
- OpenStack
- Rancher
- REST
- SDC ONAP
- ServiceNow
- SevOne
- Tivoli Application Dependency Discovery Manager
- Viptela
- VMware NSX
- VMware vCenter
- Zabbix

Integrations by class

- General (connector) integrations
- Event Manager (probe) integrations
- Observer integrations

- Custom integrations

Netcool integrations

IBM Cloud Pak for AIOps can integrate with IBM Tivoli Netcool/OMNIbus and IBM Tivoli Netcool/Impact to leverage existing event data (IBM Tivoli Netcool/OMNIbus), and policies (IBM Tivoli Netcool/Impact).

With these integrations in place IBM Cloud Pak for AIOps provides AI and visualization features in a modernized UI with an IBM Tivoli Netcool/OMNIbus integration providing additional data and an IBM Tivoli Netcool/Impact integration providing additional automation. You can continue to take advantage of all of your alerts and integrations for IBM Tivoli Netcool/OMNIbus and IBM Tivoli Netcool/Impact to obtain and send data.

Note: Each Netcool integration that you configure requires an additional 1 vCPU and 3GB memory. For example, if you configure three Netcool integrations then you will require an additional 2 vCPUs and 6GB memory.

Integrating with IBM Tivoli Netcool/OMNIbus

You can integrate IBM Cloud Pak for AIOps with an existing IBM Tivoli Netcool/OMNIbus deployment to send event data through the integration to IBM Cloud Pak for AIOps. With this integration, you can see IBM Tivoli Netcool/OMNIbus event data in the IBM Cloud Pak for AIOps Alert Viewer. These events are converted into alerts on the IBM Cloud Pak for AIOps side.

You can then take advantage of the IBM Cloud Pak for AIOps AI features to manage and visualize the alerts. For example, event data can be correlated into incidents, data can be viewed and analyzed in a topology, and runbooks can be used to automate operations for managing the alerts.

To integrate IBM Cloud Pak for AIOps with IBM Tivoli Netcool/OMNIbus, you need to create an IBM Netcool Operations Insight ObjectServer integration with the IBM Cloud Pak for AIOps UI Integrations page. This integration defines the settings for communication from IBM Cloud Pak for AIOps to IBM Tivoli Netcool/OMNIbus. For more information about how to define this integration, see Creating IBM Netcool Operations Insight ObjectServer integrations.

After you integrate with IBM Tivoli Netcool/OMNIbus, use IBM Cloud Pak for AIOps and the IBM Cloud Pak for AIOps UI for AI and visualization features. Use IBM Tivoli Netcool/OMNIbus and the associated UI tools for obtaining and sending data to IBM Cloud Pak for AIOps.

Note: If you have integrated IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus, and you are not yet ready to move to using the IBM Cloud Pak for AIOps UI, you can continue to use the IBM Tivoli Netcool/OMNIbus DASH custom dashboards and Web GUI event viewer as you transition to using the IBM Cloud Pak for AIOps UI. If you need to continue to use IBM Tivoli Netcool/OMNIbus tools, you can configure launch-in-context links from the Web GUI Event Viewer and the main DASH navigation to the IBM Cloud Pak for AIOps UI to help users access and use the IBM Cloud Pak for AIOps UI.

Integrating with IBM Tivoli Netcool/Impact

You can integrate IBM Cloud Pak for AIOps with an existing IBM Tivoli Netcool/Impact deployment so that IBM Cloud Pak for AIOps can then run automation policies that you manage with IBM Tivoli Netcool/Impact. New and existing policies can continue to be used to run operations in IBM Tivoli Netcool/Impact for data retrieval, drive automation, create tickets, leverage other integrations with IBM Tivoli Netcool/Impact, and more. These policies can be invoked by IBM Cloud Pak for AIOps or send events to IBM Cloud Pak for AIOps.

IBM Tivoli Netcool/Impact policies can reference the extension framework API for IBM Cloud Pak for AIOps. With these policies, you can write automation processes in IBM Tivoli Netcool/Impact to complete operations in IBM Cloud Pak for AIOps, such as to query alerts and incidents, and request changes to alerts.

In some cases, an event source might have an API through which it can retrieve the events. In this case, IBM Tivoli Netcool/Impact can be employed to retrieve these events. IBM Tivoli Netcool/Impact is commonly used to retrieve data from third-party systems for the purposes of event enrichment. The IBM Tivoli Netcool/Impact policy **`GetHTTP`** function is key for interacting with APIs. For more information about IBM Tivoli Netcool/Impact functions, see List and overview of functions.

To create this integration between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/Impact, you need to first create an IBM Tivoli Netcool/Impact integration with the IBM Cloud Pak for AIOps UI Integrations page. This integration defines the settings for communication from IBM Cloud Pak for AIOps to IBM Tivoli Netcool/Impact. For more information about how to define this integration, see Creating IBM Tivoli Netcool/Impact integrations.

Then to complete the integration, you need to edit the RESTful DSA configuration in IBM Tivoli Netcool/Impact. This configuration defines the settings for communicating from IBM Tivoli Netcool/Impact to IBM Cloud Pak for AIOps. For more information about editing the RESTful DSA configuration, see Configuring an integration in IBM Tivoli Netcool/Impact for IBM Cloud Pak for AIOps.

Key event integrations

IBM Cloud Pak for AIOps deployments might include requirements to integrate with third-party systems, such as a source of information for event enrichment or for ticketing purposes. You can use the following two ways to achieve key event integrations:

- **Events in ObjectServer**: If the nature of the integration is a simple or a one-time action, events can be picked up by IBM Tivoli Netcool/Impact directly from the IBM Tivoli Netcool/OMNIbus ObjectServer. Any updates to the event, such as custom field enrichment or a ticket number can be written back to the event. These updates are propagated to IBM Cloud Pak for AIOps by using the Netcool Connector.
- **Alerts in IBM Cloud Pak for AIOps**: An alternative approach is to create an IBM Cloud Pak for AIOps automation policy that sends the event to IBM Tivoli Netcool/Impact by using the IBM Tivoli Netcool/Impact integration under the preferred criteria. IBM Tivoli Netcool/Impact policy can write any updates or enrichments back to the IBM Tivoli Netcool/OMNIbus ObjectServer or back to IBM Cloud Pak for AIOps directly.

Related links

Migrating Netcool correlations to IBM Cloud Pak for AIOps

Creating IBM Tivoli Netcool/Impact integrations

You can integrate your existing IBM Tivoli Netcool/Impact policies with IBM Cloud Pak for AIOps through a IBM Tivoli Netcool/Impact integration. This integration connects the IBM Tivoli Netcool/Impact cluster with IBM Cloud Pak for AIOps.

- About this integration
- Before you begin
- Creating IBM Tivoli Netcool/Impact integrations
- Editing IBM Tivoli Netcool/Impact integrations
- Deleting IBM Tivoli Netcool/Impact integrations
- Enriching an incident by using IBM Tivoli Netcool/Impact policy

About this integration

By connecting with IBM Tivoli Netcool/Impact, IBM Cloud Pak for AIOps can then leverage your existing IBM Tivoli Netcool/Impact deployment to run automation policies that you manage with IBM Tivoli Netcool/Impact. New policies that you create in IBM Tivoli Netcool/Impact can reference the extension framework API for IBM Cloud Pak for AIOps. With these policies, you can write automation processes in IBM Tivoli Netcool/Impact to complete operations in IBM Cloud Pak for AIOps, such as to query alerts and incidents, and request changes to alerts.

Note: You can create multiple integrations between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/Impact. Each integration must point to a unique Impact endpoint.

When you create the integration in IBM Cloud Pak for AIOps to IBM Tivoli Netcool/Impact, the integration automatically creates the RESTful DSA data source in IBM Tivoli Netcool/Impact. To complete the integration with IBM Tivoli Netcool/Impact integration after creating the integration in IBM Cloud Pak for AIOps, you need to edit this RESTful DSA data source in IBM Tivoli Netcool/Impact to define your authentication settings for communicating with IBM Cloud Pak for AIOps API. After editing and verifying the authentication settings, you can begin to create and run your automation policies from IBM Tivoli Netcool/Impact.

Note: If the IBM Tivoli Netcool/Impact cluster uses a non-default cluster name ("NCICLUSTER"), the integration might fail to validate the integration. For more information, see Impact integration fails for Impact server with non-default cluster name.

Before you begin

- Ensure that you have IBM Tivoli Netcool/Impact Version 7.1 with fix pack level 33 or later and IBM Cloud Pak for AIOps 4.8.0 installed.
- Obtain the following details for your IBM Tivoli Netcool/Impact deployment:
  - The endpoint for the GUI server (example: https://noi-impact.fyre.ibm.com:16311)
  - The endpoint(s) for the primary Impact server and any secondary servers (example: https://noi-impact.fyre.ibm.com:9081)
  - If Impact is deployed with Event Manager on OpenShift, you can use the `oc describe noi` command to retrieve the endpoints.
  - The username and password for the IBM Tivoli Netcool/Impact administrator (example: impactadmin) for connecting to the IBM Tivoli Netcool/Impact endpoint.
  - The set of CA certificates for IBM Tivoli Netcool/Impact if the endpoint is secured by SSL/TLS. If self-signed this means all the certificates from all the servers.

    You can extract these certificates from your IBM Tivoli Netcool/Impact deployment with `openssl`:

    - For the GUI server (port 16311)

      ```
      openssl s_client -showcerts -servername noi-impact.mycluster.com -connect noi-impact.mycluster.com:16311
      </dev/null
      ```

    - For the backend server (port 9081)

      ```
      openssl s_client -showcerts -servername noi-impact.mycluster.com -connect noi-impact.mycluster.com:9081
      </dev/null
      ```

    To combine multiple certificates (there is a minimum requirement of two) in the integration settings, you can list them sequentially in no specific order. The combined certificates resembles the following structure:

    ```
    -----BEGIN CERTIFICATE-----
    ABCDEfGHIjklMnOPQrSTUVWXYZABCdefghiJ1k2LMNoPQRSTUVwxYZABCDEFGhij
    ... Server 1 ...
    r+wGXNPwrw==
    -----END CERTIFICATE-----
    -----BEGIN CERTIFICATE-----
    ABCDEfGHIjklMnOPQrSTUVWXYZABCdefghiJ1k2LMNoPQRSTUVwxYZABCDEFGhij
    ... Server 2 ...
    ZYXWVUTSRqpoNnLKJiHGFEDCBAZYXwvutsrQ1p4ONMlKJIHGFEdcBAZYXWVUTsrq
    -----END CERTIFICATE-----
    ```

Creating an IBM Tivoli Netcool/Impact integration

To create an IBM Tivoli Netcool/Impact integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.

2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **IBM Tivoli Netcool/Impact** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section (Events).
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: The display name of your integration. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
   - **GUI Server URLs of Impact endpoint**: Enter the GUI URL or URLs for the endpoint. For high availability environments with multiple URLs, enter the URLs as a comma-separated list. For example, `https://impact-1, https://impact-2`.
   - **Backend Server URLs of Impact endpoint**: Enter the backend URL or URLs for the endpoint. For high availability environments with multiple URLs, enter the URLs as a comma-separated list. For example, `https://impact-1, https://impact-2`.
   - **User ID**: Enter the user ID for the IBM Tivoli Netcool/Impact administrator (example: impactadmin) for connecting to the IBM Tivoli Netcool/Impact endpoint.
   - **Password**: Enter the password for the IBM Tivoli Netcool/Impact administrator for connecting to the IBM Tivoli Netcool/Impact endpoint.
   - **Impact Certificates**: Enter the valid set of CA certificates for IBM Tivoli Netcool/Impact to use for verifying the SSL/TLS connection to the REST services for IBM Tivoli Netcool/Impact.
   - **Deployment options**: This is where the integration connector is deployed, not the location of IBM Tivoli Netcool/Impact. Select one of the options, **Local** or **Remote**. **Local** deploys the integration in the same cluster and project (namespace) where IBM Cloud Pak for AIOps is installed. Alternatively, **Remote** deploys the integration in your chosen location, such as a different network region, on SaaS, or a remote on-premises VM, cluster, or container.
7. Click **Done**.
8. If you selected **Remote**, you are presented with an *Install Cloud Pak for AIOps remote integration* window.
   1. Click **Download shell script** and copy the bootstrap script to a remote VM.

      The bootstrap script requires Podman to run. You can either run it as root or non-root.

      Note: To run Podman as non-root, see the Podman installation.
   2. Run the bootstrap script on the remote VM to complete the deployment.
9. You are redirected to the *IBM Tivoli Netcool/Impact* page for the integration.

   Your new integration is listed in the table on the page. The integration has an intial status of *Initializing*. After a few minutes, the status changes to *Running* if your integration is successful. With your integration successful and running, you now need to configure the integration in IBM Tivoli Netcool/Impact.
10. Configure the integration in IBM Tivoli Netcool/Impact.

To complete the integration, you need to edit the RESTful DSA data source in IBM Tivoli Netcool/Impact.

Editing an IBM Tivoli Netcool/Impact integration

After you create your integration, you can edit the integration. For example, if you want to add or change the listed GUI or backend URLs for the Impact endpoint, or to update the User ID or password. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Tivoli Netcool/Impact** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *IBM Tivoli Netcool/Impact* page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited.

Deleting an IBM Tivoli Netcool/Impact integration

If you no longer need your integration and want to delete it entirely, you can delete the integration from the console.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.

3. Click the **IBM Tivoli Netcool/Impact** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *IBM Tivoli Netcool/Impact* integrations page, click the *Delete* icon for the integration that you want to delete.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

   Your integration is deleted in IBM Cloud Pak for AIOps. To fully remove the integration, log into the IBM Tivoli Netcool/Impact GUI and delete the IBM Cloud Pak for AIOps data source and policies from IBM Tivoli Netcool/Impact.

Enriching an incident by using IBM Tivoli Netcool/Impact policy

Cloud Pak for AIOps ingests data from many sources and is a central source of truth for a site reliability engineer (SRE) to manage all their disparate systems. You might need to automatically enrich Cloud Pak for AIOps incidents with data from external systems, such as state, ticket number or priority. The additional data gives more insight for an SRE to manage and resolve an incident.

To enrich incidents with data from external systems, use the following steps:

1. Create IBM Tivoli Netcool/Impact policy, which uses the Cloud Pak for AIOps API to enrich the incidents.
2. Create Cloud Pak for AIOps policy to send new incidents to IBM Tivoli Netcool/Impact and invoke the IBM Tivoli Netcool/Impact policy.

Create an IBM Tivoli Netcool/Impact policy to enrich the incidents

From IBM Tivoli Netcool/Impact console, create a JavaScript policy. The code is provided in the following example:

```
Log(
  'Enriching Incident(' + EventContainer.incident.id +
  ') with ticket number from external ITSM system'
);

// This is the DSA created when you create the Netcool/Impact integration with AIOps
var dsa = 'AIOps-jaunpuri1-f243ea05-c344-4d51-9e47-ec7e5031dfea';

// The experimental API being utilised
var AIOPS_INCIDENT_PATH = 'api/issue-resolution/v1alpha1/incidents/';

// The incoming Incident ID from AIOps
var incidentId = EventContainer.incident.id;
var path = AIOPS_INCIDENT_PATH + incidentId;

// In practice this would be a function call to get the URL from the external ITSM system
var ticketLink='https://github.com/oneill/myproject/issues/1489';

// Like ticketLink, this would be a function call. Hardcoded here to simplify
var ticketNumber = '1489';

// Like ticketLink. Hardcoded here to simplify.
var priority = 2;

var incidentUpdateRequest = {
  priority: priority,
  insights: [
    {
      id: String(JavaCall('java.util.UUID', null, 'randomUUID', [])),
      type: 'aiops.ibm.com/insight-type/itsm/metadata',
      source: 'myGithubProject',
      details: {
        id: String(JavaCall('java.util.UUID', null, 'randomUUID', [])),
        name: 'Github',
        type: 'aiops.ibm.com/insight-type/metadata',
        permalink: ticketLink,
        ticket_num: ticketNumber
      }
    }
  ]
};

// Call the experimental API to enrich the AIOps incident
result = RESTfulAPIPATCH(dsa, path, {
  Data: JSON.stringify(incidentUpdateRequest)
});

Log('Enrich Result: ' + result);
if (result.status > 202) {
  throw new Error(
    'Error ' + result.status + ' patching incident ' + id + '\n' +
    result.statusText
  );
}
```

Note: A few values in the preceding code are hardcoded. In actual implementation, your business logic needs to be coded to generate an external ticket, then retrieving the URL and ticket number from an external system. These details can then be effectively incorporated into your policy.

Create Cloud Pak for AIOps policy

Create Cloud Pak for AIOps policy to invoke your IBM Tivoli Netcool/Impact when new incidents are created in Cloud Pak for AIOps.

Policy triggers and policy actions within Cloud Pak for AIOps can resemble the following examples:

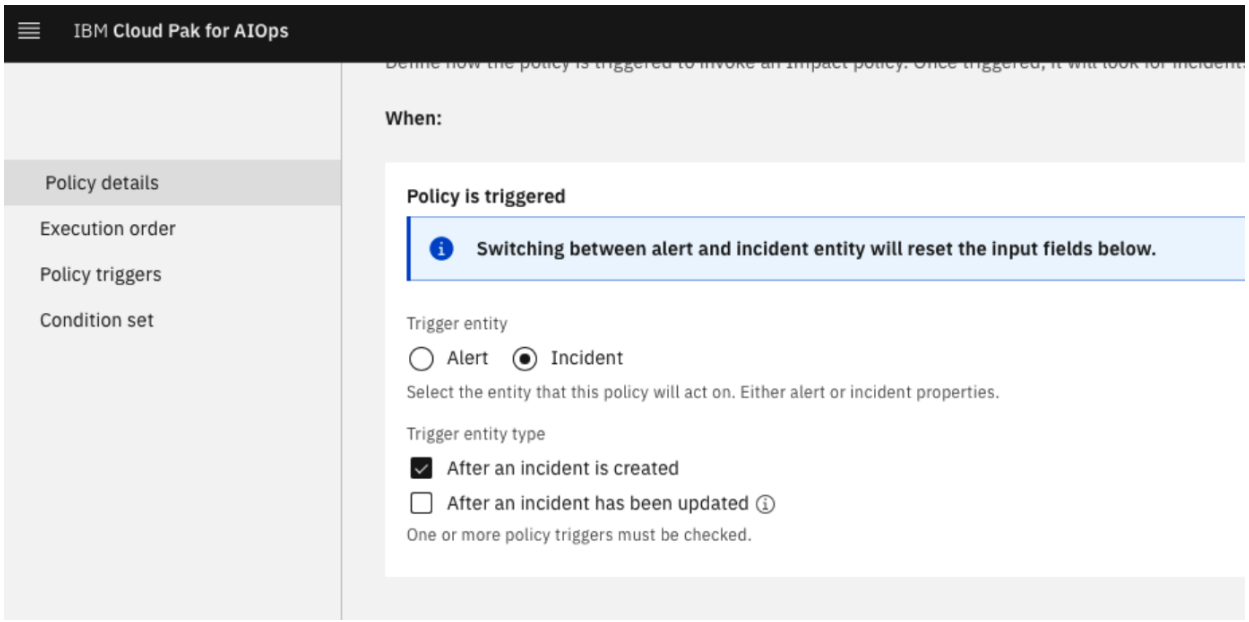


Figure. Policy triggers

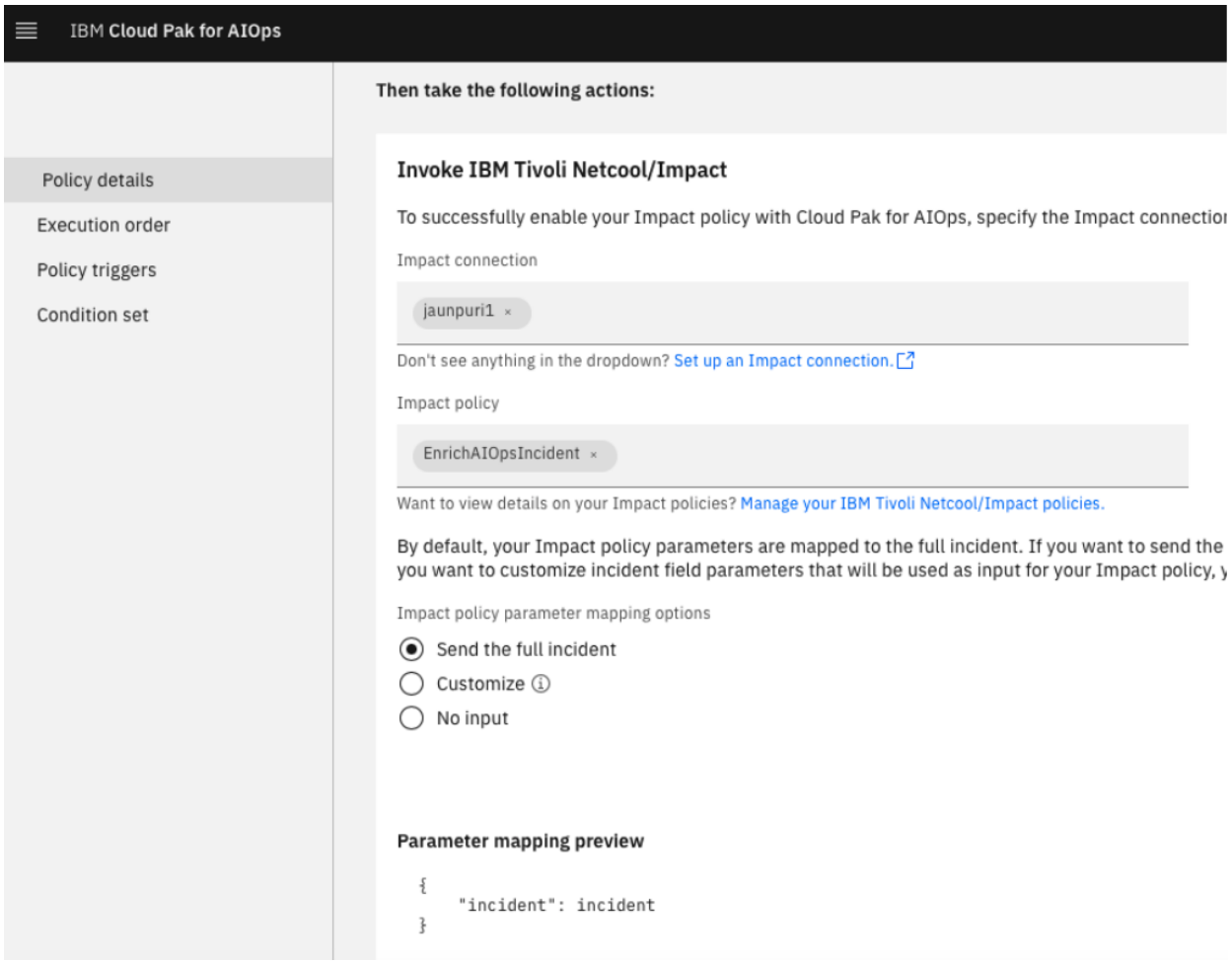


Figure. Policy actions

When new incidents are created in Cloud Pak for AIOps, the incidents automatically invoke your IBM Tivoli Netcool/Impact policy and the incident is updated with the ticket number and any other edits you decided to make to state, priority, or owner.

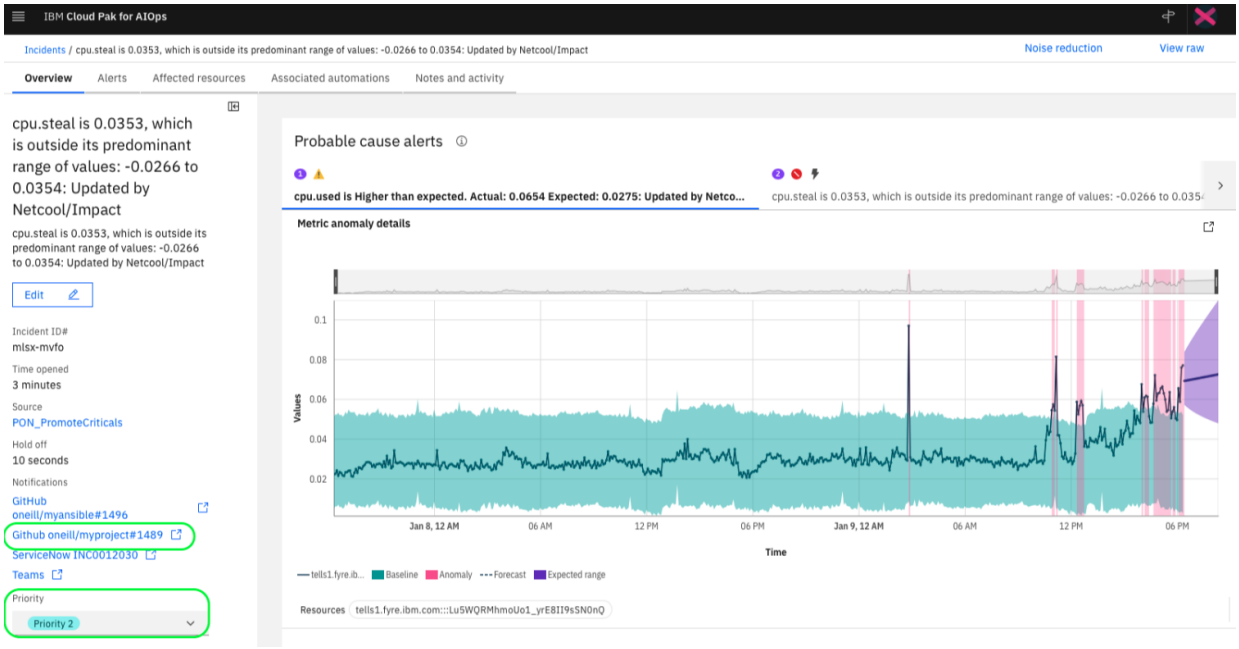


Figure. Incident overview

Note: You can also create further Cloud Pak for AIOps policies to invoke extra IBM Tivoli Netcool/Impact policies when edits are made to the incidents in Cloud Pak for AIOps. For example, if you change the priority or state in Cloud Pak for AIOps and want to write that back to the external ticketing system. For this scenario, in the **Trigger entity type** field in the **Policy details** tab, select **After an incident has been updated**.

Additional information

You can use the following IBM Tivoli Netcool/Impact code snippet to pull out all the ITSM metadata from an incoming Cloud Pak for AIOps incident. This code might be useful to you for further processing in IBM Tivoli Netcool/Impact.

```
var insights = EventContainer.incident.insights;

for (var i = 0; i < insights.length; i++) {
  var thisInsight = insights[i];

  if (thisInsight.type === 'aiops.ibm.com/insight-type/itsm/metadata') {
    Log("Insight: "+thisInsight.details.ticket_num);
  }
}
```

Configuring an integration in IBM Tivoli Netcool/Impact for IBM Cloud Pak for AIOps

Edit the RESTful DSA data source in IBM Tivoli Netcool/Impact for an integration to IBM Cloud Pak for AIOps to finish configuring the integration and integration between IBM Tivoli Netcool/Impact and IBM Cloud Pak for AIOps.

- Before you begin
- Configuring IBM Tivoli Netcool/Impact RESTful DSA data source
- Verifying the integration

Before you begin

1. Ensure that you have created the IBM Tivoli Netcool/Impact integration in the IBM Cloud Pak for AIOps UI. The integration can be created from the Integrations page in the UI. For more information, see Creating IBM Tivoli Netcool/Impact integrations.
2. Obtain the following details for your IBM Cloud Pak for AIOps deployment:
   - Your IBM Cloud Pak for AIOps hostname (cluster_address). For more information about obtaining this information, see Accessing your cluster by using the console.
   - Your encoded (base64 format) Platform UI API Key (**ZenApiKey**) for accessing IBM Cloud Pak for AIOps APIs. If you do not have an existing API key and need to generate a new API key, complete the following steps:

     Accessing APIs - Get authentication by using Platform UI API Key

     You can verify that your API key works by running the following command:

     ```
     curl -kv -H "Authorization: ZenApiKey ${ZenApiKey}" -H "x-tenant-id: ${tenant_id}"
      "https://${CPD_ROUTE}/aiops/api/issue-resolution/v1/alerts/"
     ```

Important: If you lose your API key, repeat the preceding steps to generate a new API key. Your old API key becomes invalid, and any applications or scripts cannot authenticate to the platform until you provide your new API key. For more information, see Accessing APIs.

- The Red Hat OpenShift Container Platform certificate for IBM Cloud Pak for AIOps.

Note: When you create an IBM Tivoli Netcool/Impact integration, the integration creates a Data Source name (DSA) in the IBM Tivoli Netcool/Impact Data Model (as before). The DSA name is created in the format of ***`AIOPs-ConnectorName-ConnectorID`***. Any IBM Tivoli Netcool/Impact policies will now use the new DSA instead of the old one. Therefore, you must update the new DSA with your cluster **Host Name** and **Authorization** (see step 4 below).

Configuring the RESTful DSA data source

1. Open the IBM Tivoli Netcool/Impact GUI.
2. Click and view the **Policies** tab.

   If the integration in IBM Cloud Pak for AIOps is successful, a RESTful DSA data source is already created. In addition, the following default policies are already created.

   - **`AIOPS_HandleAction`** - This file defines the wrapper that is used when IBM Cloud Pak for AIOps invokes an action in IBM Tivoli Netcool/Impact.
   - **`AIOPS_Mappings`** - This file defines how events are sent, and mapped, to IBM Cloud Pak for AIOps.
   - **`AIOPS_Utils`** - This file lists the APIs that are used for communicating with IBM Cloud Pak for AIOps.

   Ensure that these policies exist before proceeding. If they are missing, re-create the integration in IBM Cloud Pak for AIOps.
3. Click the **Data Model** tab.

   An **`AIOps-<identifier>`** RESTful DSA data source is now listed. Click this data source to view the Restful API Data Source Editor for the data model.
4. Go to the **Source** section and update the **Host Name** field with the name of the CPD route value.
5. Update the Request Header authorization setting.
   1. Go to the **Protected Request Headers** section.
   2. Select the **Authorization** header and click **Edit** in the table header.
   3. Update the **Header Value** field to include your encoded Platform UI API Key (**`ZenApiKey`**). This value must be in the following format:

      ```
      ZenApiKey <ZenApiKey>
      ```

      Where **`<ZenApiKey>`** is your encoded Platform UI API Key.
   4. Click **OK**.
6. Import the Red Hat OpenShift Container Platform certificate for IBM Cloud Pak for AIOps.

   To get the certficate, use the following steps:
   1. Export certificate from the IBM Cloud Pak for AIOps API Host:

      ```
      openssl s_client -showcerts -servername <cpd_route_name> -connect <cpd_route_name>:443 </dev/null
      ```

      Note: There are two similar routes, **`cp`** and **`cpd`**. You must use the **`cpd`** route.

      This returns the certificate in the display. The certificate has the following format:

      ```
      -----BEGIN CERTIFICATE-----
      ABCDEfGHIjklMnOPQrSTUVWXYZABCdefghiJ1k2LMNoPQRSTUVwxYZABCDEFGhij
      ... certificate text...
      ... Server 1 ...
      r+wGXNPwrw==
      -----END CERTIFICATE-----
      ```
   2. Copy the text into a certificate file. Then, import the file to the IBM Tivoli Netcool/Impact server and the IBM Tivoli Netcool/Impact UI:

      ```
      $IMPACT_HOME/sdk/bin/keytool -import -v -keystore
      $IMPACT_HOME/wlp/usr/servers/ImpactUI/resources/security/trust.jks -storepass  <impactadmin password> -alias
      aiops -file  <exported_aiops_certificate>
      $IMPACT_HOME/sdk/bin/keytool -import -v -keystore $IMPACT_HOME/wlp/usr/servers/<NCI
      NAME>/resources/security/trust.jks -storepass  <impactadmin password> -alias aiops -file
      <exported_aiops_certificate>
      ```
7. Click the **Save** icon to save your RESTful DSA data source configuration. Then, test your integration.

Verifying the integration

1. Create a test event policy in IBM Tivoli Netcool/Impact to create an alert in IBM Cloud Pak for AIOps.

   Your IBM Tivoli Netcool/Impact policies must be written in JavaScript to work with IBM Cloud Pak for AIOps. The Impact Policy Language (IPL) is not supported through the integration.

   You can use the following example JavaScript structure to help you create your own test policy. Replace the variables **`<variable>`** with the values for your own test event and system details.

```
Load('AIOPS_Utils');
var occurrenceTime = new Date();
var myEventInstance = "<instance_name>";
var event = {
    deduplicationKey: myEventInstance + " <event_name>",
    sender: {
      service: "<service>",
      name: "<sender_name>",
      type: "Success",
      location: "<location>",
      cluster: "\"quotes\"",
      controller: "<script>alert(\"<sender_controller>\");</script>"
    },
    resource: {
      name: myEventInstance + "<resource_name>",
      type: myEventInstance + " <type>",
      service: myEventInstance + " <service>",
      location: "<location>",
      controller: "<resource_controller>",
      interface: "<interface>",
      ipAddress: "<ip_address>",
      hostname: "<hostname>",
      cluster: "<cluster>"
    },
    type: {
      classification: myEventInstance + " <classification>",
      eventType: "<event_type>"
    },
    severity: 6,
    summary: myEventInstance + " <summary>",
    occurrenceTime: occurrenceTime.toISOString()
  };
var res = aiopsUtils.postEventDSA( "AIOps-<UUID>" , event );
Log("res: " + res);
```

Where **<event_type>** can be set to **problem** or **resolution**.

To obtain the **UUID** value to use for the **aiopsUtils.postEventDSA("AIOps-<UUID>" , event)** setting, run the following command:

```
oc get connectorconfiguration -o json | jq -r '.items[] | (.metadata.name + " - " + .metadata.uid)'
```

Note: You need **jq** to run this command. To install **jq**, see Download jq.

Output:

```
impact-netcool - 123abcd4-1ab2-1a23-1a2b-1aabbc234de5
```

The value **123abcd4-1ab2-1a23-1a2b-1aabbc234de5** is the UUID value.

2. Run the policy to test the integration from the IBM Tivoli Netcool/Impact UI **Policies** tab.

   Open the test policy that you created, and click the **Run** icon in the toolbar. Wait a few minutes, then proceeed to the next step.

3. Open the IBM Cloud Pak for AIOps UI and from the main navigation click **Incidents and alerts**. Click the **Alerts** tab to open the Alert Viewer.

   The alert for your test event should now display.

What to do next

- Create policies in IBM Tivoli Netcool/Impact. For more information about setting up these policies, see the IBM Tivoli Netcool/Impact documentation.
- Create a policy in IBM Cloud Pak for AIOps and invoke an IBM Tivoli Netcool/Impact policy. For more information, see:
  - Creating policies
  - Invoke an Impact policy

Creating IBM Netcool Operations Insight ObjectServer integrations

You can send event data from an existing IBM Tivoli Netcool/OMNIbus environment to IBM Cloud Pak for AIOps through an IBM Netcool Operations Insight ObjectServer integration. This integration connects the IBM Tivoli Netcool/OMNIbus ObjectServer with IBM Cloud Pak for AIOps.

With this integration, you can complete the following tasks:

- You can see IBM Tivoli Netcool/OMNIbus event data in the Alert Viewer in IBM Cloud Pak for AIOps. Events are converted into alerts on the IBM Cloud Pak for AIOps side.
- You can take advantage of the IBM Cloud Pak for AIOps AI features. For example, your event data is correlated into incidents, and you can associate topology with your data. You can also create runbooks to automate operations based on your data.

For more information about working with IBM Netcool Operations Insight ObjectServer integrations, see the following sections:

- About IBM Netcool Operations Insight ObjectServer integrations
- Before you begin
- Creating integrations

- Editing integrations
- Deleting integrations

About IBM Netcool Operations Insight ObjectServer integrations

- Though an IBM Netcool Operations Insight ObjectServer integration is bidirectional, not all data is synchronized. For example, if you close an alert in IBM Cloud Pak for AIOps, the alert does not get closed on the IBM Tivoli Netcool/OMNIbus ObjectServer. The **`AIOpsState`** column in the ObjectServer gets updated to closed but the alert continues to stay open on the ObjectServer. The operator must write a trigger on the ObjectServer to close such alerts. However, if an alert is closed on the ObjectServer, the alert does get closed in IBM Cloud Pak for AIOps.
- If you plan to create multiple IBM Netcool Operations Insight ObjectServer integrations, the set of events from each integration must be unique and not overlap with other integrations. For example, if you want to retrieve two separate sets of events, such as odd and even ServerSerial events, then you must use two different integrations. The connection filter helps to group events that you want to synchronize. For more information about the filter, see Filter clause.
    - If you have multiple ObjectServer pairs, you must create a corresponding IBM Netcool Operations Insight ObjectServer integration for each pair. Each ObjectServer pair requires a separate IBM Netcool Operations Insight ObjectServer integration. Do not reconfigure an existing IBM Netcool Operations Insight ObjectServer integration to access another ObjectServer pair.

      Note: An ObjectServer pair must synchronize with only one IBM Cloud Pak for AIOps instance through one or more IBM Netcool Operations Insight ObjectServer integrations. If you want to repoint an ObjectServer pair to a different IBM Cloud Pak for AIOps instance, clear the **`AIOpsAlertID`** and **`AIOpsState`** columns in the **`alerts.status`** table on the ObjectServer before you start the new deployment.
    - Parallel IBM Netcool Operations Insight ObjectServer integrations can be useful for distributing event load. However, use such a deployment only if necessary.

Before you begin

To configure an IBM Netcool Operations Insight ObjectServer integration, you need the following information:

- ObjectServer credentials
- Hostname and port for the primary and backup ObjectServers
- Custom ObjectServer column mapping

Creating IBM Netcool Operations Insight ObjectServer integrations

To create an IBM Netcool Operations Insight ObjectServer integration for IBM Tivoli Netcool/OMNIbus, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **IBM Netcool Operations Insight ObjectServer** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. To configure the ObjectServers with IBM Cloud Pak for AIOps columns, complete the following steps under *Configure ObjectServer*:
    - Open a command line.
    - Connect to the Netcool ObjectServer Host.
    - Save the following SQL commands in a file called *add_trigger.sql*:

```
-- Add IBM Cloud Pak for AIOps required fields ---
alter table alerts.status add column AIOpsAlertId varchar(64);
go

alter table alerts.status add column AIOpsState varchar(16);
go

-- Add the Netcool integration automation trigger group --
create or replace trigger group NetcoolConnectorAutomation;
go

-- Enable the trigger --
alter trigger group NetcoolConnectorAutomation set enabled true;
go

-- Add the clear automation trigger --
create or replace trigger netcool_connector_clear_alerts
group NetcoolConnectorAutomation
priority 1
comment 'Netcool integration clear alerts that are closed in IBM Cloud Pak for AIOps.'
every 10 seconds
when get_prop_value('ActingPrimary')%='TRUE'
begin
   -- Set Sev to 0  if AIOpsState is closed
```

```
   for each row problem in alerts.status where
       problem.Type = 1 and problem.Severity > 0 and problem.AIOpsAlertId != '' and problem.AIOpsState =
'closed'
   begin
       update alerts.status via problem.Identifier set Severity = 0;
   end;
end;
go
```

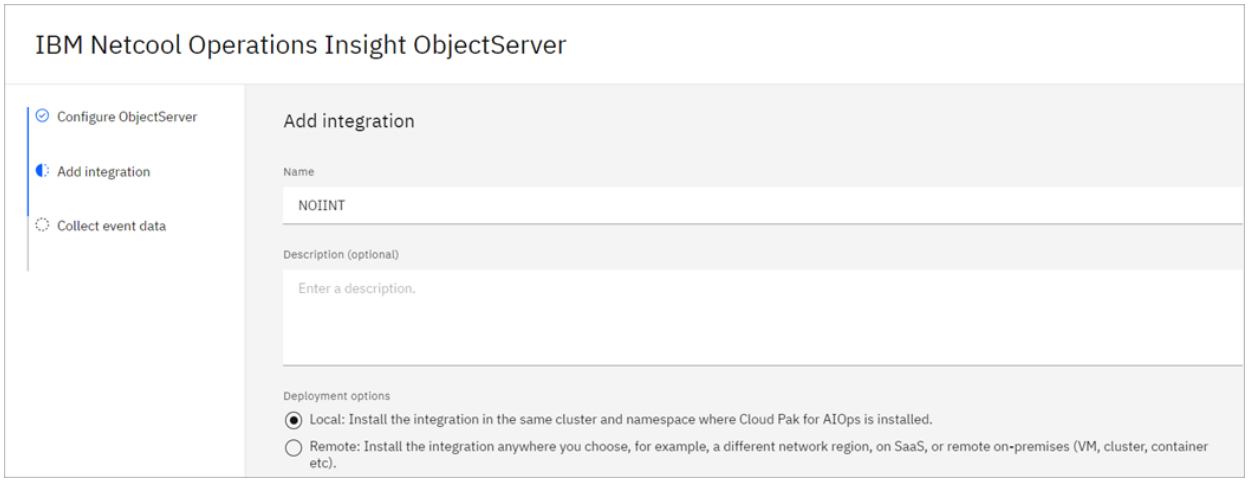
Note: The ScopeID column is available by default on Red Hat OpenShift Container Platform. However, use the following command for setting it up for on-premises ObjectServer:

```
alter table alerts.status add column ScopeID varchar(255);
go
```

- Run the following command to install the required columns and trigger by using the *add_trigger.sql* file:

```
$OMNIHOME/bin/nco_sql -server <object_server_name> -username root -password <password> < add_trigger.sql
```

7. Click **Next**.
8. Enter the following *Add integrations* information:
   - **Name**: The display name of your integration.
   - **Description**: An optional description for the integration.
   - **Deployment options**: Select **Local** or **Remote** option.



   Note: If you select local, the deployment is in the same cluster where IBM Cloud Pak for AIOps is installed. If you select remote, the deployment happens wherever you choose, for example, on a different network region, on Saas, or remotely on premises (VM, cluster, or container).

9. Fill the *ObjectServer pair* fields. For more information about the fields and values to be entered, see Table 1.
10. Click **Test Connection**. If successful, a *Test succeeded* message is displayed. If it fails, one of the following reasons appears:

- Filter is not valid
- Unable to connect to ObjectServer
- IBM Cloud Pak for AIOps fields not present

Note: If the test fails, check:

- if the *TLS certificate* is correct.
- if the *NetworkPolicy* is created with the correct ports and in the correct namespace.
- if the *ExternalName* service is created with correct service name, ports, and in the correct namespace.

For more information about *TLS certificate*, *NetworkPolicy*, and *ExternalName* service, see TLS-enabled connection.

11. Click **Next**.
12. Enter the following *Collect event data* information to define how active IBM Tivoli Netcool/OMNIbus events are collected.
    - **Enable data collection**: This toggle is set to *on* by default. It starts to collect event data from IBM Tivoli Netcool/OMNIbus immediately on creation of the integration.

- **Mapping**: This field provides mapping for standard IBM Tivoli Netcool/OMNIbus event columns that use Jsonata syntax. You can add your own custom ObjectServer columns to this mapping. If they have custom alert columns in your ObjectServer, you must add them to the Jsonata mapping provided in the Mapping textbox.

  Note: To customizing the mapping, copy the text out of the mapping textbox and into your preferred text editor. Then, add the mapping for those custom columns there and check the code syntax. When done, copy the Jsonata code back into the mapping textbox. If successful, a *Valid Jsonata configuration* message is displayed. Be aware that Jsonata is case sensitive. For details about Jsonata syntax and usage see JSONata Documentation.

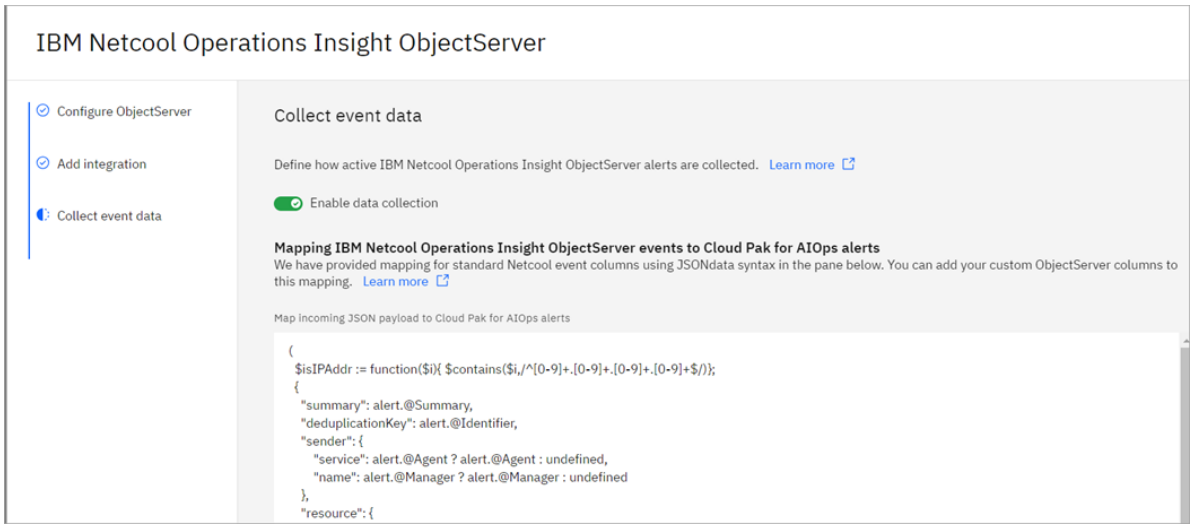


Figure. Collect event data

13. Click **Save**.

You have created an IBM Netcool Operations Insight ObjectServer integration in your instance, and a message appears on the integrations page that states *Connection completed. IBM Cloud Pak for AIOps has successfully processed your request*.

After you integrate with IBM Tivoli Netcool/OMNIbus, use IBM Cloud Pak for AIOps and the IBM Cloud Pak for AIOps UI for AI and visualization features. Use IBM Tivoli Netcool/OMNIbus and the associated UI tools for obtaining and sending data to IBM Cloud Pak for AIOps.

Notes:

- If you are integrated and are not yet ready, or able, to move to using IBM Cloud Pak for AIOps and the IBM Cloud Pak for AIOps UI fully. You can continue to use the IBM Tivoli Netcool/OMNIbus DASH custom dashboards and Web GUI event viewer as you transition to IBM Cloud Pak for AIOps. If you need to continue to use IBM Tivoli Netcool/OMNIbus tools, you can configure launch-in-context links from the Web GUI to the IBM Cloud Pak for AIOps UI to help users access and use IBM Cloud Pak for AIOps UI. For more information, see Configuring launch-in-context from IBM Tivoli Netcool/OMNIbus
- If you plan to connect with an IBM Tivoli Netcool/OMNIbus deployment in an environment that is enabled to support Federal Information Processing Standards (FIPS), you need to configure your integration to also enable the support. For more information, see Enabling support for Federal Information Processing Standards (FIPS).

ObjectServer pairs

The following table outlines the ObjectServer pair fields and their values.

Table 1. ObjectServer pair fields

ObjectServer pair fields	Description	Values
Admin username	Enter a username to authenticate to the ObjectServer.	$ NOI_OMNI_USERNAME
Admin password	Enter a password to authenticate to the ObjectServer.	$ NOI_OMNI_PASSWORD
Filter (optional)	Define the alerts that you want to synchronize by using ObjectServer SQL where clause syntax. Without filter, the integration takes all alerts from the database table.	For example, Severity > 4 or Manager LIKE 'CEM'. For more examples, see Filter clause.
Is TLS enabled?	Enable or disable the toggle button. For more information about TLS-enabled connection, see TLS-enabled connection.	**Yes** or **No**
Primary ObjectServer hostname	Enter a hostname for the primary ObjectServer integration.	$ NOI_PROXY_PRIMARY_SVC
Primary ObjectServer port	Specify a port.	$ NOI_PROXY_PRIMARY_PORT
Backup ObjectServer hostname (Optional)	Enter a hostname for the backup ObjectServer integration.	$ NOI_PROXY_BACKUP_SVC
Backup ObjectServer port (Optional)	Specify a port.	$ NOI_PROXY_BACKUP_PORT
Trusted certificates (if TLS is enabled)	Enter the TLS certificate that is used to verity the SSL/TLS connection to the ObjectServers.	See TLS-enabled connection

TLS-enabled connections

A TLS-enabled connection provides end-to-end security to data that is transmitted through it. To activate a TLS-enabled connection, you must provide a Transport Layer Security (TLS) certificate. A TLS certificate is a type of digital certificate, issued by a certificate authority (CA), verifying the ownership of the domain name.

You can create a TLS-enabled connection to either Red Hat® OpenShift® Container Platform ObjectServer, or the On-premises IBM Tivoli Netcool/OMNIbus ObjectServer. For more information, see:

- Creating a TLS-enabled integration to a Red Hat® OpenShift® Container Platform ObjectServer

Editing IBM Netcool Operations Insight ObjectServer integrations

After you create your integration, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Netcool Operations Insight ObjectServer** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the IBM Netcool Operations Insight ObjectServer integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration.



   Note: While you can modify all other fields, you cannot alter the actual integration name at this stage, as it is already created and thus shows as read-only.

6. Click **Save** when you are done editing.

Your integration is edited. If you did not previously enable or disable your application, you can enable or disable the integration directly from the console. For more information about deleting a integration, see Deleting IBM Netcool Operations Insight ObjectServer integrations.

Deleting IBM Netcool Operations Insight ObjectServer integrations

If you no longer need your IBM Netcool Operations Insight ObjectServer integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable data collection before deleting your integration.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Netcool Operations Insight ObjectServer** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the IBM Netcool Operations Insight ObjectServer integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

If you want to remove columns and automation triggers from Netcool ObjectServer, complete the following steps:

1. Open a terminal.
2. Connect to the Netcool ObjectServer host.
3. Save the following SQL commands into a file called *remove_trigger.sql*:

```
alter trigger group NetcoolConnectorAutomation set enabled false;
go

drop trigger netcool_connector_clear_alerts;
go

drop trigger group NetcoolConnectorAutomation;
go

alter table alerts.status drop column AIOpsAlertId;
go
```

```
alter table alerts.status drop column AIOpsState;
go
```

4. Run the following command to install the required columns and trigger by using the remove_trigger.sql file:

```
$OMNIHOME/bin/nco_sql -server <object_server_name> -username root -password <password> < remove_trigger.sql
```

Configuring launch-in-context from IBM Tivoli Netcool/OMNIbus

If you have integrated IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus, you can configure launch-in-context links from the Web GUI Event Viewer and the main DASH navigation to the IBM Cloud Pak for AIOps UI to help your users access and use the IBM Cloud Pak for AIOps UI.

With these links, your users can use the Web GUI Event Viewer and DASH to manage the gathering of event data, which is then sent to IBM Cloud Pak for AIOps. These users can then launch the IBM Cloud Pak for AIOps UI for AI analysis and visualization of the data.

Creating an integration between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus to send event data to Cloud Pak for AIOps is prerequisite. For more information, see Creating IBM Netcool Operations Insight ObjectServer integrations.

For more information about configuring launch-in-context, see:

- Adding incidents and alerts to the DASH navigation menu
- Creating tools for launch-in-context from Netcool/OMNIbus Event Viewer

Adding incidents and alerts to the DASH navigation menu

If you have integrated IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus, you can add a launch-in-context link to the IBM Cloud Pak for AIOps Incidents and alerts page to the DASH navigation menu for IBM Tivoli Netcool/OMNIbus.

Procedure

1. Navigate to the JAZZSM_HOME directory and open the following file:
   <JAZZSM_HOME>/profile/config/cells/JazzSMNode01Cell/applications/isc.ear/deployments/isc/isclite.war/WEB-INF/navigation.xml

2. At the end of the content wrapped between the following tags:

   ```
   <navigation:navigation-tree> {...} </navigation:navigation-tree>
   ```

   Append the following XML snippet, replacing the values marked as **<CP4WAIOPS_STORIES_URL>** and **<CP4WAIOPS_ALERTS_URL>** with valid URLs to the Cloud Pak for AIOps Incidents and Alerts pages:

   ```
   <navigation:nav-element formPersistence="false" hidden="false" iconClass="cloudIcon" isWscNode="false"
   layoutUpdatedAtRuntime="false" moduleID="com.ibm.isclite.global.custom.module"
   moduleIDOrig="com.ibm.isclite.global.custom.module" nodeType="label" securityUpdatedAtRuntime="false"
   stateMaintained="false" taskType="hiddenTask" uniqueName="com.ibm.isclite.global.custom.module-SPSVS-
   label.W8DQOK4vA5VBn4RGGldV-u11650881799360" wscRole="all authenticated portal users" wscRoleType="User">
     <navigation:title>
       <base:nls-string lang="en">CP4WAIOps</base:nls-string>
     </navigation:title>
     <navigation:parentTree parentTreeRef="root"/>
     <navigation:parameter name="com.ibm.console.resource.previewImage">
       <base:value>
       <base:string>CP4WAIOps</base:string>
       </base:value>
     </navigation:parameter>
     <navigation:parameter name="com.ibm.console.resource.description">
       <base:value>
       <base:string>CP4WAIOps</base:string>
       </base:value>
     </navigation:parameter>
     <navigation:page-params/>
     <navigation:nav-element formPersistence="false" hidden="false" isWscNode="false" layout-element-
   ref="com.ibm.isclite.admin.Freeform.layoutElement.A__1650882838344" layoutUpdatedAtRuntime="true"
   moduleID="com.ibm.isclite.global.custom.module" moduleIDOrig="com.ibm.isclite.global.custom.module" nodeType="url"
   securityUpdatedAtRuntime="false" stateMaintained="true" taskType="multiTask"
   uniqueName="com.ibm.isclite.global.custom.module-SPSVS-
   com.ibm.isclite.admin.Freeform.navigationElement.pagelayoutA.modified.9L7ci90Q2Z7x3uGKVJ2v5uh1650882838339" url="
   <CP4WAIOPS_STORIES_URL>" wscRole="iscadmins" wscRoleType="Editor">
       <navigation:title>
       <base:nls-string lang="US">Incidents</base:nls-string>
       </navigation:title>
       <navigation:parentTree parentTreeRef="label.W8DQOK4vA5VBn4RGGldV-u11650881799360"/>
     </navigation:nav-element>
     <navigation:nav-element formPersistence="false" hidden="false" isWscNode="false" layout-element-
   ref="com.ibm.isclite.admin.Freeform.layoutElement.A__1651490860014" layoutUpdatedAtRuntime="true"
   moduleID="com.ibm.isclite.global.custom.module" moduleIDOrig="com.ibm.isclite.global.custom.module" nodeType="url"
   securityUpdatedAtRuntime="false" stateMaintained="true" taskType="multiTask"
   uniqueName="com.ibm.isclite.global.custom.module-SPSVS-
   com.ibm.isclite.admin.Freeform.navigationElement.pagelayoutA.modified.taRbaKSlGPLajMfN2BbRVfB1651490860008" url="
   <CP4WAIOPS_ALERTS_URL>" wscRole="iscadmins" wscRoleType="Editor">
       <navigation:title>
   ```

```
    <base:nls-string lang="US">Alerts</base:nls-string>
    </navigation:title>
    <navigation:parentTree parentTreeRef="label.W8DQOK4vA5VBn4RGGldV-u11650881799360"/>
  </navigation:nav-element>
</navigation:nav-element>
```

3. Save the `navigation.xml` file.
4. Log out of DASH, then log back in again.

Creating a tool for launch-in-context from IBM Tivoli Netcool/OMNIbus Event Viewer

To use a launch-in-context tool, you must add it to the list of available IBM Tivoli Netcool/OMNIbus tools by using the **Tool Configuration** option within the IBM Tivoli Netcool/OMNIbus Web GUI. You must then add the tool to the list of tools that appear on the pop-up menu in the IBM Tivoli Netcool/OMNIbus Event Viewer by using the **Menu Configuration** option.

Complete the steps in the following procedure to create the tool for launch-in-context from selected events in the Event Viewer.

Procedure

1. As a user with the admin role, log in to your DASH web application.
2. Open the Netcool/OMNIbus Web UI tool configuration page: **Administration > Event Management Tools > Tool Configuration**.
3. Click the create tool icon to create a new tool.
4. Enter a name for the tool, for example "View in IBM Cloud Pak for AIOps".
5. From the **Type** drop-down field, select **CGI/URL**.
6. Paste the following URL in the field provided:

   ```
   <aiops hostname>:<aiops port>/aiops/cfd95b7e-3bc7-4006-a4a8-a73a79c71255/resolution-hub/alerts?
   selectedRowIds=alerts.status.undefined:{$selected_rows.AIOpsAlertId}&selectedColumnId=summary
   ```

   Where `<aiops hostname>:<aiops port>` is the URL of your IBM Cloud Pak for AIOps instance in the format of protocol://hostname:port.
7. Click **Save**.
8. Open the Netcool/OMNIbus Web UI menu configuration page: **Administration > Event Management Tools > Menu Configuration**.
9. Click the **Alerts** menu and select **Modify**.
10. Locate the tool that you created in the available list and add it to the list of selected tools.
11. Click **Save**.

This tool should now be available in the Event Viewer when you right-click on an Event. Selecting this tool launches into the Cloud Pak for AIOps alerts for the selected event.

Creating a TLS-enabled IBM Netcool Operations Insight ObjectServer integration to an Red Hat OpenShift ObjectServer

If you are connecting IBM Cloud Pak for AIOps to IBM Tivoli Netcool/OMNIbus and you have TLS enabled for your Red Hat OpenShift ObjectServer, you need to extract the TLS certificate to add to IBM Cloud Pak for AIOps.

A TLS-enabled connection between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus provides end-to-end security to data that is transmitted through it. To create a TLS-enabled connection to a Red Hat OpenShift ObjectServer, complete the following steps:

1. Get the host and port for primary and backup ObjectServers from the TLS Proxy service:

   ```
   NOI_INSTANCE=<NOI_instance_name>
   NOI_NAMESPACE=<NOI_namespace>
   NOI_PROXY_SVC=${NOI_INSTANCE}-proxy
   NOI_PROXY_PRIMARY_SVC=${NOI_PROXY_SVC}.${NOI_NAMESPACE}.svc
   NOI_PROXY_PRIMARY_PORT=$(oc describe svc -n $NOI_NAMESPACE $NOI_PROXY_SVC | grep aggp-proxy-port | awk -F' ' '{
   print $3 }' | awk -F'/' '{ print $1 }')
   NOI_PROXY_BACKUP_SVC=${NOI_PROXY_SVC}.${NOI_NAMESPACE}.svc
   NOI_PROXY_BACKUP_PORT=$(oc describe svc -n $NOI_NAMESPACE $NOI_PROXY_SVC | grep aggb-proxy-port | awk -F' ' '{ print
   $3 }' | awk -F'/' '{ print $1 }')
   ```

   Where, `<NOI_instance_name>` is the Netcool instance name that is installed in the cluster and `<NOI_namespace>` is the namespace where Netcool is installed. For example, `NOI_NAMESPACE=noi`.
2. Create a *NetworkPolicy* in the Netcool namespace to allow the probe to access the TLS Proxy service and port in the IBM Netcool Operations Insight namespace. To create the *NetworkPolicy*, complete the following steps:

1. Set the environment variables.

```
NOI_INSTANCE=<NOI_instance_name>
NOI_NAMESPACE=<NOI_namespace>
NOI_OBJSERV_AGG_PRIMARY_SVC=$NOI_INSTANCE-objserv-agg-primary
NOI_OBJSERV_AGG_BACKUP_SVC=$NOI_INSTANCE-objserv-agg-backup
NOI_OBJSERV_AGG_PRIMARY_PORT=$(kubectl get svc -n $NOI_NAMESPACE $NOI_OBJSERV_AGG_PRIMARY_SVC -o
jsonpath='{.spec.ports[?(@.name=="primary-tds-port")].port}')
NOI_OBJSERV_AGG_IDUC_PORT=$(kubectl get svc -n $NOI_NAMESPACE $NOI_OBJSERV_AGG_PRIMARY_SVC -o
jsonpath='{.spec.ports[?(@.name=="primary-iduc-port")].port}')
```

   Where **<NOI_instance_name>** is the Netcool instance name that is installed in the cluster and **<NOI_namespace>** is the namespace where Netcool is installed.

2. Run the following commands:

```
cat << EOF | tee >(oc apply -f -) | cat
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
   name: noi-allow-objectserver
   namespace: ${NOI_NAMESPACE}
spec:
   podSelector:
      matchLabels:
         app.kubernetes.io/instance: ${NOI_INSTANCE}
         appClass: objserv
   ingress:
      - ports:
         - protocol: TCP
           port: ${NOI_OBJSERV_AGG_PRIMARY_PORT}
         - protocol: TCP
           port: ${NOI_OBJSERV_AGG_IDUC_PORT}
   policyTypes:
      - Ingress
EOF
```

   Note: Check to see whether the existing *NetworkPolicy* allows the probe to access the TLS Proxy service and port in Netcool namespace. The *NetworkPolicy* must show both the primary and IDUC ports.

3. Create an *ExternalName* service for the primary and IDUC ports. It maps the namespace scoped service to the Netcool service in the namespace. To create the *ExternalName* service, complete the following steps:

   1. Set the environment variables.

```
AIMANAGER_NAMESPACE=<AIManager_namespace>
oc project $AIMANAGER_NAMESPACE
```

      Where **<AIManager_namespace>** is the namespace where IBM Cloud Pak for AIOps is installed.

   2. Run the following commands.

```
cat << EOF | tee >(oc apply -f -) | cat
apiVersion: networking.k8s.io/v1
kind: Service
apiVersion: v1
metadata:
  name: ${NOI_INSTANCE}-objserv-agg-primary
  namespace: ${AIMANAGER_NAMESPACE}
spec:
  type: ExternalName
  externalName: ${NOI_INSTANCE}-objserv-agg-primary.${NOI_NAMESPACE}.svc.cluster.local
  ports:
  - port: 4101
    name: primary-iduc-port
---
kind: Service
apiVersion: v1
metadata:
  name: ${NOI_INSTANCE}-objserv-agg-backup
  namespace: ${AIMANAGER_NAMESPACE}
spec:
  type: ExternalName
  externalName: ${NOI_INSTANCE}-objserv-agg-backup.${NOI_NAMESPACE}.svc.cluster.local
  ports:
  - port: 4101
    name: backup-iduc-port
---
EOF
```

4. Obtain a TLS certificate. To obtain the TLS certificate for the ObjectServer, run the following command:

```
oc extract secret/$NOI_INSTANCE-proxy-tls-secret -n $NOI_NAMESPACE   --to=. --keys=tls.crt
```

Filter clauses for an IBM Netcool Operations Insight ObjectServer integration

For an integration between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus, you can use the filter clause to define the alerts that you want to send to IBM Cloud Pak for AIOps. You can create filter clauses that are suitable for your environment by using an ObjectServer SQL **where** clause syntax.

To generate filter clauses, use fields from the ObjectServer **alerts.status** table, such as **ServerName**, **Agent**, or **Manager** for the syntax. The following example shows this syntax:

```
ServerName IN ('Server1','Server2')
```

Filter clause examples

The following table outlines some of the examples of **FILTER** clause and their description:

Filter clause examples

Filter examples	Description
`Agent != 'OMNIbus SelfMonitoring'`	This filter clause filters all events in the ObjectServer **alerts.status** table with the **Agent** field not equal to **OMNIbus SelfMonitoring**.
`Manager not like 'OMNIbus Self Monitoring*'`	This filter clause filters all events in the ObjectServer **alerts.status** table with the **Manager** field not starting with **OMNIbus Self Monitoring**. For more examples of **Manager** field, see Note.
`Severity >= 2`	This filter clause filters all events in the ObjectServer **alerts.status** table with **Severity** field equal to or beyond the warning level. Thus, the integration selects only those alerts that have severity more than or equal to 2. For more information about severity level representation, see Event severity levels.
`ServerName IN ('AGG_P','AGG_B')`	This filter clause filters all events in the ObjectServer **alerts.status** table with ServerName equal to **AGG_P** and **AGG_B**. Thus, the integration selects only those alerts that originate from **'AGG_P'** and **'AGG_B'**.

The following are some examples of the **Manager** field:

```
OMNIbus Self Monitoring @AGG_P
OMNIbus Self Monitoring @AGG_B
```

Where **AGG_P** and **AGG_B** are server names.

You can use Boolean operators such as **AND, OR** to join conditions in the filter clause syntax. For example,

```
Acknowledged>0 AND ServerName IN ('NCOMS_US','NCOMS_CA')
```

Netcool event load management

The event rate that the ObjectServer can handle is greater than a Netcool connector can handle. As a consequence, you may need to run more than one Netcool connector instance and distribute the event load amongst the them.

The need to distribute the event load arises when the Netcool connectors deployed are showing signs of the SAF file count growing constantly. This indicates that Netcool event load is breaching the connector's processing limit. Unmitigated piling of SAF files will cause issues in the pod's file system and the database for Netcool data files encompassing cache file, outbound SAF, and inbound SAF. This can cause an event delay or missing events in IBM Cloud Pak for AIOps.

To prevent storage issues, high event load must be distributed across multiple connectors.

The best practice is to split the event load and enable each connector's IDUC query on alternate alert rows to achieve evenness in event sourcing. In the Netcool connector UI configuration, append the following clause to the filter definition.

```
(mod(ServerSerial,X)=Y)
```

Where **X** is the total number of connectors deployed and **Y** is the index for each connector.

For example, if there are five connectors to be used, X=5 and Y ranges from 0 to 4:

```
connector_1: FILTER AND (mod(ServerSerial,5)=0)
connector_2: FILTER AND (mod(ServerSerial,5)=1)
connector_3: FILTER AND (mod(ServerSerial,5)=2)
connector_4: FILTER AND (mod(ServerSerial,5)=3)
connector_5: FILTER AND (mod(ServerSerial,5)=4)
```

Important: The filter definition must be same across this group of connectors.

Measuring event load and SAF file growth manageable by one connector

A connector's IDUC occurs every 30 seconds to fetch Netcool events from ObjectServer. In one IDUC cycle the number of SAF files generated is proportional to the events fetched. Each SAF file can buffer up to 1000 entries of events. The connector's SAF processor consumes SAF files for event data mapping and forwarding. SAF processor can complete processing two 100%-filled SAF files in a minute. Complexity of custom mapping rules and Kafka endpoint performance will be to some degree lengthning the processing time to get every event mapped and forwarded, but this SAF per minute baseline should serve as a guideline for planning the number of connectors to be deployed.

Clean up for a fresh start

You must clean up is mandatory on Netcool connector and AIOps when there are changes in any of the following items:

- Number of Netcool connector instances
- Filter definition

For details of cleaning up the deployment see the steps for resynchronizing the alerts from Netcool and AIOps described in this procedure

Netcool connector troubleshooting

If you experience any problems when creating a Netcool connector, use the following troubleshooting steps:

1. When you create the connector instance, ensure data flow is enabled and that the following fields are added on the ObjectServer: **`AIOpsAlert ID`**, **`AIOpsAlertState`**, and **`ScopeID`**.
2. If you are unable to get a successful test connection from the IBM Cloud Pak for AIOps UI, use the following stpes:
    1. Check you have a valid entry for the connector in the ObjectServer **`omni.dat`** file.
    2. Ensure the port is not already in use.
    3. Check whether there are any firewall rules that maybe blocking the connection.
3. On successful creation of a Netcool connector, you should see the following objects created on the OpenShift Console:
    - Netcool-conn-pod
    - Netcool PVC (Check that the PVC size default is set to 2Gi.)
4. Check the connector bridge pods.
    - If you notice connector bridge pods restarting many times, or they are down, check the connector bridge pod logs for exceptions:

      ```
      oc logs <connector-bridge-xxx>
      ```

    - If you see the following message in the connector bridge pods:

      ```
      containerStatuses:
          lastState:
            terminated:
              exitCode: 137
              finishedAt: "2024-06-07T02:48:52Z"
              reason: OOMKilled
      ```

    This is an indication that the load incoming into the system is higher than the connector bridge can manage. The connector fails with the message: **`out of memory: OOMKilled`**
5. If events are not flowing into IBM Cloud Pak for AIOps, ensure the components are running:

    ```
    oc get installation -o yaml
    ```

    This shows all that components that are in Ready state.
    1. Ensure all kafka pods are in running state and not restarting.
    2. Ensure the lifecycle pods are running as expected:

       ```
       cp4waiops-eventprocessor-eve-29ee-ep-jobmanager
       cp4waiops-eventprocessor-eve-29ee-ep-taskmanager
       ```

    3. Ensure the following job is completed:

       ```
       cp4waiops-eventprocessor-eve-29ee-ep-taskmanager
       job.batch/aiops-ir-lifecycle-create-policies-job
       job.batch/aiops-ir-lifecycle-policy-registry-svc-job
       job.batch/aiops-ir-lifecycle-policy-upgrade-job
       job.batch/aiops-ir-lifecycle-sync-es-cassandra-cronjob-28654560
       ```

    4. Ensure all the workloads are running as expected. If any of the workloads are in failure state, you would not see events in IBM Cloud Pak for AIOps.
6. Custom Mapping.
    1. If you still do not see any events coming in, review your mapping in the connector config.
    2. Test with the default mappings to see if events are flowing in.
    3. Ensure that all custom fields (if integers) are type casted to string value.
7. Resync the connector.
    1. At the connector UI, disable the collector's data collection.
    2. In the connector pod, delete all files under /bindings/netcool-connector/omnibus/var/G_CONNECTOR.
    3. At the ObjectServer alerts.status table, reset the AIOpsAlertId and AIOpsState columns.
    4. At AIOps, remove all alerts from the ObjectServer.
    5. At the connector UI, enable the collector's data collection. **Note:** if the columns in the NOI objects server AIOpsAlertId and AIOpsState columns are not reset you may not see events in AIOPS as these mark that the connector has already actioned this events
8. Clearing Events and resetting Lifecycle state. Reset lifecycle state after running the following command on the **`must-gather`** script.

    ```
    waiops-mustgather.sh -DR -C clear-lifecycle-state.sh
    ```

    Note: Clearing the lifecycle state is generally done after recovery of the Issue Resolution Lifecycle operator or clearing events in Cloud Pak for AIOps and resetting lifecycle state.

Netcool event load splitting calculation

When you deploy IBM Netcool Operations Insight ObjectServer integrations, it is important to understand the event rate of domain alerts and calculate the number of integrations that are required to share the overall load. A high event load can lead to breakdowns due to the integration pod's file storage that is filled with unprocessed Store-and-Forward (SAF) files. By planning and sizing your Netcool Operations Insight integrations deployment, you can help ensure that it runs smoothly and efficiently.

IDUC and SAF files

- Netcool Operations Insight integration triggers an IDUC query to ObjectServer for latest alert events every thirty seconds.
- The events that are fetched by IDUC are buffered in SAF files. This activity can be seen as SAG staging.
- When a SAF file passes the staging phase, it is consumed by the integration's data forwarding process.
- Each SAF file can hold up 1,000 entries of Netcool Operations Insight events, but the maximum size of a fully filled SAF file is determined by the table columns that are involved in IDUC.
- The columns of interest for IDUC data replication are configured in the integration's mapping. More columns require more space usage per SAF file, but the file size alone is not the factor of file storage depletion. The unmitigated growth of SAF files is the main cause to the storage issue.
- Multiple SAF files might be produced in one IDUC. These files might be partially filled because the SAF staging process marks the file ready for the SAF processing (data forwarding process) to consume when either of the following conditions is met:
    - The SAF file is fully filled. For example, 1000 rows.
    - The wait time for more rows that are coming in is expired.

Optimal rate of SAF file growth

- To prevent the pile up of SAF files, all the SAF files that are produced in the N-th cycle of IDUC need to be processed by the data forwarding process before the end of the (N+1)-th cycle.
- The data forwarding process is running in parallel to IDUC and SAF staging.
- If there are more than 1,000 changed rows that are detected in an IDUC query, then multiple batching is required to complete the pulling of all the changed rows from the ObjectServer.

Understanding IDUC, SAF staging, and SAF processing

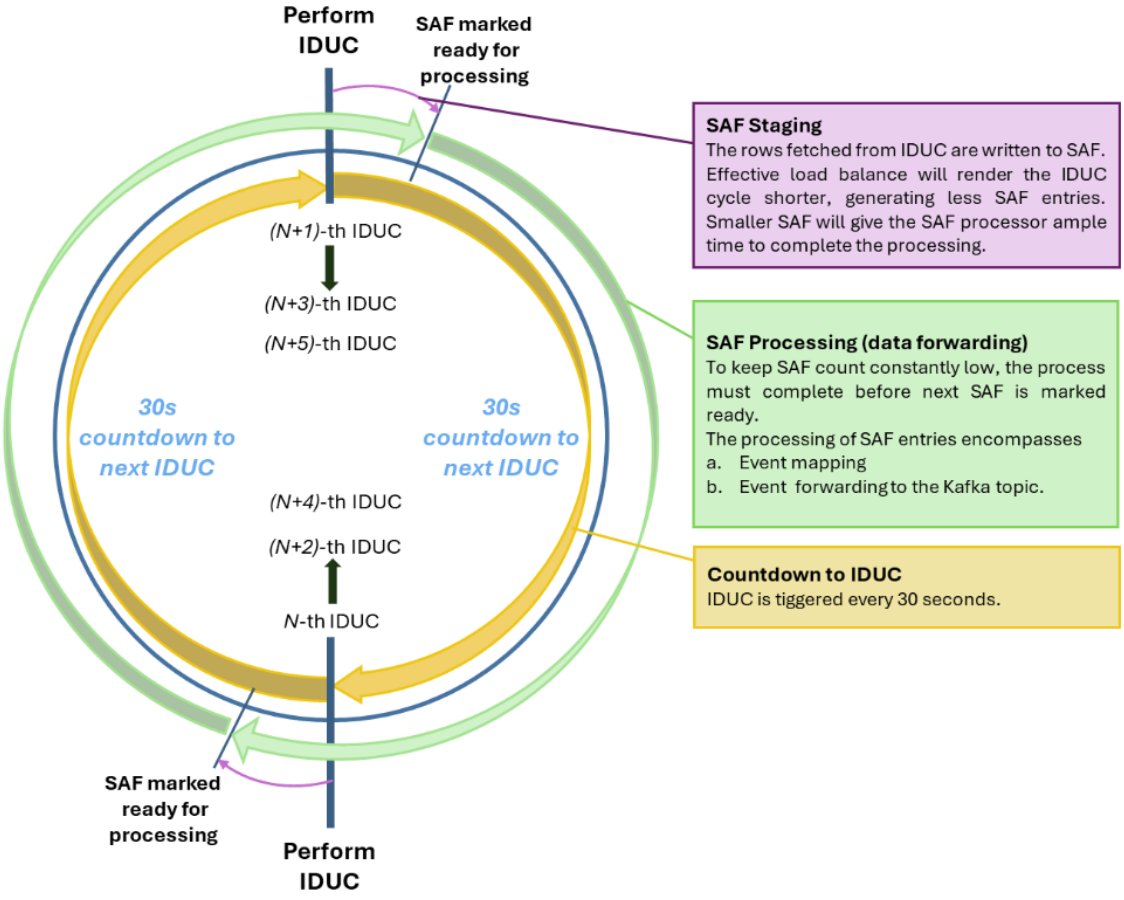


Figure. Cycles of IDUC, SAF staging, and processing projected in clock dial view

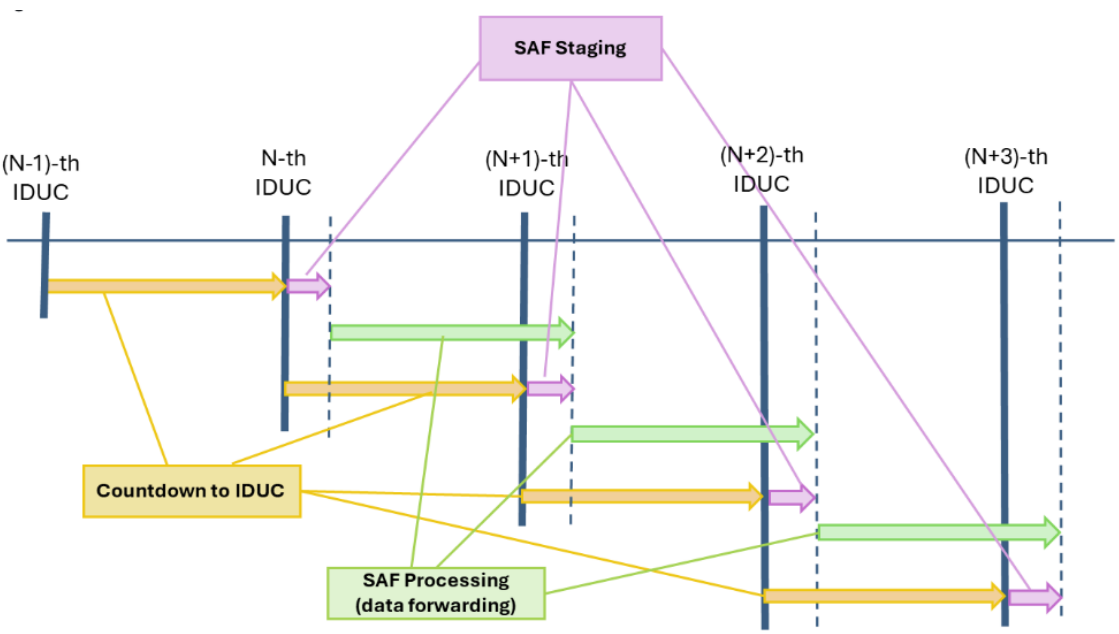


Figure. Cycles of IDUC, SAF staging, and processing projected in linear time view

Collecting event load statistics

1. Edit the following **objsvr_cred.sh** file to configure the ObjectServer username and password.

```
#!/bin/bash

USER=root
PASS=password

echo $USER $PASS
```

2. Edit the following **os_get_event_count.sh** file:

   - In the file, specify the time span for event count query. Configure the **query_interval_min** field.
   - In the file, specify an alert group for event count query. Configure the **MY_ALERT_GROUP** field.

```
#!/bin/bash

me="$(basename "$(test -L "$0" && readlink "$0" || echo "$0")")"

if [ $# -ne 1 ]; then
   echo "illegal number of parameters"
   echo "Usage: $me <object_server_name>"
   echo
   exit 0
fi

OBJSERV_NAME=$1

CRED=`./objsvr_cred.sh`
USER=`echo $CRED | awk -F' ' '{ print $1 }'`
PASS=`echo $CRED | awk -F' ' '{ print $2 }'`

# Configure the alert criteria of your choice
MY_ALERT_GROUP="AlertGroup='ConnectionStatus'"

omnihome="${OMNIHOME}"
sqllocation="./tmpsql"
mkdir -p $sqllocation

total_all_event_count=0
total_specific_event_count=0

start_time=0
query_interval_min=10 #in minutes
prev_epoch=0
sleep_interval=20 #in seconds

echo "Event count query for $query_interval_min mins"
query_interval_secs=$(( $query_interval_min * 60 ))

curr_timestamp=`date`
curr_epoch=`date +%s`
time_to_stop=$(( $curr_epoch + $query_interval_secs ))
echo "Current time: "${curr_epoch} "("${curr_timestamp}")"
echo "Time to stop: "${time_to_stop}
```

```
while :
do
   rand=`echo $RANDOM | base64 | head -c 20; echo`
   curr_epoch=`date +%s`

   if [[ $curr_epoch -gt $time_to_stop ]]
   then
      curr_timestamp=`date`
      echo "Query ends here. Time: "${curr_timestamp}
      exit 1
   fi
   if [[ $prev_epoch -eq 0 ]]
   then
      start_time=$curr_epoch
      prev_epoch=$(( $curr_epoch - $sleep_interval ))
   fi

   sttchg_window="StateChange>${prev_epoch} AND StateChange<=${curr_epoch}"
   select_condition=${sttchg_window}

   get_count_sqlfile_all="${sqllocation}/${rand}.get_count_all.sql"
   get_count_sqlfile_specific="${sqllocation}/${rand}.get_count_specific.sql"

   echo "select count(*) from alerts.status where ${select_condition}" >> ${get_count_sqlfile_all}
   echo "go" >> ${get_count_sqlfile_all}

   echo "select count(*) from alerts.status where ${select_condition} AND ${MY_ALERT_GROUP}" >>
${get_count_sqlfile_specific}
   echo "go" >> ${get_count_sqlfile_specific}

   echo "Show ${get_count_sqlfile_all} content:"
   cat ${get_count_sqlfile_all}
   echo
   echo "Show ${get_count_sqlfile_specific} content:"
   cat ${get_count_sqlfile_specific}
   echo

   all_count_result_file="${sqllocation}/${rand}.all_count_result.txt"
   ${omnihome}/bin/nco_sql -server ${OBJSERV_NAME} -user "${USER}" -password "${PASS}" < ${get_count_sqlfile_all} >
${all_count_result_file}

   specific_count_result_file="${sqllocation}/${rand}.specific_count_result.txt"
   ${omnihome}/bin/nco_sql -server ${OBJSERV_NAME} -user "${USER}" -password "${PASS}" <
${get_count_sqlfile_specific} > ${specific_count_result_file}

   all_evt_cnt_str=`sed -n '3p' < ${all_count_result_file}`
   all_curr_event_count=$(($all_evt_cnt_str))
   total_all_event_count=$(( $total_all_event_count + $all_curr_event_count ))

   specific_evt_cnt_str=`sed -n '3p' < ${specific_count_result_file}`
   specific_curr_event_count=$(($specific_evt_cnt_str))
   total_specific_event_count=$(( $total_specific_event_count + $specific_curr_event_count ))

   window_len=$(( $curr_epoch - $prev_epoch ))
   echo "Time window interval (secs): "${window_len}
   echo "All group event count for StateChange (${prev_epoch}:${curr_epoch}]: "${all_curr_event_count}
   echo "Total all group event count: "${total_all_event_count}
   echo "Specific group event count for StateChange (${prev_epoch}:${curr_epoch}]: "${specific_curr_event_count}
   echo "Total specific group event count: "${total_specific_event_count}

   rm -f $get_count_sqlfile_specific
   rm -f $get_count_sqlfile_all
   rm -f $all_count_result_file
   rm -f $specific_count_result_file

   prev_epoch=$curr_epoch

   echo "Sleep for "$sleep_interval" secs..."
   sleep $sleep_interval
   done
```

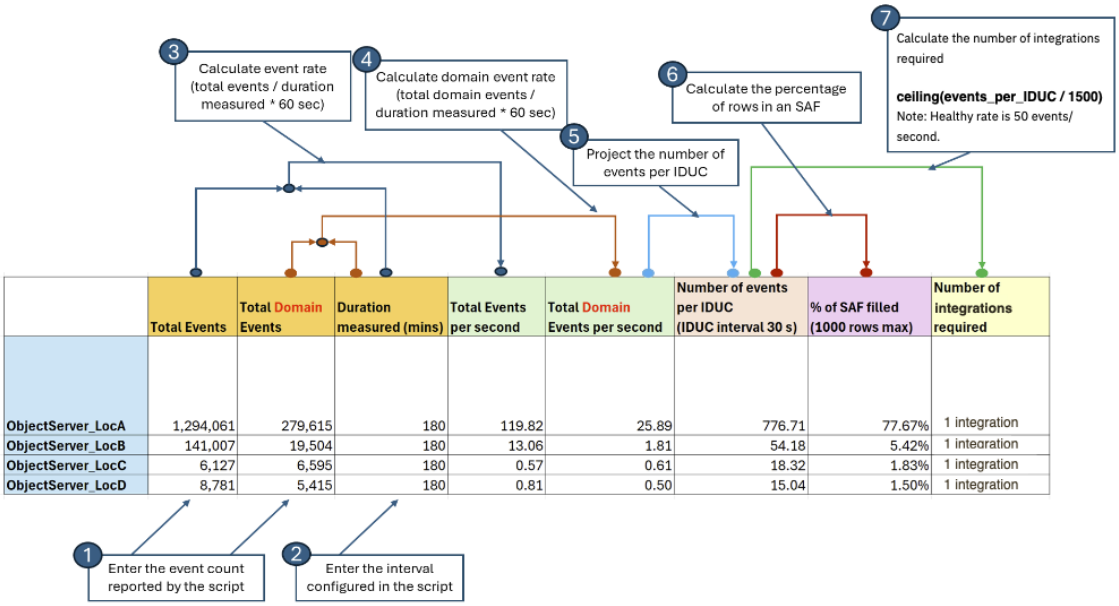
3. Run the **os_get_event_count.sh** script to collect event count statistics for all alerts and specific alerts.

   ```
   os_get_event_count.sh <ObjectServer_Name> > <output_file>
   ```

   Where

   - **<ObjectServer_Name>** is the name of your IBM Netcool Operations Insight ObjectServer. For example, **NCOMS**
   - **<output_file>** is the name of the output file where log of the script is saved. For example, **NCOMS_event_count.log**.

Calculating the number of integrations that are required for given load of domain alerts



	Total Events	Total Domain Events	Duration measured (mins)	Total Events per second	Total Domain Events per second	Number of events per IDUC (IDUC interval 30 s)	% of SAF filled (1000 rows max)	Number of integrations required
ObjectServer_LocA	1,294,061	279,615	180	119.82	25.89	776.71	77.67%	1 integration
ObjectServer_LocB	141,007	19,504	180	13.06	1.81	54.18	5.42%	1 integration
ObjectServer_LocC	6,127	6,595	180	0.57	0.61	18.32	1.83%	1 integration
ObjectServer_LocD	8,781	5,415	180	0.81	0.50	15.04	1.50%	1 integration

Figure. Number of integrations required to process the given load of domain alerts

Use the following steps to calculate the number of integrations that are required for given loads of domain alerts:

1. Enter the event count reported by the script.
2. Enter the interval configured in the script.
3. Calculate the event rate (total events / duration measured * 60 sec).
4. Calculate the domain event rate (total domain events / duration measured * 60 sec).
5. Project the number of events per IDUC.
6. Calculate the percentage of rows in an SAF.
7. Calculate the number of integrations required.

   Note: The following calculation assumes that you are using the default IDUC interval, which is 30 seconds. The formula is displayed:

   `ceiling(events_per_IDUC / 50 * IDUC interval)` => `ceiling(events_per_IDUC / 50 * 30)`

Configuring Integrations

Integrations enable IBM Cloud Pak for AIOps to collect data from various sources, then communicate how your IT operations are performing against a baseline model that is established by your AI models. To have a fully functioning instance of IBM Cloud Pak for AIOps, you must configure both outgoing and incoming integrations. It is worth noting that ServiceNow is bidirectional.

Note: Each integration that you configure requires an additional 1 vCPU and 3GB memory. For example, if you configure five integrations then you will require an additional 5 vCPUs and 15GB memory.

For more information about integrations in general, click the following links:

- Integrations UI
- Integration types

For more information about integrations in general, including at a minimum how to create, edit, and delete a particular integration, click the corresponding link in the following list:

- Ansible integrations
- Appdynamics integrations
- AWS CloudWatch integrations
- Custom integrations
- Dynatrace integrations
- ELK integrations
- Falcon LogScale integrations
- Instana integrations
- Kafka integrations
- Mezmo integrations
- Microsoft Teams integration
- Netcool integrations
- New Relic integrations

- Observer jobs
- Optional integrations
- PagerDuty integrations
- Slack integration
- ServiceNow integration
- Splunk integrations
- SSH integrations
- Zabbix integrations

For ancillary information on integrations, see:

- Prerequisites for loading data from WebSphere log data sources
- Certificate management

Integrations

Using the Integrations capability you can connect data sources to provide data that helps gather insights for your environment.

The main Integrations page now has two tabs: the *Manage integrations* tab, and the *Manage observer jobs* tab.

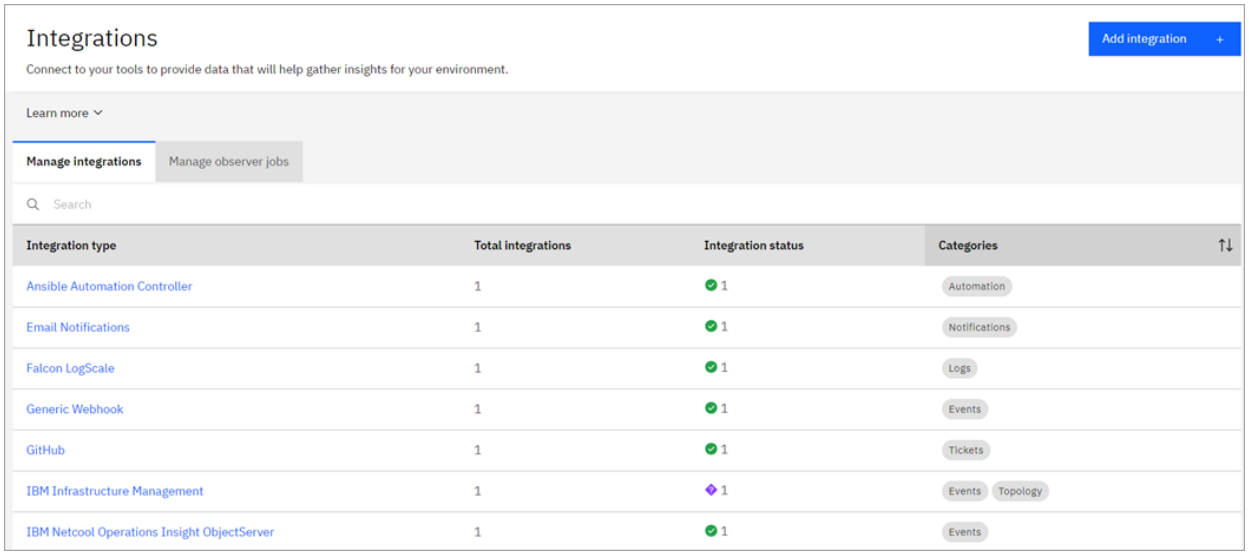


Figure. Manage integrations page

On the *Manage observer jobs* tab, you can configure, schedule and manage jobs, and use integrations and observer jobs to collect topology data.

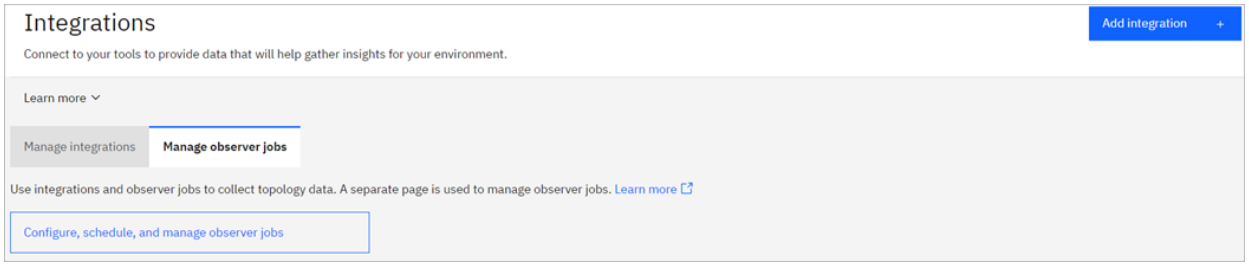
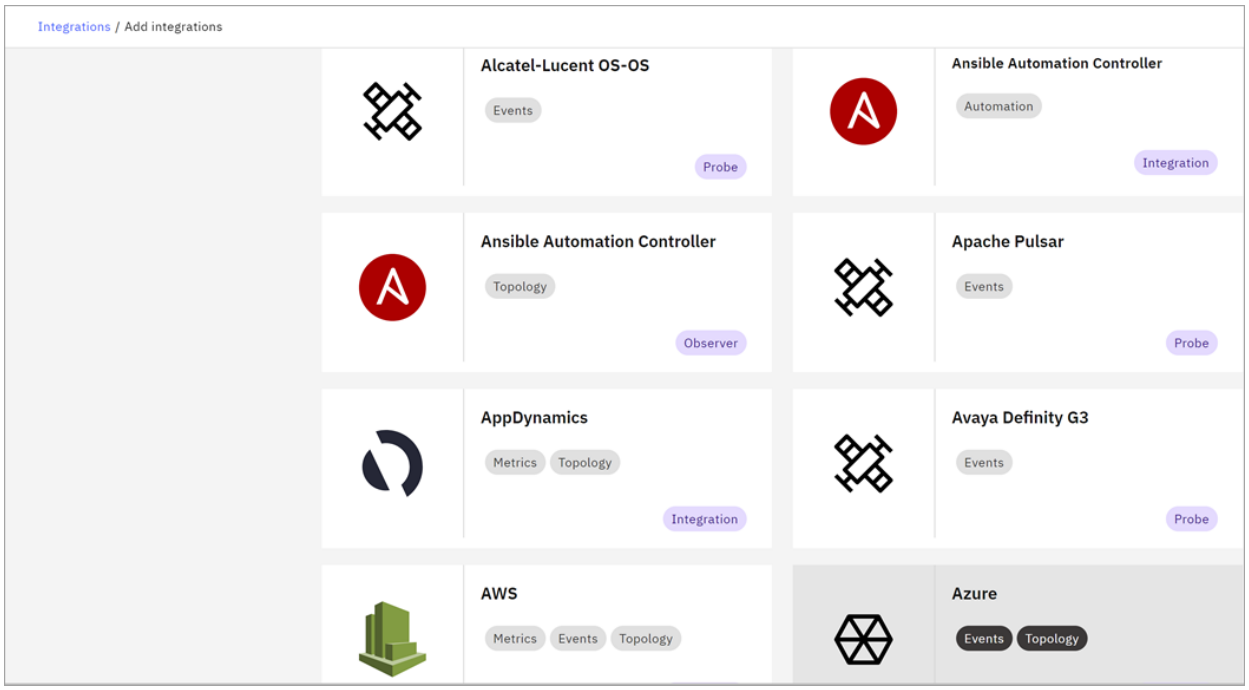


Figure. Manage observer jobs page

Incoming integrations provide data to IBM Cloud Pak for AIOps. Outgoing integrations report findings from IBM Cloud Pak for AIOps to supported ChatOps interfaces and ServiceNow. Some integrations are bidirectional. For example, to have a successfully operating instance of IBM Cloud Pak for AIOps, you might configure a Slack outgoing integration for ChatOps, a Falcon Logscsale incoming integration for log data, and a ServiceNow integration for change risk and similar ticket data.



Integrations UI

Outgoing integrations must be publicly accessible. If your cluster is behind a firewall, for example, it must be tunneled through something, such as Secure Tunnel. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps. If you previously exposed your host for one outgoing integration, you do not need to configure it for other outgoing integrations.

For more information about prerequisites for specific outgoing integrations, see the following topics:

- Configuring Microsoft Teams for integration
- Configuring Slack applications for integration
- Configuring the ServiceNow App for IBM Cloud Pak for AIOps
- Mapping data

Note: When you configure integrations for IBM Cloud Pak for AIOps, your browser might give you an option to save key, token, or password-type fields. For security purposes, do not save these values. Turn off autofill or autosave settings in your browser, or exercise caution with these features.

Integration types

Use this information to understand which data integrations to use to ingest different types of content.

Click the respective integration type that you want to learn more about creating, editing, and deleting that particular type of integration.

Integration types

Data source	Type of content supported	Link
Ansible	Runbooks, Remote Access	Ansible integrations
Appdynamics	Metrics	Appdynamics integrations
AWS CloudWatch integrations	Metrics	AWS CloudWatch integrations
Custom	Logs	Custom integrations
Dynatrace	Metrics	Dynatrace integrations
ELK	Logs	ELK integrations
Falcon LogScale	Logs	Falcon LogScale integrations
GitHub	Incident tickets	GitHub integrations
Instana	Events, Metrics, Topology	Instana integrations
Jira	Incident tickets	Jira integrations
Kafka	Logs, Events	Kafka integrations
Mezmo	Logs	Mezmo integrations
Microsoft Teams	ChatOps	Microsoft Teams integrations
Netcool	Events	Netcool integrations
New Relic	Metrics	New Relic integrations
PagerDuty	Events	PagerDuty integrations
ServiceNow	Change tickets, Incident tickets	ServiceNow integrations
Slack	ChatOps	Slack integrations
Splunk	Logs	Splunk integrations
SSH	Runbooks, Remote Access	SSH integrations

Data source	Type of content supported	Link
Zabbix	Metrics	Zabbix integrations

Outgoing integrations

Outgoing integrations enable IBM Cloud Pak for AIOps to communicate how your IT operations are performing against a baseline model that is established by you during model training. IBM Cloud Pak for AIOps uses this comparison to report anomalous activity to you in near real time.

Outgoing integrations must be publicly accessible. If your cluster is behind a firewall, for example, it must be tunneled through something, such as Secure Tunnel. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps. If you previously exposed your host for one outgoing integration, you do not need to configure it for other outgoing integrations.

When you set up your outgoing ChatOps integration, you must configure two channels: a *proactive channel* that you want to use for notifications (from change risk, for example), and a *reactive channel* that you want to use for incidents and alerts.

Note: The ServiceNow integration type has optional incoming and outgoing facets.

For more information about configuring ServiceNow, Slack, and Microsoft Teams for your integrations, see the following topics:

- Microsoft Teams integration
  - Configuring Microsoft Teams applications for integration
- Slack integration
  - Configuring Slack applications for integration
- ServiceNow integration
  - Configuring the ServiceNow App for IBM Cloud Pak for AIOps
- SSH integration
- Email integration
- Ansible integration

Creating a Microsoft Teams integration

A Microsoft Teams integration lets you and your team use collaborative ChatOps features to receive incident notifications. You can only connect to one ChatOps tool at a time, so if you already configured another ChatOps integration (such as Slack), you cannot connect to Microsoft Teams.

Prerequisites

Before you set up your Microsoft Teams integration, you must create your Azure Bot, and configure the Microsoft Teams application to communicate with IBM Cloud Pak for AIOps. For more information about configuring Microsoft Teams applications for IBM Cloud Pak for AIOps, see Configuring Microsoft Teams applications for integration.

Important: If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps or an environment that uses a proxy server, then you cannot use the ChatOps Microsoft Teams integration. The use of a proxy with the ChatOps Microsoft Team integration is not supported.

Note: If the URL of the on-premises IBM Cloud Pak for AIOps instance can't be accessed by Microsoft Teams, you can use Secure Tunnel to expose the IBM Cloud Pak for AIOps instance to either a cluster (Red Hat OpenShift or IBM Kubernetes Service cluster) or a host machine (virtual machine or physical machine), which is accessible from the public network. Then, you can get a URL that can be accessed by Microsoft Teams. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps.

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating a Microsoft Teams integration

Note: To implement your Microsoft Teams integration, you need the Client Secret and ID information available either from Microsoft Teams or Microsoft Azure. Follow the steps that are outlined in Configuring Microsoft teams application for integration to acquire these variables.

To create a Microsoft Teams integration, complete these steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Microsoft Teams** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.

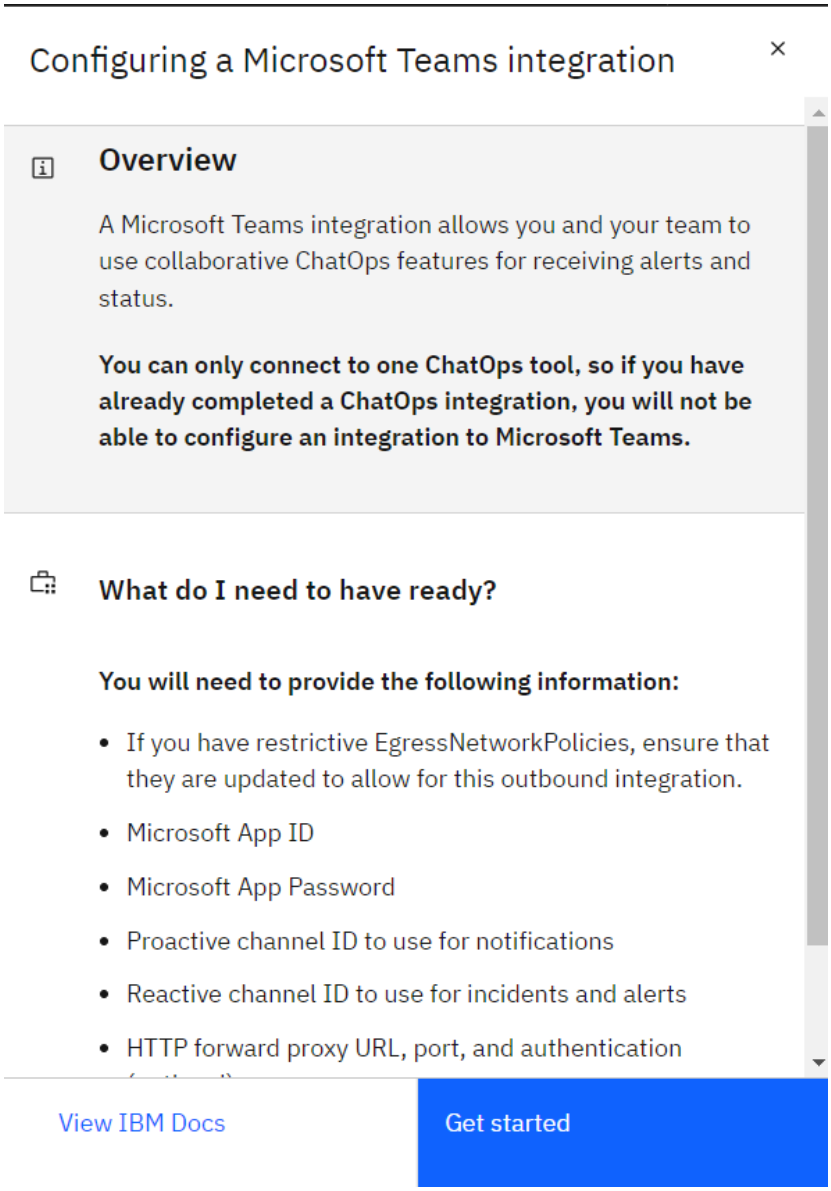


Figure. Configure Microsoft Teams integration

6. On the *Add integration* page, define the general integration details:

- **Name**: The display name of your integration.
- **Description**: An optional description for the integration.
- **Microsoft App ID**: Token for the bot that is associated with the app that is installed in a channel. You can copy this value from the *Microsoft APP ID* field on the Microsoft Teams **Settings** > **Configuration** page.
- **APP Password**: Enter the Azure Bot client secret for the integration.

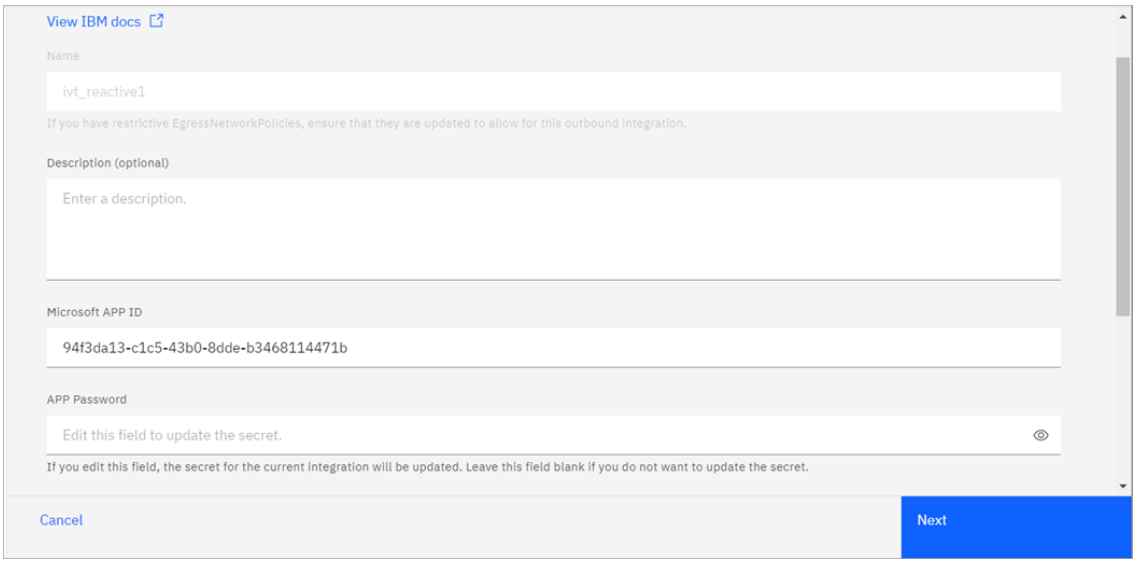


Figure. Create Microsoft Teams integration

- **Language ID**: Select your language from the drop-down list.
- **Proactive channel**: The channel ID for the channel that you want to use for notifications (from change risk, for example). For more information about obtaining channel ID values, see Obtaining links to your channels.
- **Reactive channel**: The channel ID for the channel that you want to use for incidents and alerts. For more information about obtaining channel ID values, see Obtaining links to your channels.

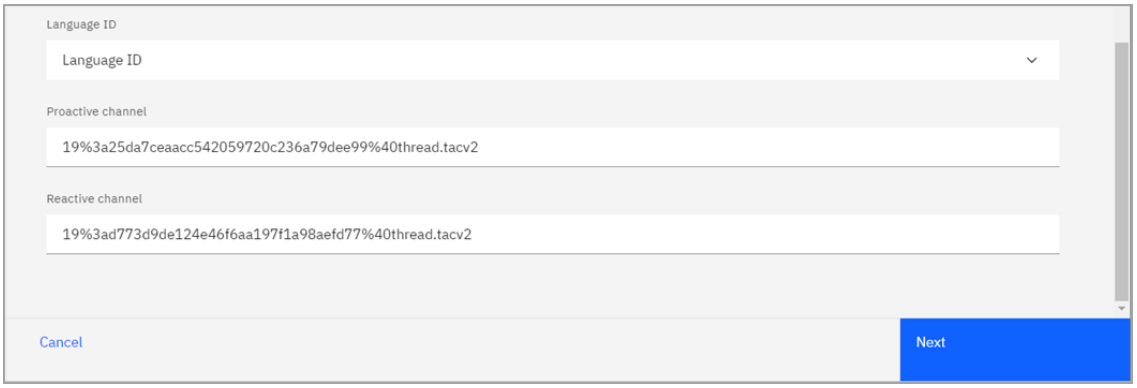


Figure. Microsoft Teams integration

Click **Next**.

Important: On the optional **Configure Proxy** page, do not toggle the *Enable Proxy* selection to configure a proxy. This does not work for Microsoft Teams integrations. For more information, see Known Issues.

You can test your integration by clicking **Test connection**.

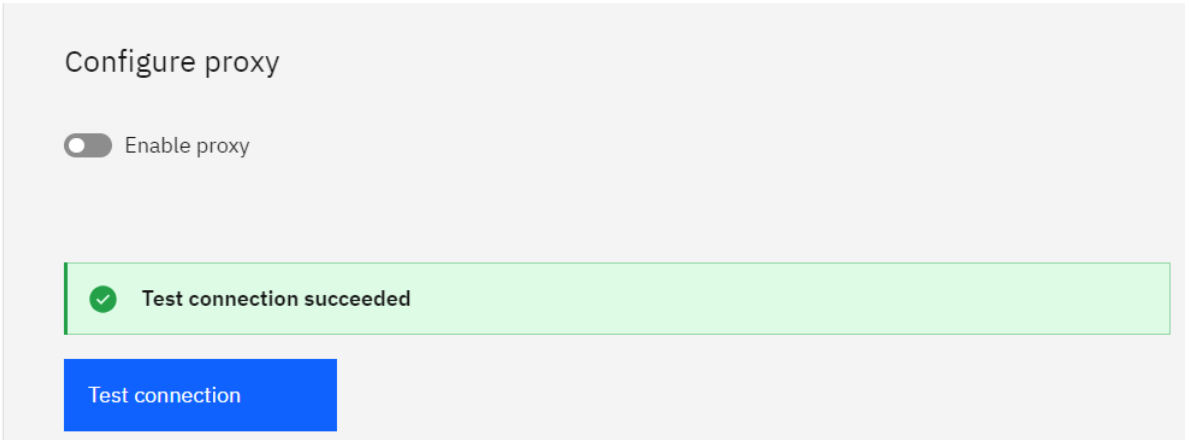


Figure. Test connection

Note: Your secret must be fully registered on Microsoft Azure for the integration to pass. If you test your integration immediately after creating the secret, wait a couple of minutes, then try again.

7. Click **Save**.

You have created a Microsoft Teams integration in your instance. Unlike a log or event integration, you do not need to enable a ChatOps integration after creating it. Once you create the integration and set up Microsoft Teams, you can verify that everything is working by issuing test commands in Microsoft Teams. For more information about verifying your integration, see Enabling your bot and adding it to your channel.

To create more integrations, see Configuring Integrations.

For more information about working with the insights provided by your integrations, see ChatOps insight management.

Connecting the Azure Bot to IBM Cloud Pak for AIOps

Before you can finish implementing your Azure Bot in Microsoft Teams, you must create your integration at the instance level in IBM Cloud Pak for AIOps. Creating an integration in IBM Cloud Pak for AIOps provides the URL for the messaging endpoint. After this step, you can edit the Microsoft Teams integration to complete setting it up, but the URL for the messaging endpoint is required to complete your Azure Bot registration.

1. Create a Microsoft Teams integration.
2. From the Cloud Pak for AIOps console, click the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Microsoft Teams** integration type in the *Manage integrations* tab. On the *Microsoft Teams* page, copy the request URL from the *Details* column. This URL is required to enable two-way communication between Microsoft Teams and IBM Cloud Pak for AIOps.

   Note: If you used Secure Tunnel to expose the IBM Cloud Pak for AIOps for Microsoft Teams integration, you need to replace the **`request URL`** with the application mapping URL.

   To get the application mapping URL, go to **Administration** > **Secure Tunnel** from the navigation menu (four horizontal bars). On the **Manage integrations** tab, find the tunnel integration, and click the integration name. In the **Application mappings** page, copy the application mapping URL from the **IBM Cloud Pak for AIOps mapping URL** field. For more information, see Viewing Secure Tunnel.
4. Log back into your Microsoft Azure home page and navigate to the *Configuration* section of your Azure Bot resource.
5. Paste the request URL from the *Details* column of the IBM Cloud Pak for AIOps *Microsoft Teams instances* page into the *Messaging endpoint* field on the *Configuration* section of your Azure Bot resource in Microsoft Azure. Click **Save**.

Enabling your Azure Bot and adding it to your channel

After you set up your Microsoft Teams integration, you can download the Microsoft Teams Azure Bot through the instance interface in the IBM Cloud Pak for AIOps console.

Note: The same Azure Bot must be added to both your reactive and proactive channels.

To download the Azure Bot, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Microsoft Teams* tile, click **1 instance**.
4. On the *Microsoft Teams instances* page, click the options menu (three vertical dots), then click **Download bot app**. Save **`IBM_Cloud_Pak_AIOps.zip`** to an accessible local directory.
5. Log in to Microsoft Teams to upload your Azure Bot to your channel. Click **Apps** in the sidebar.
6. Click **Upload a custom app**. Select the compressed file that you downloaded. The IBM Cloud Pak for AIOps Azure Bot appears in the *Built by your org* section.
7. Click the IBM Cloud Pak for AIOps tile, then click **Add**.
8. The tile is on the *Apps* page. Click the IBM Cloud Pak for AIOps tile on the *Apps* page. The IBM Cloud Pak for AIOps details page opens.
9. Click the drop-down arrow next to *Open*, then click **Add to a team**. Select the team that you designated to be the appropriate channel to post to (either your reactive or proactive channel). Click **Set up a bot**.

You have edited your integration.

The channel that you set up opens and the Azure Bot is fully functional.

You can verify that everything is operating as expected by issuing the welcome command for the Azure Bot in Microsoft Teams:

- **`@<bot_name> /waiops_welcome`**: provides a welcome message about using the bot, where **`bot_name`** is the name of your Azure Bot.

Editing a Microsoft Teams integration

After you create your integration, you can edit the integration. For example, if you want to change which channel your proactive messages are routed to, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.

3. Click the **Microsoft Teams** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Microsoft Teams integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration. Click **Next** to go through the integration configuration pages.

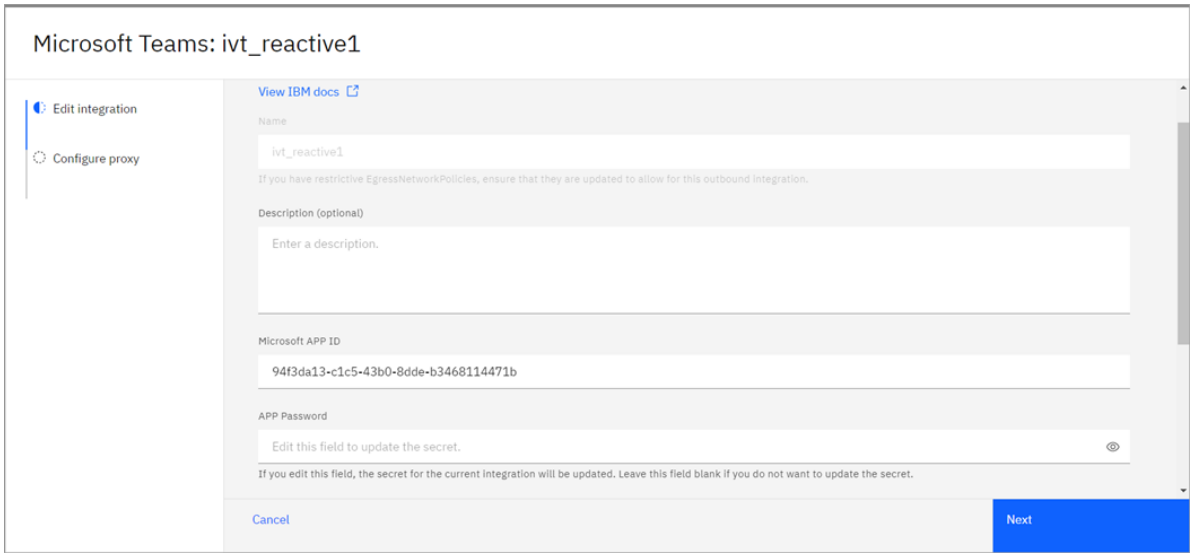


Figure. Edit Microsoft Teams integration

6. Click **Save** when you are done editing.

Note: Updating a ChatOps integration with a new Microsoft Teams Azure bot can result in disconnected ChatOps incidents. Existing incidents that were posted by a different bot do not receive any more updates. You are recommended to use the same bot resource when possible.

Deleting a Microsoft Teams integration

If you no longer need your Microsoft Teams integration, you can delete the integration from the console.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Microsoft Teams** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Microsoft Teams integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Configuring Microsoft Teams application for integration

IBM Cloud Pak for AIOps supports a direct integration to your enterprise Microsoft Teams installation.

To complete your Microsoft Teams integration in IBM Cloud Pak for AIOps, you need to obtain these credentials:

- Client secret
- Microsoft App ID
- Proactive channel ID
- Reactive channel ID

Prerequisites

- A Microsoft Azure account with the ability to create an Azure Bot and client secrets for your bot
- A Microsoft Teams account that you can create channels and add applications to

For more information about completing your Microsoft Teams integration, see Creating a Microsoft Teams integration.

Configuring Microsoft Teams

For a Microsoft Teams application integration, you must

1. Create an Azure bot resource
2. Enable Teams integration
3. Acquire integration credentials

1. Create an Azure bot resource

1. Log in to your Microsoft Azure home page.
2. Click **Create a resource**.
3. In the **Create a resource** search box, enter *Azure Bot* and select it from the list of available resources.
4. On the **Azure Bot** page, click **Create**.
5. When the create page opens, give the bot a name in the *Bot handle* field. This handle is used for tracking in Azure.
6. Next, from the dropdown menus, select a valid **Subscription** and **Resource** group.
7. In *Type of app*, select **Multi Tenant**.
8. At the bottom of the page, click **Next: Tags >**.
9. On the next page, enter any tags that you want to add in the **Tags** tab or skip and click **Review+create** tab.
10. Your submitted configuration is now validated and you can click the **Create** button.

The Azure Bot resource is now created. If you return to the Microsoft Azure home page, you see that the new Azure Bot resource is in the *Recent resources* section.

After creating and validating the Azure Bot resource, a notification appears about the deployment status. Click the notification **Deployment in progress**... to view the status of the resource, then **Go to resource** from the notification to manage your new resource.

2. Enable Teams communication

Retrieve the Azure Bot resource by going to the Microsoft Azure home page and clicking the *Recent resources* section. Alternatively, find the resource by entering the Azure Bot handle of the resource into the search box. Then, complete these steps:

1. From the resource page, click **Settings**. Go to the *Configuration* section.
2. Click **Channels**.
3. In the *Add a featured channel* section, click the Microsoft Teams icon. The *Configure Microsoft Teams* page opens.
4. Click **Save**, then accept the Terms of Service and click **Agree**.

The Azure Bot is now ready to connect to Microsoft Teams but before you implement the Azure Bot you must obtain the remaining information for your Microsoft Teams integration in IBM Cloud Pak for AIOps.

3. Acquire integration credentials

Retrieve the credentials of the Azure Bot resource by going to the Microsoft Azure home page and clicking the *Recent resources* section. Alternatively, find the resource by entering the Azure Bot handle of the resource into the search box.

From the resource page, then click **Settings** to access each of the credentials

- *Microsoft App ID*: Click **Configuration** > **Microsoft App ID**
- *Client Secret*: Click **Configuration** > **Microsoft App ID**. Click **Manage** > **New client secret**. (**Note:** This information is viewable only once, so make sure you copy the actual secret and not the ID.)

Proactive & Reactive Channel IDs: The instructions for both are the same, just with two different channels:

1. Log in to Microsoft Teams.
2. Click the options icon (three horizontal dots) next to the channel name for the channel that you want to use for your integration (proactive or reactive). Click **Get link to the channel**.
3. The channel ID is not the entire URL provided by the preceding step. To extract the channel ID from the URL, grab the string between channel/ and the next /.

   Example: `https://teams.microsoft.com/l/channel/19%3a7b1d2sldfglkjsdfhg73402d63%40thread.tacv2/...` In this example, the channel ID would be: `19%3a7b1d2sldfglkjsdfhg73402d63%40thread.tacv2`.

Now that you have created your Azure Bot and noted down the credentials, you have the required information for Creating a Microsoft Teams integration.

Creating a Slack integration

A Slack integration allows you and your team to use collaborative ChatOps features for receiving incident alerts and status. You can only connect to one ChatOps tool at a time, so if you already configured another ChatOps integration (such as Microsoft Teams), you won't be able to connect to Slack.

On the *Manage integrations* tab you can create, edit, and delete integrations.

Before you set up your Slack integration, you must create your bot and configure Slack to communicate with IBM Cloud Pak for AIOps. For more information about configuring a Slack application, see Configuring Slack applications for integration.

Note: If the URL of the on-premises IBM Cloud Pak for AIOps instance can't be accessed by Slack, you can use Secure Tunnel to expose the IBM Cloud Pak for AIOps instance to either a cluster (Red Hat OpenShift or IBM Kubernetes Service cluster) or a host machine (virtual machine or physical machine), which is accessible from the public network. Then, you can get a URL that can be accessed by Slack. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps.

For more information about working with a Slack integration, see the following sections:

- Creating a Slack integration
- Editing a Slack integration
- Deleting a Slack integration

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating a Slack integration

To complete your Slack integration, you need the following information:

- Signing secret
- Bot User OAuth Access Token
- Proactive channel ID
- Reactive channel ID
- If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps , then you must have the details of a proxy server that you have set up for IBM Cloud Pak for AIOps to connect to Slack.

Obtaining these values require you to have the following prerequisites:

- Slack
- API authentication to your Slack instance
- If you have an online deployment of IBM Cloud Pak for AIOps, network connectivity between the IBM Cloud Pak for AIOps instance and the Slack Enterprise edition instance

  Note: If a direct endpoint cannot be established, you can use Secure Tunnel to securely tunnel between the two. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps.

Important: You can only create one ChatOps integration (Microsoft Teams or Slack) per instance.

For more information about setting up and configuring Slack for integration, see Configuring Slack for integration.

To create a Slack integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Slack** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: The display name of your integration. If you have any restrictive EgressNetworkPolicies in place, ensure that they are updated to allow for this outbound integration.
   - **Description**: An optional description for the integration.
   - **Signing secret**: The secret that is generated for the client. You can copy this value from the *App Credentials* section inside the *Basic Information* page of the Slack application. For more information about obtaining this value, see Configuring your Slack app. If you edit this field, the secret for the current integration is updated. Leave this field blank if you do not want to update the secret.
   - **Bot token**: Token for the bot that is associated with the app that is installed in a workspace. You can copy this value from the *Install App* section in the Slack application. For more information about obtaining this value, see Configuring your Slack app. If you edit this field, the secret for the current integration is updated. Leave this field blank if you do not want to update the secret.
   - **Language ID**: A dropdown list offering various different languages options for the integration.
   - **Proactive channel**: Enter a proactive channel for the integration. The channel ID for the channel that you want to use for notifications (from change risk, for example). For more information about obtaining channel ID values, see Obtaining links to your channels.
   - **Reactive channel**: Enter a reactive channel for the integration. The channel ID for the channel that you want to use for incidents and alerts. For more information about obtaining channel ID values, see Obtaining links to your channels.
7. Click **Next**.

   On the optional *Configure Proxy* page, toggle the *Enable Proxy* selection to on if you require a proxy server to connect to Slack, and enter the following information:

   - **Proxy URL**: The hostname of your proxy server
   - **Proxy Port**: The port number
   - **Proxy username**: The proxy server username
   - **Proxy password**: The proxy server password



Figure. Create Slack integration - configure proxy

You can test your integration by clicking **Test connection to Slack.com**.

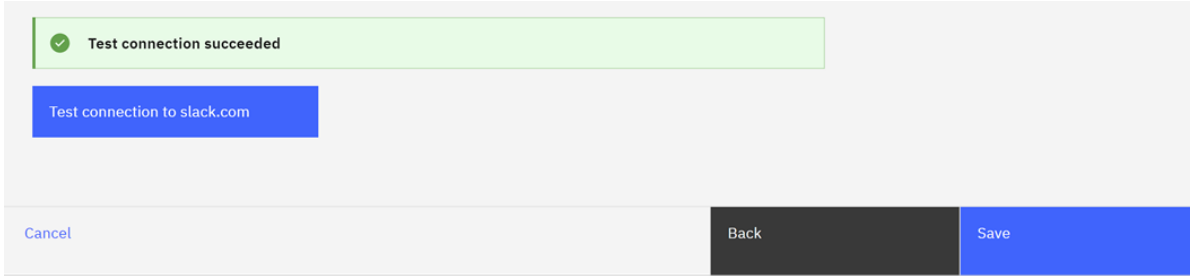


Figure. Test connection

8. Click **Done**.

You create a Slack integration in your instance. Unlike log or event integration, you do not need to enable ChatOps integrations after creating them.

To create more integrations, see Configuring Integrations.

For more information about working with the insights provided by your integration, see ChatOps insight management.

Editing a Slack integration

After you create your integration, your can edit the integration. For example, if you want to change which channel your proactive messages are routed to, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Slack** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Slack integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.

   Note: The integrations status in the *Manage integrations* tab might not display the status of your Slack integration. When you are editing a Slack integration, click the **Test connection** button to ensure that your credentials are correct.

   You can also test your integration by going to the target channel in Slack and entering `/welcome`. IBM Cloud Pak for AIOps responds with a welcome message in Slack.
5. Edit your integration. Click **Next** to go through the integration configurationn pages. Click **Save** when you are done editing.

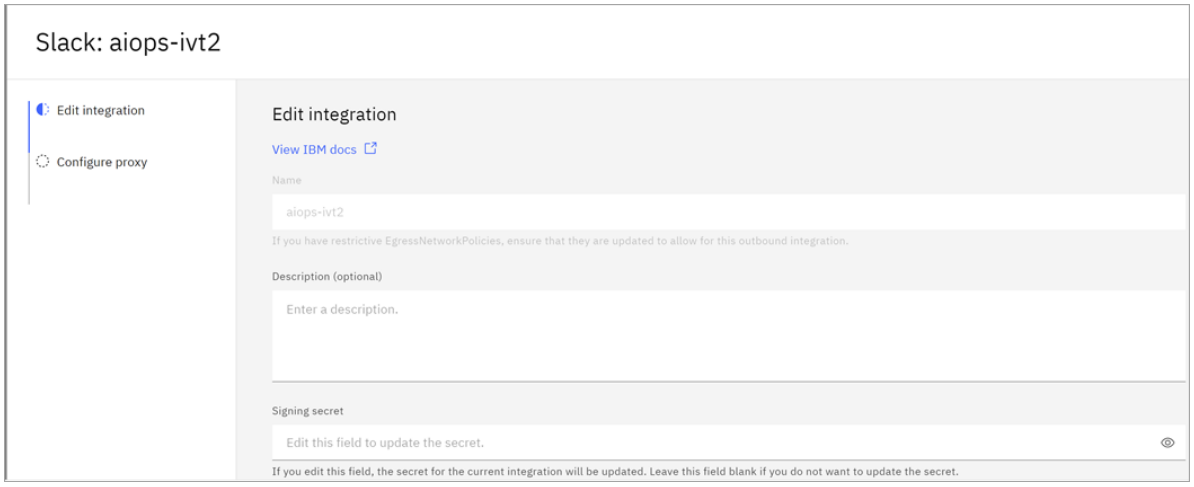


Figure. Edit Slack integration

Your integration is now edited. For more information about deleting a integration, see Deleting a Slack integration.

Note: Updating a ChatOps integration with a new Slack app can result in disconnected ChatOps incidents. Existing incidents that were posted by a different app do not receive any more updates. You are recommended to use the same app resource when possible.

Deleting a Slack integration

If you no longer need your Slack integration, you can delete the integration from the console.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Slack** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Slack integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Configuring Slack applications for integration

IBM Cloud Pak® for AIOps supports a direct integration to your enterprise Slack installation.

Prerequisites

The following requirements are necessary to setting up a Slack integration:

- Slack
- API authentication to your Slack instance
- A valid certificate (not self-signed) for your publicly exposed Red Hat OpenShift Container Platform route
- If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps , then you must have the details of a proxy server that you have set up for IBM Cloud Pak for AIOps to connect to Slack.
- If you have an online deployment of IBM Cloud Pak for AIOps, network connectivity between the IBM Cloud Pak for AIOps instance and the Slack Enterprise edition instance.

  Note: If a direct endpoint cannot be established, you can use Secure Tunnel to securely tunnel between the two. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps.

Creating your Slack app

Complete the following steps to configure your Slack integration.

Note: Be careful which workspace that you use. You must either be a workspace administrator – or wait for approval from the workspace administrator – when you interact with some of Slack's administrative settings.

Important: IBM Cloud Pak for AIOps requires two Slack channels, one for *proactive* notifications and one for *reactive* notifications. You must configure two channels, one for each purpose.

1. Identify the Slack channel that you want to connect:
    1. Create or use an existing Slack channel.
    2. Retrieve the **`channel ID`**. Right-click the channel name, then click **Copy Link** to get the web URL link. The last part of the link is the **`channel ID`**, starting with *C* for public channels or *G* for private channels.
2. Create a Slack app on https://api.slack.com/apps
    1. Click **Create New App**.
    2. Enter an App Name, and specify the workspace that you want to connect the app to, then click **Create App**.

Configuring your Slack app

1. After you create your Slack app from the Slack API page, click the name of the app that you created, then click **Features** > **App Manifest**.
2. Replace the contents of the manifest file (YAML) with the following (make sure you update the **`<app name>`** to your own app name) and save the changes.

```
_metadata:
   major_version: 1
   minor_version: 1
display_information:
   name: <app name>
features:
   bot_user:
         display_name: <app name>
         always_online: false
oauth_config:
   scopes:
      user:
         - channels:history
         - chat:write
         - groups:history
      bot:
         - app_mentions:read
         - channels:manage
         - channels:read
         - chat:write
         - files:write
         - groups:read
         - groups:write
         - users:read
         - users:read.email
         - commands
         - mpim:read
settings:
   org_deploy_enabled: false
```

   This manifest adds the following bot token to your application:

   - **`app_mentions:read`**
   - **`channels:manage`**
   - **`channels:read`**
   - **`chat:write`**
   - **`files:write`**
   - **`groups:read`**
   - **`groups:write`**
   - **`users:read`**
   - **`users:read.email`**
   - **`commands`**

   To validate your changes were saved, you can navigate to **Features** > **OAuth, & Permissions** and review the *Bot Scopes* section.
3. Go to the *Settings* > *Install App* page and go to the OAuth Tokens for Your Workspace section.
    1. Click **Install to Workspace**.
    2. Review the information and click **Allow**. Note that you must reinstall your application every time you change the scope.
    3. Record the Bot User OAuth Token found in the *OAuth Tokens for Your Workspace* section.
4. On the **Settings** > **Basic information** page, go to select *App Credentials* section and record the *Signing Secret* value.
5. Using your Slack desktop application, create and name two Slack channels under your Slack workspace:
    - Proactive channel (for example, **`aiops-proactive`**)
    - Reactive channel (for example, **`aiops-reactive`**)

   Retrieve the channel ID for both channels. Right-click the channel name, then click **Copy link** to get the web URL link. The last part of the link is the channel ID (Such as **`C020EA7QWLS`**). You need these values in following steps.
6. Log in to the IBM Cloud Pak for AIOps console.
7. From the main navigation menu (four horizontal bars in the upper-left), expand **Define**, and click **Integrations**. On the *Integrations* page, click **Add integration**.
8. Select *ChatOps* in the *Category* list (on the left), and click the **Slack** tile. On the side-panel, review the instructions and when ready to continue, click **Next**.
9. Enter the following information on the Slack integration definition page:
    - **`Name`**: The display name of your integration (Such as Slack Chat Ops)

- **`Description`**: An optional description for the integration.
- **`Signing secret`**: The signing secret generated for the client. You can copy this value from the App Credentials section inside the *Basic Information* page of the Slack application.
- **`Bot token`**: This is the token value for the bot that you recorded earlier. You can copy this value from the *Settings* > *Install App* page section in the Slack application.
- **`Language ID`**: A dropdown list offering various different languages options for the integration.
- **`Proactive channel`**: The channel ID for the channel that you want to use for notifications. You recorded this value in step 5.
- **`Reactive channel`**: The channel ID for the channel that you want to use for incidents and alerts. You recorded this value in step 5.

On the optional **Configure Proxy** page, toggle the *Enable Proxy* selection to on if you require a proxy server to connect to Slack, and enter the following information:

- **`Proxy URL`**: The hostname of your proxy server
- **`Proxy Port`**: The port number
- **`Proxy username`**: The proxy server username
- **`Proxy password`**: The proxy server password

10. Test your Slack integration integration by clicking **Test connection**. This test verifies that IBM Cloud Pak for AIOps can communicate with Slack (though it does_not_verify if Slack can communicate with IBM Cloud Pak for AIOps). If the test fails, review all preceding steps described previously.

11. Click **Done**.

12. From the Slack application page, go to *Features* > *App Manifest*. Complete the following updates to the contents of the manifest YAML.

    - Under the *settings* element, add the following child items. Ensure that you update the **placeholders**. The **`<request URL>`** value is the URL that you get from the console on the Integrations. For example, **`https://myibm.com/aiops/aimanager/instances/1647499820476511/api/slack/events`**. You can also get the value from the secure tunnel section Using Secure Tunnel for ChatOps.

      ```
      event_subscriptions:
         request_url: <request url>
         bot_events:
            - app_mention
            - member_joined_channel
      interactivity:
            is_enabled: true
            request_url: <request url>
      ```

      `event_subscriptions` and **`interactivity`** are at the same level in the configuration as the **`org_deploy_enabled`** and **`socket_mode_enabled`** elements.

    - Under the **`features`** element, add the following child item. Ensure that you update the **placeholders**.

      ```
      slash_commands:
         - command: /welcome
            url: <request url>
            description: Welcome to IBM Cloud Pak for AIOps
            should_escape: false
      ```

      `slash_commands` is at the same level in the configuration as the **`bot_user`** element.

13. Verify the Request URL. Click **Click here to verify link**. If this URL verification is not successful, review all preceding steps and ensure that the Request URL is correct.

14. Using the Red Hat OpenShift command line, run the following commands for the changes to take effect:

    ```
    oc project <namespace>
    ```

    Where **`<namespace>`** is the namespace where IBM Cloud Pak for AIOps is installed.

    ```
    oc set env deployment/$(oc get deploy -l app.kubernetes.io/component=chatops-slack-integrator -o
    jsonpath='{.items[*].metadata.name }') SLACK_WELCOME_COMMAND_NAME=/welcome
    ```

15. After the Slack application is configured, it must be reinstalled, as several permissions changed. Go to *Settings* > *Install App* page and click **Reinstall to Workspace**. On the subsequent page, click **Allow**.

16. Add the Slack application to **both** channels that you created earlier, such as your *proactive* and *reactive* channels.

    - Using your Slack desktop application, open the *Details* view of the channel, click **Integration** > **Add apps**, and add the Slack application to the channel. (The channel's *Details* view is found by clicking the small down-arrow next to the channel's name in the Slack UI.)
    - After you add the application to the channel, send the message **`/welcome`** to the channel. As a response, you can get a welcome message posted to the channel, which confirms the integration was made successfully.

You can now use Slack as your ChatOps interface. For more information about using the ChatOps interface, see ChatOps insight management.

Configuring the ServiceNow App for IBM Cloud Pak for AIOps

Before you set up your ServiceNow integration in IBM Cloud Pak® for AIOps, you must set up the IBM Cloud Pak for AIOps ServiceNow App in your ServiceNow instance.

About this task

With this integration, Cloud Pak for AIOps incidents and the alerts they're comprised of, can be sent to ServiceNow. Both the incidents and alerts contain the same details as their Cloud Pak for AIOps counterparts. To set up this integration, you must set up ServiceNow and the ServiceNow integration in Cloud Pak for AIOps.

Once configured, incidents and alerts sent to ServiceNow receive live updates to relevant fields. For incidents, this includes: title, description, state, and priority values. For alerts, this includes summary, severity, state, event count, and last occurrence values. Additionally, you can set up data synchronization so that updates made from ServiceNow are sent back to Cloud Pak for AIOps. For details on configuring data synchronization, see Creating a ServiceNow integration.

Prerequisites

Before you can set up ServiceNow to connect with IBM Cloud Pak for AIOps, you need to make sure that your ServiceNow instance meets the following requirements:

- Install the IBM Cloud Pak for AIOps App into your ServiceNow instance. To get the app, go to the ServiceNow Store.
- Your ServiceNow application must be version 4.4.0 to connect with IBM Cloud Pak® for AIOps version 4.8.0. This integration is not supported for earlier versions of the ServiceNow application.

Procedure

Complete the following steps to set up ServiceNow for integration:

1. Assign the managing roles for granting access to ServiceNow.
2. Create the users that need to access and set up the integration.
3. Create the ServiceNow integration with the IBM Cloud Pak for AIOps UI

Managing roles

In ServiceNow, two preset roles can be assigned to users: **`user_admin`** and **`itil`**. These roles can be added to the existing roles in the IBM Cloud Pak for AIOps App to grant more access to ServiceNow functions.

- The **`user_admin`** role gives privileges to create new users and assign roles to users.
  - Adding this role is optional. Not having this role does not affect functioning.
- The **`itil`** role gives access to the incident and change ticket views.
  - This role is required to view the IBM Cloud Pak for AIOps incidents and change risk assessments.

1. Log in to your ServiceNow custom application.
2. In the filter field, search for **`roles`**.
3. Click **User Administration** > **Roles**.
4. Search for **`x_ibm_waiops`**, which produces a list of predefined roles.

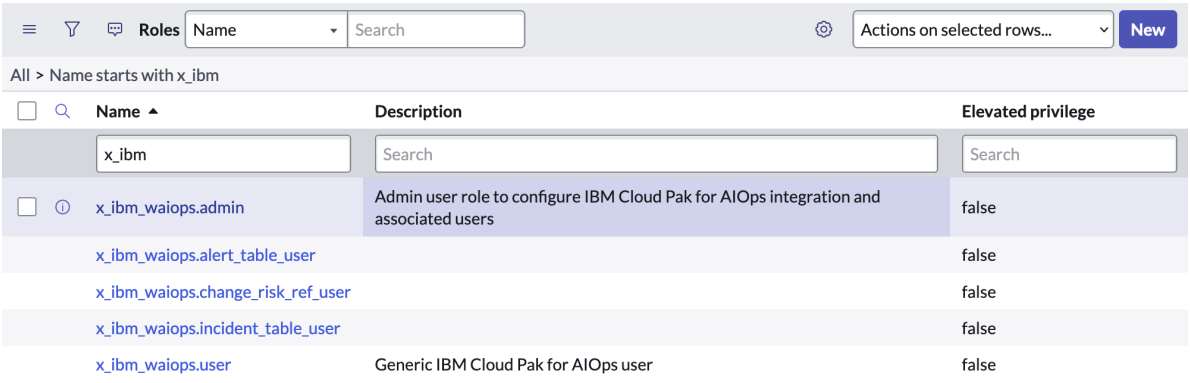


Figure. ServiceNow title list

5. Click **`x_ibm_waiops.admin`**.
6. **Optional:** Under the *Contains Roles List* box, select **Edit**, and you can add the **`user_admin`** role to the **`x_ibm_waiops.admin`** role.

   It looks like the following:

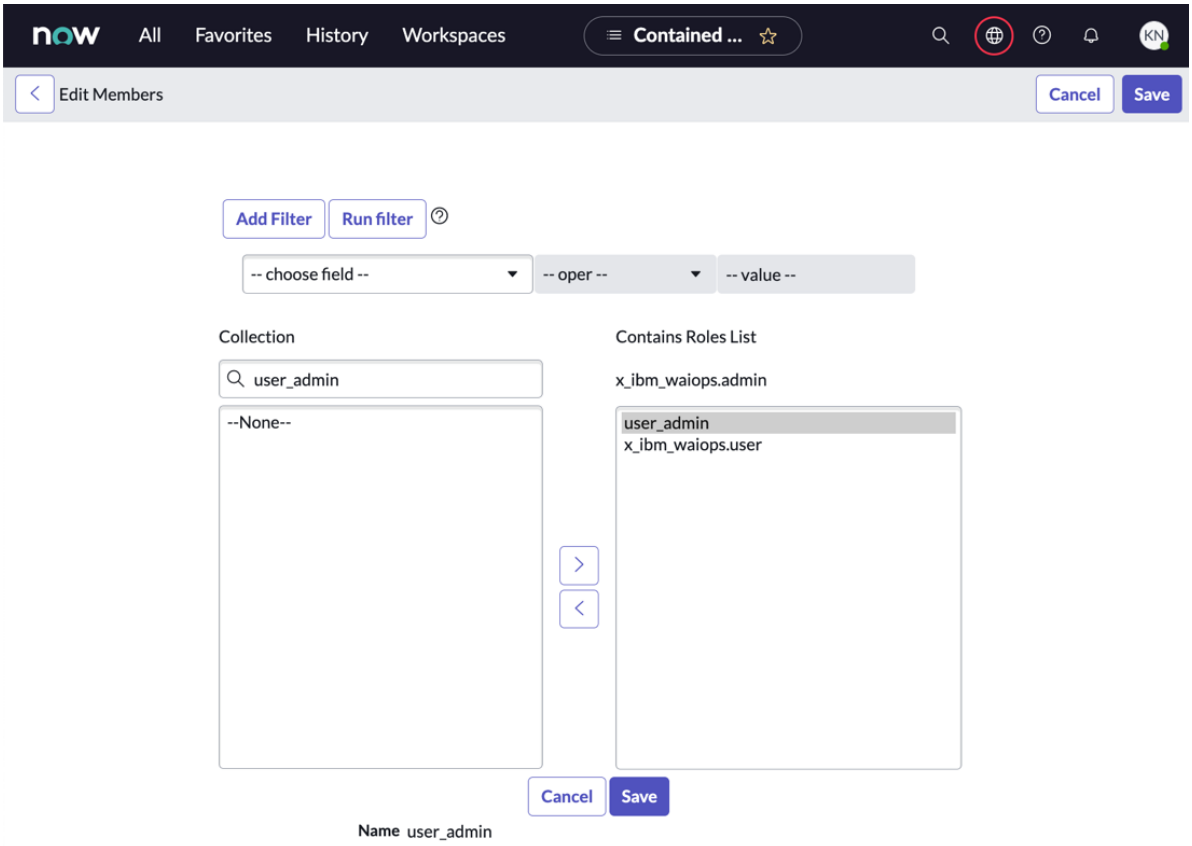


Figure. ServiceNow edit members

7. Go back to the list of predefined roles. Select **`x_ibm_waiops.user`**.
8. In the *Contains Roles List* box, select **Edit** and add the **`itil`** role to **`x_ibm_waiops.user`**. Unlike the **`user_admin`** role, this role is required.
    - Make sure that your IBM Cloud Pak for AIOps users in ServiceNow are assigned either the **`x_ibm_waiops.admin`** or the **`x_ibm_waiops.user`** role. The **`x_ibm_waiops.admin`** role includes the **`x_ibm_waiops.user`** role, so it inherits the **`itil`** role. All IBM Cloud Pak for AIOps users need the **`itil`** role to view generic incidents and change tickets. Without either the **`x_ibm_waiops.admin`** or the **`x_ibm_waiops.user`** roles, a user cannot access the augmented incidents and change tickets from the IBM Cloud Pak for AIOps ServiceNow App and receives a permissions error.
    - The other three roles available within the **`x_ibm_waiops.user`** role (all of which begin with **`x_ibm_waiops.`**) provide access to the three new tables introduced by the IBM Cloud Pak for AIOps ServiceNow App.
9. Now you see something like this:

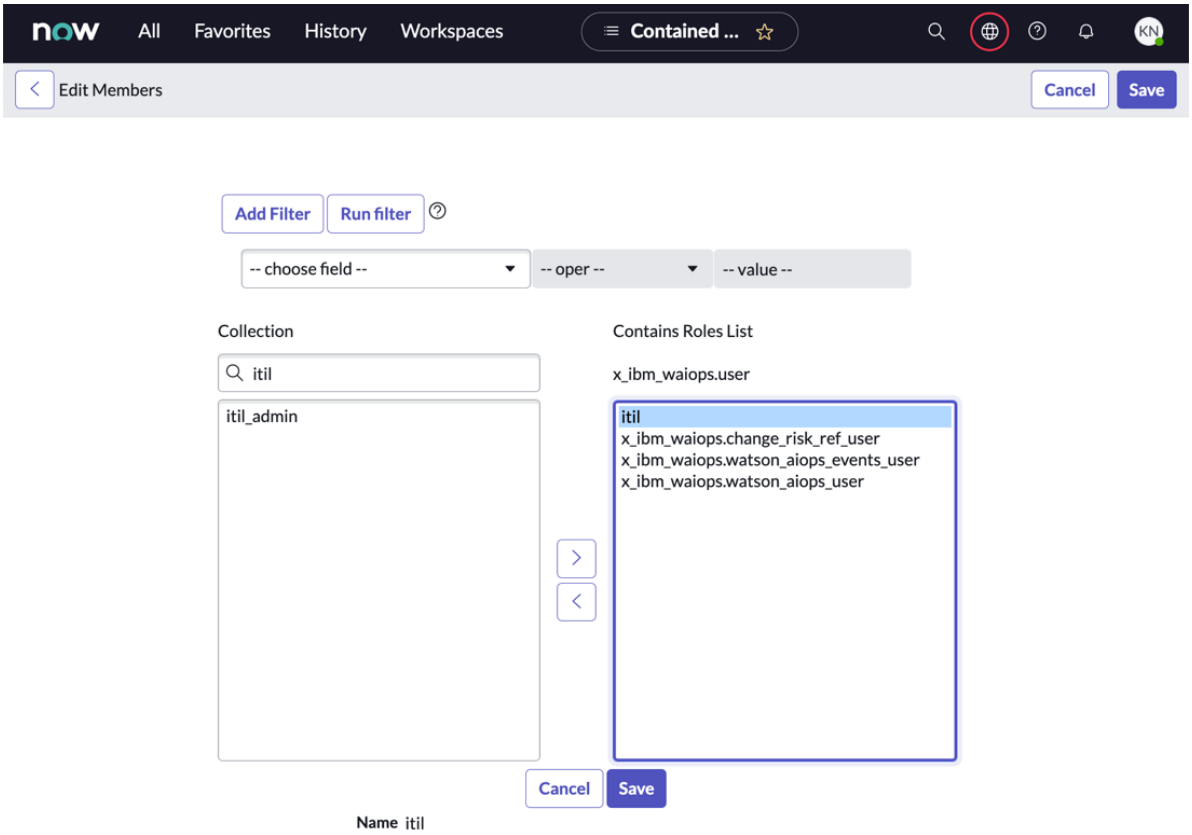


Figure. ServiceNow roles list

User roles

1. Go to **Organization** > **Users**. Create at least 2 users, an admin and a generic user to be a Site Reliability Engineer (SRE).
   - For an admin user, provide values for admin's user ID and admin's first name fields. For example, the admin's user ID can be be **`ibm.aiops`**, and the admin's first name can be **`IBM Cloud Pak for AIOps`**.
   - For the SRE user, specify values for the user ID, first, and last name.
2. For each user, set up a password by clicking **Set Password**. After you click **Set Password**, ServiceNow assigns a default password.
3. Copy and save the password. The first time that you log in with the default password, ServiceNow prompts you to set a new password. Make sure to change the default password.
4. For each user, find the **Roles** tab.

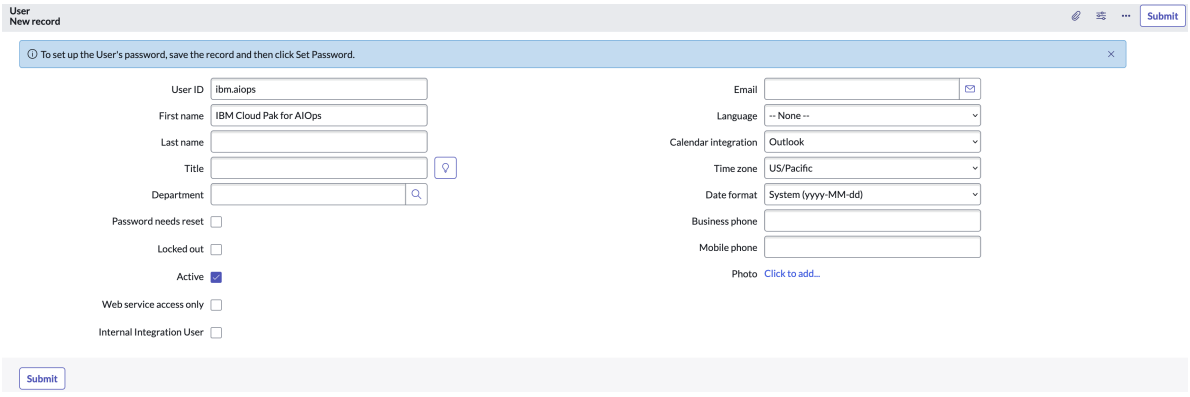


Figure. ServiceNow user creation

5. For the admin user, assign the **`x_ibm_waiops.admin`** role to be used in step 6 of Creating a ServiceNow integration.

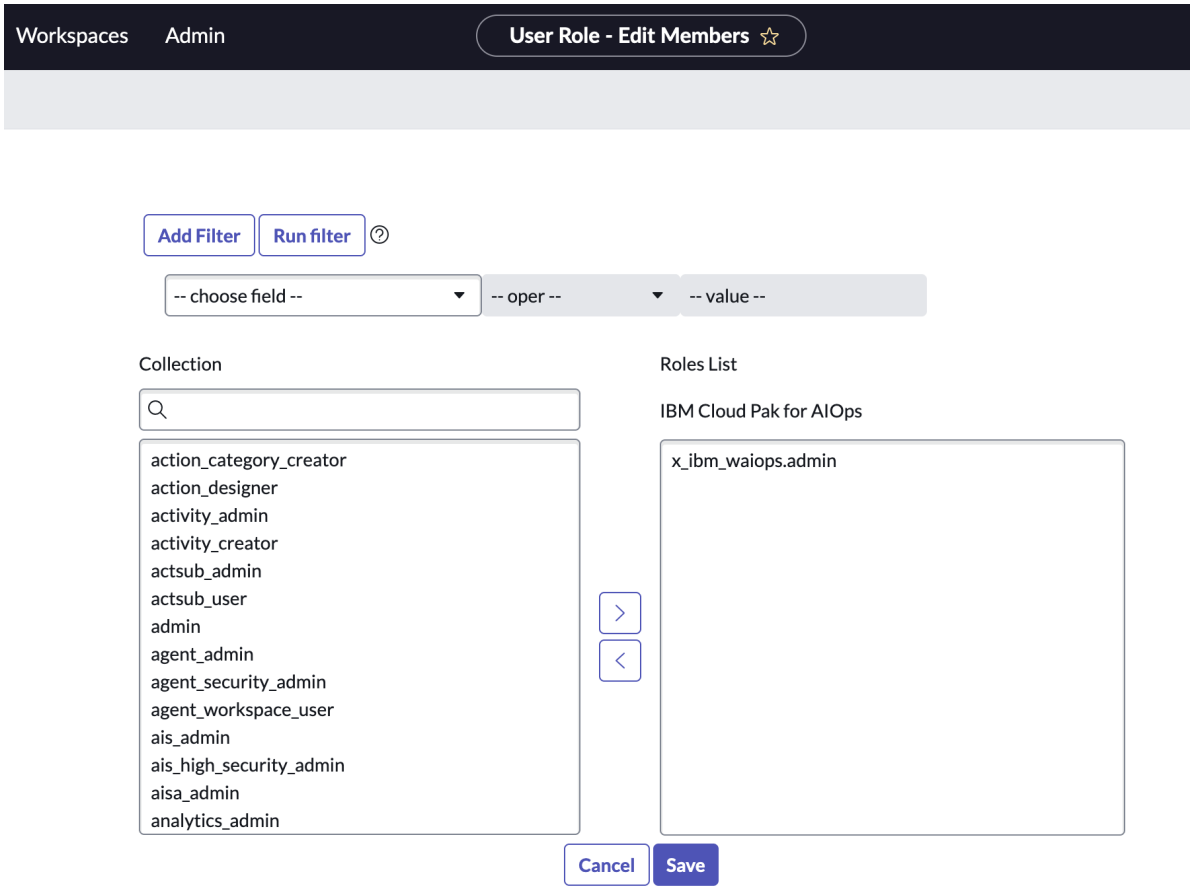


Figure. ServiceNow role assignment

This admin user will have their name, **`IBM Cloud Pak for AIOps`**, attached to the generated incidents and change ticket assessments that are created by IBM Cloud Pak for AIOps in ServiceNow.

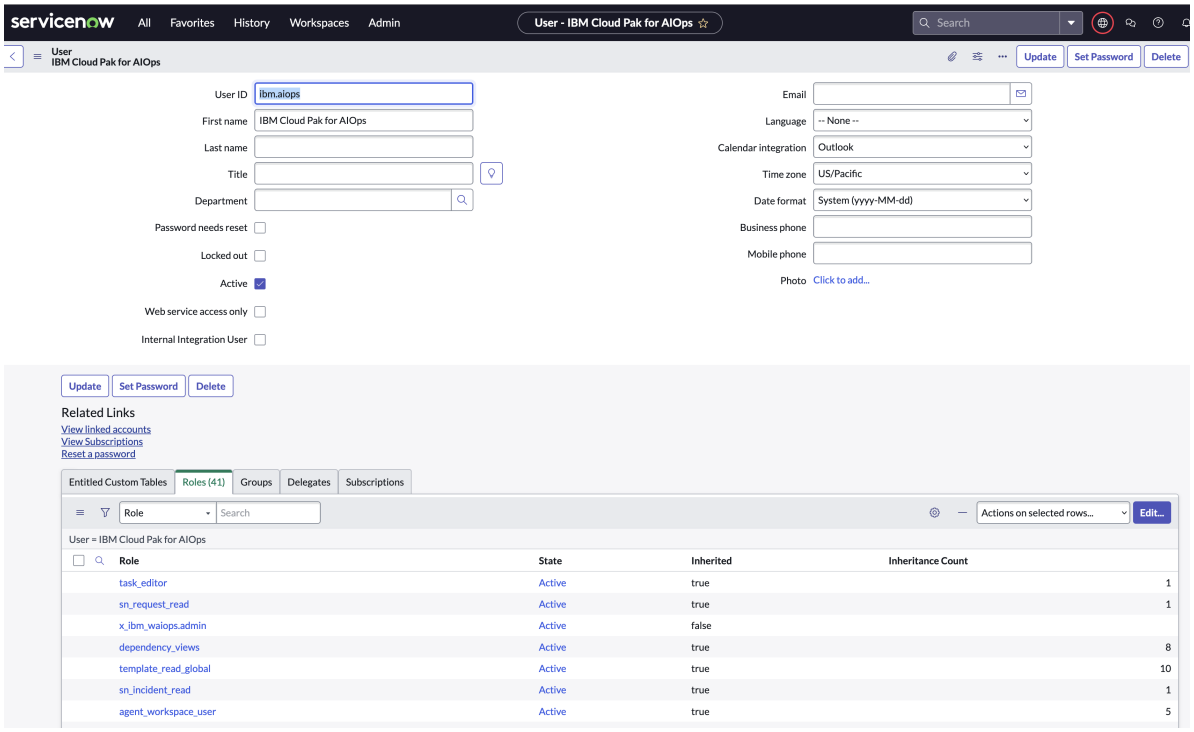


Figure. ServiceNow user roles

6. For the SRE user, assign the **`x_ibm_waiops.user`** role so that the user can view and assess ServiceNow incidents and change tickets.

Important:

- Make sure that the time zone of your *admin user* (the user value that is used to create your ServiceNow integration) matches your system time zone in ServiceNow. If the two values are not synchronized, the flow of your change request data from ServiceNow to IBM Cloud Pak for AIOps can be disrupted.

- To view the details provided by the ServiceNow integration, switch to the **IBM Cloud Pak for AIOps** view as shown in the following image:

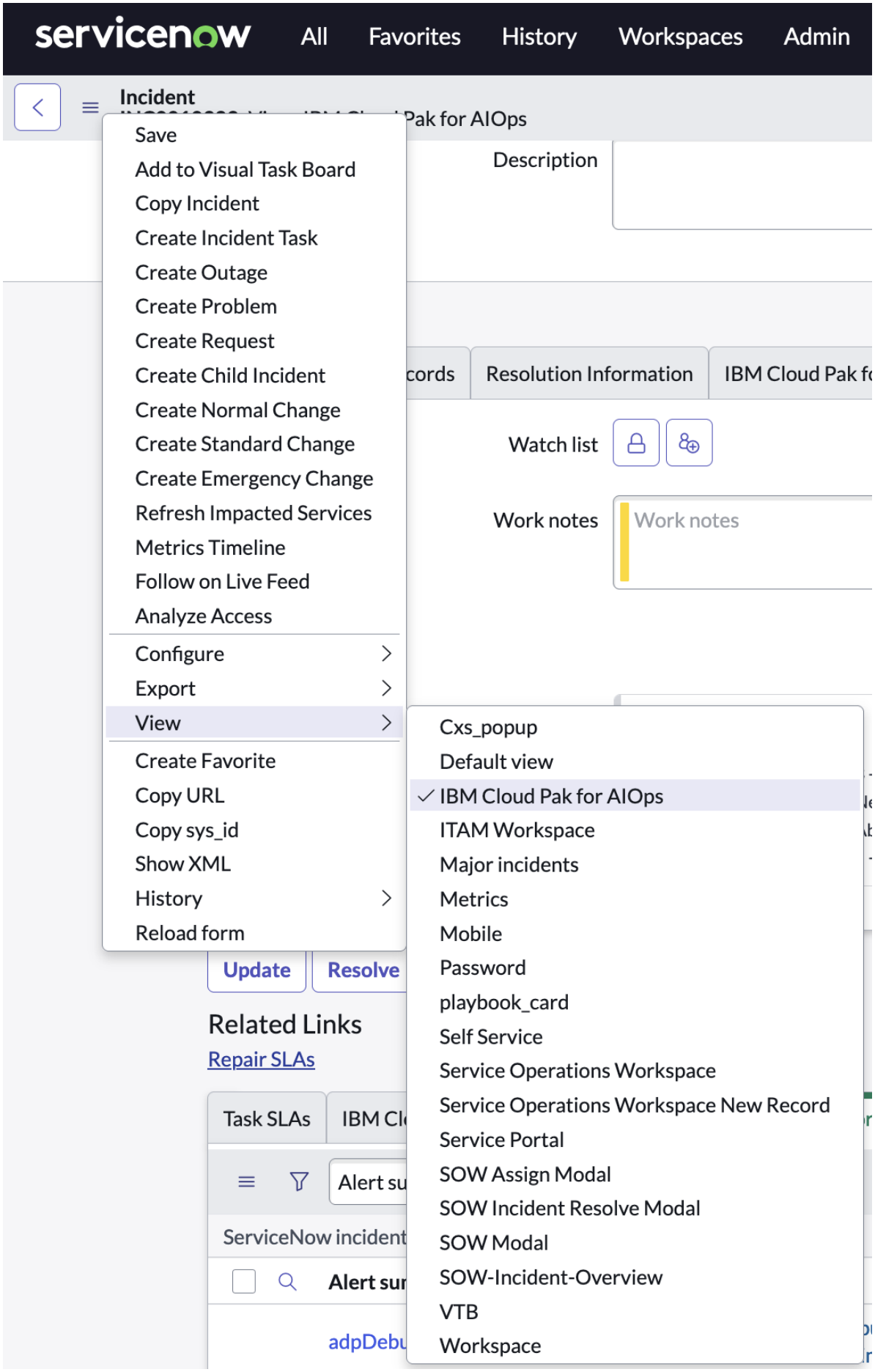


Figure. Cloud Pak View for ServiceNow integration

Create the ServiceNow integration

You also need the ServiceNow integration to be created with the IBM Cloud Pak for AIOps UI. For more information, see Creating a ServiceNow integration.

ServiceNow App configuration page

Although the **System properties for integration with IBM Cloud Pak for AIOps** page is still visible in the ServiceNow App, you do not need to configure it.

Creating a Change Risk Assessment

Complete the following steps to create and view a change request.

1. Within the IBM Cloud Pak® for AIOps WebUI, make sure ServiceNow Integration is set to the *Live data for continuous ticket data collection* mode. For more information, see step 11 of Creating a ServiceNow integration.
2. You also need a ServiceNow integration and a trained and deployed Change Risk model. For more information, see Training a Change Risk model.
3. Within your ServiceNow instance, go to **Change** > **All** to view all existing change requests.
4. Click **New** > **Create Normal Change**. Add a short description for your change. Then, click **Submit**. If ServiceNow sends you to another page, select **Change** > **All** and find the change ticket that you created.

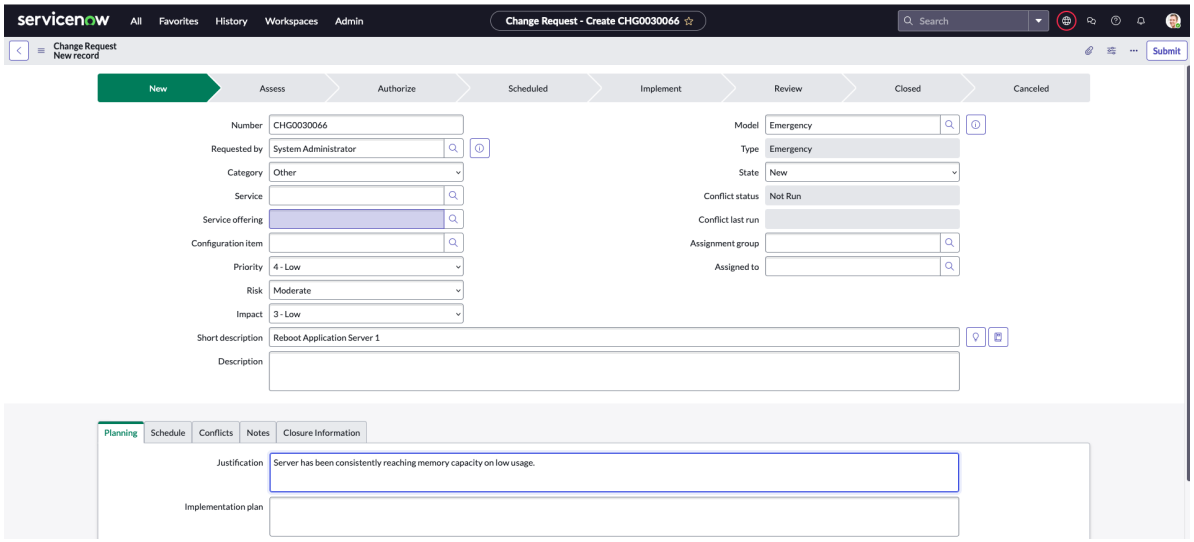


Figure. Change request short description

5. View the Change Request and the Change Risk Assessment tab.

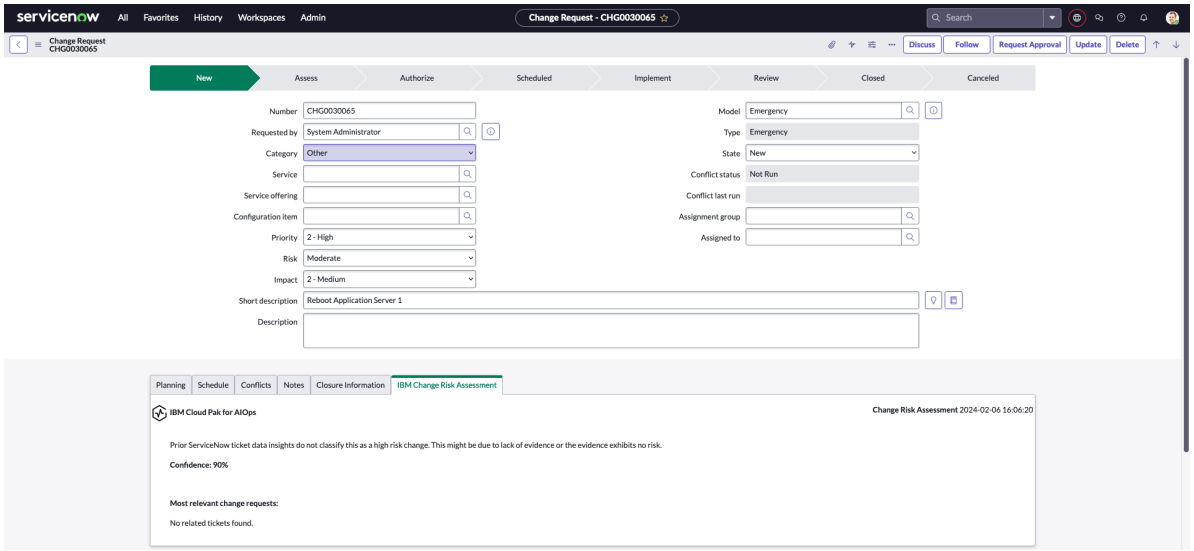


Figure. Change Request Assessment tab

ServiceNow App privacy policy

The IBM Cloud Pak for AIOps ServiceNow integration optionally collects and stores data only when the data collection is enabled. Otherwise, only the references to created records are stored in ServiceNow.

Data collection, historical data, and live data can all be enabled or disabled in the IBM Cloud Pak for AIOps ServiceNow integration UI.

Data collection and usage

Note: No data is shared with the third parties.

In ServiceNow integration the following data is collected upon enabling:

- If **historical data** is enabled, ServiceNow incident and change request records are collected and stored for AI model training with a one-time poll. Records can be excluded from the poll by specifying which ticket types need to be collected.
- If **live data** is enabled, data is collected and stored at a configurable polling interval:
- **Add risk assessments to new ServiceNow change tickets**: Change request records are collected and stored to be used for AI model training.
- **Update alerts in Cloud Pak for AIOps when corresponding incident-based alerts are updated in ServiceNow**: Alert records in ServiceNow are polled at the set interval for changes. If a relevant update exists, an alert record is sent back to IBM Cloud Pak for AIOps, and the record is not stored. For alerts, relevant updates include changes to summary, severity, state, event count, and last occurrence values.
- **Update incidents in Cloud Pak for AIOps when corresponding incident is updated in ServiceNow**: Incident records in ServiceNow are polled for changes at the set interval. If a relevant update exists, an incident record is sent back to IBM Cloud Pak for AIOps, and the record is not stored. For incidents, relevant updates include changes to title, description, state, and priority values.
- **Collect resolved ServiceNow incidents for similar tickets training**: Incident records are collected at the set interval, which is stored for AI model training, and for the similar incidents algorithm.

ServiceNow data security and management

ServiceNow data is transferred by using HTTPS protocol. The data is stored in enterprise databases, and secured within the OpenShift environment.

Note: You are responsible for the security of the IBM Cloud Pak for AIOps deployment, which includes the ServiceNow data.

Creating an SSH integration

An SSH integration provides an integration to the system where scripts and command can be run. The integration is agentless and connects directly to the target machine. It authenticates by using SSH public key authentication.

The SSH provider works for back-end systems (targets) running UNIX or Windows. For UNIX, the user running the automation must have sufficient rights to run these features.

- `bash` - A shell that is used to wrap and run the specified commands or script.
- `mktemp` - A utility that is used to create a temporary file, which is required for the script execution with this automation provider to work.
- `openssl` - A utility that is used on the target system to decrypt the transferred commands or script.

Note: You can only create one SSH integration. Also, you must have an account with administrator role to create, edit, view, or delete an SSH integration.

Before configuring you need to provide the following information:

- General integration information

If you are using a jump server (optional), you must configure it. Depending on your environment, you might require a jump server to access your target endpoints. A jump server is an SSH endpoint that is used to connect to the nested SSH endpoints. This is a common approach that is used to communicate between different network zones. To use a jump server with RBA, it must have an SSH server that is running and the `nc` command must be available. This is used to connect to nested SSH target endpoints.

Jump server configuration information:

- **Host name/IP Address:** The hostname or IP address of the jump server.
- **Port Number:** The SSH port of the jump server.
- **Username:** The username for authentication on the jump server.
- **Password:** The password for authentication on the jump server.



Figure. Create SSH integration

Any integrations to SSH target endpoints use the specified jump server.

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating an SSH integration

To create an SSH integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **SSH** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. On your target machine, register the default public key to enable access to the target endpoints through SSH for all users.

   Configuring SSH public key authentication for the UNIX root user

   The displayed public key must be added to all target machines that you plan to run scripts on through the SSH Provider. This key enables any RBA user to run script automations on the given target endpoint. The key must be added to the authorized_keys file that is usually found in the /root/.ssh/authorized_keys folder.

   Configuring SSH public key authentication for a specific UNIX user

   If you want to enforce that only a specific UNIX user can run the script on this target endpoint you should copy the key to the authorized_keys file in the home directory of the specific user, for example /home/john/.ssh/authorized_keys.

   You can regenerate the public key by clicking the **refresh** button in the upper right corner of the public key.

   Note: Regenerating the public key deletes the old key pair. If you choose to regenerate the key pair, you must exchange the public key in each target machine that you plan to access through the SSH Provider.
7. On the *Manage SSH Keys* tab that opens, you will see the SSH field default public key for SSH. You need to click the **Copy SSH key** symbol to copy this key to each target system, and append it to the authorized_keys file, for example ~/.ssh/authorized_keys.

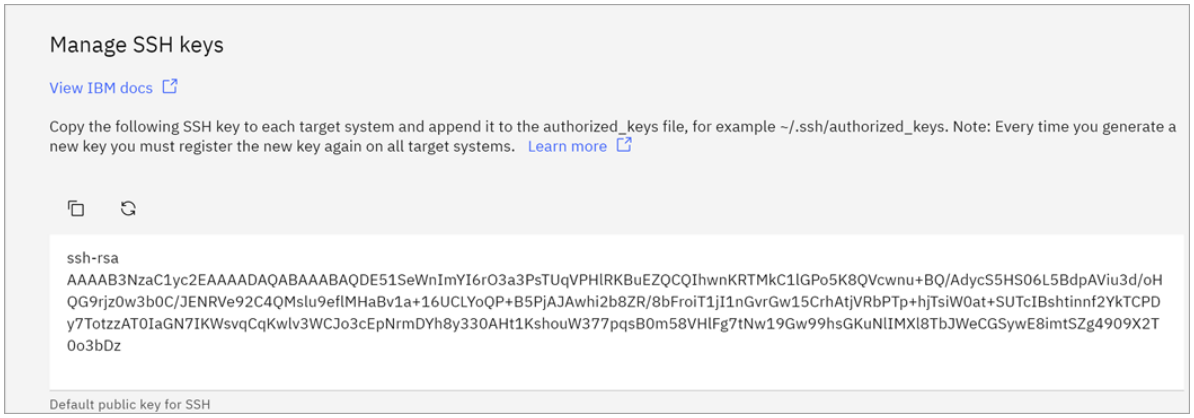


Figure. Manage SSH keys

An example of an SSH key is:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDuZkPxIyYH9NrfLEBkIGdwc6frF7WgR9vKZavE97GlTfAAZVhqoTsXO2jLo28sNC7+8wkOYnFEdfBff9tcQPx/
lc3d1df35/hJIT0a3jeHxw8YrU3/y6QIzynVvSgcQfKzB33wdN7n8xC5ZPWKXEWM7FbP58kOdzHw7f8fbBIKlPRc9SOUrC0JGvndVvpHOU7x8S3q9EJ1
D2nKaozA6yu2mcH38CLTNCBRRwbPZ+rxBxWdvJ4mMWvWtJe4lt50W2zAGCIscLKLbyyMGp/DCcJFsMhkOetBDuxAfL1ZkO7rXPT5vK5Fp6549OPDXjqf
HKEJ+9WASZD2ui1qmCdeQpUN
```

Note: Every time that you generate a new key that you must register the new key again on all target systems.

8. Click **Next**.
9. The *Define a jumpserver (Optional)* tab opens. Set the *Enable jumpserver* toggle to 'On' if you want to provide this information. The following fields show:
   - **Host name or IP address:** The hostname or IP address of the jump server.
   - **Port Number:** The SSH port of the jump server.
   - **User ID:** The username for authentication on the jump server.
   - **Password:** The Password for authentication on the jump server.
10. Click **Done**.

Editing an SSH integration

After you create your integration, your can edit the integration. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **SSH** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the SSH integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration. Click **Next** to go through the integration configurationn pages. Click **Save** when you are done editing.

Deleting an SSH integration

If you no longer need your SSH integration and want to delete it entirely, you can delete the integration from the console.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **SSH** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the SSH integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Installing new SSH key during migration

The SSH key that is associated with an SSH integration cannot be moved or copied to another deployment.

When you migrate from one deployment where you set up an SSH integration (for example, from an installation of IBM Netcool Operations Insight, or from a test installation of IBM Cloud Pak® for AIOps) to a new deployment (for example, to an installation of IBM Cloud Pak® for AIOps), then you must set up an SSH integration in the new deployment as well. This SSH integration creates its own SSH key.

Say you have installed the SSH key of the original deployment on some target systems. Now, you want to run the same SSH actions (that you migrated to the new deployment as part of the enclosing runbooks) against the same set of target systems again. You must first install the new SSH key to each of those target systems. One way of performing this administrative action is by using the sample runbooks as described in this topic.

The following procedure uses two fully automated runbooks that are delivered as sample runbooks:

- Example: Add new SSH key to target systems
- Example: Remove old SSH key from target systems

These sample runbooks have been added beginning with IBM Netcool Operations Insight 1.6.11 and IBM Cloud Pak® for AIOps 4.3.0. If the version of your original deployment does not yet contain these sample runbooks, you can import them to your original deployment. The original deployment must be at least at version IBM Netcool Operations Insight 1.6.6 or IBM Cloud Pak® for AIOps 3.5. For general instructions on exporting and importing runbooks, see Export and import runbooks.

1. Copy each of the two sample runbooks into your own new runbooks in the new deployment.
2. Export those copies of the sample runbooks from the new deployment.
3. Edit the exported JSON files and remove the `tags` property from within the `exportedAutomation` object.
4. Import the two runbooks into the original deployment.

Note:

- The scripts that are included in the actions within these runbooks require the `bash` shell to be installed on each of the target systems. If some of your target systems do not support `bash` then you need to create your own copy of the sample runbooks and their nested actions and adjust the scripts to run with `sh` only. To do this, you must modify or remove the conditional statement that uses a regular expression to do a smoke test on the SSH key.
- The scripts create a backup file of the authorized keys file that they are about to modify (for example, `authorized_keys.rbaBeforeAdd.bak` and `authorized_keys.rbaBeforeRemove.bak`).
- The two sample runbooks support Linux target systems only. You can use them as blueprints to create your own variants for target systems that use Windows Powershell.
- The two sample runbooks must be published to be executable by using the API.

Part 1: Add the new SSH key to all target systems

To get familiar with the sample runbook "Example: Add new SSH key to target systems" or if you have only a few target systems, then you can run the sample runbook from the UI of the original deployment. Each of the runbook parameters includes a description of what to specify and which format to use.

If you have many targets, then you can use the following description to define a batch of up to about 100 target hostnames and invoke the runbook by using the runbook API to add the new SSH key to each of those targets.

You must perform this part by using the original deployment ("`OLD_INSTALL`"), which has its SSH key that is already installed on all target systems. You can run the commands from any environment that has network access to your original deployment.

- **If the original deployment is IBM Cloud Pak® for AIOps**
  - Set the value of the `CURL_BASEURL_OLD_INSTALL` variable to the IBM Cloud Pak® for AIOps user interface URL and replace the path with "`/aiops/api/story-manager/rba/v1`". The following example assumes that the URL is "`https://cpd-cp4waiops.apps.mycluster.mycompany.com/zen/#/homepage`".
  - To set the value of the `CURL_AUTH_TOKEN_OLD_INSTALL` variable you must have an API key created first. If you need to generate an API key, follow the instructions in Generating API keys for authentication. The `<username>` in the command that follows is the user ID that you used to log in to the IBM Cloud Pak® for AIOps user interface when generating the API key.
  - Use `CURL_OPTIONS_OLD_INSTALL` to specify any options that you need to use for the curl command itself. For example, use "`-k`" to allow insecure SSL connections if you have not installed the deployment's certificate yet.

    ```
    export CURL_BASEURL_OLD_INSTALL="https://cpd-cp4waiops.apps.mycluster.mycompany.com/aiops/api/story-manager/rba/v1"
    export CURL_AUTH_TOKEN_OLD_INSTALL=`echo -n "<username>:<api_key_value>" | base64 -w 0`
    export CURL_AUTH_METHOD_OLD_INSTALL="ZenApiKey"
    export CURL_OPTIONS_OLD_INSTALL="-k"
    ```

- **If the original deployment is IBM Netcool Operations Insight**
  - Set the value of the `CURL_BASEURL_OLD_INSTALL` variable to the hostname where the IBM Netcool Operations Insight service is running, prefix it with "`https://`" and append "`/api/v1/rba`". For example, `https://netcool.apps.mycluster.mycompany.com/api/v1/rba`.
  - To set the value of the `CURL_AUTH_TOKEN_OLD_INSTALL` variable you must have an API key created first. If you need to generate an API key, follow the instructions in API Keys.
  - Use `CURL_OPTIONS_OLD_INSTALL` to specify any options that you need to use for the curl command itself. For example, use "`-k`" to allow insecure SSL connections if you have not installed the deployment's certificate yet.

    ```
    export CURL_BASEURL_OLD_INSTALL="https://netcool.apps.mycluster.mycompany.com/api/v1/rba"
    export CURL_AUTH_TOKEN_OLD_INSTALL=`echo -n "<api_key_name>:<api_key_password>" | base64 -w 0`
    export CURL_AUTH_METHOD_OLD_INSTALL="Basic"
    export CURL_OPTIONS_OLD_INSTALL="-k"
    ```

- **Common for all deployment types**

  Run the following command to verify the base URL and the authentication token:

  ```
  curl -X GET ${CURL_OPTIONS_OLD_INSTALL} -H "Authorization: ${CURL_AUTH_METHOD_OLD_INSTALL} ${CURL_AUTH_TOKEN_OLD_INSTALL}" "${CURL_BASEURL_OLD_INSTALL}/status"
  ```

The previous command is expected to produce some informational data about the speed of the transfer and some JSON result data similar to the following:

```
{"rbs":0,"as":0,"db":0,"connection":{"AWX":0,"IMPACT":2,"SCRIPT":0,"GITHUB":0},"errors":[],"warnings":[]}
```

Note: Some of the connection items might have a different value than "0", and the "errors" array within the JSON result data might be nonempty (if the connection is in a failed state) - this is not a problem in this context.

The values for the following set of variables must be defined as outlined here:

- `RUNBOOK_ID_ADD_SSH_KEY`: The runbook ID of the sample runbook **Example: Add new SSH key to target systems** in the original deployment.
- `PARAM_TARGET_HOSTNAMES`: A stringified array of hostnames, which defines the batch of target systems to be updated in this iteration.
- `PARAM_TARGET_USER`: The user id to be used on the target system. This user id is the same for all target systems in the batch. If different user ids need to be used, then the batches of target hostnames need to be split accordingly.
- `PARAM_AUTHORIZED_KEYS_FILE_PATH`: The relative path to the authorized keys file within the target user's home directory, or an absolute path. You can use an arbitrary file name to test the tooling (but the file needs to exist on the target). Defaults to ".ssh/authorized_keys".
- `PARAM_NEW_SSH_KEY`: The complete SSH key as copied from the SSH integration of the new deployment.
- `PARAM_OPTIONS`: Optional options for the SSH key. Must conform to the SSH specification (not validated by the tooling). Make sure to escape double quotes (see example below).
- `PARAM_COMMENT`: The comment to be saved along with `PARAM_NEW_SSH_KEY` in the `authorized_keys` file on the target systems. Must conform to the SSH specification (not validated by the tooling). Defaults to "RunbookAutomation".
- `PARAM_VERBOSE`: Set this to `true` to create verbose output for each of the editing actions on the target systems. Set this to `false` to just get a `successful` or `failed` result for the editing actions on the target systems.
- `PARAMETER_ADD_FILE_PATH`: Name of the temporary file that gets generated to contain the parameters in JSON format.

For example:

```
export RUNBOOK_ID_ADD_SSH_KEY="2f2ba8ee-4ee6-4a36-b213-2be5ec65249f"

export PARAM_RAW_TARGETS="doesnotexist1.myzone.mycompany.com"
export PARAM_RAW_TARGETS=${PARAM_RAW_TARGETS}`for i in {2..100}; do echo -n
",doesnotexist${i}.myzone.mycompany.com"; done`
export PARAM_TARGET_HOSTNAMES="[${PARAM_RAW_TARGETS}]"
export PARAM_TARGET_USER="root"
export PARAM_AUTHORIZED_KEYS_FILE_PATH=".ssh/authorized_keys_test"
export PARAM_NEW_SSH_KEY="ssh-rsa
testtesttesttestt
```

```
# Failed, reason: target system not found
doesnotexist1.myzone.mycompany.com error: Connection to target via TCP unsuccessful.
doesnotexist1.myzone.mycompany.com status: failed
```

If you have more than about 100 targets and you want to get all of your targets updated, then you must split your total list of targets into batches, and run the runbook for each batch. If you choose to write a script that loops over the set of batches, make sure to add some sleep time after you have started a batch, so the Runbook Automation service can finish the work in the background. Once you have run a few batches, you can review the actual execution time for each of them in the runbook activities view, so you can adjust the sleep time (for example, allow 30 seconds per batch).

Part 2: (Optional) Validate the new SSH key and remove the old SSH key from all target systems

You must perform this part by using the new deployment, which by now (as per part 1) has its SSH key installed on all target systems. The following steps can also be used to verify that the new key works.

Warning: Run the following steps only when the original deployment is no longer intended to run any actions on the target systems.

To get familiar with the sample runbook "Example: Remove old SSH key from target systems" or if you have only a few target systems, then you can run the sample runbook from the UI of the new deployment. Each of the runbook parameters includes a description of what to specify and which format to use.

If you want to use the runbook API to invoke the runbook, then you can run the following commands from any environment that has network access to your new deployment ("`NEW_INSTALL`").

The same comments on creating batches of target systems from part 1 apply to this part.

- The values for the following set of variables must be defined as outlined here:
  - `CURL_BASEURL_NEW_INSTALL`: Set analogously to `CURL_BASEURL_OLD_INSTALL` from part 1.
  - `CURL_AUTH_TOKEN_NEW_INSTALL`: Set analogously to `CURL_AUTH_TOKEN_OLD_INSTALL` from part 1.
  - `CURL_AUTH_METHOD_NEW_INSTALL`: Set analogously to `CURL_AUTH_METHOD_OLD_INSTALL` from part 1.
  - `CURL_OPTIONS_NEW_INSTALL`: Set analogously to `CURL_OPTIONS_OLD_INSTALL` from part 1.
  - `RUNBOOK_ID_REMOVE_SSH_KEY`: The runbook ID of the sample runbook **Example: Remove old SSH key from target systems** in the new deployment.
  - `PARAM_TARGET_HOSTNAMES`: See part 1.
  - `PARAM_TARGET_USER`: See part 1.
  - `PARAM_AUTHORIZED_KEYS_FILE_PATH`: See part 1.
  - `PARAM_OLD_SSH_KEY`: The complete SSH key as copied from the SSH integration of the old deployment.
  - `PARAM_VERBOSE`: See part 1.
  - `PARAMETER_REMOVE_FILE_PATH`: Name of the temporary file that gets generated to contain the parameters in JSON format.

  For example:

```
# If new deployment is an IBM Cloud Pak for AIOps installation
export CURL_BASEURL_NEW_INSTALL="https://cpd-cp4waiops.apps.mycluster.mycompany.com/aiops/api/story-manager/rba/v1"
export CURL_AUTH_TOKEN_NEW_INSTALL=`echo -n "<username>:<api_key_value>" | base64 -w 0`
export CURL_AUTH_METHOD_NEW_INSTALL="ZenApiKey"
export CURL_OPTIONS_NEW_INSTALL="-k"

# If new deployment is an IBM Netcool Operations Insight installation
export CURL_BASEURL_NEW_INSTALL="https://netcool.apps.mycluster.mycompany.com/api/v1/rba"
export CURL_AUTH_TOKEN_NEW_INSTALL=`echo -n "<username>:<api_key_value>" | base64 -w 0`
export CURL_AUTH_METHOD_NEW_INSTALL="Basic"
export CURL_OPTIONS_NEW_INSTALL="-k"

# Verify that the new installation can be reached
curl -X GET ${CURL_OPTIONS_NEW_INSTALL} -H "Authorization: ${CURL_AUTH_METHOD_NEW_INSTALL}
${CURL_AUTH_TOKEN_NEW_INSTALL}" "${CURL_BASEURL_NEW_INSTALL}/status"

# Set other environment variables
export RUNBOOK_ID_REMOVE_SSH_KEY="50fc56e5-55a9-429d-8c1e-7f0c59f0557f"

export PARAM_RAW_TARGETS="doesnotexist1.myzone.mycompany.com"
export PARAM_RAW_TARGETS=${PARAM_RAW_TARGETS}`for i in {2..100}; do echo -n
",doesnotexist${i}.myzone.mycompany.com"; done`
export PARAM_TARGET_HOSTNAMES="[${PARAM_RAW_TARGETS}]"
export PARAM_TARGET_USER="root"
export PARAM_AUTHORIZED_KEYS_FILE_PATH=".ssh/authorized_keys_test"
export PARAM_OLD_SSH_KEY="ssh-rsa
testtesttesttestt
```

```
curl -X POST ${CURL_OPTIONS_NEW_INSTALL} -s -H "Authorization: ${CURL_AUTH_METHOD_NEW_INSTALL}
${CURL_AUTH_TOKEN_NEW_INSTALL}" -H "Expect:" -H "Content-Type: application/json; charset=utf-8" -d
@${PARAMETER_REMOVE_FILE_PATH} "${CURL_BASEURL_NEW_INSTALL}/runbooks/${RUNBOOK_ID_REMOVE_SSH_KEY}/execute"
```

SSH algorithms

SSH actions can connect to SSH servers configured with the following algorithms.

Table. SSH algorithms

Ciphers	Compression algorithms	MAC algorithms	Key exchange algorithms	Server host key formats
`chacha20-poly1305@openssh.com` (not available when cluster is FIPS enabled)	`none`	`hmac-sha2-256-etm@openssh.com`	`curve25519-sha256` (not available when cluster is FIPS enabled)	`ssh-ed25519` (not available when cluster is FIPS enabled)
`aes128-gcm@openssh.com`	`zlib@openssh.com`	`hmac-sha2-512-etm@openssh.com`	`curve25519-sha256@libssh.org` (not available when cluster is FIPS enabled)	`ecdsa-sha2-nistp256`
`aes256-gcm@openssh.com`	`zlib`	`hmac-sha1-etm@openssh.com` (not available when cluster is FIPS enabled)	`ecdh-sha2-nistp256`	`ecdsa-sha2-nistp384`
`aes128-ctr` (not available when cluster is FIPS enabled)		`hmac-sha2-256`	`ecdh-sha2-nistp384`	`ecdsa-sha2-nistp521`
`aes192-ctr` (not available when cluster is FIPS enabled)		`hmac-sha2-512`	`ecdh-sha2-nistp521`	`rsa-sha2-512`
`aes256-ctr` (not available when cluster is FIPS enabled)		`hmac-sha1` (not available when cluster is FIPS enabled)	`diffie-hellman-group-exchange-sha256` (not available when cluster is FIPS enabled)	`rsa-sha2-256`
			diffie-hellman-group14-sha256	`ssh-rsa` (not available when cluster is FIPS enabled)
			diffie-hellman-group15-sha512	
			diffie-hellman-group16-sha512	
			diffie-hellman-group17-sha512	
			diffie-hellman-group18-sha512	

Creating email notification integrations

Use email notification integrations to send custom email messages when incidents are created and closed. Once an email notification integration is set up and running, you can select it from the Notifications area of your Promote alerts to an incident policies.

- About this task
- Prerequisites
- Fields available for inclusion in custom emails
- Creating an email integration
- Editing an email integration
- Deleting an email integration

About this task

Create one or more integrations by defining the integration details that are outlined here. You must provide a source email address (to send the incident notifications from) and the email addresses of recipients. Also, specify what incident details are sent in an email when incidents are created and closed. A default template is provided.

Prerequisites

You need the following information to create an email notification integration:

- Hostname or IP address of your SMTP server
- Port of your SMTP server
- Source (sender) email address
- SMTP username (optional)
- SMTP credentials for source email address (optional)
- Email addresses of the intended recipients
- Use of TLS to encrypt email

Note: Email notification integration supports the Simple Mail Transfer Protocol (SMTP) authentication with email addresses or with optional **SMTP username**. This approach ensures seamless integration with a wide range of email services.

Amazon Simple Email Service (SES) is supported with email notification integration. You need to use your AWS region SMTP server for example, **`email-smtp.us-east-1.amazonaws.com`**. You need to use a secure port, not 25, for example, 587. The **`Source email address`** needs to be a verified email address in your AWS SES account. The **`SMTP username`** and **`SMTP password`** fields are mandatory if they are required by Amazon SES SMTP server.

For more information about how to obtain SES SMTP credentials, see Obtaining Amazon SES SMTP credentials.

For more information about requirements to send email over SMTP by using SES, see Requirements to send email over SMTP.

Fields available for inclusion in custom emails

The following dynamic fields are availble for you to include in custom emails:

Field	Type	Description
`id`	string	A unique identifier for the incident.
`createdTime`	string	Time at which the incident was created.
`createdBy`	string	Person who created the incident.
`title`	string	Short description of the incident.
`description`	string	Longer description of the incident.
`langId`	string	ISO 639-3 language ID.
`priority`	integer	Indication of the relative impact of the issue that caused this incident to be raised.
`state`	string	The current state of the incident.
`lastChangedTime`	string	The time at which the incident last changed.
`owner`	string	The owner assigned to the incident.
`team`	string	The team assigned to the incident.
`alertIds`	array	The set of related alert id's that have triggered the creation of the incident. These alerts affect and are affected by the lifecycle of the incident. The incident will only be automatically resolved if all of the alerts are cleared. When the incident is resolved all of these alerts will be cleared.
`contextualAlertIds`	array	The set of alert id's that provide additional context to the incident due to relationships with the triggering alerts or other contextual alerts. These alerts do not affect or are affected by the lifecycle of the incident.
`insights`	array	An array of insights associated with the incident with one or more other entities or groupings.

To include a dynamic field, use the following syntax:

```
$string(incident.<field_name>)
```

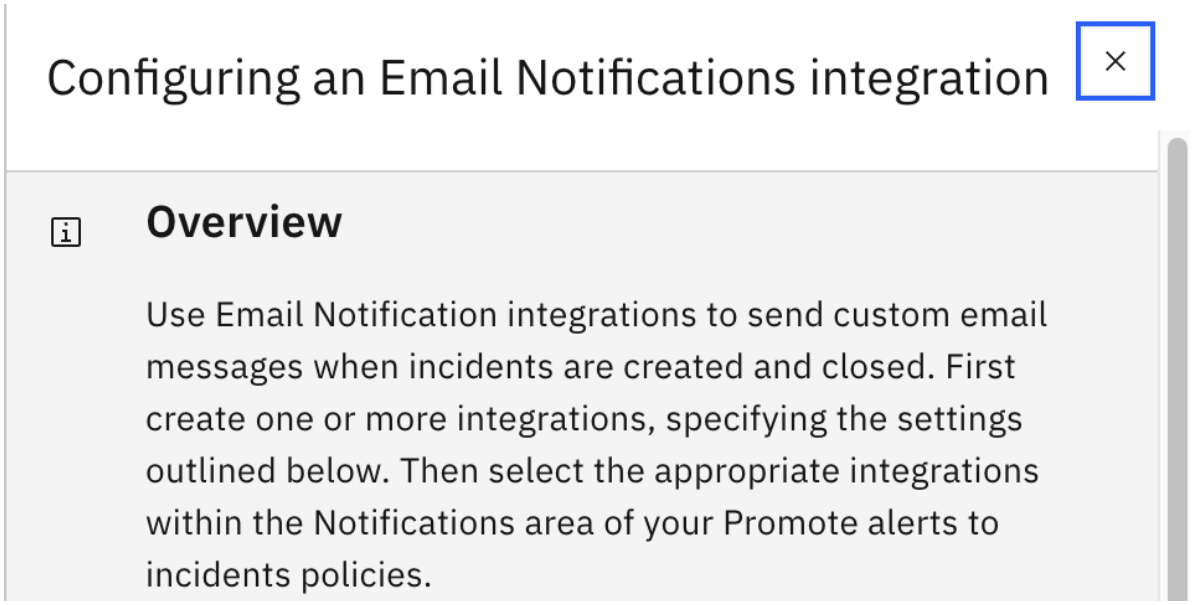
For example, if you want to include the time at which the incident was created, include the following code in the email template:

```
$string(incident.createdTime)
```

Creating an email integration

To create an email integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Email Notifications** tile.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.



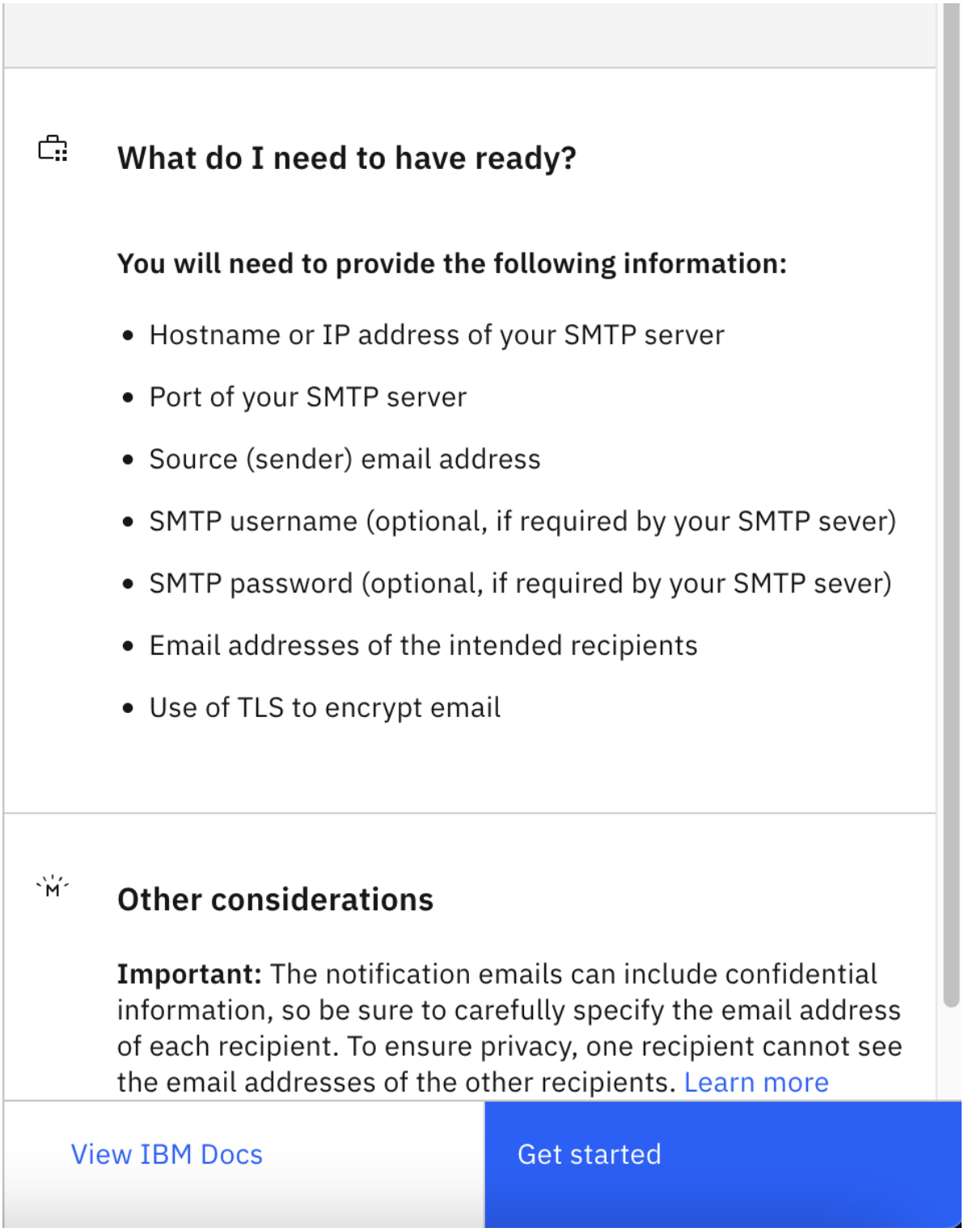


Figure. Configure Email Notifications integration

6. On the *Add integration* page, define the general integration details:

- **Name**: The display name of your integration. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
- **Description** (optional): Optionally, enter a description of the integration.
- **Hostname or IP address of SMTP server**: Enter the hostname or IP address of the SMTP server. **Note**: If running behind a proxy server, enter the proxy server hostname or IP address.
- **Port of SMTP server**: Enter the port of the SMTP server.

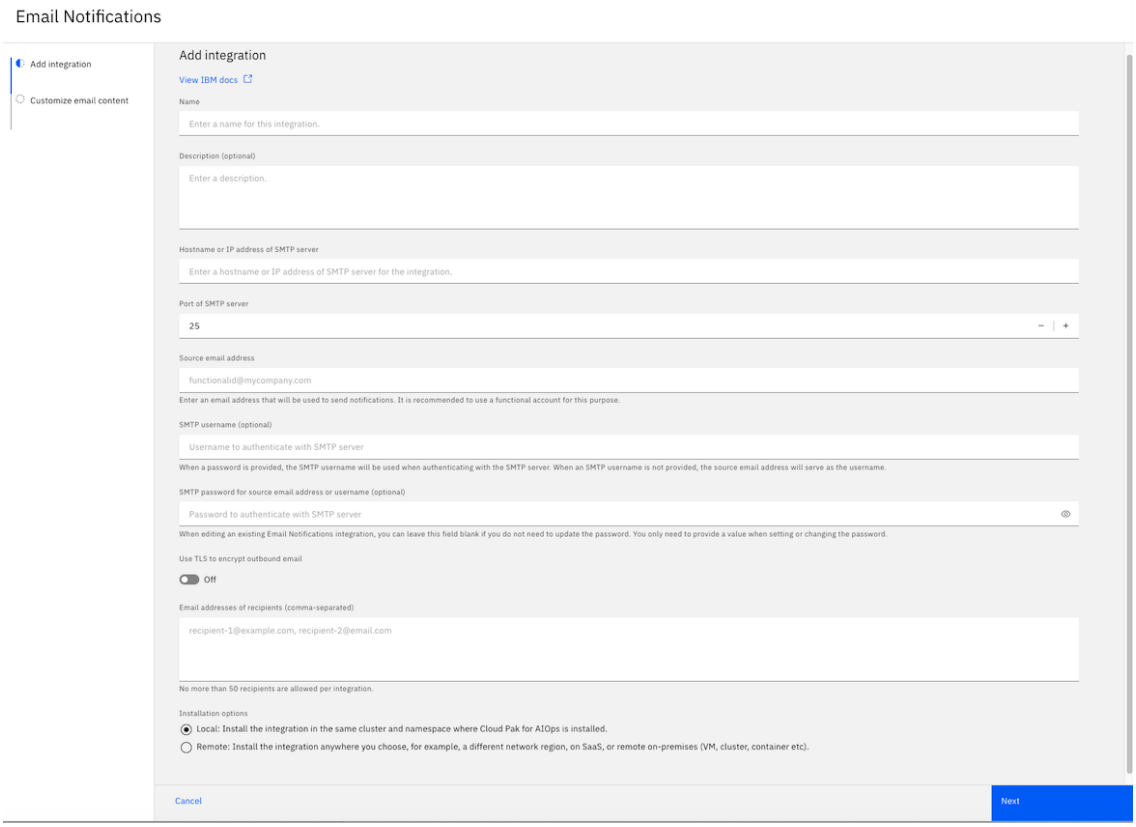


Figure. Add Email Notifications integration

- **Source email address**: The email address that the new incident notifications will be sent from. It is recommended to use a functional email account for this purpose.
- **SMTP username (optional)**: Enter the SMTP username. When a SMTP username is not provided, the source email address is used as the username.
- **SMTP password for source email address** (optional): Optionally, enter an SMTP password for the source email address.
- **Use TLS to encrypt outbound email** (optional): Select from **On** or **Off**.
- **Email addresses of recipients**: Enter the target email addresses (comma-separated). No more than 50 recipients are allowed per integration.

  Important: The notification emails can include confidential information, so be sure to carefully specify the email address of each recipient. To ensure privacy, one recipient cannot see the email addresses of the other recipients.
- **Deployment options**: Select one of the options, **Local** or **Remote**. **Local** deploys the integration in the same cluster and project (namespace) where IBM Cloud Pak for AIOps is installed. Alternatively, **Remote** deploys the integration in your chosen location, such as a different network region, on SaaS, or a remote on-premises VM, cluster, or container.

  Note: If you select **Remote** after adding the integration, you are provided with a bootstrap script to finish the deployment. Download and run the bootstrap script on the remote cluster to complete the deployment.

7. Click **Next**.
8. Customize the content for both the 'Incident created' and the 'Incident closed' emails. You can use the default email templates as-is, or customize them to include different content. A subject and a message must be included; otherwise, an email will not be sent.

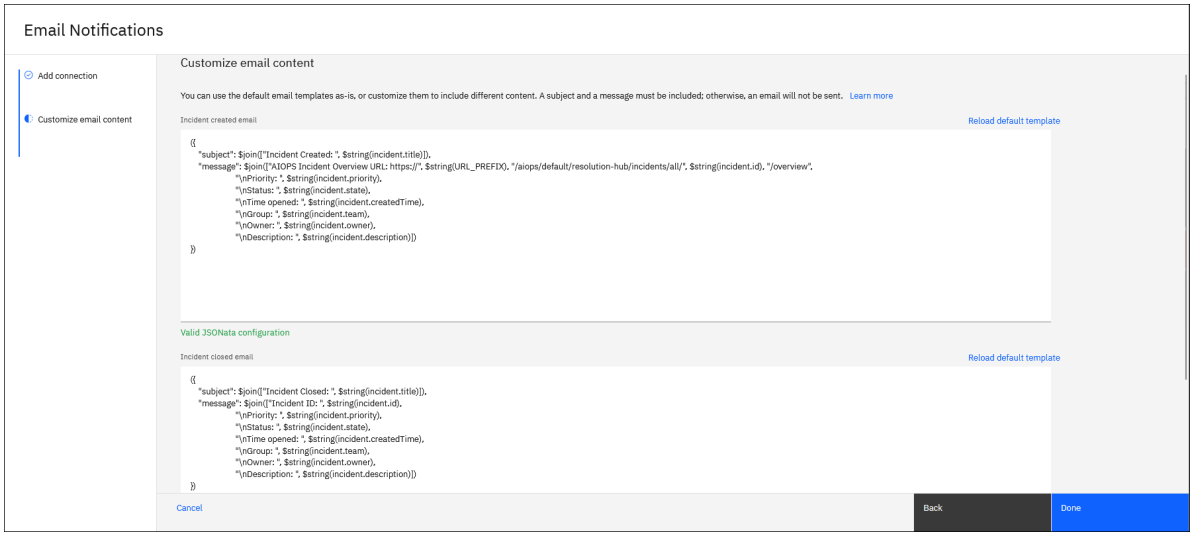


Figure. Customize Email Notifications content

9. Click **Done**. You are redirected to the *email notification* page for the integration.

Your new integration is listed in the table on the page. The integration has an initial status of *Initializing*. After a few minutes, the status changes to *Running* if your integration is successful.

Editing an email integration

After you create your integration, you can edit the integration. For example, if you want to add or change the listed recipient email addresses. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Email Notifications** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *Email Notifications* page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.

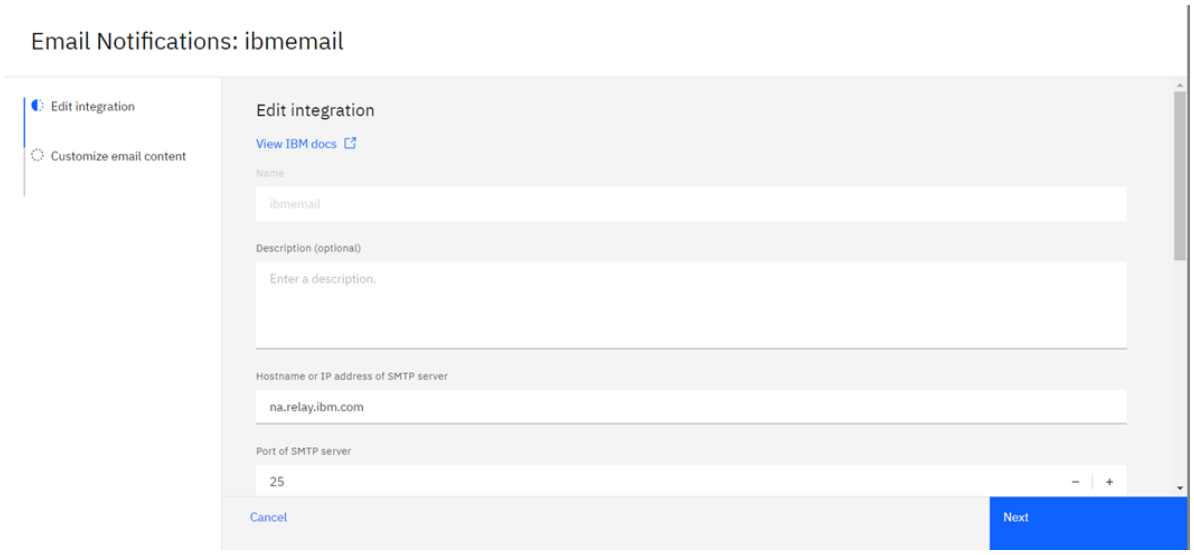


Figure. Edit Email Notifications integration

5. Edit your integration as required. Click **Save** when you are done editing.

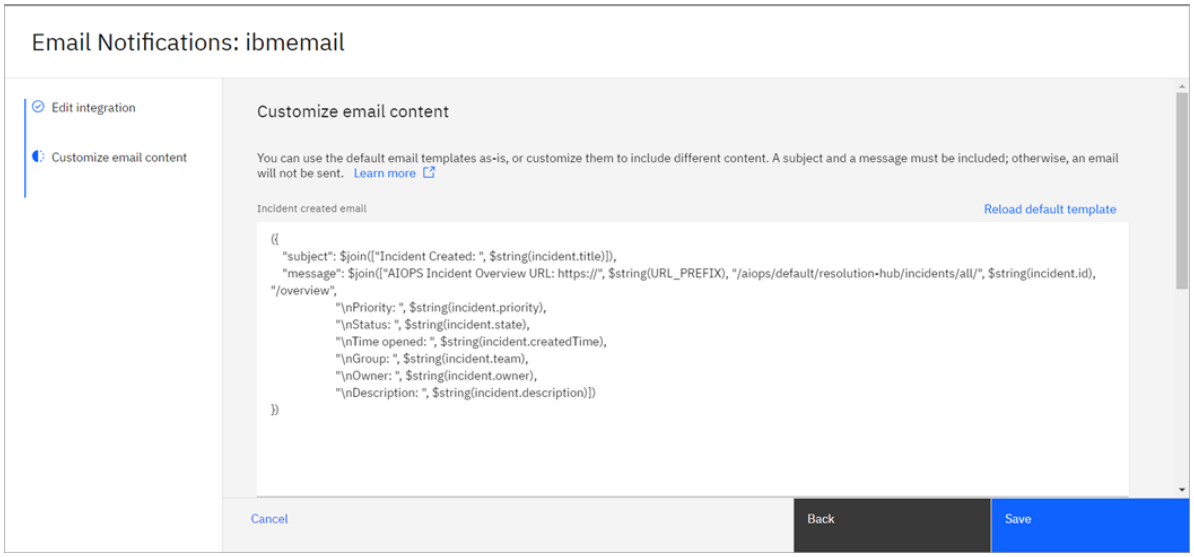


Figure. Edit Email Notifications integration - customize

Your integration is now edited.

Deleting an email integration

If you no longer need your integration and want to delete it entirely, you can delete the integration from the console.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Email Notifications** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *Email Notifications* integrations page, click the *Delete* icon for the integration that you want to delete.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

   Your integration is deleted in IBM Cloud Pak for AIOps.

Creating GitHub integrations

Adding a GitHub integration enables your organization to create tickets when incidents are created in IBM Cloud Pak® for AIOps, discover details about similar tickets, and assess the risk of implementing proposed changes. Once a GitHub integration is set up and running, you can select it from the Notifications area of your Promote alerts to an incident policies.

- About this task
- Prerequisites
- Best practices
- Creating a GitHub integration
- Editing a GitHub integration
- Deleting a GitHub integration

About this task

Create one or more integrations by defining the integration details that are outlined here. Provide the URL and personal access token of your GitHub server, the GitHub repository, username, and the repository owner or organization. Optionally, specify field-mapping information for the incident details sent to GitHub when incidents are created. A default template is provided.

Prerequisites

You need the following information to create a GitHub integration:

- URL of your GitHub server
- Personal access token for your GitHub server
- GitHub username
- GitHub Owner or organization
- GitHub repository

Best practices

- When creating multiple GitHub connectors, use a different username and token to facilitate speedier data collection. There is a rate limit for GitHub API calls of about 5000/hr per individual and 15000/hr at enterprise level.
- Use a personal access token that does not expire.
- When Historical data collection is selected (see step 9), select a start date close to the date that will be used for AI training. Selecting a date further back might result in more API calls to GitHub, especially if there is a large volume of issues.

Creating a GitHub integration

To create a GitHub integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **GitHub** tile.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. On the *Add integration* page, define the general integration details:
    - **Name**: The display name of your integration. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
    - **Description** (optional): Optionally, enter a description of the integration.
    - **URL of GitHub server**: Enter the URL of the GitHub server, for example `https://api.github.com/`. If you have any restrictive egress network policies in place, ensure that the policies are updated to allow for this outbound integration.
    - **Personal access token for GitHub server**: This token is used to authenticate your requests to access GitHub APIs. The owner of the token must have write access to the repository.



Figure. Add Github integration

Important: The selected scope of your GitHub repository must be "**repo**" with **Full control of private repositories**. To configure this in your GitHub repository, navigate to your profile > **Settings** > **Developer settings** > **Personal access tokens**.

- **Username**: Enter your GitHub username.
- **Owner or organization**: Enter the GitHub repository owner or organization.
- **Repository**: Enter the GitHub repository.

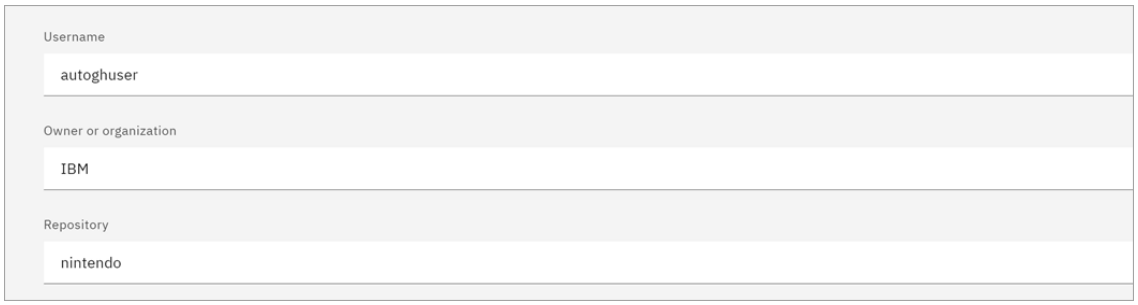


Figure. Add Github integration

7. Click **Next**.
8. You can use the default issue template as-is, or customize it to include different information. A title is required or the issue cannot be created. In addition to the fields included by default, you can also specify 'state' and 'assignees' for the incident. Incident 'priority' must be set in the label, otherwise the priority

will be appended and not replaced.

The owner of the token must have push access to the repository to set 'labels' and 'assignees' for new issues. Changes to 'labels' and 'assignees' are silently dropped if the user does not have push access to the repository.



Figure. Customize GitHub issue template

Note: **`$string(URL_PREFIX)`** reads from the **`URL_PREFIX`** environment variable to create the URL. The value for this variable is **`hostname`**.

9. Define how ticket data is collected:

   - **Data collection**: toggle to **Off** or **On**. The default is **On**. Note, if the rate limit for GitHub API calls is reached, an alert is sent to notify users that the connector is sleeping until the reset time from GitHub.
   - **Mode**: choose between **Live** or **Historical**. Live is the frequency at which data is sampled for GitHub issues in minutes (1-60 minutes). The default value is 1. For **Historical**, select a start date to retrieve closed GitHub issues since their last updated time.

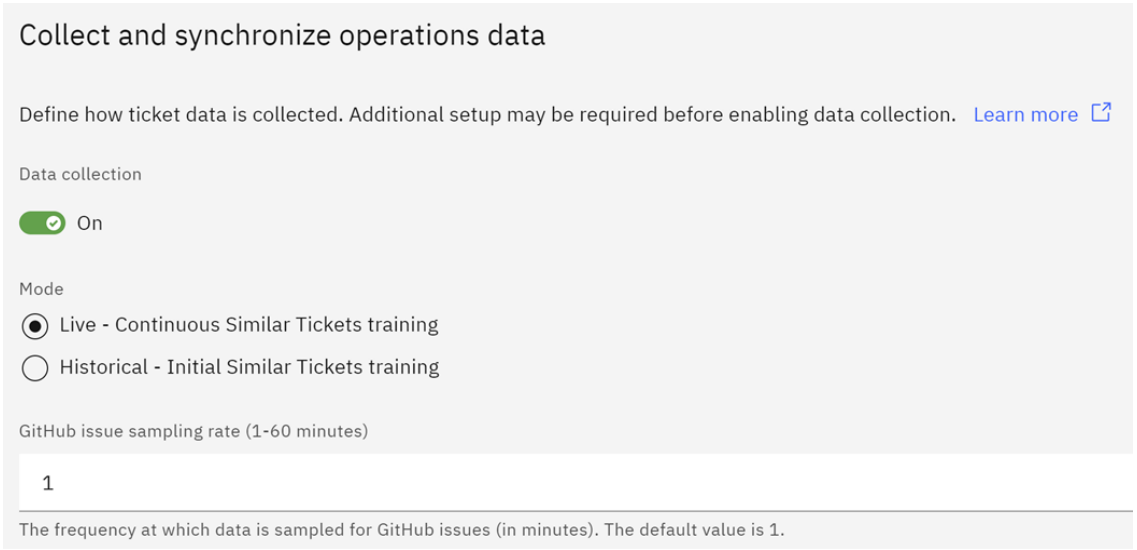


Figure. Collect and synchronize operations data

   Once the historical data is collected, you can train the data in **AI model management** using similar tickets training.

10. Click **Done**. You are redirected to the *GitHub integration* page.

Your new integration is listed in the table on the page. The integration has an initial status of *Initializing*. After a few minutes, the status changes to *Running* if your integration is successful. Once the integration is running, you can select GitHub as a ticketing option in your *Promote alerts to an incident* policies.

Editing a GitHub integration

After you create your integration, you can edit the integration. For example, if you want to customize the mappings in the issue template. To edit an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **GitHub** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *GitHub integration* page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited.

Deleting a GitHub integration

If you no longer need your integration and want to delete it entirely, you can delete the integration from the console.

To delete an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **GitHub** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *GitHub* integrations page, click the *Delete* icon for the integration that you want to delete.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

   Your integration is deleted in IBM Cloud Pak for AIOps.

Ansible Automation Platform integrations

You can create an integration to Ansbile Automation Platform. Red Hat Ansible Automation Platform provides tools for building and operating automation.

IBM supports Ansible as part of the Ansible Automation Platform. For more information about this platform, see Red Hat Ansible Automation Platform.

Note: You can create only one Ansible Automation Platform controller integration. Also, you must have an account with administrator role to create, edit, view, or delete an Ansible Automation Platform integration.

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating Ansible Automation Platform integrations

Complete the following steps to create an integration to an Ansible Automation Platform controller server.

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Ansible Automation Controller** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. Enter the base URL of your Ansible Automation Platform controller server. This URL must contain the protocol, for example: `https://ansible.mycompany.com:443`.
7. Choose an authentication type. You can select **User ID/Password** to connect with username and password, or **API Token** to use a bearer token, previously created with **Write Scope** in the Ansible.
8. Enter the chosen authentication information.
9. Optional: Enter the Ansible Automation Platform controller server root CA certificate, or the Ansible Automation Platform controller self-signed certificate.

   If the Ansible server certificate is based on a root CA certificate, then this root CA certificate must be entered (in PEM format). Some Ansible installations provide a self-generated root CA certificate (for example, at `<aap_install_dir>/tls/ca.cert`).

   If the Ansible server certificate is a self-signed certificate, then make sure the common name (CN) of the certificate matches the fully qualified domain name of the host system (a certificate with another CN like "localhost" will not be accepted). If needed, replace that certificate with a certificate that is issued for the actual domain name you are using.

   On Linux systems, enter the following command to receive the self-signed certificate:

   ```
   echo -n | openssl s_client -servername <ANSIBLE_TOWER_HOSTNAME> -connect <ANSIBLE_TOWER_HOSTNAME>:
   <ANSIBLE_TOWER_PORT> -showcerts | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > file.cert
   ```

   If the command does not work in your environment, use the following variant of the command:

   ```
   ex +'g/BEGIN CERTIFICATE/,/END CERTIFICATE/p' <(echo | openssl s_client -showcerts -servername
   <ANSIBLE_TOWER_HOSTNAME> -connect <ANSIBLE_TOWER_HOSTNAME>:<ANSIBLE_TOWER_PORT>) -scq > file.cert
   ```

   If errors occur, make sure your exported certificate that is stored in `file.cert` contains a full and valid certificate. Errors like `verify error:num=20:unable to get local issuer certificate` occur due to a missing CA root certificate for the DigiCert CA.

   The resulting file.cert might contain one or more certificates and each certificate begins and ends as follows:

   ```
   -----BEGIN CERTIFICATE-----
   ...
   -----END CERTIFICATE-----
   ```

   On Windows systems, use your preferred browser to export the certificate or certificate chain.

10. Click **Done** to complete the Ansible Automation Platform integration.

Note: See Ansible actions for additional information on the supported Ansible versions and some other restrictions.

Creating an IBM Db2 integration

BETA The IBM Db2 integration is available only for technology preview, and is not for production usage.

Adding an IBM® Db2® integration enables your organization to store incident data in a Db2 database. The incident data can then be included in custom dashboards and reports that are created in IBM Cognos® Analytics.

- Prerequisites
- Creating an IBM Db2 integration
- Editing an IBM Db2 integration
- Deleting an IBM Db2 integration

Prerequisites

You need the following information to create an IBM Db2 integration:

- Data source type
- Name of the IBM Db2 database
- Hostname or IP address at the database
- Port number for incoming client connections
- Username and password to access the database

Note: The IBM Db2 integration is only supported on Red Hat® OpenShift® Container Platform clusters. IBM Db2 integration is not supported on IBM Cloud Pak for AIOps on Linux.

Creating an IBM Db2 integration

To create an IBM Db2 integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **IBM Db2** tile.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. On the *Add integration* page, define the general integration details:
    - **Name**: The display name of your integration. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
    - **Description** (optional): Optionally, enter a description of the integration.
    - **Data source type**: From the drop-down menu, select the type of IBM Db2 database.
    - **Database hostname or IP address**: Enter the fully qualified hostname or IP address of the IBM Db2 database, for example, `myserver.abc.com` or `192.0.2.24`.
    - **Database port (1024 - 65535)**: Enter the database port number.
    - **Database name**: Enter the name of the IBM Db2 database.

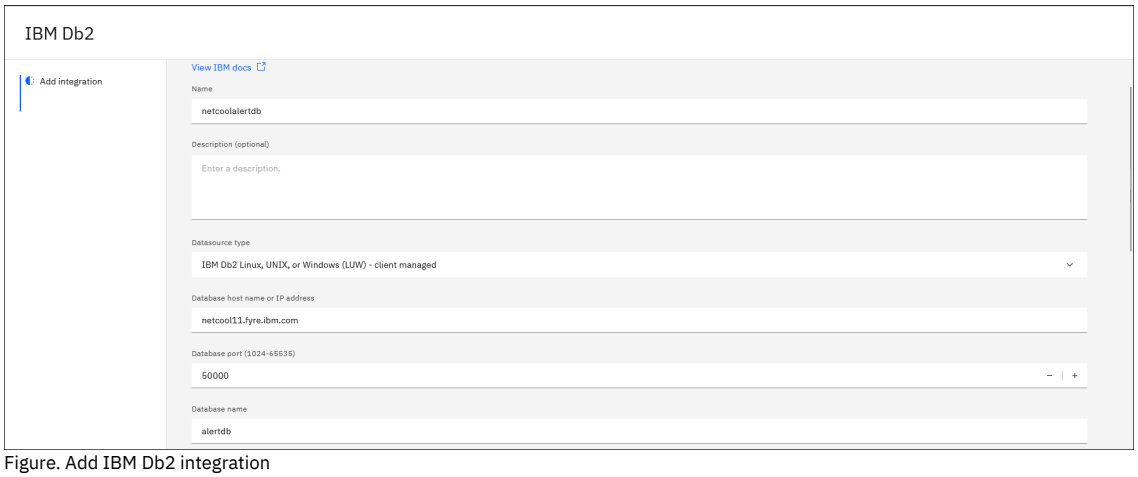


Figure. Add IBM Db2 integration

- **Username**: Enter the username to access the IBM Db2 database.
- **Password**: Enter the password for the IBM Db2 database username.
- **Schema qualifier (optional)**: Enter the schema qualifier for the database objects to be accessed. If blank, the username is used.
- **Maximum pool size (0 - 100000)**: Enter the maximum number of connections allowed within a single connection pool. The default value is 0, which indicates that connection pooling is switched off for the integration server.
- **Additional database parameters (optional)**: Enter additional database parameters in the format `name1=value1;name2=value2`.

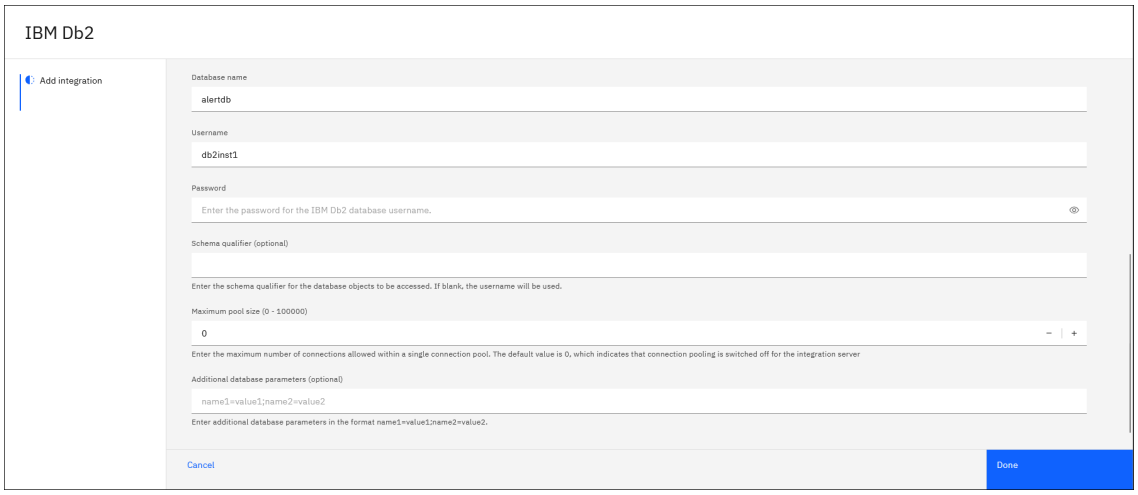


Figure. Add IBM Db2 integration

7. Click **Done**. You are redirected to the *IBM Db2 integration* page.

   Your new integration is listed in the table on the page. The integration has an initial status of *Initializing*. After a few minutes, the status changes to *Running* if your integration is successful.

Create a policy to populate a dashboard or report

Once an IBM Db2 integration is set up and running, you can create a policy to send alerts or incidents from Cloud Pak for AIOps to populate a dashboard or report in IBM Cognos® Analytics. For more information, see Populate a dashboard or report.

Editing an IBM Db2 integration

After you create your integration, you can edit the integration. To edit an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Db2** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *IBM Db2 integration* page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited.

Deleting an IBM Db2 integration

If you no longer need your integration and want to delete it entirely, you can delete the integration from the console.

To delete an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Db2** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *IBM Db2* integrations page, click the *Delete* icon for the integration that you want to delete.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

   Your integration is deleted in IBM Cloud Pak for AIOps.

Incoming integrations

Incoming integrations enable IBM Cloud Pak for AIOps to collect log, metric, and event data from various sources.

IBM Cloud Pak for AIOps uses this collected data to establish baseline models during AI model training. These models are then used to compare against live incoming data from these integrations to report anomalous activity to you in near real time by way of your outgoing integrations. In live data mode, the integration might need a delay to offset the query time windows to provide time buffer to prevent retrieving partial real-time data. For more information, see Delay configuration in data integrations.

The recommended value for source parallelism should be equal to the number of days. For example, if you are pulling historical data for training for 7 days, the source parallelism should be around 7 or higher, so that you can pull data for multiple days simultaneously.

Unlike outgoing integrations, you can set up multiple sources of incoming integrations. For example, if you have a single source of Falcon LogScale data for logs, and PagerDuty configured for events, you can create integrations for both of those. You can also use the custom and Kafka integrator for greater flexibility. The custom integration type collects data from a single specific data source (like any other integration type). If you want to collect log data from different systems, and then route that information through a forwarding agent, such as an Apache Kafka topic (like Sysdig or FluentBit), you can use the Kafka integration type.

Note: The ServiceNow integration type has optional incoming and outgoing facets.

For more information about incoming integrations, see the following topics:

- Administrator process integrations
- Appdynamics integrations
- AWS CloudWatch integrations
- Custom integrations
- Dynatrace integrations
- ELK integrations
- Falcon LogScale integrations
- Generic webhook integrations
- IBM Netcool Operations Insight ObjectServer integrations
- IBM Tivoli Netcool/Impact integrations
- Instana integrations
- Kafka integrations
- Mezmo integrations
- New Relic integrations
- Observer integrations
- PagerDuty integrations
- Splunk integrations
- Zabbix integrations

Creating AppDynamics integrations

An AppDynamics integration can provide Cloud Pak for AIOps metric and topology data for the applications or infrastructure that you monitor.

You need to have both an endpoint and basic authentication ready. The AI Algorithm that analyzes AppDynamics data is Metric anomaly detection. AppDynamics SaaS controller 22.5 is the version that is supported for the AppDynamics integration.

For more information about working with AppDynamics integrations, see the following sections:

- Creating AppDynamics integrations
- Enabling AppDynamics integrations
- Editing AppDynamics integrations
- Deleting AppDynamics integrations
- Adding an AppDynamics data retention period

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating AppDynamics integrations

To create an AppDynamics integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **AppDynamics** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and select from the following options:

   - **Select the data types that you need to collect**: Select the relevant checkbox for *Metrics* or *Topology*.
   - **Specify whether you'd like to install the integration**: Select between the *Cluster running Cloud Pak for AIOps* or *Self-hosted with network visability to AppDynamics* radio buttons.

- **Select Cloud Pak for AIOps integration**: You can leave this at the default *Use the integration that is recommended by Cloud Pak for AIOps*, which will be whichever you selected out of *Metrics* or *Topology* in the *Select the data types that you need to collect* field, or you can deselect this and then select between just *Metrics* or *Topology*.

  The below steps are for selecting *Metrics* but if you do select *Topology* then you will be brought to a *New observer job* page. An AppDynamics integration provides topology data through an observer that supports the discovery of business applications, nodes and tiers. When an AppDynamics observer job is run, it will gather and read data from the AppDynamics Controller via REST API. For more information, see Configuring an AppDynamics observer job.

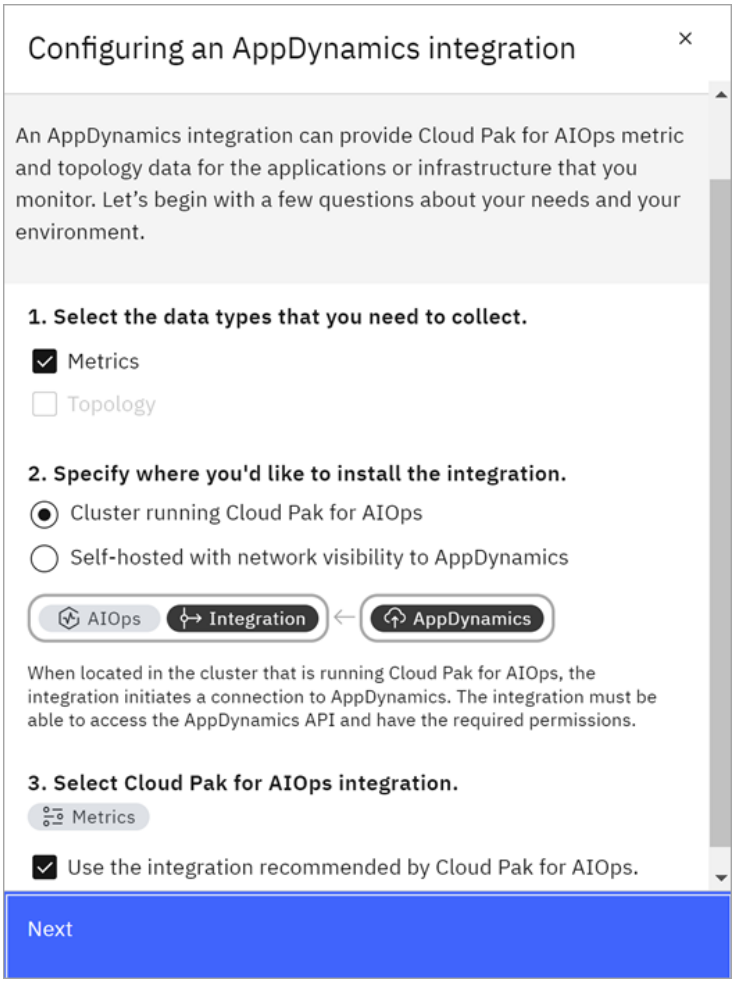


Figure. Configuring an AppDynamics integration

6. Then click **Next**.

   You see an overview, further direction such as *what you need to have ready* and also *what algorithms analyze AppDynamics data*.

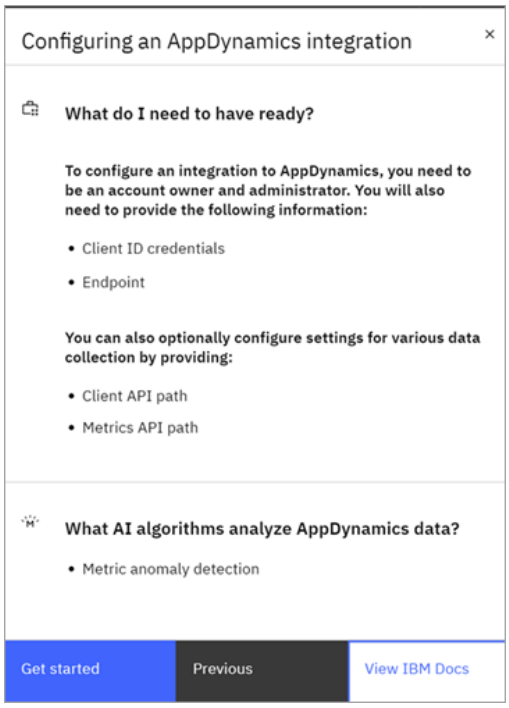


Figure. Get started

7. Click **Get Started**.
8. On the *Add integration* page, define the general integration details:
    - **Name**: The display name of your integration. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
    - **Description**: An optional description for the integration.
    - **Endpoint**: Enter an endpoint for the integration, which in this case is the AppDynamics data source base URL of your account. For example `https://myinstance.saas.appdynamics.com`.
    - **Client ID**: Client ID is generated from the AppDynamics controller, in the format 'ApiClientName@AccountName', for example 'aiops@ibm', where 'aiops' is the API Client name, and 'ibm' is the AppDynamics account name.

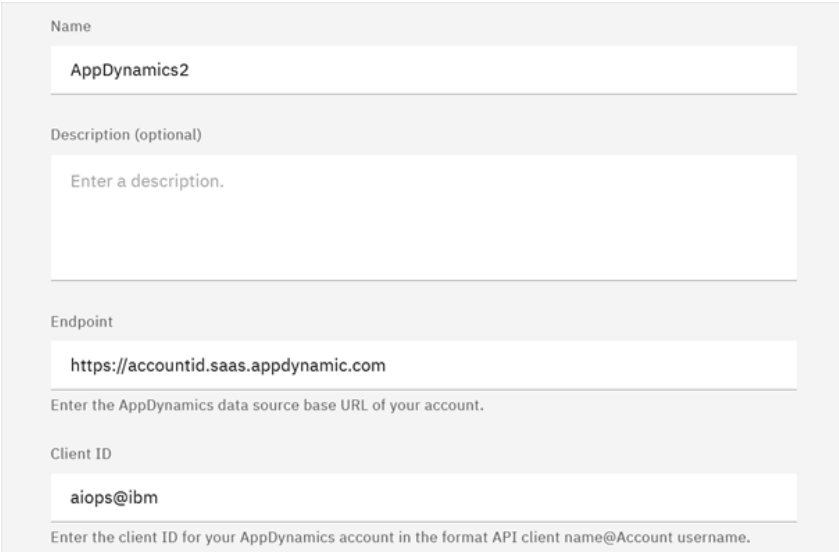


Figure. Add AppDynamics integration

- **Client Secret**: Generated from **AppDynamics UI -> Settings -> Administration -> API Clients**. Enter the Client secret for the application. Also, ensure that the Client Secret created has the following roles: Account Owner, Administrator, or Applications and Dashboards Viewer, which is the minimum role. If not, the API calls might not return valid application IDs.
- **Deployment options**: Select between 'Local' and 'Remote' radio buttons. If you pick 'Remote', then after you add the integration, you are provided with a bootstrap command to finish the orchestration. Run the bootstrap script on the remote server where you want to deploy the integration remotely.

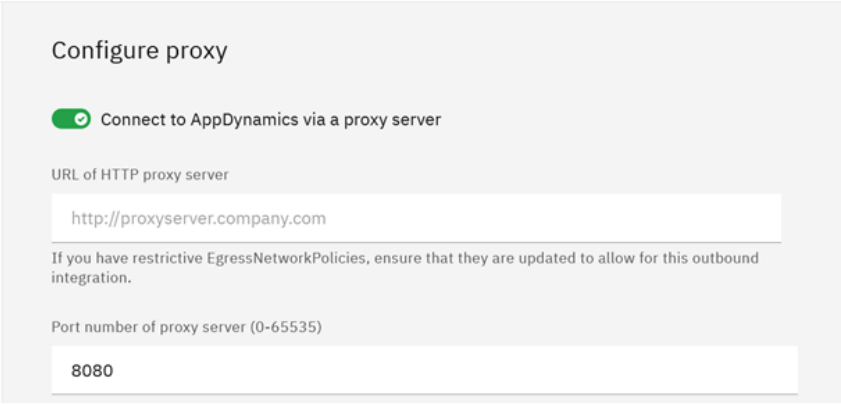
  Important: Podman must be installed to execute the remote deployment script with a recommended minimum version of 4.4.1. Read and write access are required for the folder where the remote deployment script will be executed and the script also must have execution permissions in order to be run. No more than one remote deployment script should be executed per host.

9. To verify the SSL/TLS configuration:
    1. Set the **Verify the SSL/TLS certificate for the REST service** toggle to *on*.
    2. Either enter a CA certificate used to verify the SSL/TLS integration to the REST service or click **Fetch SSL certificate** to fetch the certificate automatically.
10. Click **Test Connection**. If successful, a Test succeeded message is displayed.



Figure. Test connection

11. Click **Next**.
12. On the *Configure Proxy* (Optional) page complete the following fields:
    - **Connect to AppDynamics via a proxy server**: Set the toggle button to *on*.
    - **URL of HTTP proxy server**: Enter a URL for the HTTP proxy server. **Note:** The URL for the metrics-only integration should have the protocol in the URL.
    - **Port number of proxy server**: Enter a port. The default is 8080.



    Note: Only unauthenticated HTTP proxy is supported.
13. Click **Next**.
14. Enter the following *Collect metric data**: Define how you want to collect metric data.

- **Enable data collection**: Set the toggle button to on.
- **Fill out Code field**: Enter Application name and corresponding Metric paths as JSON, using the following format:

```
{"Application Name":"[metric path1, metric path2, metric path3]"
}
```

Example

```
{
"helloworld":"[Overall Application Performance|Calls per Minute,Overall Application Performance|Average Response Time (ms),Overall Application Performance|mytier|Average Response Time (ms)]",
"products-ecommerce":"[Overall Application Performance|*,Business Transaction Performance|Business Transactions|DwightVM|*|*]",
"Database Monitoring":"[Databases|admin2|KPI|*,Databases|mysql-admin| Server Statistic|*]"
}
```

To get the metric path, go to the AppDynamics Console, select the application > Metrics Browsers > choose the metrics that you want to monitor > right-click and select 'Copy Full Path'.

15. You can chose between collecting historical or live data. If you want to collect live data you need to click the **Only click Live data once the integration is established** checkbox. If you do not you need to fill out the following fields for historical data collection:
    - **Enter Start date**: Enter the start date *mm:dd:yyyy*.
    - **Enter Start time**: Enter the start date and time in the format *hh:mm* (hours and minutes).
    - **Set LocalTime**: Pick the time zone required from the dropdown list.

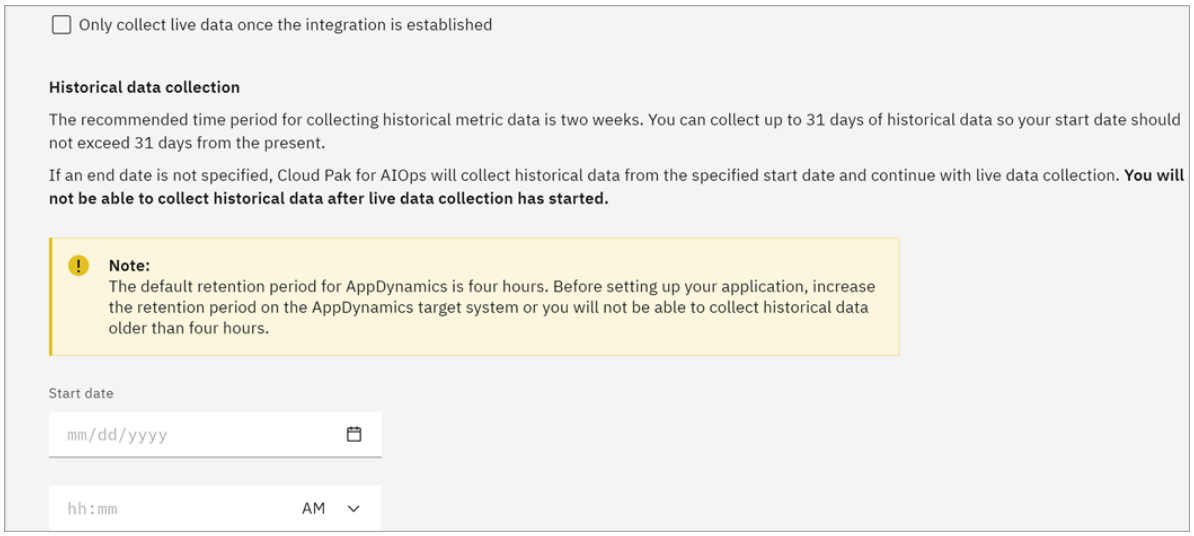


Figure. Collect metric data

Click the **Add end date** button if you want to add an end date. There is also a *Remove End date* button.The following fields appear:

- **End date**: Enter the end date *mm:dd:yyyy*.
- **End time**: Enter the end date and time in the format *hh:mm* (hours and minutes).
- **Scheduler run time (minutes)**: The frequency at which the live scheduler should run. The default is 5 mins.

Note: For historical metric data collection, the recommended time period is two weeks. You can collect up to 31 days of historical data so your start date should not exceed 31 days from the present. If an end date is not specified, Cloud Pak for AIOps will collect historical data from the specified start date and continue with live data collection. You will not be able to collect historical data after live data collection has started. The default retention period for AppDynamics is four hours. Before setting up your application, increase the retention period on the AppDynamics target system or you will not be able to collect historical data older than four hours.

If you click the *Live* data radio button the field to complete is:

- **Scheduler run time (minutes)**: The frequency at which the live scheduler should run. The default is 5 mins.

16. Click **Done**.

For more information about adding an AppDynamics data retention period, see Adding an AppDynamics data retention period.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

You created an AppDynamics integration in your instance. After you create your integration, you must enable the data collection to connect your integration with the AI of IBM Cloud Pak for AIOps. For more information about enabling your integration, see Enabling AppDynamics integrations.

Note: An error that might show if no data is set for particular metrics or applications can resemble the following message.

```
2022-06-22 04:59:07,578 INFO [Default Executor-thread-3]
com.ibm.watson.aiops.connectors.service.AppdynamicsWebClientService : Bad Request
[err] javax.ws.rs.BadRequestException: BadRequest
[err] java.lang.NullPointerException
```

In order to avoid this error, ensure the application name and metrics path configured are valid. Also, ensure that the Client Secret created previously has the following roles: Account Owner and Administrator. If not, the API calls might not return valid application IDs.

To create more integrations (such as a ChatOps integration), see Configuring Integrations.

For more information about working with the insights provided by your integrations, see ChatOps insight management.

Enabling and disabling AppDynamics integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **AppDynamics** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *Collect metric data* section. Set **Enable data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting an integration, see Deleting AppDynamics integrations.

Editing AppDynamics integrations

After you create your integration, your can edit the integration. For example, if you specified *Historical mode* but now want your integration to pull in live data for continuous monitoring, you can edit it. To edit an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **AppDynamics** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the AppDynamics integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

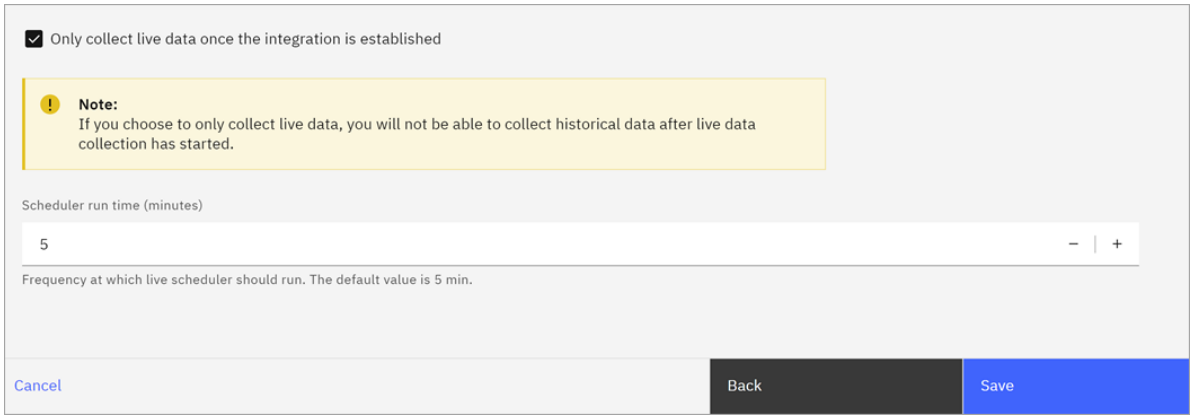


Figure. Edit AppDynamics integration

Your integration is now edited. If your application was not previously enabled or disabled, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling AppDynamics integrations. For more information about deleting an integration, see Deleting AppDynamics integrations.

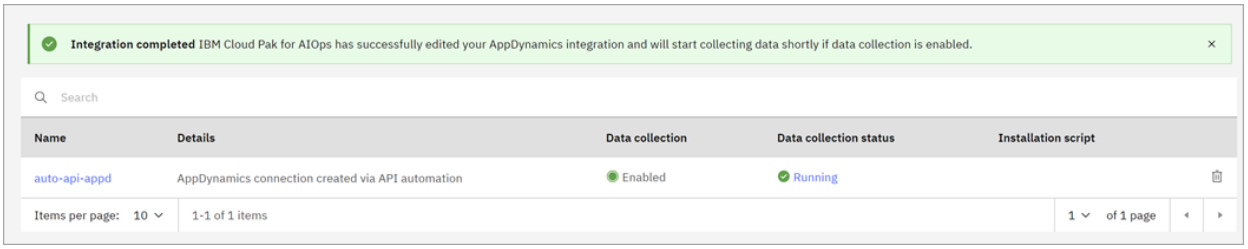


Figure. AppDynamics integration created

Deleting AppDynamics integrations

If you no longer need your AppDynamics integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

To delete an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **AppDynamics** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the AppDynamics integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Adding an AppDynamics data retention period

The default data retention period for AppDynamics is 4 hours. Metric anomaly detection uses up to two weeks of data to build models and without historical data to collect, it is several days to a week before anomalies start to be detected and raised without having historical data to quickly train up the algorithms.

Consider whether it is possible to increase the AppDynamics data retention period in advance so that the analytics can train up more quickly. To increase the data retention period in AppDynamics, consult the relevant AppDynamics retention documentation.

Creating AWS CloudWatch integrations

An AWS integration can provide CloudPak for AIOps metric, event, and topology data for the applications or infrastructure that you monitor.

To activate the integration, you must have an endpoint, an access key, and a secret key ready. You can select the metrics that you want to monitor by choosing the associated namespaces in the IBM Cloud Pak for AIOps console. The Metric anomaly detection AI algorithm analyzes the AWS CloudWatch data.

For more information about working with AWS CloudWatch integrations, see the following sections:

- Creating AWS CloudWatch integrations
- Enabling and disabling AWS CloudWatch integrations
- Editing AWS CloudWatch integrations
- Deleting AWS CloudWatch integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating AWS CloudWatch integrations

To create an AWS CloudWatch integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **AWS CloudWatch** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and select your required options from the following sections:
   - **Select the data types that you need**: Select the relevant checkbox for *Metrics*, *Events*, or *Topology*.
   - **Specify whether you'd like to install the integration**: Select between the *Cluster running Cloud Pak for AIOps* or *Self-hosted with network visability to Dynatrace* radio buttons.
   - **Select Cloud Pak for AIOps integration**: You can leave this at the default *Use the integration recommended by Cloud Pak for AIOps*, or deselect it, and change to just *Metrics*, *Events*, or *Topology*.

     The below steps presuppose you select *Metrics* but if you do select *Events* then there will be a *Collect event data* page to complete, and if you select *Topology* then you will be brought to a *New observer job* page. The AWS observer reads services data from AWS through the AWS SDK and generates a topology. The AWS Observer supports multiple Amazon Web Services (AWS) services such as EC2 for its compute services. For more information, see Configuring an AWS observer job.
6. Then click **Next**.

   You see an overview, further direction such as *what you need to have ready* and also *what algorithms analyze Dynatrace data*.

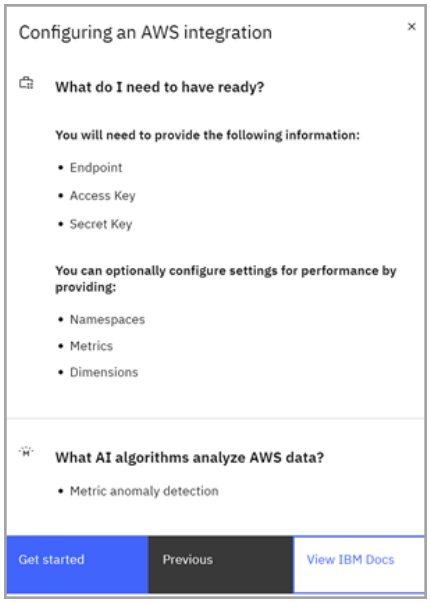


Figure. Configure AWS integration - get started

7. Click **Get Started**.
8. On the *Add integration* page, define the general integration details:
   - **Name**: Type in the display name of your integration.
   - **Description**: Provide an optional description for the integration.
   - **Endpoint**: Enter an endpoint for the integration. For example `https://monitoring.us-east-1.amazonaws.com`.



Figure. Add AWS integration

- **Access Key**: Enter an Access Key for the integration.
- **Secret Key**: Enter a Secret Key for the integration. For getting access and secret keys, contact admin.
- **AWS CloudWatch region**: Select a region from the available regions in AWS, for example, us-east-1.
- **Deployment options**: Select one of the following options: **Local** or **Remote**.

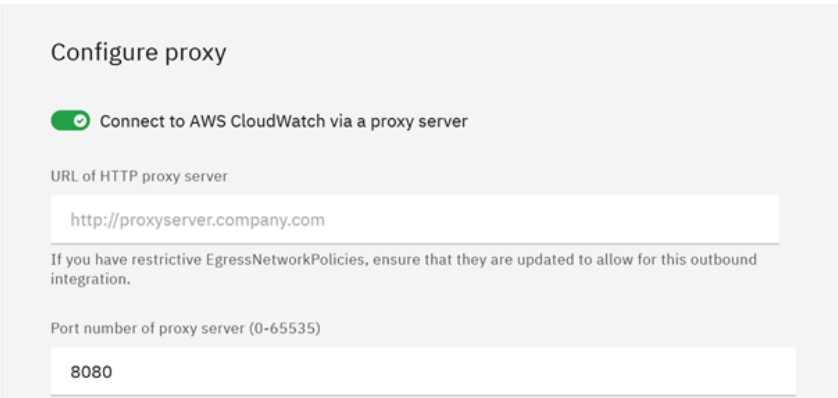
  The **Local** option deploys the integration in the same cluster where IBM Cloud Pak for AIOps is installed. Alternatively, the **Remote** option deploys it in your chosen location. If you select the **remote** option, you are provided with a bootstrap command after adding the integration to finish the deployment.

9. To verify the SSL/TLS configuration:
   1. Set the **Verify the SSL/TLS certificate for the REST service** toggle to *on*.
   2. Either enter a CA certificate used to verify the SSL/TLS integration to the REST service or click **Fetch SSL certificate** to fetch the certificate automatically.
10. Click **Test connection**. A message stating whether the test succeeded or failed is displayed.



Figure. Test connection

11. Click **Next**.
12. On the *Configure Proxy* (Optional) page complete the following fields:
    - **Connect to AWS CloudWatch via a proxy server**: Set the toggle button to *on*.
    - **URL of HTTP proxy server**: Enter a URL for the HTTP proxy server.
    - **Port number of proxy server**: Enter a port. The default is 8080.



    Note: Only unauthenticated HTTP proxy is supported.

13. Click **Next**.
14. Enter the following *Collect metric data* information. It defines how you want to collect metric data.
    - **Enable data collection**: Set the toggle button to on.

Note: If you disable data flow, a new integration is created, but metrics are not collected.

- **Namespace**: Select one or more namespaces from the list to see its associated metrics in the filter table. By default, all metrics are selected in the table. You can search and select or remove metrics from it.
- **Aggregation interval (minutes)**: Enter a value in minutes. This value is the frequency at which data is collected. The default value is 5 minutes.
- Select live or historical data collection

  Note: If you choose to collect only live data, you cannot collect historical data after the live data collection starts. However, if you collect historical data, you can continue live data collection afterwards.

  - Live data collection: To collect live data, complete the following steps:
    - **Only collect live data once the integration is established**: Select the checkbox.
    - **Scheduler run time (minutes)**: Enter a value in minutes. The value represents the frequency at which live scheduler runs. For example, if you set the value to 10 minutes, the live scheduler runs every 10 minutes. The default value is 5 minutes.
  - Historical data collection: To collect historical data, complete the following steps:
    - **Start date and time**: Select the start date from the calendar and enter the time in the 'hh:mm' format (hours and minutes).

      Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.
    - **Time zone**: Select your time zone from the dropdown list.
    - **End date and time**: Select a date from the calendar. Also, enter the time in the hh:mm format (hours and minutes). If you want to the end date, click **Remove end date**.

      Note: If you do not set an end date, IBM Cloud Pak for AIOps collects historical data from the specified start date and continues with live data collection. You cannot collect historical data after the live data collection starts.
    - **Scheduler run time (minutes)**: Enter a value in minutes. The value represents the frequency at which live scheduler runs. The default value is 5 minutes.

      Note: Scheduler run time is only applicable to live mode. You must configure the scheduler run time only if you do not add an end date as the integration will continue to run in live mode after historical mode.

15. Click **Done**.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

Enabling and disabling AWS CloudWatch integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **AWS CloudWatch** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *Collect metric data* section. Set **Enable data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting a integration, see Deleting AWS CloudWatch integrations.

Editing AWS CloudWatch integrations

After you create your integration, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **AWS CloudWatch** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the AWS CloudWatch integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Deleting AWS CloudWatch integrations

If you no longer need your AWS CloudWatch integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.

2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **AWS CloudWatch** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the AWS CloudWatch integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating custom integrations

IBM Cloud Pak for AIOps provides many connectors, probes, and observers by default for setting up integrations with common IBM and third-party systems and services. If a default integration is not available for connecting to a system or service that you want to use with IBM Cloud Pak for AIOps, you can set up a custom integration to connect to that system or service.

You can use a custom integration to ingest user-defined log data, which is used to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts and events and published to your ChatOps interface, which help you determine the cause and resolution of a problem.

Due to each user-defined data source having different formats and sizes, not all log sources can be ingested directly with this custom integration. For more information, see Data source requirements.

Important: Custom integrations cannot use historical data for initial AI training.

Note: As an alternative to using the **Custom** integration that is provided by default, you can create your own integration connector to interact with IBM Cloud Pak for AIOps. For instance you can create a custom connector for sending events, metrics, and topology data to IBM Cloud Pak for AIOps from an external source when a default integration is not available. For more information see Creating custom integrations using the integration SDK

For more information about working with custom integrations, see the following sections:

- Data source requirements
- Creating custom integrations
  - Customizing `GET` and `POST` integrations
  - Custom mappings
- Enabling custom integrations
- Editing custom integrations
- Deleting custom integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Data source requirements

Before creating the integration, you should be aware of the following information.

- **Format of source data**: The data that is ingested with the log integration must be in newline-delimited JSON format. Each individual log entry must be on a separate line and each log entry must be less than 1 MB, otherwise the custom integration is not able to ingest the data. The following code snippet shows an example of the newline-delimited JSON format:

  ```
  {"message": "exampleMessage", "time": 500}

  {"message": "exampleMessage", "time": 500}
  ```

  Custom integrations support the ingestion of flat JSON source data but do not support nested key-value pairs. The following examples illustrate what is supported and what is not.

  Supported JSON source data: the following code snippet shows an example of flat JSON source data.

  ```
  {
     "message": "exampleMessage",
     "time": "..."
  }
  ```

  Unsupported JSON source data: the following code snippet shows an example of nested key-value pairs.

  ```
  {
     "log":["message": "exampleMessage","timezone": "exampleTimezone"],
     "time": "..."
  }
  ```

  Important: If your data does not conform with the preceding format, the custom integration is not be able to ingest the data. Review the requirements before creating your custom integration.

- **Load**: To prevent this integration placing an inordinate load on your data source and potentially impacting your logging operations, this integration only connects to one API with a default data frequency of 60 seconds. This is controlled by using the **Sampling rate** setting in the Procedure section.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting in the Procedure section.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Note: If you want to collect log data from multiple systems and then route that information through a forwarding agent, such as an Apache Kafka topic (for example, Sysdig or FluentBit), you must use the Kafka integration type. For more information about Kafka integrations, see Creating Kafka integrations.

Creating custom integrations

The custom integration type collects newline-delimited JSON from a single specific data source.

Procedure

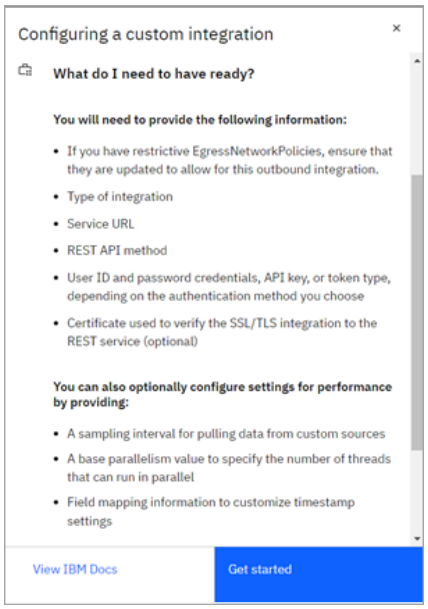
To create a custom integration from a specific source, step through the following sections:

- Adding a Custom integration
- Specifying integration parameters
- Specifying field mapping
- Specifying how log data is collected for AI training

Adding a Custom integration

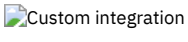
1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Custom** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.



Specifying integration parameters

1. On the *Add integration* page enter the following integration information:
   - **Name**: The display name of your integration.
   - **Description**: An optional description for the integration.
   - **Extract data from an array of log lines embedded in JSON**: When enabled, logs are extracted from an array that is embedded within a JSON object that is returned by the REST service. This option works only for an array of flat JSON objects that is embedded within a JSON object.

     
     Custom integration
   - **Embedded array path**: Type the path to the array from which you want to extract log data. The following examples, show how to configure the array path:

     ```
     {"results": [{"message": "myMessage"}]} would be results[*]
     {"results": {"events": [{"message": "myMessage"}]}} would be results.events[*]
     {"results": [{"events": [{"message": "myMessage"}]}]} would be results[0].events[*]
     ```
   - **REST method**: The method by which your data gets transferred, either `GET` or `POST`. If you select *POST* a *POST JSON (Optional)* field appears also.



- **Authentication type**: Select one of the following values:
  - **User ID/password**: The user ID and password that is required for accessing the REST service. You must enter both in the integration configuration. The HTTPS authentication header is sent to the Service URL in the format 'Authorization: Basic MyCredential'. 'MyCredential' is base64 encoded and in the format 'username:password'.
  - **API key**: The REST service is authenticated with an API key. The HTTPS authentication header is sent to the Service URL in the format 'Authorization: MyAPIKey'.
  - **Token**: The REST service is authenticated with a temporary token. The HTTPS authentication header is sent to the Service URL in the format 'Authorization: Bearer MyToken'.
  - **Query Key**: The REST service is authenticated with a query key. The HTTPS authentication header is sent to the Service URL in the format **`x-Query-Key: MyQueryKey`**.
  - **Custom**: Your custom REST service has specific authentication requirements that are not covered by the other authentication methods. You must enter both a Header and Value to configure your authentication. Note, both the Header and Value are plain text properties that are case-sensitive and does not remove spaces. The HTTPS authentication header is sent to the Service URL in the format 'Authorization: MyHeader MyValue'.
  - **None**: Your custom REST service requires no authentication. No Authorization header is sent to the target system.
- **Certificate (optional)**: Certificate used to verify the SSL/TLS connection to the REST service.
- **UI URL**: The hostname or IP address of the UI of your custom source.
- **Service URL**: The URL for the service that you want to integrate. For more information about customizing your *Service URL*, see Customizing GET and POST integrations

  Custom integration
- **Base parallelism**:It is recommended to use a higher value than 1 so that you can process data in parallel. In a small environment the available flinks slots is 16 and for a large environment, the maximum available slots is 32.
- **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is **`60`**.
- **JSON processing option**: Select a JSON processing option.
  - **None**: The default option. The JSON is not processed or modified.
  - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
  - **Filter**: This option extracts the JSON object and replaces it with an empty string.

- For more information about the options, see Managing embedded JSON.

Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see Improving data streaming performance for log anomaly detection.

2. Click **Test connection**.
3. Click **Next** to move to the next page.

Specifying field mapping

1. On the *Field mapping* page you can improve search performance by mapping the fields from your implementation fields to standard fields within IBM Cloud Pak for AIOps.

   - For more information about how field mappings are defined, see Mapping data from incoming sources.
   - For more information about using mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data using regular expressions.

   The following code snippet displays an example of field mapping that uses the supported format. When you are coding your mapping, use this example to help you.

```
{
  "codec": "custom",
  "message_field": "message",
  "log_entity_types": "kubernetes.container_name",
  "instance_id_field": "_app",
  "timestamp_settings":{
     "timestamp_field":"_ts", <MANDATORY>
     "multiplier": 1000, <OPTIONAL>
     "pattern": "yyyy-MM-dd hh:mm:ss z" <OPTIONAL, SPECIFIC TO RAW DATA OUTPUT>
  }
}
```

   If you are configuring an integration to WebSphere, then you must use the following mapping. In this mapping, all fields must be specified exactly as shown, except for the **log_entity_types** parameter. This can be empty, as shown in the following example.

```
{
"codec": "custom",
"rolling_time": 10,
"instance_id_field": "ibm_serverName",
"log_entity_types": "",
"message_field": "message",
"timestamp_settings": {
   "timestamp_field": "ibm_datetime",
   "pattern": "yyyy-MM-dd'T'HH:mm:ss.SSSZZZZZ"
                      }
}
```

   For more information about specific custom mappings, see Custom mappings.

   Alternatively, if you want to include specific fields from your WebSphere record within the log alert **resource** field to be generated by the log anomaly detection algorithms, then specify those WebSphere fields in the **log_entity_types** parameter, as shown in the example below.

```
{
"codec": "custom",
"rolling_time": 10,
"instance_id_field": "ibm_serverName",
"log_entity_types": "host,ibm_userDir,ibm_serverName,module",
"message_field": "message",
"timestamp_settings": {
   "timestamp_field": "ibm_datetime",
   "pattern": "yyyy-MM-dd'T'HH:mm:ss.SSSZZZZZ"
                      }
}
```

   For more information about the log alerts generated by the log anomaly detection algorithms, see the following topics:

   - Natural language examples
   - Statistical baseline examples

2. Click **Next** to move to the next page.

Specifying how log data is collected for AI training

1. On the *AI training and log data* page, select how you want to manage collecting data for use in AI training and anomaly detection. Click the *Data collection* toggle to turn on data collection then select how you want to collect data:

   - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your integration is used to both train AI models and analyze your data for anomalous behavior.

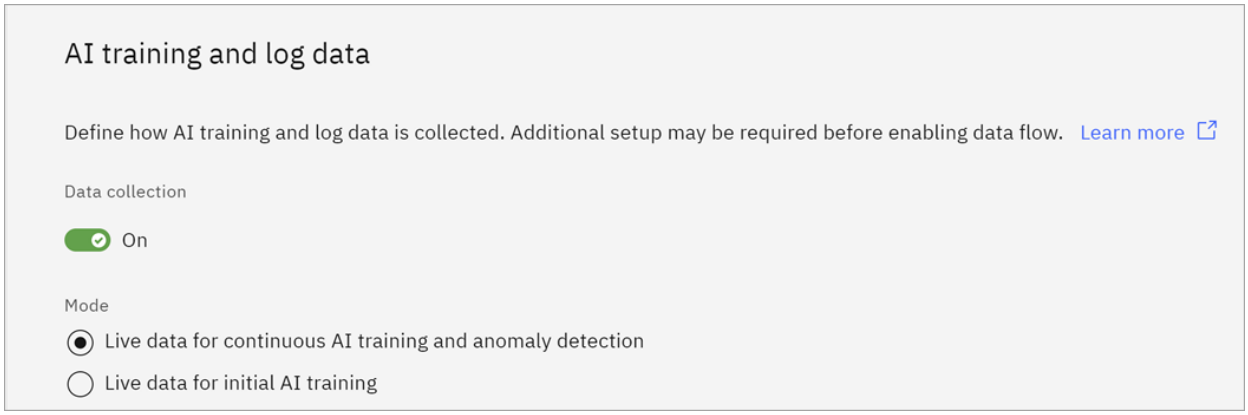
     Note: After an initial installation, there is no data at all in the system. If you select this option, then the two different log anomaly detection algorithms behave in the following ways:

     - Natural language log anomaly detection does not initially detect anomalies as no model has been trained. If **Live data for continuous AI training and anomaly detection** is set to on, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.

- Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.

- **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your integration.

  Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.



2. **Important**: Keep in mind the following considerations when you select your data collection type:

   - Anomaly detection for your integration occurs if you select *Live data for continuous AI training and anomaly detection*.
   - Custom integrations cannot use historical data for AI training.
   - Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.
3. Click **Next**.
4. On the *Resource requirements* page, you can review the slot usage for your log integrations to see if there are enough slots to fully support the integration for multizone high availability.

   If you set the **Data collection** toggle to **On**, you will see the resource management overview.

   - If your current usage and other usage are less than the provisioned slots, but the HA slots exceed the provisioned slots, you will be able to create the integration, but will see a warning that you do not have enough slots. The integration will not have multizone high availability.
   - If your projected usage exceeds the provisioned slots, you will not be able to create the integration because you do not have enough slots on your system for log data integrations.
   - If your total slots, including HA slots, are within the provisioned slots, the integration will have multizone high availability.

     Note: HA operation assumes high availability for three zones.

   If you set the **Data collection** toggle to **Off**, you will see a message stating that you need to enable logs data collection to see the resource management overview. When data collection is off, no slots are used by that integration.
5. Click **Done**.

You have created a custom integration in your instance. After you create your integration, you must enable the data collection to connect your integration with the AI of IBM Cloud Pak for AIOps. For more information about enabling your integration, see Enabling custom integrations.

To create more integrations (such as a ChatOps integration), see Configuring Integrations.

For more information about working with the insights provided by your integrations, see ChatOps insight management.

Customizing `GET` and `POST` integrations

`GET` integrations

If you customize the *Service URL* of your integration, you must include the start and end time in the URL with properly formatted timestamps (`SimpleDateFormat`). For epoch timestamps, milliseconds and seconds are supported. The following timestamp identifiers are supported in the **Service URL** field:

- <start-epoch-ms>, <end-epoch-ms>. For example, `http://exampleurl.com/v1/export?to=<end-epoch-ms>&from=<start-epoch-ms>`.
- <start-epoch-s>, <end-epoch-s>. For example, `http://exampleurl.com/v1/export?to=<end-epoch-s>&from=<start-epoch-s>`.
- <start-`SimpleDateFormat`>, <end-`SimpleDateFormat`>. For example, `http://exampleurl.com/v1/export?to=<end-yyyy-MM-dd'T'HH:mm:ss.SSSXXX>&from=<start-yyyy-MM-dd'T'HH:mm:ss.SSSXXX>`.

`POST` integrations

For `POST` integrations, you do not need to customize the *Service URL* value. For these integrations, the payload is in the body of the JSON and you can customize the format, but must include the start and end times with properly formatted timestamps (`SimpleDateFormat`). For epoch timestamps, milliseconds and seconds are supported. The following timestamp identifiers are supported in the **Service URL** field:

- <start-epoch-ms>, <end-epoch-ms>. For example, `{"queryString":"exampleQuery","start":"<start-epoch-ms>", "end":"<end-epoch-ms>","isLive":false}`
- <start-epoch-s>, <end-epoch-s>. For example, `{"queryString":"exampleQuery","start":"<start-epoch-s>", "end":"<end-epoch-s>","isLive":false}`
- <start-`SimpleDateFormat`>, <end-`SimpleDateFormat`>. For example, `{"query": {"range": {"@timestamp": {"gte": "<start-yyyy-MM-dd'T'HH:mm:ss.SSSXXX>","lt": "<end-yyyy-MM-dd'T'HH:mm:ss.SSSXXX>"}}},"size": 10000}`

Custom mappings

Custom mappings allow more flexibility regarding timestamp settings. You must provide the mapping information about the timestamp field and how to parse the timestamp. Custom mappings support unnested JSON inputs.

The basic building blocks are similar to other integrations. The `rolling_time`, `instance_id_field`, `log_entity_types`, and `message_field` mappings are defined in the same way. For custom integrations, an extra `timestamp_settings` mapping needs to be specified. It contains the required `timestamp_field` to tell what field contains the timestamp, and two optional fields `pattern` and `multiplier`. The `pattern` field must be used when the timestamps must be parsed for a `String` representation, and the m`ultiplier` field must be used when the timestamp is represented as the `Double`type.

When timestamps are formatted correctly, as 13-digit UNIX® timestamps, you do not need to specify the `pattern` or `multiplier` fields. You need to define the `timestamp_field` field as in the following example:

```
{
    "rolling_time": 10
    "instance_id_field": "_app",
    "log_entity_types": "kubernetes.container_name",
    "message_field": "message",
    "timestamp_settings":{
      "timestamp_field":"_ts"
    }
{
```

Custom mapping with String type timestamps

When timestamps are represented as Strings, they must be parsed and converted to 13-digit UNIX timestamps. In this case, the `pattern` field must contain the parsing pattern. For example, if the timestamp is `2020-12-31 23:59:59 PST`, the integration mapping can be defined as follows:

```
{
    "rolling_time": 10,
    "instance_id_field": "_account",
    "log_entity_types": "kubernetes.container_name",
    "message_field": "_line",
    "timestamp_settings":{
      "timestamp_field":"_ts",
      "pattern": "yyyy-MM-dd hh:mm:ss z"
    }
}
```

Custom mapping with Double type timestamps

It might occur that timestamps are not represented in milliseconds. In this case, you must multiply or divide the input timestamps to obtain the milliseconds. The following example illustrates how you can use the `multiplier` field to obtain milliseconds.

Consider the following source timestamp without milliseconds:

```
{
   "_ts": 1569873609.539,
   "_account": "...",
   "_cluster": "...",
   "_host": "...",
   "_logtype": "...",
   "_line": "...",
   "_id": "...",
   "__key": "...",
}
```

You can set the `multiplier` field in the mapping to show the timestamps in milliseconds:

```
{
    "rolling_time": 10,
    "instance_id_field": "_account",
    "log_entity_types": "kubernetes.container_name",
    "message_field": "_line",
    "timestamp_settings":{
      "timestamp_field":"_ts",
      "multiplier": 1000
    }
}
```

Custom mapping with static instance ID support

You can input a static value for the `instance_id_field` in the IBM Cloud Pak for AIOps pipeline. See the following example:

```
{
   "rolling_time":10,
   "instance_id_field":"%aStaticInstanceId%",
   "log_entity_types":"kubernetes.container_name",
   "message_field":"_line",
   "timestamp_settings":{
```

```
      "timestamp_field":"_ts",
      "pattern":"yyyy-MM-dd hh:mm:ss z"
   }
}
```

The custom mapping places the value that is contained between `%...%` in the **`instance_id_field`** field throughout the IBM Cloud Pak for AIOps data collection. The following example illustrates the resulting data from the preceding custom mapping:

```
{
   "message":"exampleLogMessage",
   "instance_id":"aStaticInstanceId",
   "entities":{
      "pod":"examplePod",
      "_cluster":"exampleCluster",
      "container":"exampleContainer"
   },
   ...,
   "timestamp":1605195377154
}
```

Supported custom-mapping fields

Custom mapping fields

Field Name	Description	Default Value
rolling_time	The recurring time interval to capture data from in seconds.	10
instance_id_field	The field name in the incoming data that contains the application name.	kubernetes.namespace_name, kubernetes.host, kubernetes.container_name
log_entity_types	The field name in the incoming data that contains the entity to extract and analyze.	kubernetes.container_name
message_field	The field name in the incoming data that contains the log message.	`@rawstring`
standard_entity_types	Standardized fields. If declared, the performance for event grouping can be improved.	`None`
standard_entity_types .pod_name	Standardized field. If declared, the performance for event grouping can be improved.	`pod`
standard_entity_types .node_name	Standardized field. If declared, the performance for event grouping can be improved.	`node`
timestamp_settings	Customizable settings on timestamps from source's raw log output.	`None`
timestamp_settings.timestamp_field	The field name in the incoming data that contains the data's timestamp.	`_ts`
timestamp_settings.pattern (Optional)	Must be used when the timestamps must be parsed for a string representation. Custom integrations support `SimpleDateFormat` for timestamps in incoming data.	`yyyy-MM-dd hh:mm:ss z`
timestamp_settings.multiplier (Optional)	Used when epoch timestamps are represented as a double.	`None`

Enabling and disabling custom integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. If you selected *Live data for initial AI training* when you created your integration, you must disable the integration before AI model training. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **Custom** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

Your integration is now enabled or disabled. For more information about deleting a integration, see Deleting custom integrations.

Editing custom integrations

After you create your integration, you can edit the integration. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Custom** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Custom integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited. If you have not enabled or disabled your application, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling custom integrations. For more information about deleting a integration, see

Deleting custom integrations.

Deleting custom integrations

If you no longer need your custom integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable data collection before deleting your integration. For more information about disabling data collection, see Enabling and disabling custom integrations.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Custom** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Custom integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating custom integrations using the integration SDK

You can create your own integration connectors to interact with IBM Cloud Pak for AIOps, such as for sending events, metrics, and topology data to IBM Cloud Pak for AIOps from an external source when a default integration is not available.

The framework for creating integration connectors is documented in the following repository, which also hosts the Java template for new custom integration connectors for IBM Cloud Pak for AIOps: IBM/cp4waiops-connectors-java-template. This template provides you with a working example for extending IBM Cloud Pak for AIOps. You can use this example as a starting base for developing your own custom integration. Details are included in the repository documentation to help you understand the framework architecture and to help you proceed from development to deployment.

The Java template uses the integration SDK, which can be obtained from Maven central. This SDK abstracts away the gRPC logic, provides security, and provides common methods to help you accelerate your custom integration development. Check Maven central for the latest version of the SDK before you create or update your custom integrations.

With the SDK and template, you can create your own integration connector for sending the following types of data to IBM Cloud Pak for AIOps:

- Events
- Metrics
- Topology

Creating dashboards using the dashboard extension toolkit

You can create your own dashboards to integrate with IBM Cloud Pak for AIOps, such as for summarizing alerts and incidents data with the IBM Cloud Pak for AIOps console.

A guide for creating dashboards is documented in the following repository, which also hosts dashboard examples for IBM Cloud Pak for AIOps: IBM/aiops-ui-extension-template. You can use these examples as a starting base for developing your own custom dashboards. Details are included in the repository documentation to help you understand the dashboard extension framework and to help you proceed from development to deployment.

Within dashboards created using the toolkit, you can leverage a vast variety of data from IBM Cloud Pak® for AIOps directly from the APIs. For more information, see APIs.

Creating Dynatrace integrations

A Dynatrace integration can provide metric, event, and topology data for the applications, services, process groups, and infrastructure that you monitor with Dynatrace. Multiple Dynatrace integration technologies are available. You need to have both an endpoint and an API Key ready. The AI Algorithms that analyze Dynatrace data are probable cause, and temporal correlation. Dynatrace SaaS 1.247 with v2 Metrics API is the version that is supported for the Dynatrace integration.

For more information about working with Dynatrace integrations, see the following sections:

- Creating Dynatrace integrations
- Enabling Dynatrace integrations
- Editing Dynatrace integrations
- Deleting Dynatrace integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating Dynatrace integrations

There are three data collection methods that can be used when creating Dynatrace integrations. The first method outlined here shows how to gather metric data only, while the second shows a new method of gathering both metric and event data together. Note that metric and event data collection is available only for technology preview, and is not for production usage. The third method is for collecting topology data.

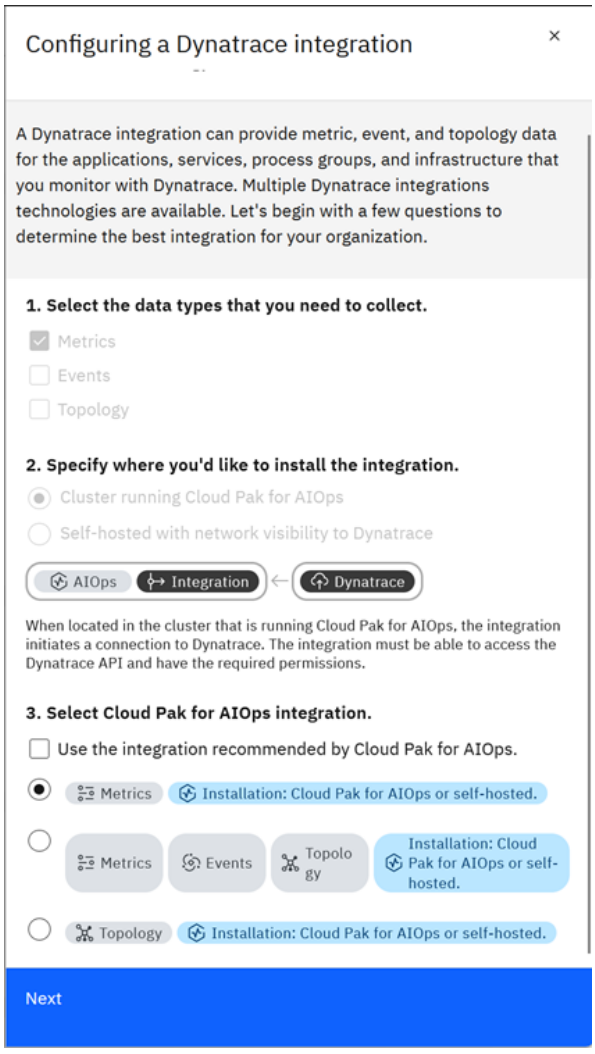
- Method 1: Metric-only data collection
- Method 2: Metric, event, and topology data collection
- Method 3: Topology-only data collection

Method 1: Metric-only data collection

To create a Dynatrace integration, collecting metric data only, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of all integrations, find and click the **Dynatrace** tile.

   Note: If you do not immediately see Dynatrace, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section, or alternatively start typing *Dynatrace* in the search bar and the Dynatrace tile appears.
5. On the side-panel, review the instructions and select your required options from the following sections:
   - **Select the data types that you need to collect**: In this case select the *Metrics* checkbox, and not *Events* or *Topology*.
   - **Specify whether you'd like to install the integration**: Select between the *Cluster running Cloud Pak for AIOps* or *Self-hosted with network visability to Dynatrace* radio buttons.
   - **Select Cloud Pak for AIOps integration**: Deselect the **Use the integration recommended by Cloud Pak for AIOPs** checkbox.
   - Select the **Metrics** option at the top.



6. Then, click **Next**.

You see an *Overview*, further direction such as *what you need to have ready* and also *what AI algorithms analyze Dynatrace data* information.

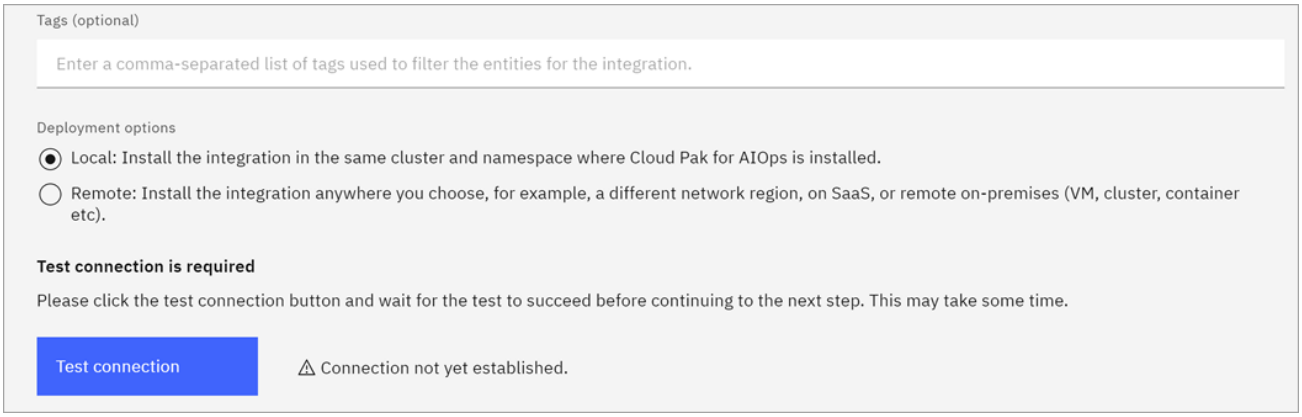
7. Click **Get Started**.
8. On the *Add integration* page, define the following general integration details:
   - **Name**: The display name of your integration. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
   - **Description**: An optional description for the integration.
   - **Endpoint**: Dynatrace data source base url, for example `https://<your-environment-id>.live.dynatrace.com`.
   - **Access Token (API key)**: Enter an API key for the integration.



   - **Tags** (optional): Enter a comma-separated list of tags used to filter the entities for the integration.

     Note: It is recommended to configure *Tags* in order to collect data only for tags you are interested in.
   - **Deployment options**: Select between *Local* and *Remote* radio buttons.
     - Local: Install the integration in the same cluster and namespace where Cloud Pak for AIOps is installed.
     - Remote: Install the integration anywhere you choose, for example in a different network region, on SaaS, or remote on-premises (VM, cluster, container, and so on).

       Note: If you select *Remote*, you are be provided with a remote deployment script to establish your integration after adding it.
9. To verify the SSL/TLS configuration:
   1. Set the **Verify the SSL/TLS certificate for the REST service** toggle to *on*.
   2. Either enter a CA certificate used to verify the SSL/TLS integration to the REST service or click **Fetch SSL certificate** to fetch the certificate automatically.
10. Click **Test connection**. It is best to wait for the test to succeed before continuing to the next step. This can take some time.

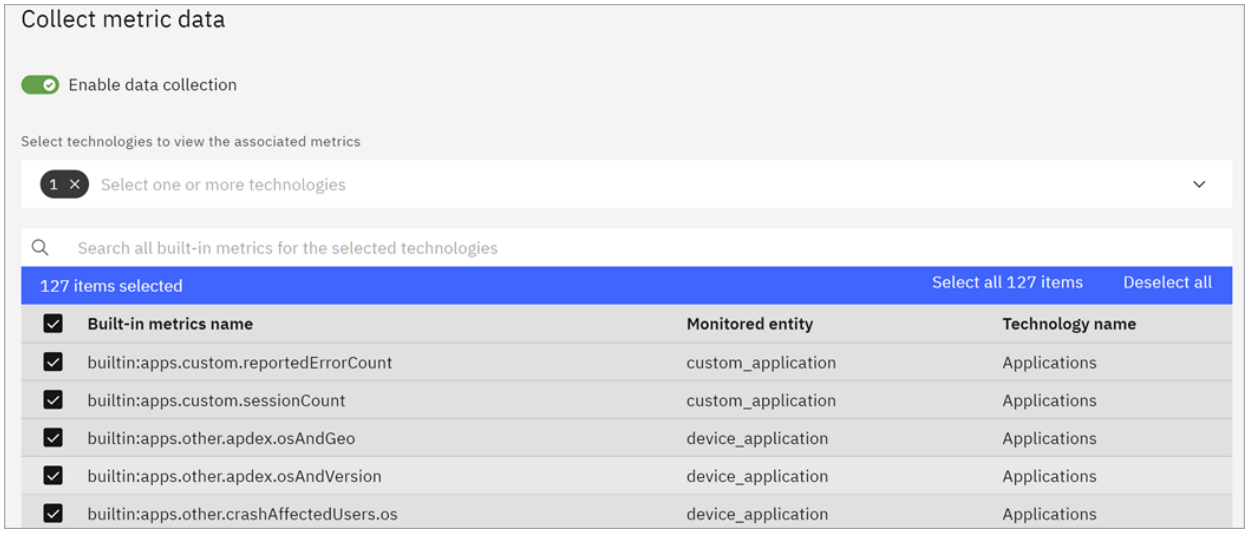


11. Click **Next**.
12. On the *Configure Proxy* (Optional) page complete the following fields:
    - **Connect to Dynatrace via a proxy server**: Set the toggle button to *on*.
    - **URL of HTTP proxy server**: Enter a URL for the HTTP proxy server. **Note:** The URL for the metrics-only integration should have the protocol in the URL.
    - **Port number of proxy server**: Enter a port. The default is 8080.



Note: Only unauthenticated HTTP proxy is supported.

13. Click **Next**.
14. Enter the following *Collect metric data* to define how you want to collect metric data:
    - **Enable data collection**: Set the *Enable data collection* toggle button to *on*.
    - **Select technologies to view the associated metrics**: From this dropdown menu you can click the *Select all technologies* checkbox, or pick and choose technologies from the list as you need. You then see a list of all built-in metrics names for the options you selected. You can leave all checked or pick and choose from the list by clicking in the respective checkbox to deselect particular metrics for a technology. It is recommended to only select the metrics that are needed.



    - **Aggregation intervals (minutes)**: The frequency at which data is collected. The default is 5.
    - **Only collect live data once the integration is established**: Click this checkbox if you want to only collect live data, as you are not be able to collect historical data after live data collection has started. If unchecked you are able to collect historical data.

      Note: The recommended time period for collecting historical metric data is two weeks. You can collect up to 31 days of historical data so your start date should not exceed 31 days from the present. If the end date is not specified, Cloud Pak for AIOps collects historical data from the specified start date and continue with live data collection. You are not be able to collect historical data after live data collection has started.
    - **Start date**: Enter in the format *mm/dd/yyyy*.
    - **Start time**: Enter in the format *hh:mm*.
    - **Timezone**: Set to local time.
    - **Add End Date**: click the *Add End Date* button if you want to set an end date.
15. Click **Done**.

Note: The API limit can cause a delay in receiving metrics from Dynatrace. Each Dynatrace token is limited to 50 API requests per minute. Multiple API requests might be needed for each metric data collection interval based on the amount of data that Dynatrace returns. If the required requests to the API exceed the limit, the metrics-only integration throttles the requests until more requests are available in the next minute. Review the performance considerations for metric integrations and the filtering options available to optimize this behavior. For more information, see Performance considerations for metric data collection.

A message appears to indicate that the integration is complete, and that IBM Cloud Pak for AIOps has successfully created your Dynatrace integration and will start collecting data shortly.

Method 2: Metric, event, and topology data collection

To create a Dynatrace integration collecting metric and event data together, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.

2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of all integrations, find and click the **Dynatrace** tile.

   Note: If you do not immediately see Dynatrace, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section, or alternatively start typing *Dynatrace* in the search bar and the Dynatrace tile appears.

5. On the side-panel, review the instructions and select your required options from the following sections:

   - **Select the data types that you need to collect**: Select the relevant checkbox from *Metrics*, *Events*, or *Topology*, which in this case is *Metrics*, *Events* or *Topology*.
   - **Specify whether you'd like to install the integration**: Select between the *Cluster running Cloud Pak for AIOps* or *Self-hosted with network visability to Dynatrace* radio buttons.
   - **Select Cloud Pak for AIOps integration**: You can leave this at the recommended default *Use the integration recommended by Cloud Pak for AIOps*, which collects both metric and event data.

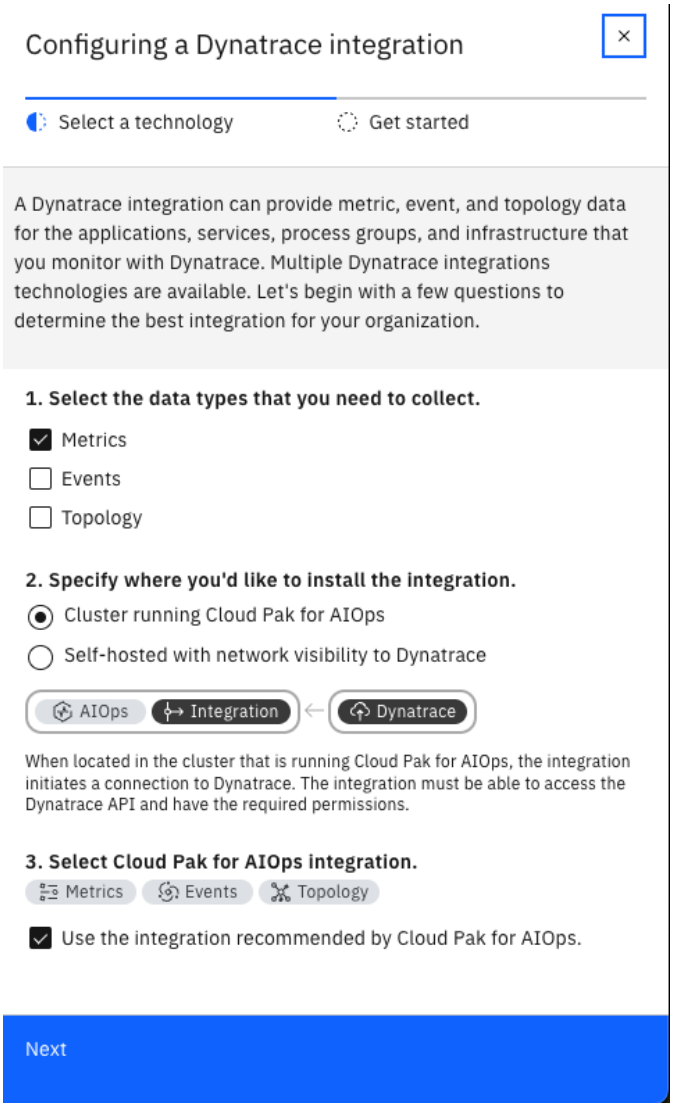


Figure. Metric selected

   Important: Depending on which of the data types that you choose to collect the following sections vary. This integration does not support historical data collection. However it is worth noting that if did select *Topology* you would be brought to a *New observer job* page. Using the Dynatrace observer, you can query a specified Dynatrace environment for information about its applications, services, process groups, and infrastructure entities. For more information, see Configuring a Dynatrace Observer job.

6. Then, click **Next**.

   You see an *overview*, further direction such as *what you need to have ready* and also *what algorithms analyze Dynatrace data* information.

7. Click **Get Started**.
8. On the *Add integration* page, define the following general integration details:

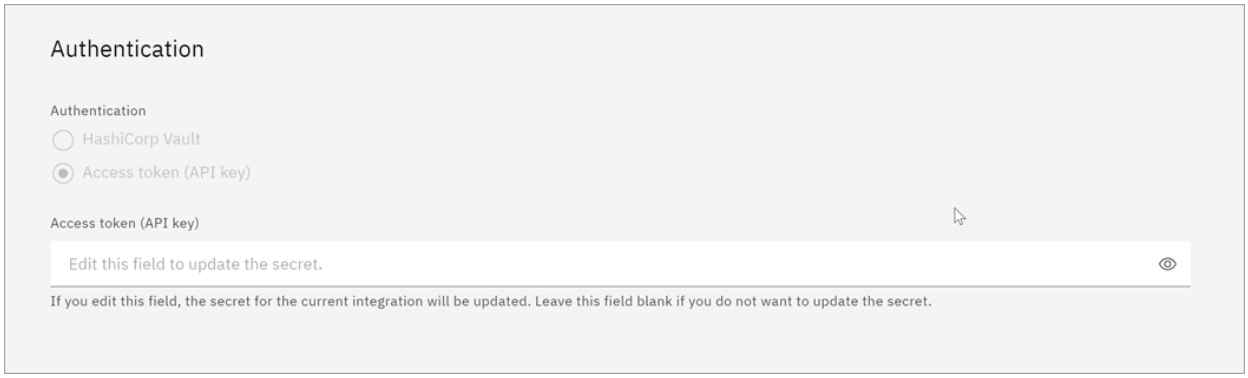
   - **Name**: The display name of your integration. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.

- **Description**: An optional description for the integration.
- **Endpoint**: Dynatrace data source base url, for example **`https://<your-environment-id>.live.dynatrace.com`**.
- **Dynatrace zone** (optional): Enter a single Dynatrace zone to monitor for the integration.
- **Tags** (optional): Enter a comma-separated list of tags used to filter the entities for the integration.

  Note: It is recommended to configure one of either *Dynatrace zone* or *Tags* in order to collect data only for tags or management zones you are interested in.
- **Deployment options**: Select between *Local* and *Remote* radio buttons.
  - Local: Installs the integration in the same cluster and namespace where Cloud Pak for AIOps is installed.
  - Remote: Installs the integration anywhere you choose, for example, a different network region, on SaaS, or remote on-premises (VM, cluster, container, and so on) and enter a name in the **Installation name** field.

    Important: Podman must be installed to execute the remote deployment script with a recommended minimum version of 4.4.1. Read and write access is required for the folder where the remote deployment script is executed and the script also must have execution permissions in order to be run. No more than one remote deployment script should be executed per host.

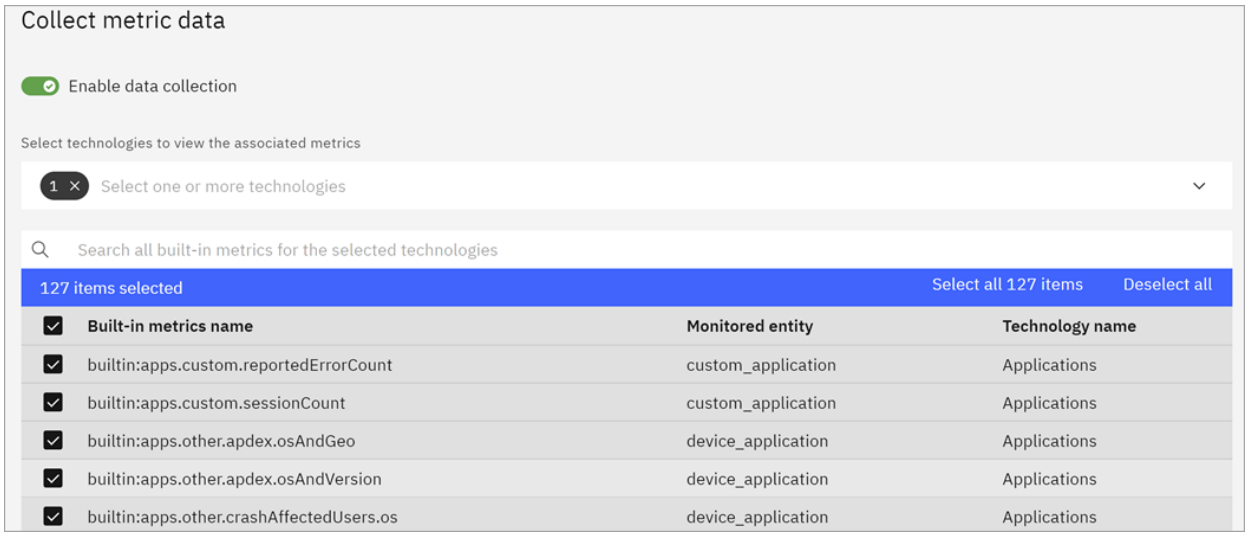
9. Click **Next**.
10. On the *Authentication* page, define the following general integration details:
    - **Authentication**: Select between the *Hashicorp Vault* and *Access token (API key)* radio buttons.



    If you select *Access token (API key)* you need to enter an API key for the integration, and if you select *Hashicorp Vault* you need to complete the following fields:

    - **Endpoint (URL)**: Endpoint URL of the Hashicorp Vault server instance.
    - **HashiCorp Vault access token**: Enter a Hashicorp Vault access token for this integration.
    - **HashiCorp Vault secret path**: Enter a Hashicorp Vault secret path for this integration.
    - **CA HTTPS Certificate**: Enter a CA certificate used for HTTPS with Hashicorp Vault.
    - **HashiCorp Vault secret refresh rate**: The HashiCorp Vault secret is refreshed automatically at a fixed hourly interval. The default refresh rate is every 24 hours.

      Note: Only HTTPS for Vault is supported, and not HTTP.
11. Click **Next**.
12. Enter the following *Collect metric data* (optional) to define how you want to collect metric data:
    - **Enable data collection**: Set the *Enable data collection* toggle button to *on*.
    - **Select technologies to view the associated metrics**: From this dropdown menu you can click the *Select all technologies* checkbox, or pick and choose technologies from the list as you need. You then see a list of all built-in metric names for the options you selected. You can leave all checked or pick and choose from the list by clicking in the respective checkbox to deselect particular metrics for a technology. It is recommended to only select the metrics that are needed.
    - **Polling interval (seconds)**: The frequency at which data is collected (in seconds). The default value is 300.



13. Click **Next**.
14. Enter the following *Collect event data* (optional) to define how you want to collect event data:
    - **Enable data collection**: Set the *Enable data collection* toggle button to *on*.
    - **Select technologies to view the associated events**: From this dropdown menu you can click the *Select all technologies* checkbox, or deselect this, and pick and choose from the list as you need. You then see a list of *Event names* for the technologies you selected. You can leave all checked or pick and choose from the list by clicking in the respective checkbox. It is recommended to only select the events that are needed.
    - **Polling interval (seconds)**: The frequency at which data is collected (in seconds). The default value is 60.
15. Click **Done**.

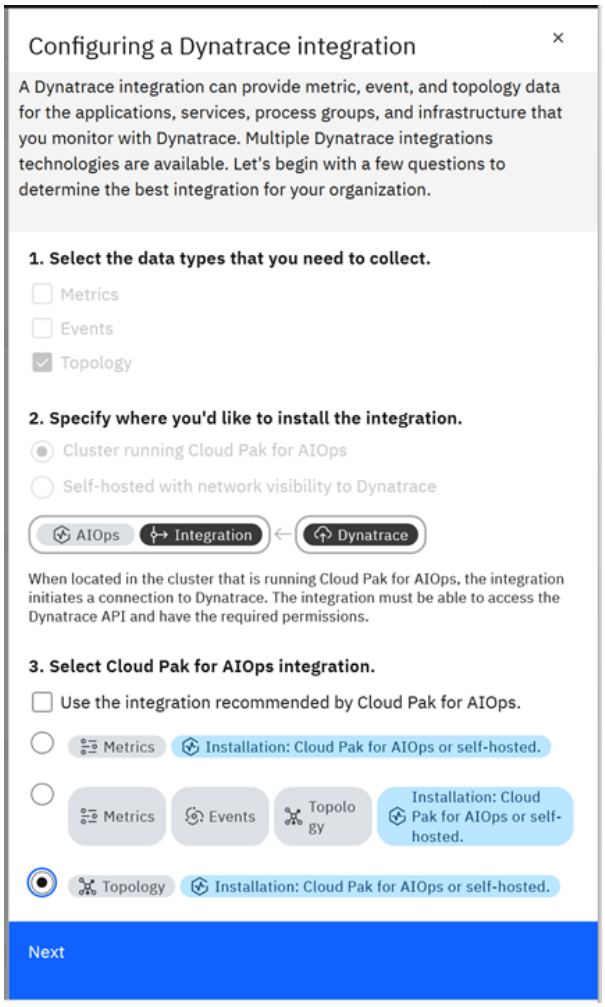
A message appears to indicate that the integration is complete, and that IBM Cloud Pak for AIOps has successfully created your Dynatrace integration and will start collecting data shortly.

Method 3: Topology-only data collection

To create a Dynatrace integration, collecting topology data, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of all integrations, find and click the **Dynatrace** tile.

   Note: If you do not immediately see Dynatrace, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section, or alternatively start typing *Dynatrace* in the search bar and the Dynatrace tile appears.
5. On the side-panel, review the instructions and select your required options from the following sections:
    - **Select the data types that you need to collect**: In this case select the *Topology* checkbox.
    - **Specify whether you'd like to install the integration**: Select between the *Cluster running Cloud Pak for AIOps* or *Self-hosted with network visability to Dynatrace* radio buttons.
    - **Select Cloud Pak for AIOps integration**: Deselect the **Use the integration recommended by Cloud Pak for AIOPs** checkbox.
    - Select the **Topology** option at the bottom.



6. Then, click **Next**.

   You see an *Overview*, further direction such as *what you need to have ready* and also *what AI algorithms analyze Dynatrace data* information.

7. Click **Get Started**.

8. On the *Add integration* page, define the following general integration details:

   - **Name**: The display name of your integration. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
   - **Description**: An optional description for the integration.
   - **Endpoint**: Dynatrace data source base url, for example `https://<your-environment-id>.live.dynatrace.com`.
   - **Dynatrace zone** (optional): Enter a single Dynatrace zone to monitor for the integration.
   - **Tags** (optional): Enter a comma-separated list of tags used to filter the entities for the integration.

     Note: It is recommended to configure one of either *Dynatrace zone* or *Tags* in order to collect data only for tags or management zones you are interested in.

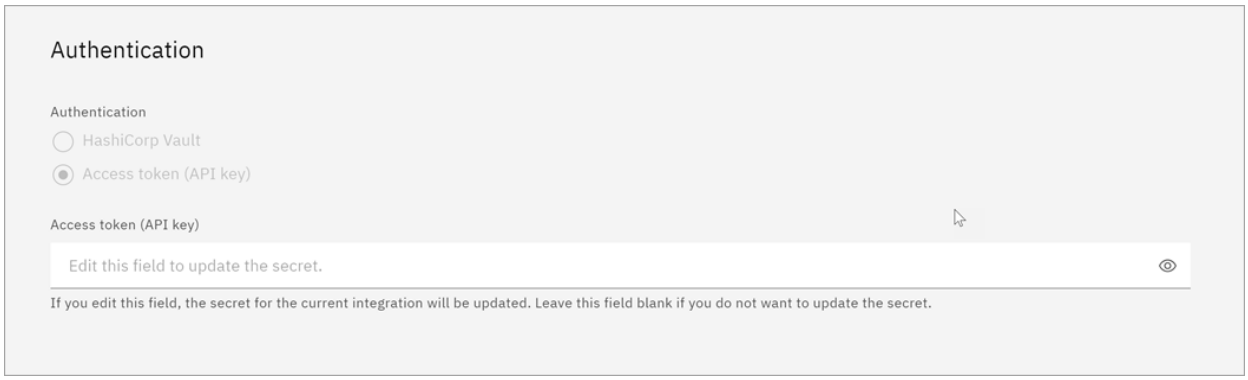
   - **Deployment options**: Select between *Local* and *Remote* radio buttons.
     - Local: Installs the integration in the same cluster and namespace where Cloud Pak for AIOps is installed.
     - Remote: Installs the integration anywhere you choose, for example, a different network region, on SaaS, or remote on-premises (VM, cluster, container, and so on) and enter a name in the **Installation name** field.

       Important: Podman must be installed to execute the remote deployment script with a recommended minimum version of 4.4.1. Read and write access is required for the folder where the remote deployment script is executed and the script also must have execution permissions in order to be run. No more than one remote deployment script should be executed per host.

9. Click **Next**.

10. On the *Authentication* page, define the following general integration details:

    - **Authentication**: Select between the *Hashicorp Vault* and *Access token (API key)* radio buttons.



If you select *Access token (API key)* you need to enter an API key for the integration, and if you select *Hashicorp Vault* you need to complete the following fields:

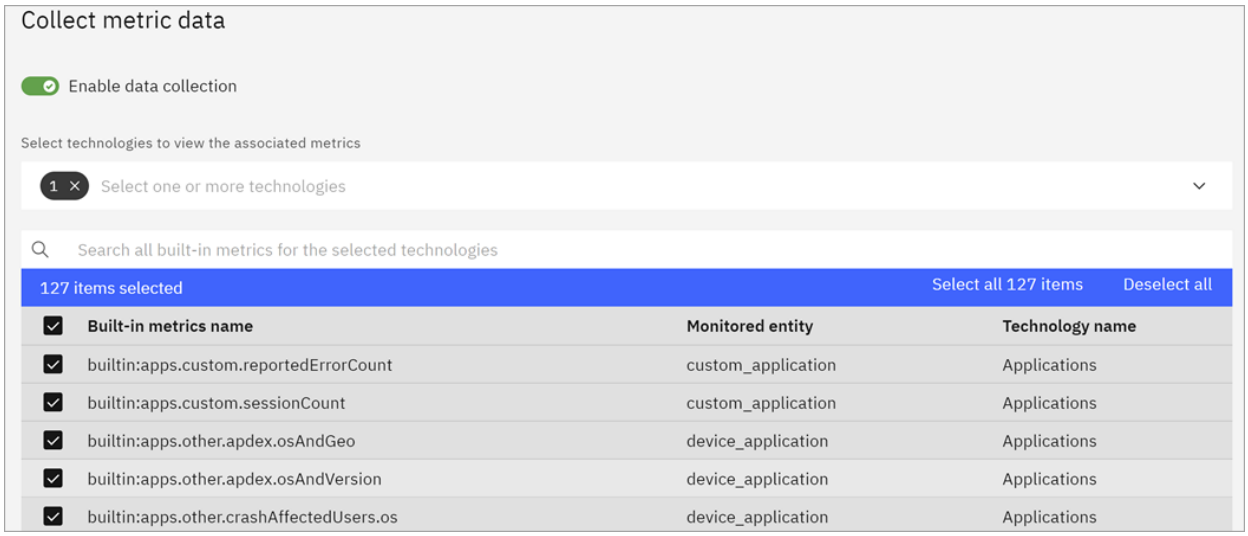
- **Endpoint (URL)**: Endpoint URL of the Hashicorp Vault server instance.
- **HashiCorp Vault access token**: Enter a Hashicorp Vault access token for this integration.
- **HashiCorp Vault secret path**: Enter a Hashicorp Vault secret path for this integration.
- **CA HTTPS Certificate**: Enter a CA certificate used for HTTPS with Hashicorp Vault.
- **HashiCorp Vault secret refresh rate**: The HashiCorp Vault secret is refreshed automatically at a fixed hourly interval. The default refresh rate is every 24 hours.

  Note: Only HTTPS for Vault is supported, and not HTTP.

11. Click **Next**.
12. On the *Configure Proxy* (Optional) page complete the following fields:
    - **Connect to Dynatrace via a proxy server**: Set the toggle button to *on*.
    - **URL of HTTP proxy server**: Enter a URL for the HTTP proxy server. **Note:** The URL for the metrics-only integration should have the protocol in the URL.
    - **Port number of proxy server**: Enter a port. The default is 8080.



13. Click **Next**.
14. Enter the following *Collect metric data* (optional) to define how you want to collect metric data:
    - **Enable data collection**: Set the *Enable data collection* toggle button to *on*.
    - **Select technologies to view the associated metrics**: From this dropdown menu you can click the *Select all technologies* checkbox, or pick and choose technologies from the list as you need. You then see a list of all built-in metric names for the options you selected. You can leave all checked or pick and choose from the list by clicking in the respective checkbox to deselect particular metrics for a technology. It is recommended to only select the metrics that are needed.
    - **Polling interval (seconds)**: The frequency at which data is collected (in seconds). The default value is 300.



15. Click **Next**.
16. Enter the following *Collect event data* (optional) to define how you want to collect event data:
    - **Enable data collection**: Set the *Enable data collection* toggle button to *on*.
    - **Select technologies to view the associated events**: From this dropdown menu you can click the *Select all technologies* checkbox, or deselect this, and pick and choose from the list as you need. You then see a list of *Event names* for the technologies you selected. You can leave all checked or pick and choose from the list by clicking in the respective checkbox. It is recommended to only select the events that are needed.
    - **Polling interval (seconds)**: The frequency at which data is collected (in seconds). The default value is 60.
17. Click **Next**.
18. Enter the following *Collect topology data* (optional) to define how you want to collect topology data:
    - **Enable data collection**: Set the *Enable data collection* toggle button to *on*.
    - **Cloud Pak for AIOps topology entity selector**: This mapping configures the entities and filters used to ingest topology data from Dynatrace to Cloud Pak for AIOps. The entity selector from the Observer Jobs UI can be used in this field.
    - **Data collection mode**: Select between the **Collect topology data continuously at a specified time interval** and **Collect topology data once** radio buttons.
    - **Schedule topology data collection**: Select between the **Start data collection after creating the integration** and **Start data collection at a scheduled date and time** radio buttons.
    - **Time interval of data collection**: Specify how frequently to run the topology data collection.
    - **Unit of time**: Specify the unit of time for topology data collection.
    - **Number of records to be fetched per page**: Specify a value between 100 and 4000. The default value is 4000.
    - **Cloud Pak for AIOps topology edge mapping** (optional): This mapping defines the relationship between the Cloud Pak for AIOps topology edges and Dynatrace. Cloud Pak for AIOps will use a default mapping for edges by default, but edge mapping values can be overrided in this field.
    - **Cloud Pak for AIOps entity mapping** (optional): This mapping defines the relationship between the Cloud Pak for AIOps topolopy entities and Dynatrace entity types. Cloud Pak for AIOps will use a default mapping for the entity relationship by default, but the entity types can be overrided in this field.
    - **Grouping vertex preference** (optional): Cloud Pak for AIOps will create a defined topology for the discovered vertices. By default, there is no grouping vertex preference.
    - **Connection timeout** (optional): The connection timeout for the topology collection in milliseconds. The default is 5000 milliseconds.
    - **Access scope** (optional): The access scope provides the scope for the topology resources. The access scope can be used to aid the mapping of alerts to resources when resources in different scopes share the same matchTokens. Examples of scope would be locations, project names, or namespaces.
19. Click **Done**.

A message appears to indicate that the integration is complete, and that IBM Cloud Pak for AIOps has successfully created your Dynatrace integration and will start collecting data shortly.

Known issue

There is currently a known issue with the Dynatrace metrics, events, and topology integration whereby the events will show up outside of the management zone if a management zone filter is applied. As a workaround, you should use a *tags* filter. The management zone filtering for metrics is not affected.

Enabling and disabling Dynatrace integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. To enable or disable a created integration, complete the following steps:

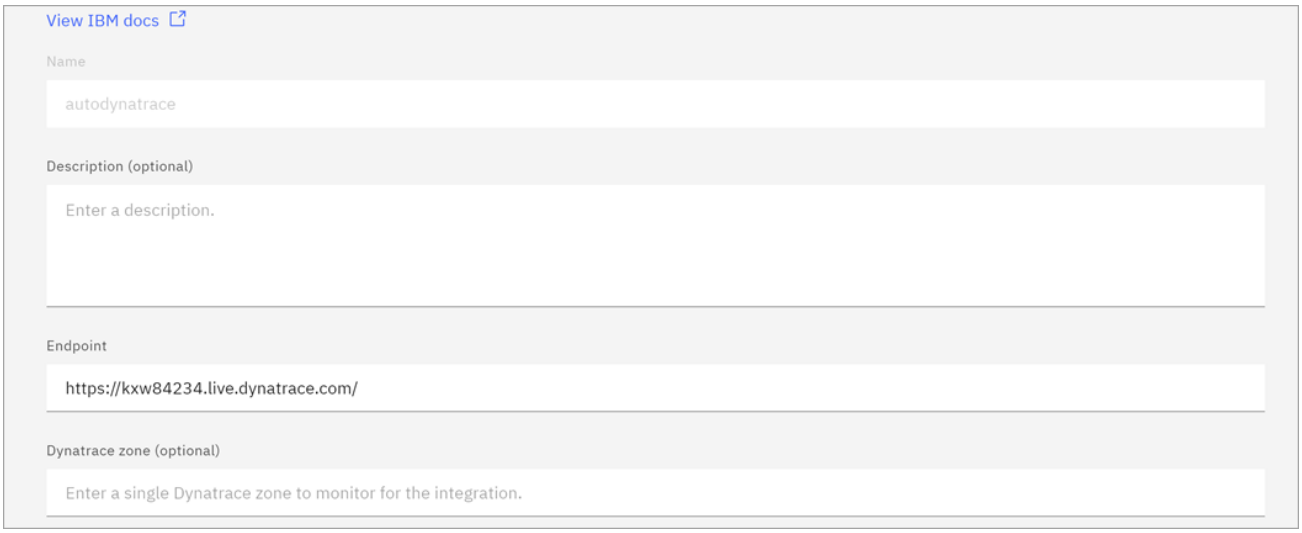
1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **Dynatrace** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *Collect metric data* section for the metrics-only Dynatrace integration, or indeed for the combined events and metrics integration go to both the *Collect metric data* and *Collect event data* sections. Set **Enable data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting an integration, see Deleting Dynatrace integrations.

Editing Dynatrace integrations

After you create your integration, your can edit the integration by completing the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Dynatrace** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Dynatrace integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.



Your integration is now edited. If your application was not previously enabled or disabled, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling Dynatrace integrations. For more information about deleting an integration, see Deleting Dynatrace integrations.

Known issue with the Dynatrace integration

When event data collection is disabled and metrics data collection is enabled you may see errors in the logs, but it will not affect data collection.

Deleting Dynatrace integrations

If you no longer need your Dynatrace integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

To delete an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Dynatrace** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Dynatrace integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**, or alternatively click the dustbin icon.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating ELK integrations

Elasticsearch, Logstash, and Kibana (ELK) integrations provide log data for anomaly detection in IBM Cloud Pak® for AIOps. An ELK integration provides log data, which is used to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts and events, and published to your ChatOps interface to help you determine the cause and resolution of a problem.

For more information about working with ELK integrations, see the following sections:

- Creating ELK integrations
- Enabling ELK integrations
- Editing ELK integrations
- Deleting ELK integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

About this task

Before creating the integration, you should be aware of the following information.

- **Load**: To prevent this integration placing an inordinate load on your data source and potentially impacting your logging operations, this integration only connects to one API with a default data frequency of 60 seconds. This is controlled by using the **Sampling rate** setting in the Procedure section.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting in the Procedure section.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Known issue with later versions of ELK

IBM Cloud Pak for AIOps only supports ELK up to version 7.12.0. For later versions of ELK, the **`event_link`** provided by AIOps does not open the correct logs in Kibana.

The reason for this incorrect link is that later versions of ELK use the **`uuid`** as the default ID for the index pattern. As a workaround, you must set the Custom index pattern ID to same value as the index pattern name.

To do so, use the following steps:

1. Log in to the ELK instance.
2. From the hamburger menu, navigate to **Stack Management**.
3. Click on **Create index pattern**.
4. Complete the required details.
5. Click on the **Show advanced settings button**.
6. Set the **Custom index pattern ID** to the same value that you specified for the **Index pattern name**. This will override the default value.

Procedure

To create an ELK integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **ELK** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.

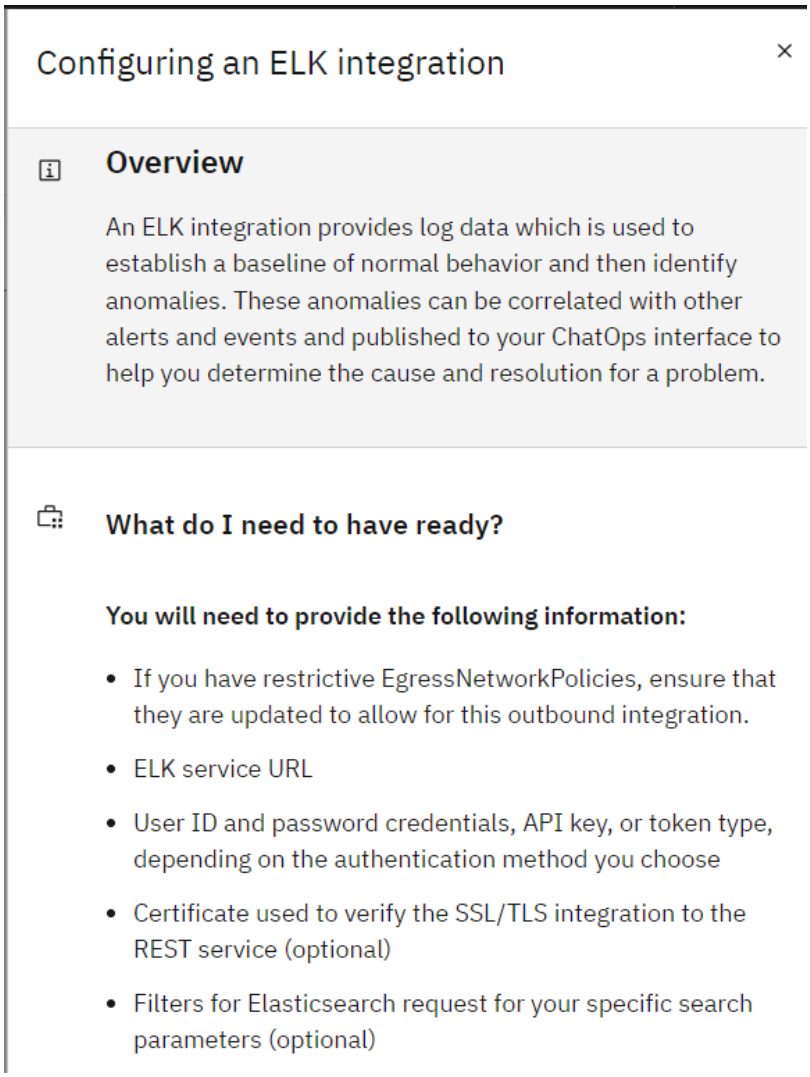


Figure. Configure ELK integration

5. On the side-panel, review the instructions and when ready to continue, click **Get started**.

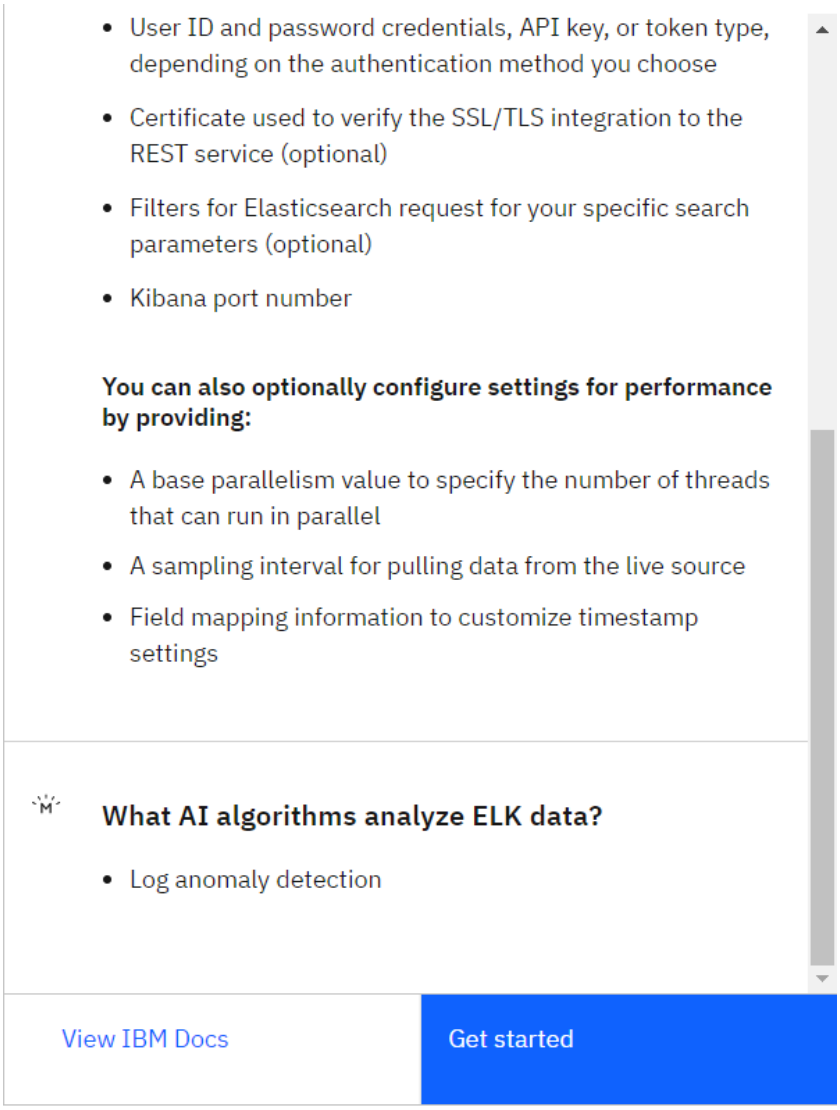


Figure. Configure ELK integration - get started

6. On the *Add integration* page, define the general integration details:

   - **Name**: The display name of your integration.
   - **Description**: An optional description for the integration.
   - **ELK service URL**: ELK service URL: The Elasticsearch host and public API port. The URL must have the target index that IBM Cloud Pak for AIOps uses to search for the data from your applications. If you want to use all data, specify `*` as your index. For example, your ELK service URL can be `https://myURL.com:8080/*`.
   - **Kibana URL**: Enter a URL for the service instance.

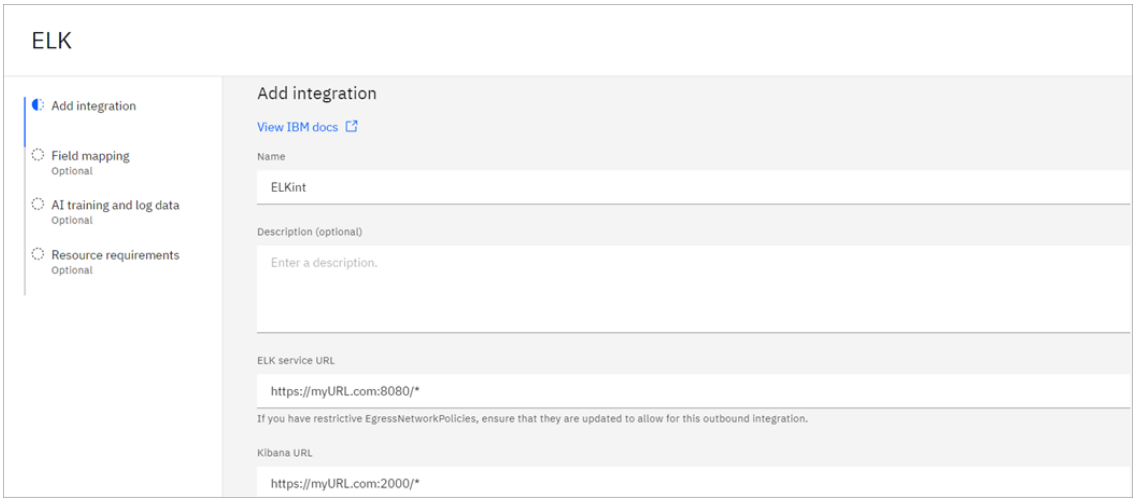


Figure. Create ELK integration

- **Authentication type**: Select one of the following values:
  - **User ID/password**: The Elasticsearch instance has a user ID and password as authentication. You must enter both in the integration configuration.
  - **API key**: The Elasticsearch instance is authenticated with an API key.
  - **Token**: The Elasticsearch instance is authenticated with a temporary token.
  - **Custom**: The Elasticsearch instance is authenticated with a header and value.
  - **None**: The Elasticsearch instance has no authentication.

    Note: If you selected API key as Authentication type value, then you need to follow the steps on ELK integration steps to use ApiKey authentication method.
- **User ID**: Enter a user ID for the integration.
- **Password**: Enter a password for the integration.
- **Certificate (optional)**: Certificate used to verify the SSL/TLS connection to the REST service.
- **Filters (optional)**: A custom Boolean Query to filter the Elasticsearch request for your specific application, terms, keywords, or other filters.
- **Time zone (optional)**: The time zone in which your data is situated. The default time is converted from the system time relative to UTC time. The default value is UTC.
- **Kibana port**: The port of the Kibana instance that is on the same host as the Elasticsearch instance. If the Kibana instance is not exposed on the default `5601` port, change this value to the port where the Kibana instance is shown.
- **Base parallelism**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots. In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this integration, you can set this value to be equal to the **source** parallelism.
- **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is `60`.
- **JSON processing option**: Select a JSON processing option.
  - **None**: The default option. The JSON is not processed or modified.
  - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
  - **Filter**: This option extracts the JSON object and replaces it with an empty string.
  - For more information about the options, see Managing embedded JSON.

Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see Improving data streaming performance for log anomaly detection.

Note: If using the Filter option, you should not use a timestamp in the filter query. It gives a parsing error in the backend.

```
    "range": {
      "@timestamp": {
        "gte": "now-2m",
        "lt" : "now"
      }
```

Other than that, ELK filter can use any clauses so long as the fields and the values that are specified in the filter are relevant to the target endpoint data set.

Filter debugging tip: You can first test the filter by using the curl command to ELK endpoint. When using the curl command, the timestamp range is optional in the elk filter, and depends on whether you want to narrow down the search over a particular period of time. If the query gets zero records with 200 HTTP responses (for example no filter error), it is an indication that the query was not able to hit the particular set of data scoped by the filter. Users can tune filter to get the needed results.

7. You test your integration by clicking **Test connection**.
8. Click **Next**.
9. Enter *Field Mapping* information (Optional):

   You can improve search performance by mapping the fields from your implementation fields to IBM Cloud Pak for AIOps's standard fields. For more information about how field mappings are defined, see Mapping data from incoming sources. For more information about using mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data that use regular expressions. Consider the supported data schema when you create your field mapping:

```
{
  "codec": "elk",
  "rolling_time": 10,
  "instance_id_field": "application_name",
  "log_entity_types": "kubernetes.pod_name",
  "message_field": "message",
  "timestamp_field": "@timestamp",
  "resource_id": "kubernetes.pod_name"
}
```

10. Click **Next**.

11. Enter AI training and log data (Optional):

    Select how you want to manage collecting data for use in AI training and anomaly detection. Click the **Data collection** toggle to turn on data collection, then select how you want to collect data:

    - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your integration is used to both train AI models and analyze your data for anomalous behavior.

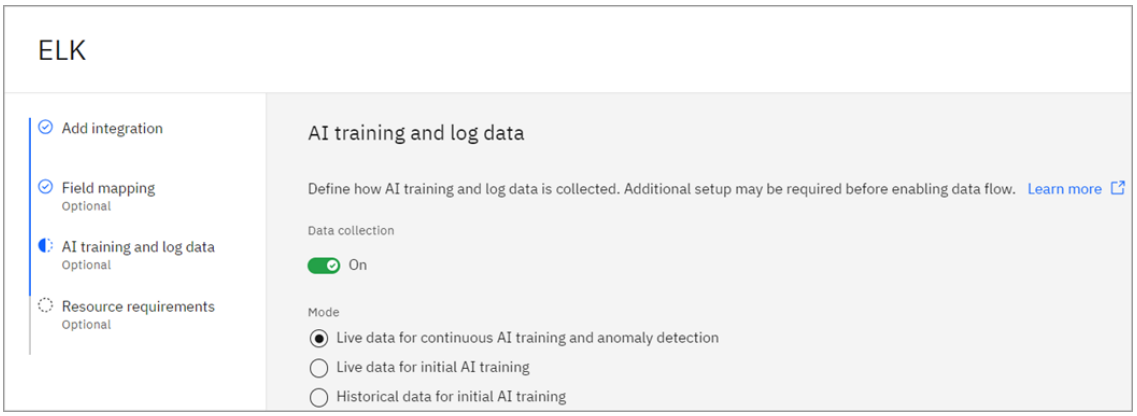


Figure. AI training

    Note: After an initial installation, there is no data at all in the system. If you select this option, then the two different log anomaly detection algorithms behave in the following ways:

    - Natural language log anomaly detection does not initially detect anomalies as no model has been trained. You can retrieve historical data (select **Historical data for initial AI training**) to speed up the retrieval of data to train on, or you can leave the **Live data for continuous AI training and anomaly detection** setting on. In the latter case, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.

    - Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.

    - **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your integration.

      Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.

    - **Historical data for initial AI training**: A single set of training data used to define your AI model. You need to give Start and End dates, and specify the parallelism of your source data. Historical data is harvested from existing logs in your integration over a specified time period in the past.

      - **Start date**: Select a start date from the calendar and enter the time in hh:mm (hours and minutes) format.

        Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.

      - **Time zone**: Select your time zone from the dropdown list.

      - **End date and time**: Click **Add end date** and select an end date from the calendar and enter the time in hh:mm format.

        Note: If you do not specify the end date, then live data collection follows the historical data collection. If you do not want to set an end date, click **Remove end date**.

      - **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

    Important: Keep in mind the following considerations when you select your data collection type:

    - Anomaly detection for your integration occurs if you select *Live data for continuous AI training and anomaly detection*.

    - Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

12. Click **Next**.

13. On the *Resource requirements* page, you can review the slot usage for your log integrations to see if there are enough slots to fully support the integration for multizone high availability.

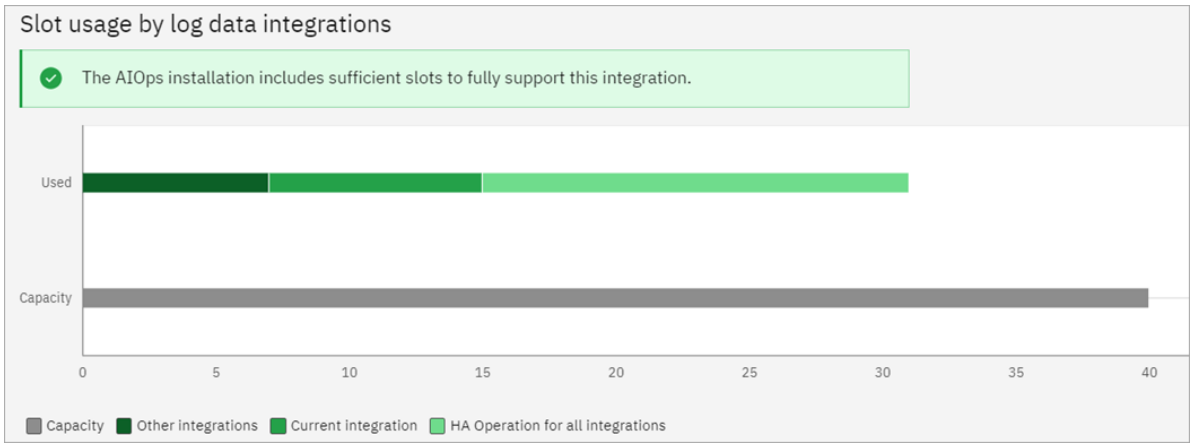


Figure. Resource requirements

If you set the **Data collection** toggle to **On**, you will see the resource management overview.

- If your current usage and other usage are less than the provisioned slots, but the HA slots exceed the provisioned slots, you will be able to create the integration, but will see a warning that you do not have enough slots. The integration will not have multizone high availability.
- If your projected usage exceeds the provisioned slots, you will not be able to create the integration because you do not have enough slots on your system for log data integrations.
- If your total slots, including HA slots, are within the provisioned slots, the integration will have multizone high availability.

  Note: HA operation assumes high availability for three zones.

If you set the **Data collection** toggle to **Off**, you will see a message stating that you need to enable logs data collection to see the resource management overview. When data collection is off, no slots are used by that integration.

14. Click **Done**.

You created an ELK integration in your instance. After you create your integration, you must enable the data collection to connect your integration with the AI of IBM Cloud Pak for AIOps. For more information about enabling your integration, see Enabling ELK integrations.

To create more integrations (such as a ChatOps integration), see Configuring Integrations.

For more information about working with the insights provided by your integrations, see ChatOps insight management.

Enabling and disabling ELK integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. If you selected *Live data for initial AI training* when you created your integration, you must disable the integration before AI model training. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **ELK** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data integration** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting a integration, see Deleting ELK integrations.

Editing ELK integrations

After you create your integration, your can edit the integration. For example, if you specified *Historical data for initial AI training* but now want your integration to pull in live data for continuous monitoring, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **ELK** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the ELK integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited. If your application was not previously enabled or disabled, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling ELK integrations. For more information about deleting a integration, see

Deleting ELK integrations.

Deleting ELK integrations

If you no longer need your ELK integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable data collection before deleting your integration. For more information about disabling data collection, see Enabling and disabling ELK integrations.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **ELK** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the ELK integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Generating an API Key for an ELK integration

When you are creating an ELK integration, you might need to generate an API key. Use the following procedure to generate the API Key.

Procedure

When you are completing these steps, replace the cluster and API key details with the details for your environment.

1. To check, or list, your existing **api_key**, run the following command:

   ```
   curl -k -X GET 'https://was911.myibm.com:9200/_security/api_key' -H 'Authorization: Basic ZWxhc3RpYzpwYXNzdzByZA==' | jq '.'
   ```

2. If you need to create an **api_key**, run the following command:

   ```
   curl -k -X POST 'https://was911.myibm.com:9200/_security/api_key' -H 'Content-Type: application/json' -H 'Authorization: Basic ZWxhc3RpYzpwYXNzdzByZA==' -d '{ "name": "my-api-key-sheikh-2", "role_descriptors": {} }' | jq '.'
   ```

3. Create the **encoded-base64-api-key** with the following command. Keep **"id": "LoRMmn4BCbCvoljFvMzq"** and **"api_key": "ZZEnRG93Q1i6wOCVW2XWug"**. These settings are needed to create the **encoded-base64-api-key**.

   ```
   echo -n "{id}:{api_key}" | base64
   ```

   For example, see the following example:

   ```
   echo -n "LoRMmn4BCbCvoljFvMzq:ZZEnRG93Q1i6wOCVW2XWug" | base64
   TG9STW1uNEJDYkN2b2xqRnZNenE6WlpFblJHOTNRMWk2d09DVlcyWFd1Zw==
   ```

4. Test the ELK query by using the **encoded-base64-api-key**:

   ```
   curl -k -X 'GET' 'https://was911.myibm.com:9200/my-index' -H 'accept: /' -H 'Authorization: ApiKey {encoded-base64-api-key}}'
   ```

5. Test the key with the IBM Cloud Pak for AIOps console:

   1. Create an ELK integration with **api_key** as **encoded-base64-api-key**.
   2. Test the integration.
   3. Turn on data collection and save.
   4. Finally, ensure that your integration is saved.

Creating Elasticsearch, Fluentd, and Kibana (EFK) integrations

You can connect to, and use, an Elasticsearch, Fluentd, and Kibana (EFK) stack as a log aggregator tool for IBM Cloud Pak for AIOps.

EFK is a suite of tools that functions as a log aggregation tool. Similar to an Elasticsearch, Logstash, and Kibana (ELK) integration with this type of integration, you can collect log data to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts to help you determine the cause and resolution of a problem.

You can connect to an Elasticsearch, Fluentd, and Kibana (EFK) stack as a variant option to connecting to an Elasticsearch, Logstash, and Kibana (ELK) stack.

There are multiple ways of hosting an Elasticsearch, Fluentd, and Kibana (EFK) cluster, one of which is the Red Hat OpenShift Logging library. The following instructions reference this library, however, the same configuration also works with other hosted EFK stacks.

For more information about working with EFK integrations, see the following sections:

- About this task
- Prerequisites
- Creating EFK integrations
- Enabling EFK integrations
- Editing EFK integrations
- Deleting EFK integrations
- Troubleshooting

About this task

Before creating the integration, you should be aware of the following information.

- **Load**: To prevent this integration placing an inordinate load on your data source and potentially impacting your logging operations, this integration only connects to one API with a default data frequency of 60 seconds. This is controlled by using the **Sampling rate** setting.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Prerequisites

As a prerequisite, you need to make sure that you have an Elasticsearch, Fluentd, and Kibana (EFK) set up on a cluster. There are many ways to set up the EFK stack. One option is to use the Red Hat OpenShift Logging library.

Note: The Red Hat OpenShift Logging library is being deprecated in OpenShift Container Platform 4.14.

Configure EFK with Red Hat OpenShift Logging library:

1. Install the Red Hat OpenShift Logging library. For more information, see the Red Hat OpenShift documentation Installing the logging subsystem for Red Hat OpenShift.

   When you are deploying the Red Hat OpenShift Logging (ClusterLogging) instance, use the following YAML definition as the base for your own definition.

   Change the values for the **`storageClassName`**, **`maxAge`**, and **`size`** depending on your own system. For example, if you are installing on Red Hat OpenShift on IBM Cloud (ROKS), use **`ibmc-file-gold-gid`** for the **`storageClassName`**. If you want to keep application logs for more than 15 days, update the **`maxAge`** value.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    retentionPolicy:
      application:
        maxAge: 15d
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: "rook-cephfs"
        size: 80G
      resources:
        limits:
            memory: "16Gi"
        requests:
            memory: "16Gi"
      proxy:
        resources:
          limits:
            memory: 256Mi
          requests:
            memory: 256Mi
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana"
    kibana:
      replicas: 1
  collection:
    logs:
      type: "fluentd"
      fluentd: {}
```

2. Verify that all pods under the **`openshift-logging`** namespace are running by running the following command.

```
oc get pods -n openshift-logging
```

   If your pods are not running, it can take several minutes for all pods to be in the **`Running`** state. For instance, the **`collector`** pods can go through several restarts before they are all running. You can have several pods for ClusterLogging, Elasticsearch, fluentd (**`collector`**), and Kibana similar to the example output:

```
NAME                                            READY   STATUS      RESTARTS   AGE
cluster-logging-operator-84854b544c-tc758       1/1     Running     0          26m
collector-5hndx                                 2/2     Running     0          10m
collector-6sfk9                                 2/2     Running     0          10m
collector-f5w84                                 2/2     Running     0          10m
collector-fgzsr                                 2/2     Running     0          10m
collector-fhsbk                                 2/2     Running     0          10m
collector-fwdhk                                 2/2     Running     0          10m
collector-hj2q9                                 2/2     Running     0          10m
collector-l4v5q                                 2/2     Running     0          10m
elasticsearch-cdm-medm3kgq-1-74fb867496-swgwx   2/2     Running     0          22m
elasticsearch-cdm-medm3kgq-2-6cb7495ff5-h5bgd   2/2     Running     0          22m
elasticsearch-cdm-medm3kgq-3-5fb8f9f4-bgqgp     2/2     Running     0          22m
elasticsearch-im-app-27288900-b8j28             0/1     Completed   0          2m3s
elasticsearch-im-audit-27288900-btw2x           0/1     Completed   0          2m3s
elasticsearch-im-infra-27288900-ndjcb           0/1     Completed   0          2m3s
kibana-5586499d9b-45x7c                         2/2     Running     0          22m
```

3. **Optional:** If you intend to limit the scope to a particular namespace in your cluster, you can complete the following steps:

    1. Update the ClusterLogging instance `managementState` field from `Managed` to `Unmanaged`. This change can give you more control of the components that are managed by the Red Hat OpenShift Logging Operator. To change the setting, run the following command to edit the instance:

        ```
        oc edit clusterlogging instance -n openshift-logging
        ```

    2. Update the `path` variable in the `collector` configmap under the container logs section.

        You can use the following command to update the configmap.

        ```
        oc edit configmap collector -n openshift-logging
        ```

        As an alternative, you can also use the Red Hat OpenShift Console. To use the console, go to **Workloads** > **ConfigMaps**. Then, from the **Projects** drop-down, select *openshift-logging*. Select the `collector` configmap and edit the YAML. By default, this variable has the following value in the `collector` configmap:

        ```
        path "/var/log/pods/*/*/*.log"
        ```

        Change the value to reference your preferred namespace, for example:

        ```
        path "/var/log/pods/qotd_qotd-*/*/*.log"
        ```

    3. If you updated the collector `ConfigMapRestart` then restart all collector pods (daemonset):

        ```
        oc get pods -n openshift-logging --no-headers=true | awk '/collector/{print $1}' | xargs  oc delete -n
        openshift-logging pod
        ```

4. Expose the log store service as a route. For more information, see the Red Hat OpenShift documentation Exposing the log store service as a route.

    When you are completing this procedure, obtain the token for the `cluster-logging-operator` service account instead of your own Red Hat OpenShift Container Platform token. You must be logged in to the cluster where you installed the `ClusterLogging` instance.

    ```
    token=$(oc sa get-token cluster-logging-operator -n openshift-logging)
    echo $token
    ```

5. When the token variable is set, run the following commands on your workstation to validate that you can connect to Elasticsearch:

    ```
    routeES=`oc get route elasticsearch -o jsonpath={.spec.host} -n openshift-logging`
    curl -tlsv1.2 --insecure -H "Authorization: Bearer ${token}" "https://${routeES}"
    curl -tlsv1.2 --insecure -H "Authorization: Bearer ${token}" "https://${routeES}/_search?scroll=1m"
    ```

    If needed, you can run the following command to get the Elasticsearch version:

    ```
    curl --insecure -H "Authorization: Bearer ${token}" https://${routeES}/
    ```

6. Index the Elasticsearch documents. This indexing is done automatically, but the process can take a few minutes in a new or updated cluster. Wait a few minutes for the changes to display in the Kibana UI.

    When all configuration steps take effect, you can see entries in the Kibana UI for only the application that you intend to collect logs from.

Creating EFK integrations

With EFK set up, you can now create the integration in IBM Cloud Pak for AIOps.

For this integration you need to select the tile for creating an Elasticsearch, Logstash, and Kibana (ELK) integration, but change some settings to connect to EFK instead of ELK.

To create the integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **ELK** tile.

    Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.

5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add integration* page, define the general integration details:
    - **Name**: The display name of your integration.
    - **Description**: An optional description for the integration.
    - **ELK service URL**: ELK service URL: The Elasticsearch host and public API port. The URL must have the target index that IBM Cloud Pak for AIOps uses to search for the data from your applications. If you want to use all data, specify ***** as your index. For example, your ELK service URL can be **`https://myURL.com:8080/*`**.

      If you are using the Red Hat OpenShift Logging library, then run the following command to find the hostname for the URL:

      ```
      oc get routes -n openshift-logging | grep elastic
      ```

      Example output:

      ```
      elasticsearch    elasticsearch-openshift-logging.apps.demo-apps.cp.mysite.com    elasticsearch    <all>
      reencrypt    None
      ```

      When you have your Elastic hostname, you can build your URL. For example, **`https://elasticsearch-openshift-logging.apps.demo-apps.cp.mysite.com/app*`**. Ensure that you also replace **`/app*`** in the URL with the value for your EFK pattern.
    - **Kibana URL**: Enter a URL for the service instance.

      If you are using the Red Hat OpenShift Logging library, then run the following command to find the hostname for the URL:

      ```
      oc get routes -n openshift-logging | grep kibana
      ```
    - **Authentication type**: Set this value to *Token* to indicate that the Elasticsearch instance is authenticated with a temporary token.

      If you are using the Red Hat OpenShift Logging library, then run the following command to obtain the token for the **`cluster-logging-operator`** service account:

      ```
      token=$(oc sa get-token elasticsearch -n openshift-logging)
      echo $token
      ```
    - **Certificate (optional)**: Certificate used to verify the SSL/TLS connection to the REST service.
    - **Filters (optional)**: A custom Boolean Query to filter the Elasticsearch request for your specific application, terms, keywords, or other filters.
    - **Time zone (optional)**: The time zone in which your data is situated. The default time is converted from the system time relative to UTC time. The default value is UTC.
    - **Kibana port:* The port of the Kibana instance that is on the same host as the Elasticsearch instance. Use **`443`** for this field.
    - **Base parallelism**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots. In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this integration, you can set this value to be equal to the **source** parallelism.
    - **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is **`60`**.
    - **JSON processing option**: Select a JSON processing option.
        - **None**: The default option. The JSON is not processed or modified.
        - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
        - **Filter**: This option extracts the JSON object and replaces it with an empty string.
        - For more information about the options, see Managing embedded JSON.

    Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see Improving data streaming performance for log anomaly detection.

    Note: If you use the Filter option, do not use a timestamp in the filter query. This setting can cause a parsing error in the backend.

    ```
        "range": {
          "@timestamp": {
            "gte": "now-2m",
            "lt" : "now"
          }
    ```

    Other than that limitation, filter can use any clauses so long as the fields and the values specified in the filter are relevant to the target endpoint data set.
7. You can test your integration by clicking **Test connection**.
8. Click **Next**.
9. Enter *Field Mapping* information (Optional):

    You can improve search performance by mapping the fields from your implementation fields to IBM Cloud Pak for AIOps's standard fields.

    Use the following mapping instead of the default mapping that is provided for an ELK integration:

    ```
    {
      "codec": "elk",
      "message_field": "message",
      "log_entity_types": "kubernetes.container_image_id, kubernetes.host, kubernetes.pod_name,
    ```

```
kubernetes.namespace_name",
  "instance_id_field": "kubernetes.container_name",
  "rolling_time": 10,
  "timestamp_field": "@timestamp"
}
```

10. Click **Next**.
11. Enter AI training and log data (Optional):

    Select how you want to manage collecting data for use in AI training and anomaly detection. Click the **Data collection** toggle to turn on data collection, then select how you want to collect data:

    - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your integration is used to both train AI models and analyze your data for anomalous behavior.

      Note: After an initial installation, there is no data at all in the system. If you select this option, then the three different log anomaly detection algorithms behave in the following ways:

      - Natural language log anomaly detection does not initially detect anomalies as no model has been trained. You can retrieve historical data (select **Historical data for initial AI training**) to speed up the retrieval of data to train on, or you can leave the **Live data for continuous AI training and anomaly detection** setting on. In the latter case, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.
      - Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.
      - Log Anomaly Golden Signals does not initially detect anomalies as no model has been trained. The system gathers training data live, and after a few days, there is enough data to train a model. When this model is deployed, it can detect anomalies that deviate from the normal.

    - **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your integration.

      Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.

    - **Historical data for initial AI training**: A single set of training data used to define your AI model. You need to give Start and End dates, and specify the parallelism of your source data. Historical data is harvested from existing logs in your integration over a specified time period in the past.

      - **Start date**: Select a start date from the calendar and enter the time in hh:mm (hours and minutes) format.

        Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.

      - **Time zone**: Select your time zone from the dropdown list.
      - **End date and time**: Click **Add end date** and select an end date from the calendar and enter the time in hh:mm format.

        Note: If you do not specify the end date, then live data collection follows the historical data collection. If you do not want to set an end date, click **Remove end date**.

      - **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

    Important: Keep in mind the following considerations when you select your data collection type:

    - Anomaly detection for your integration occurs if you select *Live data for continuous AI training and anomaly detection*.
    - Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

12. Click **Done**.

You created the integration in your instance. After you create your integration, you must enable the data collection to connect your integration with the AI of IBM Cloud Pak for AIOps.

Enabling and disabling EFK integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. If you selected *Live data for initial AI training* when you created your integration, you must disable the integration before AI model training. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **ELK** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting an integration, see Deleting EFK integrations.

Editing EFK integrations

After you create your integration, your can edit the integration. For example, if you specified *Historical data for initial AI training* but now want your integration to pull in live data for continuous monitoring, you can edit it. To edit an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **ELK** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the ELK integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited. If your application was not previously enabled or disabled, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling EFK integrations. For more information about deleting an integration, see Deleting EfK integrations.

Deleting EFK integrations

If you no longer need your EFK integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable data collection before you delete your integration. For more information about disabling data collection, see Enabling and disabling EFK integrations.

To delete an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **ELK** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the ELK integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating and managing Falcon LogScale integrations

A Falcon LogScale (previously named Humio) integration provides log data, which is used to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts and events, and published to your ChatOps interface, to help you determine the cause and resolution of a problem.

For more information about working with Falcon LogScale integrations, see the following sections:

- Creating Falcon LogScale integrations
- Enabling Falcon LogScale integrations
- Editing Falcon LogScale integrations
- Deleting Falcon LogScale integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating Falcon LogScale integrations

About this task

Before creating the integration, you should be aware of the following information.

- **Load**: To prevent this integration placing an inordinate load on your data source and potentially impacting your logging operations, this integration only connects to one API with a default data frequency of 60 seconds. This is controlled by using the **Sampling rate** setting in the Procedure section.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting in the Procedure section.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Procedure

To create a Falcon Logscale integration from a specific source, step through the following sections:

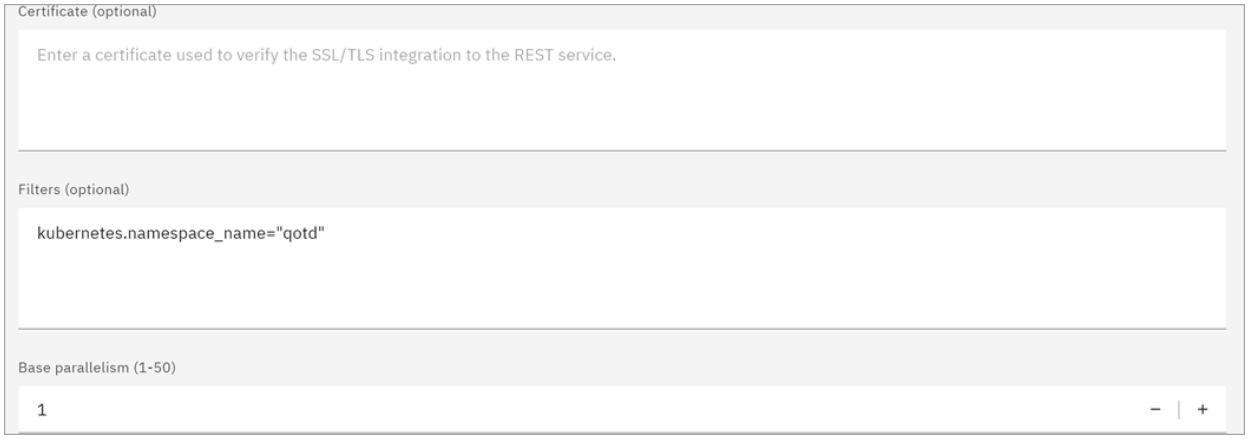
- Adding a Falcon Logscale integration
- Specifying integration parameters

- Specifying field mapping
- Specifying how log data is collected for AI training

Adding a Falcon Logscale integration

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Falcon LogScale** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: The display name of your integration.
   - **Description**: An optional description for the integration.
   - **Falcon LogScale service URL**: The fully qualified Falcon LogScale query API URL for your Falcon LogScale instance, for example: **`http://<Falcon LogScale_HOSTNAME>/api/v1/repositories/<REPO_NAME>/query`**
   - **API Token**: The integration token for your Falcon LogScale instance, which is available from the account settings page in Falcon LogScale.
   - **Certificate**: An optional certificate used to verify the SSL/TLS connection to the REST service.
   - **Filters**: Optional filters that define subsets of data that is pulled from Falcon LogScale.
   - **Base parallelism**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots. In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this integration, you can set this value to be equal to the **source** parallelism.

     caption=
   - **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is **`60`**.
   - **JSON processing option**: Select a JSON processing option.
     - **None**: The default option. The JSON is not processed or modified.
     - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
     - **Filter**: This option extracts the JSON object and replaces it with an empty string.
     - For more information about the options, see Managing embedded JSON.

   Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployment sizes, see Improving data streaming performance for log anomaly detection.
7. Test your integration by clicking **Test connection**.

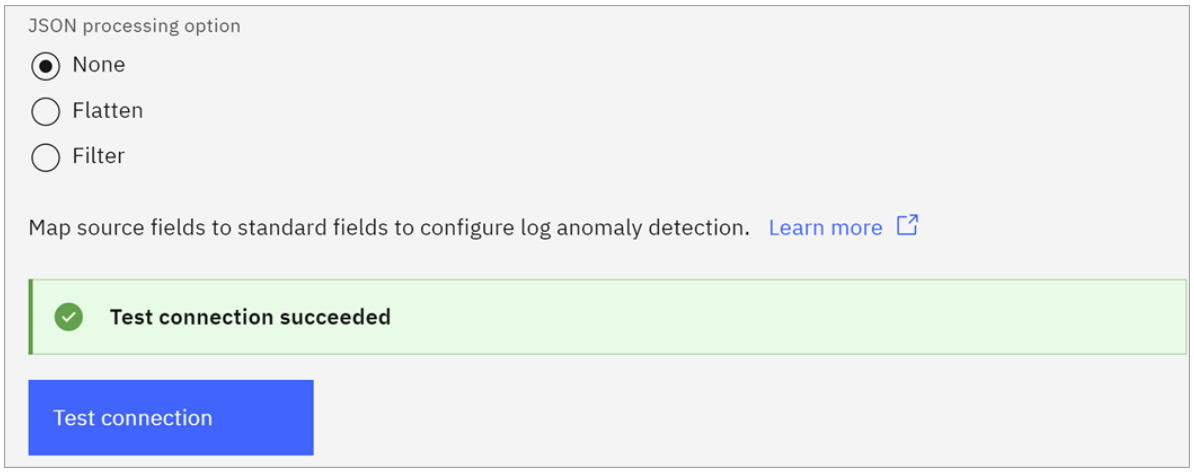


Figure. Test connection

8. Click **Next** to move to the next page.

Specifying field mapping

1. Enter **Field Mapping** information (Optional):

   The default mapping setting that is provided by UI can improve the search performance by mapping the fields from your implementation fields to IBM Cloud Pak for AIOps's standard fields. For more information about how field mappings are defined, see Mapping data from incoming sources. For more information about the use of mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data that use regular expressions. Consider the supported data schema when you create your field mapping:

   ```
   {
     "codec": "humio",
     "message_field": "@rawstring",
     "log_entity_types": "kubernetes.container_name",
     "instance_id_field": "kubernetes.container_name",
     "rolling_time": 10,
     "timestamp_field": "@timestamp",
     "resource_id": "kubernetes.pod_name"
   }
   ```

   But processing of logs with this format into log anomaly alerts can lead to false alarms in the Alert Viewer.

2. To resolve the false alarms issue in step 1, use regular expression to exclude parts of the log lines at the end of fields mapping (Optional):

   Logs ingested by Falcon LogScale usually have the following format:

   - `_timestamp_ + stdout F + _real message_`
   - `_timestamp_ + stderr F + _real message_`

   You can define the field mapping as follows:

   ```
   {
   "codec": "humio",
   "message_field": "@rawstring",
   "log_entity_types": "kubernetes.container_name",
   "instance_id_field": "kubernetes.container_name",
   "rolling_time": 10,
   "timestamp_field": "@timestamp",
      "custom_regex": [
      ".+?(stdout F )",
      ".+?(stderr F )"
      ]
   }
   ```

3. Click **Next** to move to the next page.

Specifying how log data is collected for AI training

1. Enter *AI training and log data* (Optional):

   Select how you want to manage collecting data for use in AI training and anomaly detection. Set the **Data collection** toggle to 'on', then select how you want to collect data:

   - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your integration is used to both train AI models and analyze your data for anomalous behavior.

     Note: After an initial installation, there is no data at all in the system. If you select this option, then the two different log anomaly detection algorithms behave in the following ways:

     - Natural language log anomaly detection does not initially detect anomalies as no model has been trained. You can retrieve historical data (select **Historical data for initial AI training**) to speed up the retrieval of data to train on, or you can leave the **Live data for continuous AI**

training and anomaly detection setting on. In the latter case, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.

- Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.

- **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your integration.

  Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.

- **Historical data for initial AI training**: A single set of training data used to define your AI model. You need to give Start and End dates, and specify the parallelism of your source data. Historical data is harvested from existing logs in your integration over a specified time period in the past.

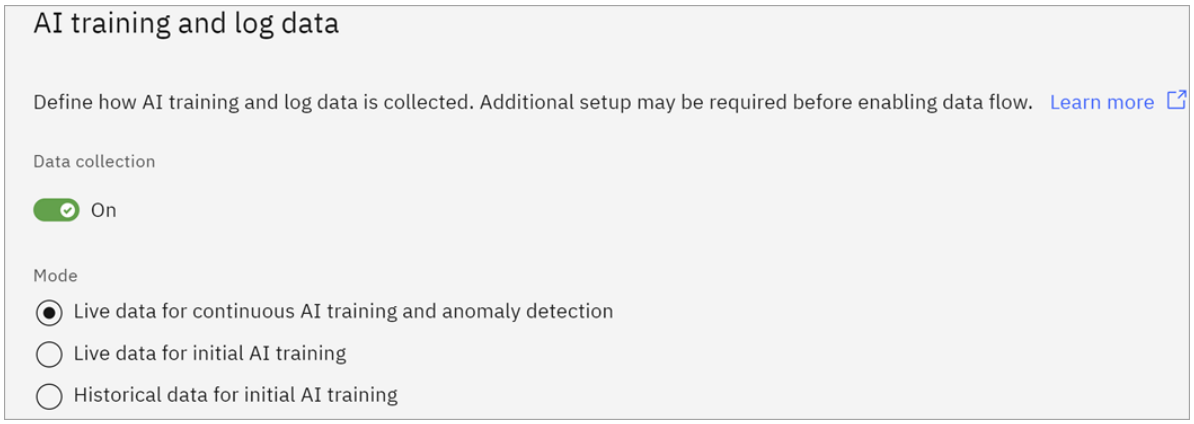


Figure. AI training

- **Start date**: Select a start date from the calendar.

  Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.

- **End date**: Select an end date from the calendar.

  Note: If you do not specify the end date, then live data collection follows the historical data collection.

- **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

  Note: To avoid issues with task manager memory, run no more than one *Historical data for initial AI training* run at a time.

Important: Keep in mind the following considerations when you select your data collection type:

- Anomaly detection for your integration occurs if you select *Live data for continuous AI training and anomaly detection*.
- Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

2. Click **Next**.
3. On the *Resource requirements* page, you can review the slot usage for your log integrations to see if there are enough slots to fully support the integration for multizone high availability.

   If you set the **Data collection** toggle to **On**, you will see the resource management overview.

   - If your current usage and other usage are less than the provisioned slots, but the HA slots exceed the provisioned slots, you will be able to create the integration, but will see a warning that you do not have enough slots. The integration will not have multizone high availability.
   - If your projected usage exceeds the provisioned slots, you will not be able to create the integration because you do not have enough slots on your system for log data integrations.
   - If your total slots, including HA slots, are within the provisioned slots, the integration will have multizone high availability.

     Note: HA operation assumes high availability for three zones.

   If you set the **Data collection** toggle to **Off**, you will see a message stating that you need to enable logs data collection to see the resource management overview. When data collection is off, no slots are used by that integration.

4. Click **Done**.

You have created a Falcon LogScale integration in your instance. After you create your integration, you must enable the data collection to connect your integration with the AI of IBM Cloud Pak for AIOps. For more information about enabling your integration, see Enabling Falcon LogScale integrations.

To create more integrations (such as a ChatOps integration), see Configuring Integrations.

For more information about working with the insights provided by your integrations, see ChatOps insight management.

Enabling and disabling Falcon LogScale integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. If you selected *Live data for initial AI training* when you created your integration, you must disable the integration before AI model training. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **Falcon LogScale** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data integration** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

You have enabled or disabled your integration. For more information about deleting a integration, see Deleting Falcon LogScale integrations.

Editing Falcon LogScale integrations

After you create your integration, your can edit the integration. For example, if you specified *Historical data for initial AI training* but now want your integration to pull in live data for continuous monitoring, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Falcon LogScale** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Falcon LogScale integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited. If your application was not previously enabled or disabled, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling Falcon LogScale integrations. For more information about deleting a integration, see Deleting Falcon LogScale integrations.

Once edited, slot usage of log data integrations can be reviewed.

```
![Falcon Logscale integration](../images/int_falconlogscale_editusage.png){: caption="Figure.  Resource requirements" caption-side="bottom"}
```

Deleting Falcon LogScale integrations

If you no longer need your Falcon LogScale integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable data collection before deleting your integration. For more information about disabling data collection, see Enabling and disabling Falcon LogScale integrations.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Falcon LogScale** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Falcon LogScale integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating Generic Webhook integrations

If you want to connect to a data source and a dedicated integration is not available for that source in the IBM Cloud Pak for AIOps Integrations UI, you can use the generic webhook integration. This integration provides an endpoint for applications to send JSON formatted events to IBM Cloud Pak for AIOps through HTTP POST requests.

In the requests you can also include JSONata that maps incoming JSON payloads to IBM Cloud Pak for AIOps events.

For more information about working with a generic webhook integrations, see the following sections:

- Before you begin
- Creating a Generic Webhook integration
- Editing a Generic Webhook integration
- Deleting a Generic Webhook integration

Before you begin

To configure a generic webhook integration, you need the following information for the application that you are connecting to IBM Cloud Pak for AIOps:

- A name and description for the webhook integration.
- If you need to secure the webhook, the appropriate user name and password.
- The JSONata that maps the incoming JSON payload to IBM Cloud Pak for AIOps events.

Creating a Generic Webhook integration

To create a Generic Webhook integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Generic Webhook** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section (Events).
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: The display name of your integration. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
   - **Description**: An optional description for the integration.
   - **Authentication type**: Select the **Username/Password** or **None** option.
   - **Username**: Enter the appropriate username to authenticate the connection.
   - **Password**: Enter the appropriate password to authenticate the connection.
   - **Webhook route**: The predefined full URL (location) of the webhook. This field is not editable.
   - **Maximum event rate**: The maximum number of events (per second) that are allowed by the Generic Webhook integration. You must specify a value between 10 and 1000 events per second. The default is 100.
     Note: The actual rate that events are collected depends on network performance.
7. Click **Next**.
8. Configure the event mapping.

   Use JSONATA to map the incoming JSON payload to the IBM Cloud Pak for AIOps event data schema. You can start from scratch or load an example mapping. The mapping configuration will be tested to ensure that it is in a valid JSONata format.

   Click **Load sample mapping** to show a selection of available samples and select an appropriate use case to load. For example, you can select **SevOne** or **Turbonomics**. Finish the mapping by clicking the **Load sample mapping** button.

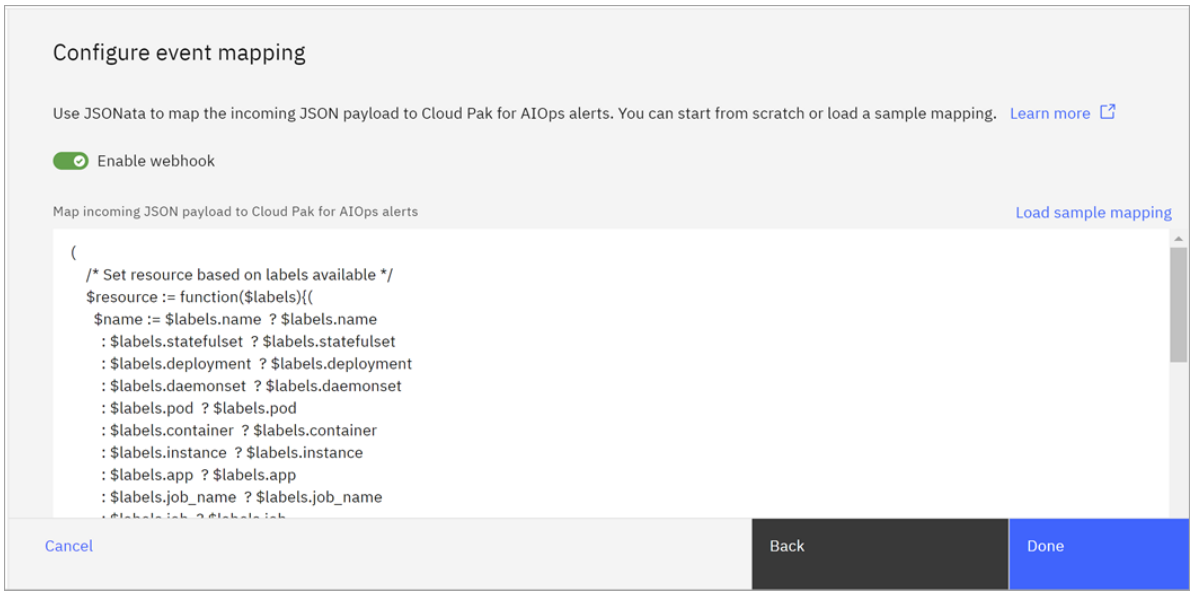


Figure. Configure event mapping

The following mapping is an example mapping for an integration with IBM SevOne Network Management System (NMS)

```
(
    {
        "severity": alertState="Alert"?6:alertState="Emergency"?6:alertState="Critical"?6:alertState="Error"?
5:alertState="Warning"?3:alertState="Notice"?2:alertState="Info"?2:alertState="Debug"?2:1,
        "summary": description,
        "resource": {
            "name": deviceName
        },
        "type": {
            "classification": policyName,
            "eventType":  alertState = "Cleared" ? "resolution" : "problem"
        },
        "sender": {
            "name": "IBM SevOne Network Management System",
            "type": "Webhook Integration"
        },
        "expirySeconds": alertId="-1"?300
    }
)
```

Note: The JSONata expression must contain the `summary`, `resource` and `type` fields. If the fields are missing, the event is invalid and might be rejected. Values mapped to the `classification` field in the data model are especially significant. Inappropriate values assigned to this field at the time of mapping has a negative impact on several downstream activities, such as Golden Signal classification, incident creation, and recommended policies. Classification should contain information summarizing the event type, for example, CPU Utilization, Node Down, or the actual metric name itself.

Review the following topics for more example mappings to help you configure your own mappings:

- Example: IBM SevOne Network Management System (NMS) generic webhook event mappings
- Example: Prometheus AlertManager (Red Hat OpenShift) generic webhook event mappings

9. Enable the webhook. Click the toggle to change the setting **Enable webhook** or **Disable webhook**. When disabled, the webhook integration discards incoming HTTP requests.
10. Click **Done**.
11. Within the application that you are connecting to IBM Cloud Pak for AIOps, complete any required configurations to support the integration.

    Important: You might need to configure a signed trusted certificate for IBM Cloud Pak for AIOps that the external system that you are connecting to trusts. When an external system is sending a payload through this webhook integration, the payload goes through a TLS handshake. If IBM Cloud Pak for AIOps is using the default configuration, IBM Cloud Pak for AIOps uses self-signed certificates that an external system will not accept if the system does not trust self-signed certificates. To configure a signed trusted certificate, If you are connecting to a SaaS system, you might need to use an internationally recognised certificate authority certificate as you might need be able to update the SaaS trusted certificate authority list. Otherwise, you can use your own certificate authority and add the certificate to the external system.

    For example, to connect to IBM SevOne Network Management System (NMS) you need to complete the following tasks:

    - Copy the Webhook URL from IBM Cloud Pak for AIOps.
    - Importing a CA signer certificate into IBM SevOne Network Management System (NMS).
    - Add the webhook destination within IBM SevOne Network Management System (NMS).

      In the IBM SevOne Network Management System (NMS) UI go to **Events** > **Configuration** > **Policy Browser** > **Configure webhook destination** > **Add webhook destination**.
    - Add the webhook into existing policies to send event data to IBM Cloud Pak for AIOps.

12. Test your integration with your connected application to verify that events are sent to IBM Cloud Pak for AIOps, and display as alerts in the Alert Viewer of the IBM Cloud Pak for AIOps UI.

Enabling or disabling a Generic Webhook integration

After you create your webhook integration, you can enable or disable the webhook by completing the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Generic Webhook** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *Generic Webhook* page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Enable or disable the webhook by clicking the toggle to change the setting to **Enable webhook** or **Disable webhook**. When disabled, the webhook integration discards incoming HTTP requests.
6. Click **Save** when you are done.

Your webhook is now enabled or disabled.

Editing a Generic Webhook integration

After you create your integration, you can edit the integration. For example, if you want to add or change the listed GUI or backend URLs for the IBM Tivoli Netcool/Impact endpoint, or to update the User ID or password. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Generic Webhook** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *Generic Webhook* page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited.

Deleting a Generic Webhook integration

If you no longer need your integration and want to delete it entirely, you can delete the integration from the console.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Generic Webhook** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the *Generic Webhook* integrations page, click the *Delete* icon for the integration that you want to delete.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Integrating with IBM SevOne Network Performance Management

You can use a generic webhook integration to integrate IBM Cloud Pak for AIOps with IBM SevOne Network Performance Management.

For details see Creating Generic Webhook integrations.

Configuring IBM SevOne Network Management System (NMS) to forward events to the webhook integration for SevOne

IBM SevOne NMS must be configured with a new webhook configuration to send alerts to the webhook integration for SevOne NMS.

Note: Instructions to configure the webhook with SevOne NMS can differ depending on your SevOne NMS version. For more information about using the SevOne Webhook Definition Manager and configuring alerts in SevOne NMS, see NMS 6.6 Guides and the SevOne NMS User Guide.

Note IBM SevOne requires the webhook destination to use a TLS certificate signed by a trusted Certificate Authority (CA).

1. Log in to the IBM SevOne console.

2. On the top menu, go to **Events > Configuration > Webhook Definition Manager**.
3. On the Policy Browser page, click **Configure Webhook Destination** on the Policies panel.
4. Complete the following steps from the Add Webhook Destination panel:
    1. Set a Webhook Destination Name, for example: IBM Cloud Pak for AIOps Webhook
    2. Set a Description, for example: Webhook integration for SevOne NMS
    3. You can obtain the webhook URL from the Integrations UI page.
    4. Select the **Allow insecure webhook connection** option if the webhook integration uses a self-signed certificate.
    5. Expand the Headers section and select **Use Basic Auth**. Then, set the **Username** and **Password** fields with the credentials set in **Step 6)** of Creating a Generic Webhook integration.
    6. In the **Request Options**, select **POST** as the Request Method, **application/json** as the Content Type, and use the following sample body:

```
  {
      "routes": ["IBM CloudPak for AIOps"],
      "host": "$deviceName",
      "description": "$alertMessage",
      "alertMessage": "$alertMessage",
      "check": "$policyName - $objectName",
      "cluster": "$clusterName",
      "alertId": "$alertId",
      "alertType": "$alertType",
      "alertState": "$alertState",
      "occurrences": "$occurrences",
      "assignedTo": "$assignedTo",
      "deviceId": "$deviceId",
      "deviceIp": "$deviceIp",
      "deviceName": "$deviceName",
      "deviceAltName": "$deviceAltName",
      "groupName": "$groupName",
      "objectId": "$objectId",
      "objectName": "$objectName",
      "objectAltName": "$objectAltName",
      "objectDescription": "$objectDescription",
      "pluginName": "$pluginName",
      "pluginDescription": "$pluginDescription",
      "policyId": "$policyId",
      "policyName": "$policyName",
      "thresholdId": "$thresholdId",
      "thresholdName": "$thresholdName",
      "linkURL" : "<DI_URL.com/redirect/v1/reports?reportName=Indicator%20Summary&$DIDataResources|View in Data
Insight>",
      "triggeringConditions": $triggeringConditions,
      "closureMessage": "$closureMessage"
  }
```

    7. Click **Test Definition** to send a test HTTP event to the webhook integration. Test both the triggering condition and clear condition webhooks to ensure that the recieved response code is `200` or `204`.
    - A `204` status code indicates that no payload is included in the HTTP response, but the integration is successful.
    - If a response code of `400 Bad request` is received, check the JSON template that is configured. Ensure that the JSON does not contain any syntax errors.
    8. Click **Save**.
5. Configure or create policies to send events to the webhook integration by selecting the webhook integration as the webhook destination. For each of the policies in the **Webhooks** section, select the webhook that was created in the previous step. Complete the selection for the **Trigger Conditions** and **Clear Conditions** tabs. By default, SevOne alerts configuration enables **One Webhook per Alert** option which only sends one webhook on the first trigger condition and not subsequent triggers. You can opt to select **Override Webhook per alert** and deselect **One Webhook per Alert** option to trigger webhook on each alert trigger to send alert updates to your webhook.

Integrating with IBM Turbonomic

You can use a generic webhook integration to integrate IBM Cloud Pak for AIOps with IBM Turbonomic.

For details see Creating Generic Webhook integrations.

Configuring IBM Turbonomic to forward events to the webhook integration for Turbonomic

IBM Turbonomic must be configured with a new webhook configuration to send alerts to the webhook integration for Turbonomic.

Creating a workflow

1. Set the following variables:

```
TURBONOMIC_HOSTNAME=<api endpoint hostname>
TURBONOMIC_ADMIN_USER=<turbonomic username>
TURBONOMIC_ADMIN_PASSWORD=<turbonomic password>
AIOPS_WEBHOOK_URL=<webhook integration URL>
BASIC_AUTH_USERNAME=<webhook basic authentication username>
BASIC_AUTH_PASSWORD=<webhook basic authentication password>
```

Note: Update the values set for **AIOPS_WEBHOOK_URL**, **BASIC_AUTH_USERNAME** and **BASIC_AUTH_PASSWORD**. You can obtain these values from the AIOps Webhook integration configuration.

2. Get the JSESSIONID token

```
JSESSIONID=$(curl \
--silent \
--cookie-jar - \
--insecure \
https://${TURBONOMIC_HOSTNAME}/api/v3/login?hateoas=true \
--data "username=${TURBONOMIC_ADMIN_USER}&password=${TURBONOMIC_ADMIN_PASSWORD}" \
| awk '/JSESSIONID/{print $7}')
```

3. Create a workflow

```
curl \
"https://${TURBONOMIC_HOSTNAME}/api/v3/workflows" \
--insecure \
--compressed \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "cookie: JSESSIONID=${JSESSIONID}" \
--request POST \
-vv \
--data @- <<EOF
{
"displayName": "Event Webhook integration",
"className": "Workflow,
"description": "Event webhook integration",
"discoveredBy": {
    "readonly": false
},
"type": "WEBHOOK",
"typeSpecificDetails": {
  "url": "${AIOPS_WEBHOOK_URL}",
  "method": "POST",
  "authenticationMethod": "BASIC",
  "username": "${BASIC_AUTH_USERNAME}",
  "password": "${BASIC_AUTH_PASSWORD}",
  "template": "\$converter.toJson(\$action)",
  "trustSelfSignedCertificates": true,
  "headers": [
     {
      "name": "Content-type",
      "value": "application/json"
     }
   ],
  "type": "WebhookApiDTO"
  }
}
EOF
```

4. Get the workflow UID for later use.

```
WORKFLOW_UUID=$(curl \
"https://${TURBONOMIC_HOSTNAME}/api/v3/workflows" \
--insecure \
--compressed \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "cookie: JSESSIONID=${JSESSIONID}" \
--request GET | jq -c ".[] | select(.typeSpecificDetails .
url | contains(\"${AIOPS_WEBHOOK_URL}\")) | .uuid" | tr -d '"')
```

Create a policy

1. Log in to the Turbonomic console.
2. Navigate to the *Settings* page and then choose **Policies**.
3. Click **New Automation Policy** and then select the policy type, **Virtual Machine** in this case. This sets the type of entity that your policy will affect. Refer to Working with policies for more information.

   Name the policy

   - **Add Scope** The scope determines which entities this policy will affect. Choose one or more groups, or create new groups and add them to the policy scope. These groups match the type of entity you have set for the policy.
   - **Policy Schedule** The *Select Schedule* fly-out lists all the schedules that are currently defined for your instance of Turbonomic. Expand a schedule entry to see further details, or skip it if you don't need the schedule.
   - **Automation and Orchestration** You can define automation and orchestration settings for different action types within the same policy.
     - **Action Type** See a list of actions that are viable for the policy, and then make your selections.
     - **Action Generation and Acceptance**
       - Do not generate actions as Turbonomic doesn't considers your selected actions in its calculations. For example, if you do not want to generate resize actions for VMs in the policy, analysis will still drive toward the desired state, but will do so without considering resizes.

- Generate Actions Turbonomic generates your selected actions to address or prevent problems. Choose from the following *Action Acceptance* modes to indicate how you would like the actions to run:
  - **Recommend** Recommend the action so a user can run it via the given hypervisor or by other means
  - **Manual** Recommend the action, and provide the option to run that action through the Turbonomic user interface
  - **Automatic** Run the action automatically For automated resize or move actions on the same entity, Turbonomic waits five minutes between each action to avoid failures associated with trying to run all actions at once. Any action awaiting execution stays in queue.
- **Before Execution and Action Execution:** By default, generated actions run without the need for orchestration. Turbonomic gives you the ability to set up orchestration to affect the execution of actions.
- **After Execution**: The default is to 'do nothing'. Other options include:
  - **Create Record in Orchestration** Turbonomic registers the action in the Orchestration log, showing that the given action has been run. For example: A final ActionID could be created and with the final ActionState, the Type and Severity would be updated accordingly in IBM Cloud Pak for AIOps. Those actions would also be updated to IBM Cloud Pak for AIOps, and it would handle with the alerts with updated state: closed or cleared or keep open.
  - **Use Action Script** Run an action script that is set up for the POST entrypoint. The action script name must match the entity type and action type.

4. Save the policy.
5. Action events will be sent to the webhook integration when an action is run.

Integrating with Zabbix

You can use the generic webhook integration to integrate Zabbix with IBM Cloud Pak for AIOps.

To do so, use the instructions in the following sections:

- Creating a Generic Webhook integration
- Creating a new webhook media type for CP4WAIOPs
- Assigning the CP4WAIOPs webhook media type into Admin user
- Creating a Zabbix Trigger Action to send notification to CP4WAIOPs

Creating a Generic Webhook integration for Zabbix

To create a Generic Webhook integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Generic Webhook** tile.
5. Click **Get started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: The display name of your integration. For example `zabbix-demo`
   - **Authentication type**: Select **None**.
7. Click **Next**.
8. Configure the event mapping.

   Use JSONATA to map the incoming JSON payload to the IBM Cloud Pak for AIOps event data schema.

   Copy the following Zabbix JSONATA expression into the text area:

```
(
   {
      "severity": severity="5"?6:severity="4"?5:severity="3"?4:severity="2"?3:severity="1"?2:1, "summary": message,
      "resource": {
         "name": deviceName,
         "hostname": hostname,
         "ipAddress": ipaddress,
         "sourceId": hostid,
         "port": hostport
   },
   "type": {
      "classification": triggerdesc,
      "eventType": recoveryValue = "0" ? "resolution" : "problem"
   },
   "sender": {
```

```
      "name": "Zabbix",
      "sourceId": $exists(actionName) ? $length(actionName) > 0 ? "Trigger action: " & actionName,
      "type": "Webhook Connector"
      }
   }
)
```

9. Click **Done**.
10. You should see the newly created **zabbix-demo** webhook integration. Copy the webhook route. You will need to paste this route into the Zabbix configuration later.

Creating a new webhook media type for CP4WAIOPs

Before creating a new webhook media type for CP4WAIOPs, you must a create a cp4waiops-webhook.yaml file that you will import into your Zabbix server.

To create the YAML file, you can use the example YAML that follows. This example supports both basic authentication and no authentication types as defined in the CP4WAIOPs Generic webhook connection.

If you want to use authentication, set **req_username** and **req_password** for basic authentication.

If you do not want to use authentication, remove the **req_username** and **req_password** variables from the YAML.

Sample CP4WAIOPS Connector Media type YAML

```
zabbix_export:
  version: '6.4'
  media_types:
    - name: 'CP4WAIOPS connector demo'
      type: WEBHOOK
      parameters:
        - name: actionName
          value: '{ACTION.NAME}'
        - name: deviceName
          value: '{HOST.NAME}'
        - name: hostid
          value: '{HOST.ID}'
        - name: hostname
          value: '{HOST.HOST}'
        - name: hostport
          value: '{HOST.PORT}'
        - name: HTTPProxy
        - name: ipaddress
          value: '{HOST.IP}'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: recoveryValue
          value: '{EVENT.RECOVERY.VALUE}'
        - name: req_password
          value: password
        - name: req_username
          value: admin
        - name: sendTo
          value: '{ALERT.SENDTO}'
        - name: severity
          value: '{EVENT.NSEVERITY}'
        - name: subject
          value: '{ALERT.SUBJECT}'
        - name: triggerdesc
          value: '{TRIGGER.DESCRIPTION}'
        - name: zabbixServer
          value: '{$ZABBIX.URL}'
      script: |
        try {

            var params = JSON.parse(value),
                req = new HttpRequest(),
                response;

            if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
                req.setProxy(params.HTTPProxy);
            }
            var data = {};
            data.message = params.message;
            data.severity = params.severity;
            data.deviceName = params.deviceName;
            data.hostname = params.hostname;
            data.hostid = params.hostid;
            data.hostport = params.hostport;
            data.ipaddress = params.ipaddress;
            data.subject = params.subject;
            data.recoveryValue = params.recoveryValue;
            data.triggerdesc = params.triggerdesc;
            data.actionName = params.actionName;
            data.sendTo = params.sendTo;
            data.zabbixServer = params.zabbixServer;
            data = JSON.stringify(data);

            req.addHeader('Content-Type: application/json');
```

```
            if (params.req_username !== '' && params.req_password !== '') {
               req.addHeader('Authorization: Basic ' + btoa(params.req_username + ':' + params.req_password));
            }

            Zabbix.log(4, '[ CP4WAIOPS webhook ] URL: ' + params.sendTo + ' data: ' + data);
            response = req.post(params.sendTo, data);
            Zabbix.log(4, '[ CP4WAIOPS webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response);

            if (response !== null) {
                try {
                    response = JSON.parse(response);
                    if (response.Status < 200 || response.Status >= 300) {
                        throw 'Request failed with status code ' + response.Status + ' error: ' + response.ErrorMessage;
                    }
                }
                catch (error) {
                    if (req.getStatus() < 200 || req.getStatus() >= 300) {
                        throw 'Request failed with status code ' + req.getStatus() + ' error: ' + response.ErrorMessage;
                    }
                    else {
                        throw 'Request success, but response parsing failed.';
                    }
                }
            }
            return 'OK';
        }
        catch (error) {
            Zabbix.log(3, '[ CP4WAIOPS webhook ] Sending failed. Error: ' + error);
            throw 'Failed with error: ' + error;
        }
      description: 'webhook media type for Cloud Pak Watson AIOPS - Webhook connector'
      message_templates:
        - event_source: TRIGGERS
          operation_mode: PROBLEM
          subject: 'Problem: {EVENT.NAME}'
          message: |
            Problem started at {EVENT.TIME} on {EVENT.DATE}
            Problem name: {EVENT.NAME}
            Host: {HOST.NAME}
            Severity: {EVENT.SEVERITY}
            Operational data: {EVENT.OPDATA}
            Original problem ID: {EVENT.ID}
            {TRIGGER.URL}
        - event_source: TRIGGERS
          operation_mode: RECOVERY
          subject: 'Resolved in {EVENT.DURATION}: {EVENT.NAME}'
          message: |
            Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE}
            Problem name: {EVENT.NAME}
            Problem duration: {EVENT.DURATION}
            Host: {HOST.NAME}
            Severity: {EVENT.SEVERITY}
            Original problem ID: {EVENT.ID}
            {TRIGGER.URL}
        - event_source: TRIGGERS
          operation_mode: UPDATE
          subject: 'Updated problem in {EVENT.AGE}: {EVENT.NAME}'
          message: |
            {USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME}.
            {EVENT.UPDATE.MESSAGE}

            Current problem status is {EVENT.STATUS}, age is {EVENT.AGE}, acknowledged: {EVENT.ACK.STATUS}.
```

To create a new webhook media type for CP4WAIOPs

1. Log in to your Zabbix server web GUI. Click **Alerts > Media types**.
2. Click **Import** in the top right corner. Click **Choose file** from the pop-out windows, select the cp4waiops-webhook.yaml file you created previously, make sure the **Create new** box is checked, and click **Import**.

Note: If you already had imported the cp4waiops webhook media type and want to update the existing one, check the **Update existing** box.

3. A new media should now be showing up in the media type list. This new media type has not yet been assigned to a trigger action, and so the **Used in action** field will be blank.
4. To test the new webhook media type, click Test in the Action column and fill in the required fields. In the **sendTo** field, paste in the details of the CPWAIOPs webhook connection route from Creating a Generic Webhook integration for Zabbix.
5. Click **Test**.
6. If the test is successful, you should receive the response `OK` and the message: `Media type test successful`. In CP4WAIOPs, you should receive the test POST message from Zabbix.

Assigning the CP4WAIOPs webhook media type into Admin user

After you have created and tested the new media type, you must assign the media type into the Admin user.

1. At the Zabbix web GUI side bar, click **Users > Users** to access the **Users** page. Click **Admin**.

2. Click the **Media** tab.
3. On the **Media** tab, click **Add**.
4. On the Media windows, select the integration that you created in Creating a Generic Webhook integration for Zabbix from the Type filed. Fill in the **Send to** field with the webhook connector route. Ensure the **Enabled** box is checked. Click **Add**.
5. You should see the new entry in media type now. Click **Update** to save it. You have now assigned the CP4WAIOPs media type to the Admin user.

Creating a Zabbix Trigger Action to send notification to CP4WAIOPs

After assigning the media type to the Admin user, you need to create a trigger action in Zabbix.

1. At the Zabbix web GUI side bar, click **Alerts > Actions > Trigger actions**. Click **Create action** in the top right corner. You need to create a trigger action that will send notifications to CP4WAIOPs using the integration that you created in Creating a Generic Webhook integration for Zabbix.
2. Complete the **Name** field, for example enter `Send CP4WAIOPs notification`.
3. Click **Add** under the **Conditions** section. A **New condition** window will pop-out. Define the condition where you want to send notification for this group of resource types. For example, select the **Host** as resource type.
4. Click **Select** from the **Hosts** section.
5. A **Hosts** window will pop-out. Click **Select** in the **Host Group** section. A **Host groups** window will pop-out to show you all available host groups that you can select from.
6. Select the host groups or hosts that you want to send notification to CP4WAIOPs. Click **Select**.
7. Click **Add** to add them into your trigger condition.
8. Now you should have set your trigger condition. Make sure check the **Enabled** box.
9. Click on the **Operations** tab. Click **Add** under the **Operations** section.
10. The **Operation details** window should pop-out. In the **Send to users** section, click **Select** and select **Admin**. In the **Send only** to section, click **Select** and select the integration that you created in Creating a Generic Webhook integration for Zabbix. Click **Add** to add the operation details.
11. Repeat Step 10. for the **Recovery operations** and **Update operations** sections. Click **Add**.
12. You should see an **Action Added** message. You have now created a trigger action with the name `Send CP4WAIOPs notification` that will trigger CP4WAIOPs notifications through the Admin user.

Example: IBM SevOne Network Management System (NMS) generic webhook event mappings

Review the following example mappings for a generic webhook integration between IBM Cloud Pak for AIOps and IBM SevOne Network Management System (NMS). Use these mappings to help you configure your own event mapping for a generic webhook integration.

- Example event (alert) JSON
- Example event mapping JSONata expression
- Example event result JSON
- Example Curl Command

Example event (alert) JSON


```
{
    "routes": [
        "IBM CloudPak for AIOps"
    ],
    "host": "hur-ka-cpb2a.net.hursley.ibm.com",
    "description": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's Out Discards: 147.98 Number
>= 120 Number totaled over 5.00 minutes",
    "alertMessage": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's Out Discards: 147.98 Number
>= 120 Number totaled over 5.00 minutes",
    "check": "AS-4.3-Nov - Ethernet1/59",
    "cluster": "Hursley-A",
    "alertId": "913284",
    "alertType": "system",
    "alertState": "Critical",
    "occurrences": "1",
    "assignedTo": "unassigned",
    "deviceId": "42",
    "deviceIp": "9.20.253.60",
    "deviceName": "hur-ka-cpb2a.net.hursley.ibm.com",
    "deviceAltName": "",
    "groupName": "Special",
    "objectId": "17325",
    "objectName": "Ethernet1/59",
    "objectAltName": "",
    "objectDescription": "fcxhurb14",
```

```
    "pluginName": "SNMP",
    "pluginDescription": "SNMP Poller",
    "policyId": "84",
    "policyName": "AS-4.3-Nov",
    "thresholdId": "39649",
    "thresholdName": "AS-4.3-Nov - hur-ka-cpb2a.net.hursley.ibm.com - Ethernet1/59",
    "linkURL": "<DI_URL.com/redirect/v1/reports?reportName=Indicator%20Summary&datasourceName=Hursley-
A&indicators=SNMP%26%26Interface%26%26hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F59%26%26ifOutDiscards&startTime=1701075605000|View in Data Insight>",
    "triggeringConditions": [
        {
            "aggregationDuration": "5.00",
            "aggregationOperation": "Total",
            "baselineValue": "0.00",
            "comparisonOperation": ">=",
            "comparisonUnits": "Number",
            "comparisonValue": "120",
            "dataUnits": "Number",
            "dataValue": "147.98",
            "indicatorDescription": "Out Discards",
            "indicatorName": "ifOutDiscards",
            "sigmaDirection": "above or below",
            "sigmaValue": "nan",
            "thresholdValue": "120.00"
        }
    ],
    "closureMessage": "N/A"
}
```

Example event mapping JSONata expression

```
(
    {
        "severity": alertState="Alert" ? 6 : alertState="Emergency" ? 6 : alertState="Critical" ? 6 : alertState="Error"
? 5 : alertState="Warning"? 3 : alertState="Notice" ? 2 : alertState="Info"? 2 : alertState="Debug" ? 2 : 1,
        "summary": description,
        "resource": {
            "name": deviceName
        },
        "type": {
            "classification": policyName,
            "eventType": ($exists(closureMessage) and closureMessage != "N/A") ? "resolution" : alertState = "Cleared" ?
"resolution" : "problem"
        },
        "sender": {
            "name": "IBM SevOne",
            "type": "Webhook Connector"
        },
        "expirySeconds": alertId="-1" ? 300
    }
)
```

Example event result JSON

```
{
    "summary": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's Out Discards: 147.98 Number >=
120 Number totaled over 5.00 minutes",
    "severity": 6,
    "sender": {
        "name": "IBM SevOne",
        "type": "Webhook Connector"
    },
    "resource": {
        "name": "hur-ka-cpb2a.net.hursley.ibm.com"
    },
    "occurrenceTime": 1701662721822,
    "type": {
        "condition": null,
        "eventType": "problem",
        "classification": "AS-4.3-Nov"
    }
}
```

Example Curl Command

1. Create a payload json file

```
cat << EOF | tee > sevone-payload.json | cat
  {
    "routes": [
        "IBM CloudPak for AIOps"
    ],
    "host": "hur-ka-cpb2a.net.hursley.ibm.com",
    "description": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's Out Discards: 147.98 Number
>= 120 Number totaled over 5.00 minutes",
    "alertMessage": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's Out Discards: 147.98 Number
>= 120 Number totaled over 5.00 minutes",
```

```
    "check": "AS-4.3-Nov - Ethernet1/59",
    "cluster": "Hursley-A",
    "alertId": "913284",
    "alertType": "system",
    "alertState": "Critical",
    "occurrences": "1",
    "assignedTo": "unassigned",
    "deviceId": "42",
    "deviceIp": "9.20.253.60",
    "deviceName": "hur-ka-cpb2a.net.hursley.ibm.com",
    "deviceAltName": "",
    "groupName": "Special",
    "objectId": "17325",
    "objectName": "Ethernet1/59",
    "objectAltName": "",
    "objectDescription": "fcxhurb14",
    "pluginName": "SNMP",
    "pluginDescription": "SNMP Poller",
    "policyId": "84",
    "policyName": "AS-4.3-Nov",
    "thresholdId": "39649",
    "thresholdName": "AS-4.3-Nov - hur-ka-cpb2a.net.hursley.ibm.com - Ethernet1/59",
    "linkURL": "<DI_URL.com/redirect/v1/reports?reportName=Indicator%20Summary&datasourceName=Hursley-
A&indicators=SNMP%26%26Interface%26%26hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F59%26%26ifOutDiscards&startTime=1701075605000|View in Data Insight>",
    "triggeringConditions": [
        {
            "aggregationDuration": "5.00",
            "aggregationOperation": "Total",
            "baselineValue": "0.00",
            "comparisonOperation": ">=",
            "comparisonUnits": "Number",
            "comparisonValue": "120",
            "dataUnits": "Number",
            "dataValue": "147.98",
            "indicatorDescription": "Out Discards",
            "indicatorName": "ifOutDiscards",
            "sigmaDirection": "above or below",
            "sigmaValue": "nan",
            "thresholdValue": "120.00"
        }
    ],
    "closureMessage": "N/A"
  }
EOF

cat << EOF | tee > sevone-payload-clear.json | cat
{
  "routes": ["IBM CloudPak for AIOps"],
  "host": "hur-ka-cpb2a.net.hursley.ibm.com",
  "description": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/29's Out Discards: 800.34 Number >=
100 Number totaled over 3.00 minutes",
  "alertMessage": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/29's Out Discards: 800.34 Number
>= 100 Number totaled over 3.00 minutes",
  "check": "AS-4.3-Nov - Ethernet1/29",
  "cluster": "Hursley-A",
  "alertId": "898689",
  "alertType": "system",
  "alertState": "Critical",
  "occurrences": "2",
  "assignedTo": "unassigned",
  "deviceId": "42",
  "deviceIp": "9.20.253.60",
  "deviceName": "hur-ka-cpb2a.net.hursley.ibm.com",
  "deviceAltName": "",
  "groupName": "Special",
  "objectId": "17296",
  "objectName": "Ethernet1/29",
  "objectAltName": "",
  "objectDescription": "fshurb1",
  "pluginName": "SNMP",
  "pluginDescription": "SNMP Poller",
  "policyId": "84",
  "policyName": "AS-4.3-Nov",
  "thresholdId": "39640",
  "thresholdName": "AS-4.3-Nov - hur-ka-cpb2a.net.hursley.ibm.com - Ethernet1/29",
  "linkURL" : "<DI_URL.com/redirect/v1/reports?reportName=Indicator%20Summary&datasourceName=Hursley-
A&indicators=SNMP%26%26Interface%26%26hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F29%26%26ifInErrors%7C%7CSNMP%26%26Interface%26%26hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F29%26%26ifOutErrors%7C%7CSNMP%26%26Interface%26%26hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F29%26%26ifInDiscards%7C%7CSNMP%26%26Interface%26%26hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F29%26%26ifOutDiscards&startTime=1700650083000|View in Data Insight>",
  "triggeringConditions": [] ,
  "closureMessage": "Alert cleared by clear condition"
}
EOF
```

2. Send the payload with a `curl` command

```
# Webhook details
WEBHOOK_URL=https://webhook-connector-1a2bc34d-1ab2-1abc-1a2b-1234a56bc789-aiops.apps.mysite.com/webhook-
connector/abcdefgh1jk
```

```
AUTH=test:test

# Send alert
curl -X POST -u ${AUTH} --insecure \
-H 'Content-Type: application/json' ${WEBHOOK_URL} \
-d @./sevone-payload.json


# Send clear event
curl -X POST -u ${AUTH} --insecure \
-H 'Content-Type: application/json' ${WEBHOOK_URL} \
-d @./sevone-payload-clear.json
```

Example: Prometheus AlertManager (Red Hat OpenShift) generic webhook event mappings

Review the following example mappings for a generic webhook integration between IBM Cloud Pak for AIOps and Prometheus AlertManager (Red Hat OpenShift). Use these mappings to help you configure your own event mapping for a generic webhook integration.

- Example event (alert) JSON
- Example event mapping JSONata expression
- Example event result JSON
- Example Curl Command

Example event (alert) JSON


```
{
  "receiver": "Default",
  "status": "firing",
  "alerts": [
    {
      "status": "firing",
      "labels": {
        "alertname": "PrometheusMissingRuleEvaluations",
        "container": "kube-rbac-proxy",
        "endpoint": "metrics",
        "instance": "10.254.40.166:9092",
        "job": "prometheus-k8s",
        "namespace": "openshift-monitoring",
        "openshift_io_alert_source": "platform",
        "pod": "prometheus-k8s-1",
        "prometheus": "openshift-monitoring/k8s",
        "rule_group": "/etc/prometheus/rules/prometheus-k8s-rulefiles-0/openshift-kube-apiserver-kube-apiserver-slos-
94dbc1dd-b771-4899-8c8f-54c0f5353e04.yaml;kube-apiserver.rules",
        "service": "prometheus-k8s",
        "severity": "warning"
      },
      "annotations": {
        "description": "Prometheus openshift-monitoring/prometheus-k8s-1 has missed 1 rule group evaluations in the last
5m.",
        "summary": "Prometheus is missing rule evaluations due to slow rule group evaluation."
      },
      "startsAt": "2023-06-01T17:02:53.691Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
g0.expr=increase%28prometheus_rule_group_iterations_missed_total%7Bjob%3D~%22prometheus-k8s%7Cprometheus-user-
workload%22%7D%5B5m%5D%29+%3E+0&g0.tab=1",
      "fingerprint": "24ad9e973e22bdce"
    },
    {
      "status": "firing",
      "labels": {
        "alertname": "AlertmanagerReceiversNotConfigured",
        "namespace": "openshift-monitoring",
        "openshift_io_alert_source": "platform",
        "prometheus": "openshift-monitoring/k8s",
        "severity": "warning"
      },
      "annotations": {
        "description": "Alerts are not configured to be sent to a notification system, meaning that you may not be
notified in a timely fashion when important failures occur. Check the OpenShift documentation to learn how to configure
notifications with Alertmanager.",
        "summary": "Receivers (notification integrations) are not configured on Alertmanager"
      },
      "startsAt": "2023-04-26T19:27:36.642Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
g0.expr=cluster%3Aalertmanager_integrations%3Amax+%3D%3D+0&g0.tab=1",
      "fingerprint": "72bc0ebbd3167d00"
    }
  ],
  "groupLabels": {
    "namespace": "openshift-monitoring"
  },
```

```
  "commonLabels": {
    "namespace": "openshift-monitoring",
    "openshift_io_alert_source": "platform",
    "prometheus": "openshift-monitoring/k8s",
    "severity": "warning"
  },
  "commonAnnotations": {},
  "externalURL": "https:///console-openshift-console.apps.example.ibm.com/monitoring",
  "version": "4",
  "groupKey": "{}:{namespace=\"openshift-monitoring\"}",
  "truncatedAlerts": 0
}
```

Example event mapping JSONata expression

```
(
    /* Set resource based on labels available */
    $resource := function($labels){(
      $name := $labels.name  ? $labels.name
        : $labels.statefulset  ? $labels.statefulset
        : $labels.deployment  ? $labels.deployment
        : $labels.daemonset  ? $labels.daemonset
        : $labels.pod  ? $labels.pod
        : $labels.container  ? $labels.container
        : $labels.instance  ? $labels.instance
        : $labels.app  ? $labels.app
        : $labels.job_name  ? $labels.job_name
        : $labels.job  ? $labels.job
        : $labels.type ? $labels.type: $labels.prometheus;
       $labels.namespace ? ($name & '/' & $labels.namespace): $name;
      )
    };
    /* Map to event schema */
    alerts.(
      {
        "summary": annotations.summary ? annotations.summary: annotations.description ? annotations.description :
annotations.message ? annotations.message,
        "severity": $lowercase(labels.severity) = "critical" ? 6 : $lowercase(labels.severity) = "major" ? 5 :
$lowercase(labels.severity) = "minor" ? 4 : $lowercase(labels.severity) = "warning" ? 3 : 1,
        "resource": {
          "name": $resource(labels)
        },
        "type": {
          "eventType": $lowercase(status) = "firing" ? "problem": "resolution",
          "classification": labels.alertname
        },
        "links": [
          {
              "url": generatorURL
          }
        ],
        "sender": {
          "name": "Prometheus",
          "type": "Webhook Connector"
        },
       "details": labels
      }
    )
  )
```

Example event result JSON

The following example shows the JSON result that creates an array of events. The generic webhook integration creates a CloudEvent for each element in the array.

```
[
    {
        "occurrenceTime": "2023-06-07T14:49:53Z",
        "summary": "Prometheus is missing rule evaluations due to slow rule group evaluation.",
        "severity": 3,
        "type": {
            "eventType": "problem",
            "classification": "PrometheusMissingRuleEvaluations",
            "condition": null
        },
        "sender": {
            "name": "Prometheus",
            "type": "Webhook Connector"
        },
        "resource": {
            "name": "prometheus-k8s-1/openshift-monitoring"
        },
        "links": [
            {
                "linkType": null,
                "name": null,
                "description": null,
                "url": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
g0.expr=increase%28prometheus_rule_group_iterations_missed_total%7Bjob%3D~%22prometheus-k8s%7Cprometheus-user-
workload%22%7D%5B5m%5D%29+%3E+0&g0.tab=1"
```

```
            }
        ],
        "details": {
            "alertname": "PrometheusMissingRuleEvaluations",
            "container": "kube-rbac-proxy",
            "endpoint": "metrics",
            "instance": "10.254.40.166:9092",
            "job": "prometheus-k8s",
            "namespace": "openshift-monitoring",
            "openshift_io_alert_source": "platform",
            "pod": "prometheus-k8s-1",
            "prometheus": "openshift-monitoring/k8s",
            "rule_group": "/etc/prometheus/rules/prometheus-k8s-rulefiles-0/openshift-kube-apiserver-kube-apiserver-slos-
94dbc1dd-b771-4899-8c8f-54c0f5353e04.yaml;kube-apiserver.rules",
            "service": "prometheus-k8s",
            "severity": "warning"
        }
    },
    {
        "occurrenceTime": "2023-06-07T14:49:53Z",
        "summary": "Receivers (notification integrations) are not configured on Alertmanager",
        "severity": 3,
        "type": {
            "eventType": "problem",
            "classification": "AlertmanagerReceiversNotConfigured",
            "condition": null
        },
        "sender": {
            "name": "Prometheus",
            "type": "Webhook Connector"
        },
        "resource": {
            "name": "openshift-monitoring/k8s/openshift-monitoring"
        },
        "links": [
            {
                "linkType": null,
                "name": null,
                "description": null,
                "url": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
g0.expr=cluster%3Aalertmanager_integrations%3Amax+%3D%3D+0&g0.tab=1"
            }
        ],
        "details": {
            "alertname": "AlertmanagerReceiversNotConfigured",
            "namespace": "openshift-monitoring",
            "openshift_io_alert_source": "platform",
            "prometheus": "openshift-monitoring/k8s",
            "severity": "warning"
        }
    }
]
```

Example Curl Command

```
# Webhook details
WEBHOOK_URL=https://webhook-connector-cc1c414f-f356-4ae8-bd61-a85e56d18cf4-aiops.apps.example.ibm.com/webhook-
connector/asy2gjgcebu
AUTH=test:test

# Send payload
curl -X POST -u ${AUTH} --insecure -d '{
  "receiver": "Default",
  "status": "firing",
  "alerts": [
    {
      "status": "firing",
      "labels": {
        "alertname": "PrometheusMissingRuleEvaluations",
        "container": "kube-rbac-proxy",
        "endpoint": "metrics",
        "instance": "10.254.40.166:9092",
        "job": "prometheus-k8s",
        "namespace": "openshift-monitoring",
        "openshift_io_alert_source": "platform",
        "pod": "prometheus-k8s-1",
        "prometheus": "openshift-monitoring/k8s",
        "rule_group": "/etc/prometheus/rules/prometheus-k8s-rulefiles-0/openshift-kube-apiserver-kube-apiserver-slos-
94dbc1dd-b771-4899-8c8f-54c0f5353e04.yaml;kube-apiserver.rules",
        "service": "prometheus-k8s",
        "severity": "warning"
      },
      "annotations": {
        "description": "Prometheus openshift-monitoring/prometheus-k8s-1 has missed 1 rule group evaluations in the last
5m.",
        "summary": "Prometheus is missing rule evaluations due to slow rule group evaluation."
      },
      "startsAt": "2023-06-01T17:02:53.691Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
```

```
g0.expr=increase%28prometheus_rule_group_iterations_missed_total%7Bjob%3D~%22prometheus-k8s%7Cprometheus-user-
workload%22%7D%5B5m%5D%29+%3E+0&g0.tab=1",
      "fingerprint": "24ad9e973e22bdce"
    },
    {
      "status": "firing",
      "labels": {
        "alertname": "AlertmanagerReceiversNotConfigured",
        "namespace": "openshift-monitoring",
        "openshift_io_alert_source": "platform",
        "prometheus": "openshift-monitoring/k8s",
        "severity": "warning"
      },
      "annotations": {
        "description": "Alerts are not configured to be sent to a notification system, meaning that you may not be
notified in a timely fashion when important failures occur. Check the OpenShift documentation to learn how to configure
notifications with Alertmanager.",
        "summary": "Receivers (notification integrations) are not configured on Alertmanager"
      },
      "startsAt": "2023-04-26T19:27:36.642Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
g0.expr=cluster%3Aalertmanager_integrations%3Amax+%3D%3D+0&g0.tab=1",
      "fingerprint": "72bc0ebbd3167d00"
    }
  ],
  "groupLabels": {
    "namespace": "openshift-monitoring"
  },
  "commonLabels": {
    "namespace": "openshift-monitoring",
    "openshift_io_alert_source": "platform",
    "prometheus": "openshift-monitoring/k8s",
    "severity": "warning"
  },
  "commonAnnotations": {},
  "externalURL": "https:///console-openshift-console.apps.example.ibm.com/monitoring",
  "version": "4",
  "groupKey": "{}:{namespace=\"openshift-monitoring\"}",
  "truncatedAlerts": 0
}' -H 'Content-Type: application/json' ${WEBHOOK_URL}
```

Creating IBM Infrastructure Management integrations

You can send event data, inventory data, and metrics data from Infrastructure Management to IBM Cloud Pak for AIOps through an Infrastructure Management integration.

- Before you begin
- Creating Infrastructure Management integrations
- Editing Infrastructure Management integrations
- Deleting Infrastructure Management integrations

Before you begin

Before you configure an Infrastructure Management integration, ensure that the following prerequisites are met:

- You have a working Infrastructure Management cluster. Infrastructure Management must be installed on an Red Hat OpenShift Container Platform cluster. The integration cannot be created for an Infrastructure Management appliance that is running on a virtual machine (VM).
- Event syndication, inventory syndication, and metrics syndication must be enabled within Infrastructure Management as required.

  For information on how to enable event, inventory, and metrics syndication, see *Enabling events, inventory and metrics syndication* within the topic General Configuration.

Note: Infrastructure Management and Cloud Pak for AIOps can be located in the same cluster or in different clusters.

Creating Infrastructure Management integrations

To create an Infrastructure Management integration, complete the following steps:

1. Log in to the IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the **Integrations** page, click **Add integration**.
4. From the list of available integrations, find and click the Infrastructure Management tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Select the **Events**, **Topology**, and **Metrics** from the **Category** list.
5. On the side panel, review the information and when ready to continue, click **Get started**.

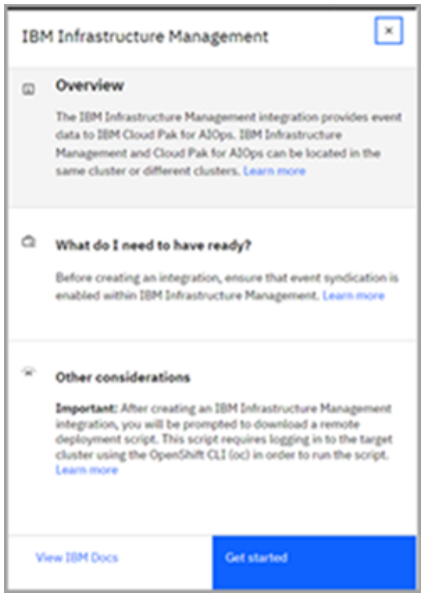


Figure. Add Infrastructure Management integration - get started

6. On the Add integration page, define the general integration details:

   Name: The display name for the integration.

   Description (optional): An optional description for the integration.

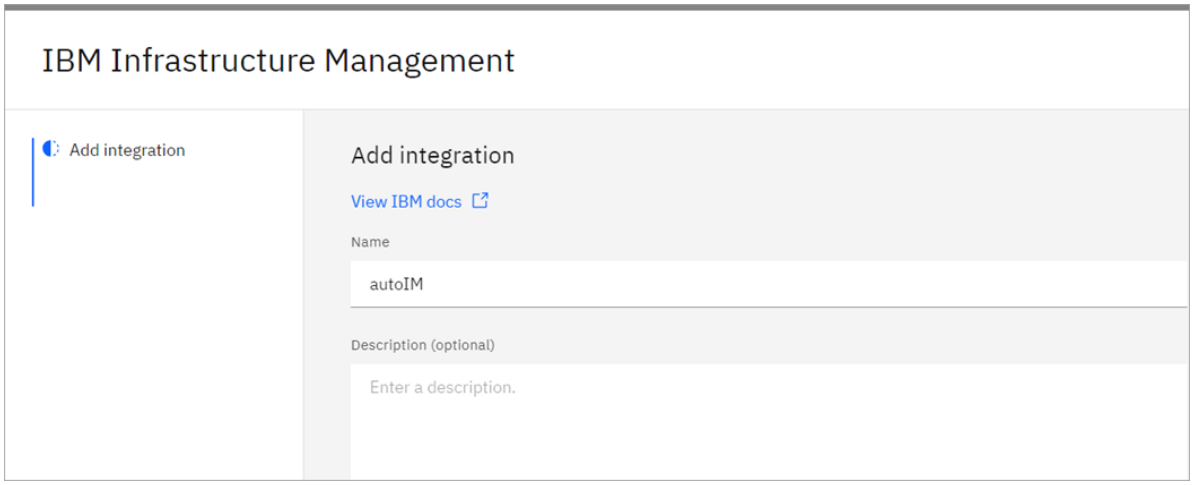


Figure. Add Infrastructure Management integration

7. Click **Done**.

   Important: After creating the integration, you are prompted to copy or download a boostrap command for running a remote deployment script. The script is generated when you create the integration in IBM Cloud Pak for AIOps. This script is required for establish the integration from Infrastructure Management to IBM Cloud Pak for AIOps.

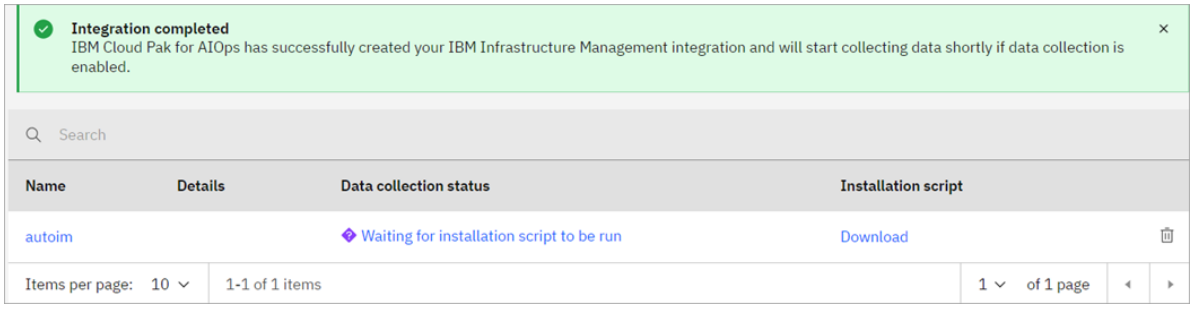


Figure. Add Infrastructure Management integration - completed

8. Copy or download the bootstrap command for the deployment script.

   You can click the **Copy to clipboard** button to copy the command, or click **Download as shell script** to download the command as a `bootstrap.sh` file.

   Note: If you do not obtain the command now, you can copy or download the command later by completing the following steps:

   1. Log in to IBM Cloud Pak for AIOps console.

2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Infrastructure Management** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the IBM Infrastructure Management page, click the **Download** link in the **Remote deployment script** column for the integration.
5. Either copy or download the bootstrap command for the deployment script.

9. Run the remote deployment script on your Infrastructure Management cluster.
    1. Get the OpenShift CLI `oc login` command from the cluster where Infrastructure Management is installed.
    2. From a command line, log in to the Infrastructure Management cluster with the `oc login` command.
    3. Switch to the project (namespace) where Infrastructure Management is installed.
    4. Run the deployment script as a `sh` (script file), such as the downloadable `bootstap.sh` file.

Editing Infrastructure Management integrations

After you create a integration, you can edit the integration to update the name or description. Complete the following steps to edit the integration:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Infrastructure Management** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the IBM Infrastructure Management page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required.
6. Click **Save** when you are done editing.

Deleting Infrastructure Management integrations

If you no longer need your IBM Infrastructure Management integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

To delete a integration, you need to complete the following steps for IBM Cloud Pak for AIOps. Then, you need to complete steps on your Infrastructure Management to remove the integration deployment. The deployment on the cluster is not removed when you delete the integration in the IBM Cloud Pak for AIOps console for IBM Cloud Pak for AIOps.

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **IBM Infrastructure Management** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the IBM Infrastructure Management integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**. Your integration is deleted.
6. Delete the integration deployment on your Infrastructure Management cluster. To delete the deployment, you must manually run OpenShift CLI (`oc`) commands against the cluster.
    1. From a command line, use the `oc login` command to log in to your Infrastructure Management cluster.
    2. Switch to the project (namespace) where Infrastructure Management is installed.

       ```
       oc project <project>
       ```

       Where `<project>` is the project (namespace) where Infrastructure Management is installed.
    3. Delete the `grpc-bridge-secret` secret.

       ```
       oc delete secret grpc-bridge-secret
       ```
    4. Delete the integration deployment

       ```
       oc delete deploy aiops-connector
       ```

       The deployment is now deleted from your Infrastructure Management cluster.
    5. Delete the `aiops-connector-allow-kafka` network policy:

       ```
       oc delete networkpolicy aiops-connector-allow-kafka
       ```

Creating Instana integrations

You can integrate with Instana to enable connectivity to Instana events, metrics, and topology by using an Instana integration. Instana assets can be combined with IBM Cloud Pak for AIOps AI, such as runbooks, metric anomaly detection, and log anomaly detection to help you gain insights.

If needed, you can create multiple Instana integrations. This deployment model has only one topology processor and multiple Instana integrations. The topology processor treats each integration as a distinct topology provider.

Important: You can use the Instana Collector or the Instana Observer to retrieve topology data from an integration with Instana, **but not both**. See Configuring Instana Observer jobs for information on using the Instana Observer.

The list of technologies is determined dynamically, based on the actual resources available in the Instana instance the integration is pulling metrics data from. Each technology includes a list of all available metrics that can be dynamically retrieved by the Instana integration. The integration also provides a set of recommended metrics to select from.

For more information about working with Instana integrations, see the following sections:

- Before you begin
- Creating Instana integrations
- Enabling Instana integrations
- Editing Instana integrations
- Deleting Instana integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Before you begin

Ensure that you have following information to hand:

- Endpoint
- API Token
- Filters to define subsets of data that is pulled from Instana (optional)

Creating Instana integrations

To create an Instana integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Instana** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and click **Get started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: The display name of your integration.
   - **Description**: An optional description for the integration.
   - **Endpoint**: Enter an endpoint for the integration. For example `https://myinstance.mydomain.com`.
   - **API token**: Enter an API token for the integration. The API Token can be fetched from the Instana web UI.



Figure. Add Instana integration

   - **Deployment options**: Select between the *local* or *remote* radio buttons. If you select remote, after adding the integration, you will be provided with a bootstrap command to finish the deployment.

- **Test connection**: click to see is the connection can be established.

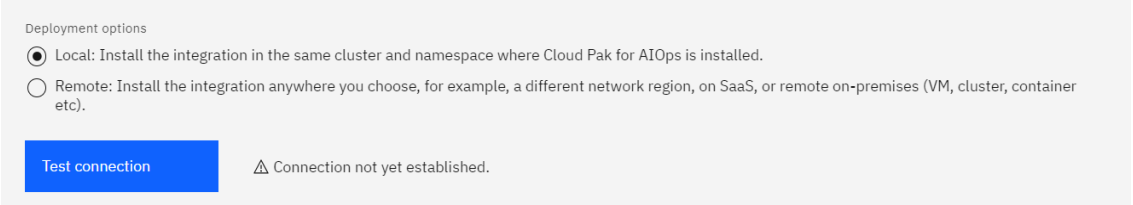


Figure. Test connection

Note: To use the Instana API for the collection of Instana data (metrics, events, and topologies), an API token is required. These API tokens can be created on the Instana user interface. Click the **Settings** icon, then **Team Settings > API Tokens**, and lastly click **Add API Token** to create a new API token.

The default configuration for a new API token is enough for this Instana integration.

7. Click **Next**.
8. If you would you like to optionally use an HTTP proxy configuration with the Instana integration, enter the following *Configure proxy* information:
   - **Proxy URL**: Enter the URL or domain name of the Proxy Server including the protocol. Do not include the port in the Proxy URL field.
   - **Proxy Port**: Enter the port as a number in the Proxy Port field.
   - **Proxy Username**: If the HTTP proxy server requires authentication, enter the proxy username. If not, you can leave this field blank to use the proxy without authentication.
   - **Proxy Password**: If the HTTP proxy server requires authentication, enter the proxy password. If not, you can leave this field blank to use the proxy without authentication.

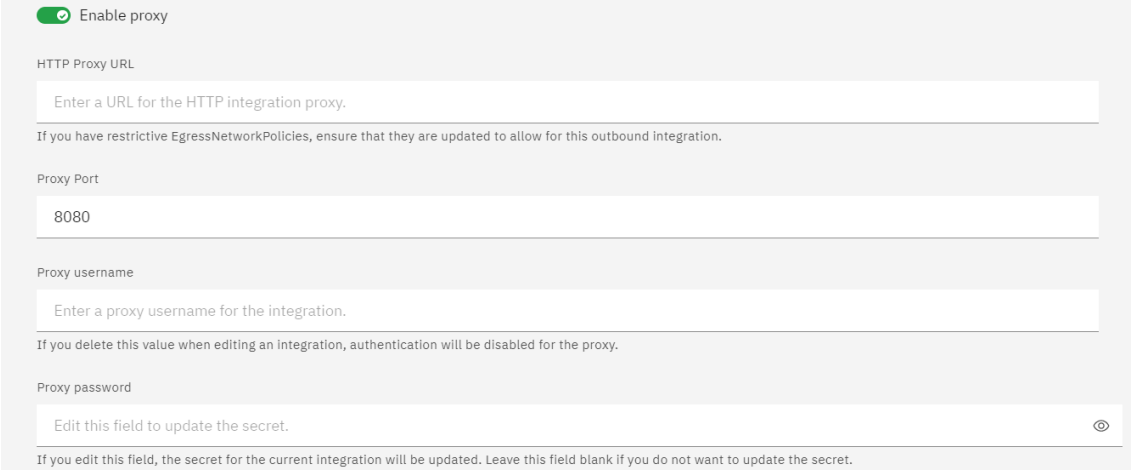


Figure. Configure proxy

Note: An HTTPS proxy server and certificate authentication are not supported for the proxy configuration. Only basic authentication (username and password) or no authentication with an HTTP proxy server are supported.

9. Click **Next**.
10. Enter the following *Collect topology data* information:
    - **Enable data collection**: Enable the toggle button.
    - **Time window (seconds)**: Reference the windowSize within the Instana API.
    - **Integration interval (seconds)**: Specify a time period for how frequently to run the job.
    - **Application name allowlist pattern**: Filter app perspective collection based on the name of the application.
    - **Import Instana application perspectives as Cloud Pak for AIOps applications**: Enable the toggle button (if required).

      Note: When this toggle button is enabled, an IBM Cloud Pak for AIOps application for each application perspective in Instana is automatically created. If it is not turned on, then you create resource groups for only the services that are part of that application in Instana. So, you have the opportunity to define your own IBM Cloud Pak for AIOps applications, by adding your own group to the application. It's important to note that if the application is created automatically by enabling the toggle button, the application is not controlled by the user. It is the product itself that controls the application. Only if you create the application yourself can you control the lifecycle of the application, and thus remove or delete it.
11. Click **Next**.
12. Enter the following *Collect event data* information:
    - **Enable data collection**: Enable the toggle button if required.

- **Types of events**: Select one or more event types for data collection from the following options:
    - Incidents
    - Issues
    - Changes

13. Click **Next**.
14. Enter the following *Collect metric data* information to define how you want to collect metric data. Topology metric data that is related to applications, endpoints, and services can be enabled and thus collected, or indeed disabled.

    - **Enable data collection**: Set the toggle button to 'on'.
    - **Monitored metrics**: You can select built-in metrics to monitor. These metrics are categorized by the technologies that they are associated with. You can also add custom-defined metrics to monitor. To select metrics, use any of the following options:
        - Enter your preferred metrics in the search bar and select their corresponding checkboxes.
        - Search and select your preferred technologies and their associated metrics in their respective filter table. To find the related metrics of a technology, click its corresponding arrow. By default, all metrics of the chosen technologies are selected.
        - Select the **Start from recommended metrics** checkbox if you are not sure which metrics to start with. This option selects all recommended technologies and their associated metrics in the filter table.
        - Clear the selected checkboxes to undo any selection. However, if you clear the **Start from recommended metrics** checkbox, all current selection is cleared.
        - All selected metrics are listed in the **Selected metrics** list. You can remove any listed metric by clicking the **Deselect Metric** icon corresponding to it.
        - You might see performance issues or gaps in your data due to the number of metrics selected. To avoid this problem, consider starting with the set of recommended metrics or those that are most important for you to monitor.

15. Click **Next**.
16. Enter the following *Metric data summary* information:

    - **Selected metrics**: The table lists all the selected technologies and metrics. You can remove any metrics from the list. To remove a listed metric, select the checkbox corresponding to it and click **Remove**.
    - **Only collect Live Data once the integration is established**: Select the checkbox to enable live data collection. **Note**: If you choose to collect only live data, you cannot collect historical data after the live data collection starts.
    - **Scheduled collection interval (minutes)**: Enter a value in minutes. This value is the frequency at which data is collected.
    - **Historical data collection**: The recommended time period for collecting historical metric data is two weeks, though it is possible to collect it for up to 31 days. Enter a start and end date and time in the following formats:
        - *Start date and time**: Select the start date from the calendar and enter the time in the 'hh:mm' format (hours and minutes).

          Note: Your start date should not exceed 31 days from the present.

        - **Time zone**: Select your time zone from the dropdown list.

        - **End date and time**: Click **Add end date** and select a date from the calendar. Also, enter the time in the 'hh:mm' format (hours and minutes). If you don't want to set an end date at this stage, click **Remove End Date**.

        - **Scheduled collection interval (minutes)**: Enter a value in minutes. This value is the frequency at which data is collected.

          Note: If you do not select an end date, IBM Cloud Pak for AIOps collects historical data from the specified start date and continues with live data collection. You cannot collect historical data after live data collection starts.

          Note: If you have already set an end date and time for an Instana integration, you cannot edit it. If you need to edit either of these properties, you must delete your integration and create a new one.

17. Click **Done**.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

You have created an *Instana* integration in your instance.

After the integration creation is complete, and if you selected the *Remote* option for integration deployment, you can download the *bootstrap script*. Select the created integration, and then click **download** from the options menu (three vertical dots). If a script is regenerated, it invalidates the credentials of previously downloaded scripts for that integration.

Note: You can either copy the script or download as an sh file to run it on a remote cluster.

Prerequisites before you run the bootstrap script on a remote cluster

- Podman needs to be installed on the remote cluster.
- Log in to the *Image registry* for the script to pull in the image by using the Podman login command.

After you create your integration, you must enable the data collection to connect your integration with the AI of IBM Cloud Pak for AIOps. For more information about enabling your integration, see Enabling Instana integrations.

To create more integrations (such as a ChatOps integration), see Configuring Integrations.

For more information about working with the insights provided by your integrations, see ChatOps insight management.

For Instana on-prem installation, the recommended rate limit is double the number of resources. This limit is enough to cover most cases, although situations might arise where the limits need to be increased even more. When a more precise way to determine the limit is required (for instance a shared multi-tenant SaaS environment), the following formula can be applied:

```
number-of-metric-API-calls-per-hour ~= (number-of-selected-technologies x 2) x (snapshots-for-selected-technologies / 30)
x (60 / collection-interval)
number-of-topology-API-calls-per-hour ~= (number-of-application-perspectives x (60 / collection-interval)) +(number-of-
```

```
services x (60 / collection-interval))
number-of-events-API-calls-per-hour = 60

total= number-of-metric-API-calls-per-hour + number-of-topology-API-calls-per-hour + number-of-events-API-calls-per-hour
```

Note: This formula is not always 100% accurate, as each technology or plug-in has a different number of metrics collected (the mean value across these is used, which is 2 collection cycles or technology). If the environment is unbalanced toward either end of the spectrum (for instance, if you have mostly hosts that define the most metrics), then the formula might underestimate the required limit.

To determine the number of resources (snapshots) for each infrastructure pluginAPI, you can use the following:

```
api/infrastructure-monitoring/snapshots?plugin=technology_name
```

Example:

```
api/infrastructure-monitoring/snapshots?plugin=host
```

For more information about Instana APIs, see Instana API.

With the following example curl commands, you can retrieve the number of the following resources:

- `snapshots-for-selected-technologies` (such as host)

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/infrastructure-monitoring/snapshots?plugin=host' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

- `number-of-application-perspectives`

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/application-monitoring/applications' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

- `number-of-services`

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/application-monitoring/services' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

Enabling and disabling Instana integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **Instana** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *Collect topology data* section. Enable the toggle button if required.
6. Go to the *Collect event data* section. Enable the toggle button if required.
7. Go to the *Collect metric data* section. Enable the toggle button if required.

Note: Disabling data collection for an integration does not delete the integration.

You have enabled or disabled your integration. For more information about deleting an integration, see Deleting Instana integrations.

Editing Instana integrations

After you create your integration, you can edit it. To edit an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Instana** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Instana integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

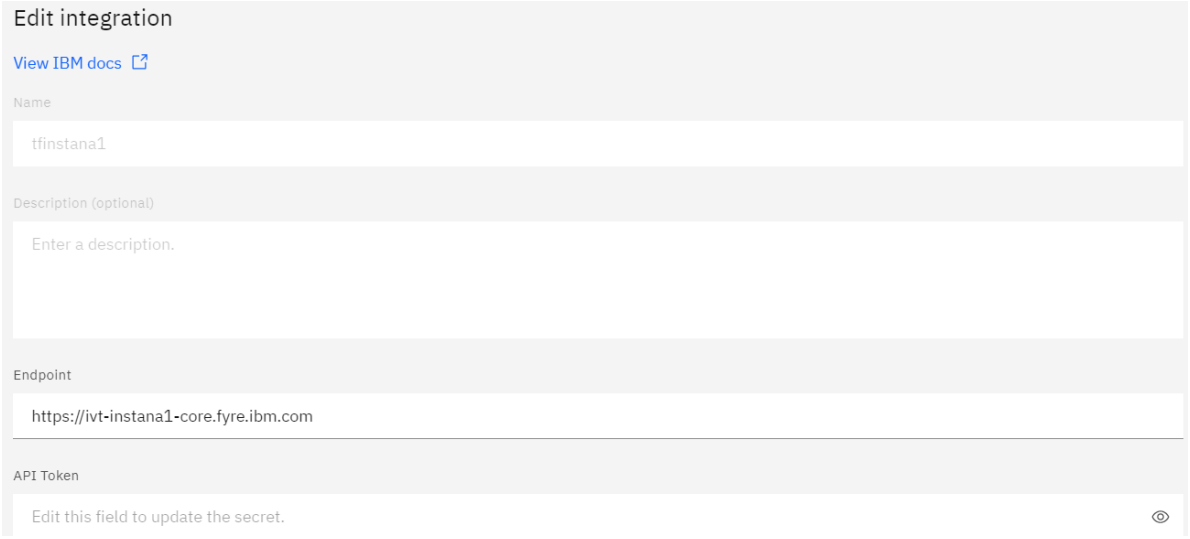


Figure. Edit Instana integration

You have edited your integration. If you have not previously enabled or disabled your application, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling Instana integrations. For more information about deleting an integration, see Deleting Instana integrations.

Deleting Instana integrations

If you no longer need your Instana integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

To delete an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Instana** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Instana integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating Jira integrations

You can integrate with Jira to enable connectivity to Jira tickets. This enables you to view the resolutions for Jira incidents within IBM Cloud Pak for AIOps based on the similar Jira tickets. You can also get Jira issues created for incidents in IBM Cloud Pak for AIOps, update the incident in AIOps, and get the same update in the corresponding Jira issue.

For more information about working with Jira integrations, see the following sections:

- Before you begin
- Best practices
- Creating Jira integrations
- Enabling and disabling data flow in Jira integrations
- Editing Jira integrations
- Deleting Jira integrations

Before you begin

Ensure that you have following information to hand:

- Domain of the Jira instance
- Key for the Jira project
- User name to access the Jira server
- Password or API token for the Jira server
- Jira issue type
- Final state of the Jira issue

Best practices

- When creating multiple Jira connectors, use a different username and token to facilitate speedier data collection.

- Use a personal access token that does not expire.
- When Historical data collection is selected, select a start date close to the date that will be used for AI training. Selecting a date further back might result in more API calls to Jira, especially if there is a large volume of issues.

Creating Jira integrations

To create an Jira integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Jira** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and click **Get started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: The display name of your integration.
   - **Description**: An optional description for the integration.
   - **Domain of Jira instance**: Enter the domain of your Jira instance, for example `https://<your-environment-id>.atlassian.net`.
   - **Jira project key**: Enter the key for Jira project that contains your tickets.
   - **User name to access Jira server**: Enter the name of a Jira user that has permissions to create and manage tickets in the specified project.
   - **Password or API token**: +Enter the password or personal access token that IBM Cloud Pak for AIOps will use when calling the Jira APIs.
7. Click **Test connection**. It is best to wait for the test to succeed before continuing to the next step. This can take some time.
8. Click **Next**.
9. Use the *Field mapping* page to map IBM Cloud Pack for AIOps incident fields to Jira ticket fields. You can use the default template, or you can customize it to suit your environment.

   Note: To customize the mapping, copy the text out of the mapping textbox and into your preferred text editor. Then add the mapping for your fields. When done, copy the text back into the mapping textbox.

   The following example maps the states that are in the `statusMap`. `Keys` are the IBM Cloud Pak for AIOps states and `values` are Jira states. You can set the status using the `statusMap` as shown below:

   ```
   (
   $statusMap := {
   "resolved": "Resolved",
   "inProgress": "In Progress",
   "assigned": "To Do",
   "closed": "Done"
   };
   {
       "summary": $string(incident.title),
       "description": $join(["Incident Id:", $string(incident.id),
                         "\nAIOPS Incident Overview URL: https://", $string(URL_PREFIX), "/aiops/default/resolution-
   hub/incidents/all/", $string(incident.id), "/overview",
                         "\nStatus: ", $string(incident.state),
                         "\nDescription: ", $string(incident.description)]),
       "labels": [$join(["priority:", $string(incident.priority)])],
       "assignee": $string(incident.owner),
       "status": $statusMap[$string(incident.state)] ? $statusMap[$string(incident.state)] : $string(incident.state)
   })
   ```

   You must enter the following information on this page according to your Jira project settings:
   - **Issue type for Jira**: Enter the type of issue that will be created in Jira. The default value is `Task`.
   - **Final state of Jira issue**: Enter the final state of the Jira issue that contains the incident. The default value is `Done`.
10. Click **Next**.
11. Enter the following *Collect ticket data* information:
    - **Data collection**: Enable the toggle button.
    - **Mode**: Select one of the following options:
      - **Live - Continuous Similar Tickets training**
      - **Historical - Initial Similar Tickets training**
    - **Jira issue sampling rate**: Specify the frequency (in minutes) at which data is sampled for Jira issues.
12. Click **Done**.

You have created a *Jira* integration in your instance.

Using a policy to open a Jira ticket

When the Jira integration is up and running, you can create a policy that opens an incident within Jira by selecting the integration from the *Open a ticket* area of your *Promote alerts to an incident* policies. See **Steps 12** and **13** of Promote alerts to an incident.

Enabling and disabling data flow in Jira integrations

If you did not enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **Jira** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *Collect ticket data* section. Enable the toggle button if required.

Note: Disabling data collection for an integration does not delete the integration.

You have enabled or disabled your integration. For more information about deleting an integration, see Deleting Jira integrations.

Editing Jira integrations

After you create your integration, you can edit it. To edit an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Jira** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Jira integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

You have edited your integration. If you have not previously enabled or disabled your application, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling Jira integrations. For more information about deleting an integration, see Deleting Jira integrations.

Deleting Jira integrations

If you no longer need your Jira integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

To delete an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Jira** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Jira integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating Kafka integrations

A Kafka integration can provide either event or log data. Log data can be used to establish a baseline of normal behavior and then identify anomalies. Event data enables the analysis and processing of different types of alerts and events. Anomalies that are identified can be correlated with alerts and events and published to your ChatOps interface to help you determine the cause and resolution of a problem.

Custom integrations can route only a single source of data at one time. Unlike the custom integration type, you can use the Kafka integration type to collect log data and event data from different systems. Then, you can route that information through a forwarding agent, such as an Apache Kafka topic like Sysdig or FluentBit.

You can also use the Kafka integration to enable training with an offline, historical data set. For more information about using Kafka topics for offline training, see Importing offline log data.

Notes:

- The Kafka replication factor is set to one replica by default. If you are implementing a production deployment of IBM Cloud Pak for AIOps, you might lose data if your Kafka pods fail or restart. If the data collection is enabled in your Kafka integration when the Kafka pods go down, you might experience a gap in the data that your integration generated during that down period.
- The maximum Kafka message size limit in Cloud Pak for AIOps is `1048588` bytes.
- If you frequently send in messages that are close to the maximum size, it can adversely impact the performance.

For more information about working with Kafka integrations, see the following sections:

- Creating Kafka integrations

- Enabling Kafka integrations
- Editing Kafka integrations
- Importing offline log data
- Deleting Kafka integrations

Creating Kafka integrations

Unlike other integration types, you can use Kafka integrations to push properly formatted historical training data to IBM Cloud Pak for AIOps. To create a Kafka integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Kafka** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: The display name of your Kafka integration.
   - **Description**: An optional description for the Kafka integration.
   - **Kafka partitions**: The number of Kafka partitions. The default value of `1` might be suitable for importing a few log records. However, if you intend to import many records, increase this number. The range is `1` to `500`. For example, for a proof of content (PoC) deployment where you need to import a large data set, you can use `48` Kafka partitions.
   - **JSON processing option**: Select a JSON processing option.
     - **None**: The default option. The JSON is not processed or modified.
     - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
     - **Filter**: This option extracts the JSON object and replaces it with an empty string.

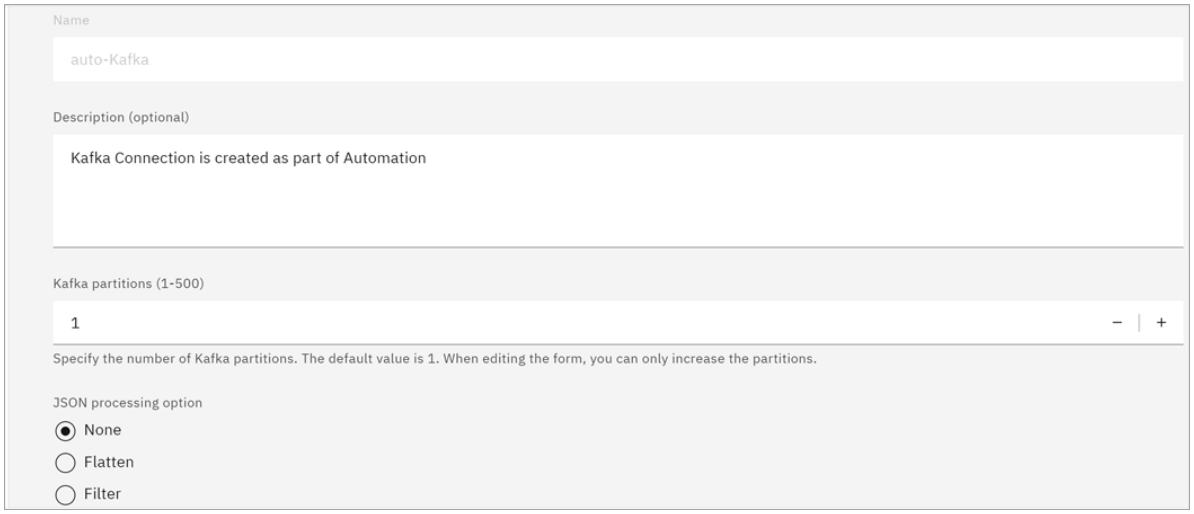


Figure. Create Kafka integration

   For more information about the options, see Managing embedded JSON.
7. Click **Next**.

Entering Field mapping information

Unlike other integration types, you can specify what kind of information you want to run through your Kafka integration. For example, if you want to collect data from multiple custom sources, you can set them up as separate Kafka integrations.

1. In the **Field mapping** section, set the **Data source** to specify the type of data incoming from the Kafka integration. You can select **Events** or **Logs**.
2. Specify the **Mapping type**, which creates the field mapping for your specified log type.
   - Mapping types for **Events** include **None**, **PagerDuty**, and **NOI**.
   - Mapping types for **Logs** are **None**, **ELK**, **Falcon LogScale**, **Mezmo**, **Custom**, and **Splunk**.
   - If you have log data that is already normalized, choose **None**.

- If your log data is in one of the supported log formats, such as ELK or Falcon LogScale, choose the corresponding log format.
- For field mapping for **Custom** types, see Creating custom integrations.

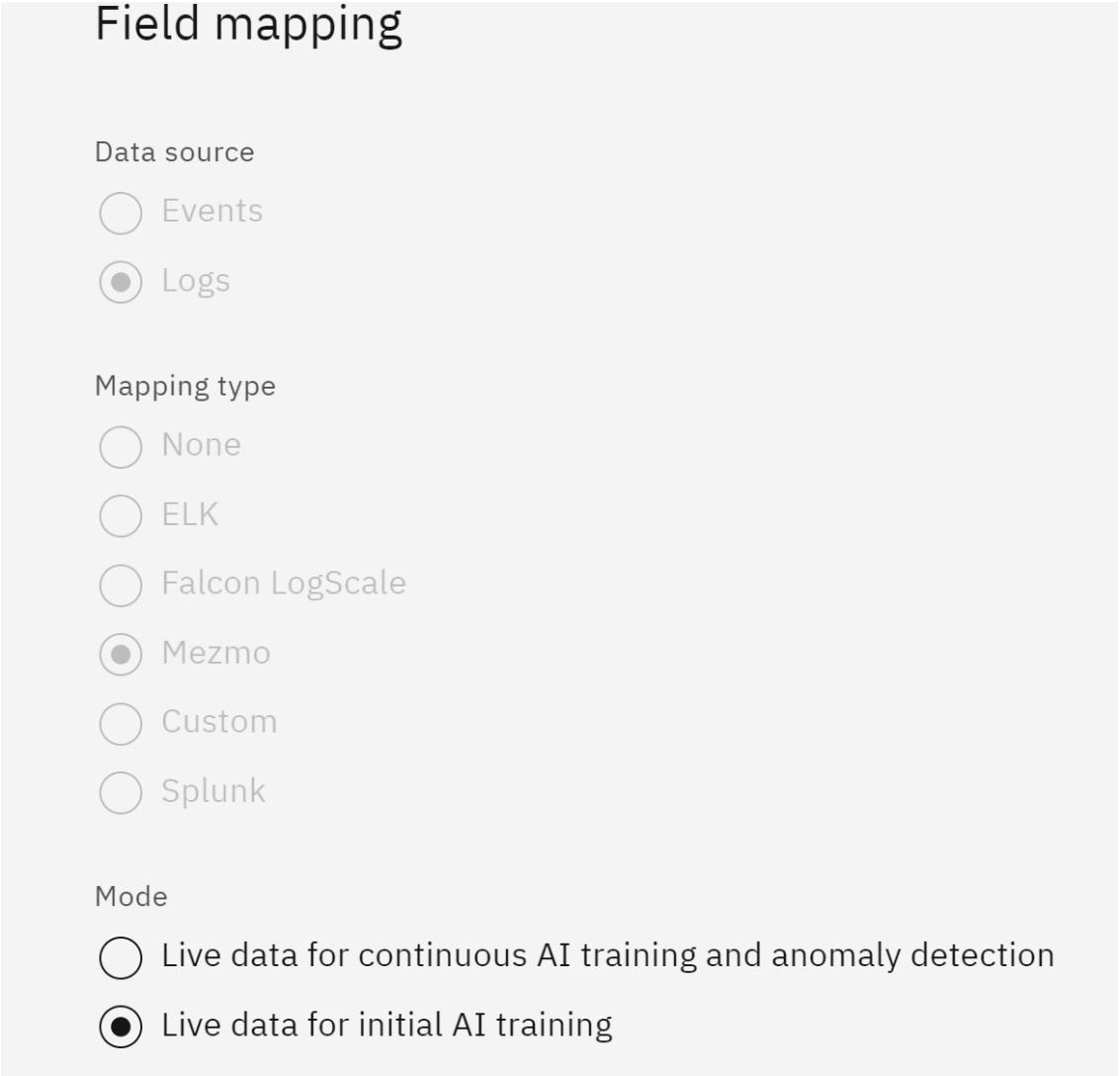


Figure. Field mapping type

3. The **Topic** name for a Kafka integration is predefined. Note this topic and send your data to it.
4. For log integrations, complete the following settings:
    1. Enter the **Maximum number of logs per second** rounded to the nearest thousand. The maximum number of logs per second is collected in increments of 1,000 up to 25,000 and is rounded up to the nearest thousand sent from Kafka to IBM Cloud Pak for AIOps.
    2. Set the **Field mapping** mode to **Live data for initial AI training** for initial training or to **Live data for continuous AI training and anomaly detection**.

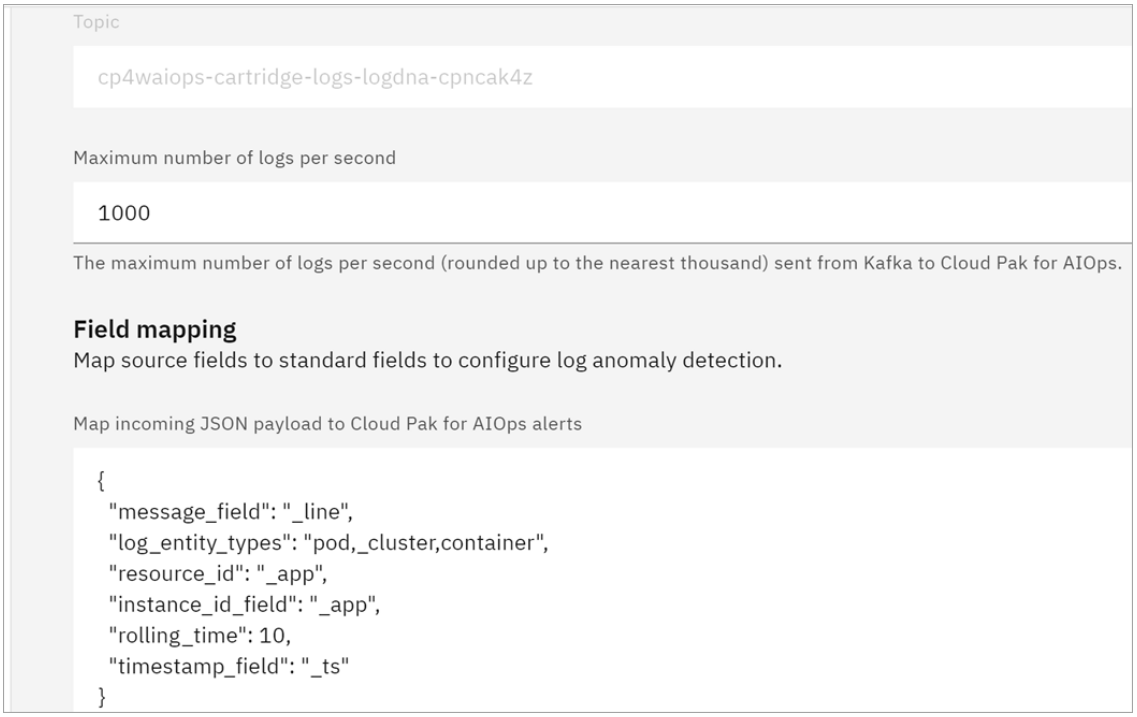


Figure. Field mapping

- You can improve search performance by mapping the fields from your implementation fields to standard IBM Cloud Pak for AIOps fields. For more information about how field mappings are defined, see Mapping data from incoming sources.
- For more information about using mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data using regular expressions.

3. In the **Mapping** section, if you use a **Mapping type** other than **None**, verify whether the JSON document that is automatically assigned to this field is valid. The validity depends on the structure of your log records. For example, for a PoC deployment, an example default **Field mapping** for Kafka can resemble the following mapping:

```
{
  "codec": "elk",
  "message_field": "@message",
  "log_entity_types": "@hostname, @bundleName, @context.Environment",
  "instance_id_field": "@properties.processtype",
  "rolling_time": 10,
  "timestamp_field": "@timestamp"
}
```

If you chose **None** for the **Mapping type**, each log record in your log files conforms to the following format. Set both of the `application_id` and `application_group_id` values to `"1000"`, and make sure that a 13-digit timestamp value is assigned to the `timestamp` field:

```
{
    "timestamp": 1581549409000,
    "utc_timestamp": "2020-02-12 23:16:49.000",
    "instance_id": "calico-node-jn2d2",
    "application_group_id": "1000",
    "application_id": "1000",
    "features": [],
    "meta_features": [],
    "level": 1,
    "message": "[64] ipsets.go 254: Resyncing ipsets with dataplane. family=\"inet\"",
    "entities": {
        "pod": "calico-node-jn2d2",
        "cluster": null,
        "container": "calico-node",
        "node": "kube-bmgcm5td0stjujtqv8m0-ai4itsimula-wp16cpu-00002c34"
    },
    "type": "StandardLog"
}
```

Important: The **Events** data that is collected must follow the **Kafka integration normalized Event schema**. For more information, see Normalized mapping rules.

5. Click **Next**.

Note: When importing events the JSON structure for an event must be on a single line in the file. For more information about the structure for importing event data, see Normalized Mapping Rules.

Entering AI training and log data information

1. If you want to enable data collection for AI training and log data, switch the **Data collection** toggle to **On**.

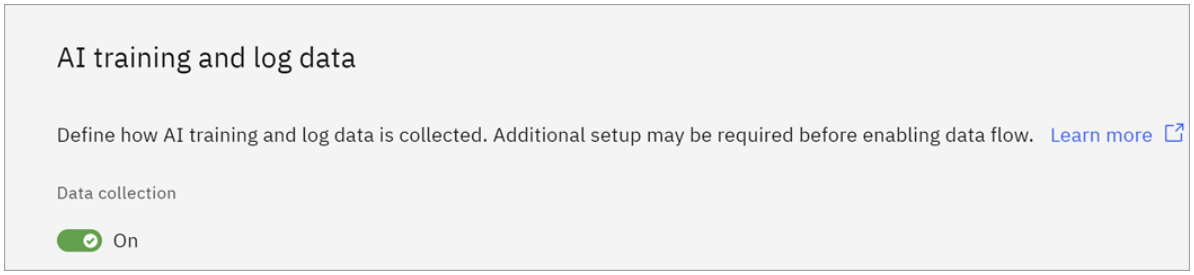


Figure. AI training

2. Click **Next**.
3. The UI displays the *Resource requirements* page.

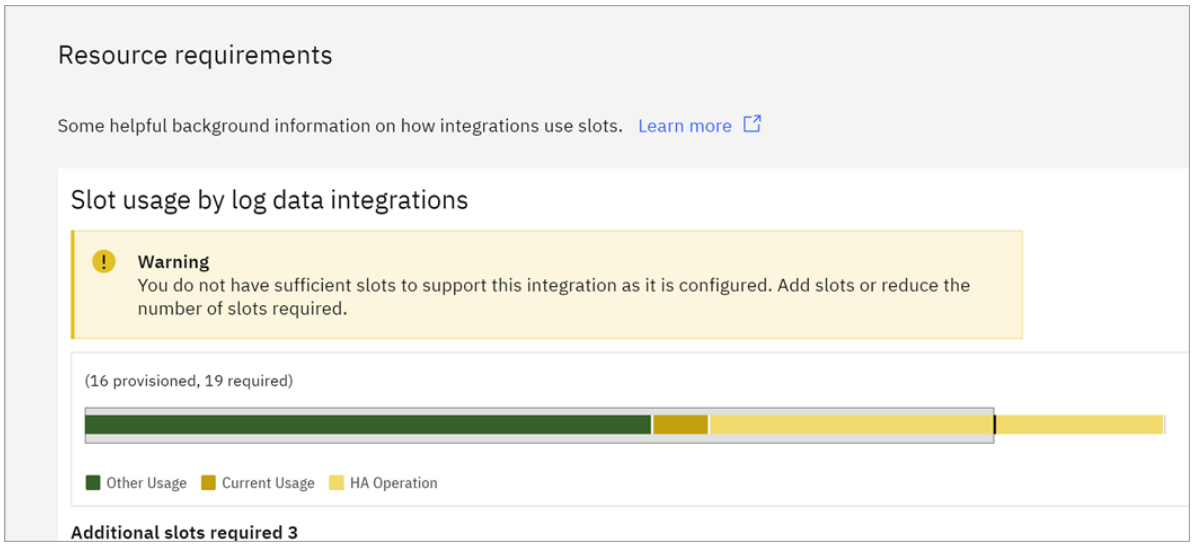


Figure. Resource requirements

Note: If the **Data Source** is set to **Events**, you will see a message stating that you need to enable logs data collection to see the resource management overview. You can skip this page when events are enabled.

On the *Resource requirements* page, you can review the slot usage for your log integrations to see if there are enough slots to fully support the integration for multizone high availability.

If you set the **Data collection** toggle to **On**, you will see the resource management overview.

- If your current usage and other usage are less than the provisioned slots, but the HA slots exceed the provisioned slots, you will be able to create the integration, but will see a warning that you do not have enough slots. The integration will not have multizone high availability.
- If your projected usage exceeds the provisioned slots, you will not be able to create the integration because you do not have enough slots on your system for log data integrations.
- If your total slots, including HA slots, are within the provisioned slots, the integration will have multizone high availability.

  Note: HA operation assumes high availability for three zones.

If you set the **Data collection** toggle to **Off**, you will see a message stating that you need to enable logs data collection to see the resource management overview. When data collection is off, no slots are used by that integration.

4. Click **Done** to save your integration.

You created a Kafka integration in your instance, and you can now use the Kafka integration as the basis for your offline AI model training. After you create your integration, enable the data collection to connect your integration with the AI of IBM Cloud Pak for AIOps.

- For more information about enabling your integration, see Enabling Kafka integrations.
- To create more integrations, such as a ChatOps integration, see Configuring Integrations.
- For more information about working with the insights provided by your integrations, see ChatOps insight management.
- For more information about AI model training, see Configuring AI models.

Importing offline log data

Before you proceed with the following steps, make sure that the data flow is enabled for the Kafka Ops integration that you previously defined. You can import offline log data into IBM Cloud Pak for AIOps in two ways. Both mechanisms publish the offline log data to the Kafka topic that is associated with the Kafka Ops integration.

Using a Kafka utility to publish log data

1. Install a Kafka utility, such as the kcat utility.

2. Log in to the Red Hat® OpenShift® cluster where IBM Cloud Pak for AIOps is installed.
3. Run the following commands from the namespace where IBM Cloud Pak for AIOps is installed:

```
oc extract secret/iaf-system-cluster-ca-cert --keys=ca.crt --to=- > ca.crt

export sasl_password=$(oc get secret cp4waiops-cartridge-kafka-auth-0 --template={{.data.password}} | base64 --
decode);

export BROKER=$(oc get routes iaf-system-kafka-bootstrap -o=jsonpath='{.status.ingress[0].host}{"\n"}'):443
```

4. Publish the log data to the Kafka topic. This example uses kcat. If you also use kcat, update the variables and run the following commands:

```
export KAFKA_TOPIC=<kafka topic>

export LOG_FILE=<log file>

kcat -X security.protocol=SASL_SSL -X ssl.ca.location=ca.crt -X sasl.mechanisms=SCRAM-SHA-512 -X
sasl.username=cp4waiops-cartridge-kafka-auth-0 -X sasl.password=$sasl_password -X
enable.ssl.certificate.verification=false -b $BROKER -P -t $KAFKA_TOPIC -l $LOG_FILE
```

Copying log data into the Kafka cluster

Use this option only if you can't use a Kafka utility such as kcat for your environment. Complete the following steps to publish the log data to the corresponding Kafka topic by copying the log data into the Kafka cluster:

1. Run the following commands from the namespace where IBM Cloud Pak for AIOps is installed to obtain the Kafka password and Kafka broker. Record both values, which you need in a following step.

```
export kafka_password=$(oc get secret cp4waiops-cartridge-kafka-auth-0 --template={{.data.password}} | base64 --
decode)

export kafka_broker=$(oc get routes iaf-system-kafka-bootstrap -o=jsonpath='{.status.ingress[0].host}{"\n"}'):443

echo $kafka_password

echo $kafka_broker
```

2. Copy the log file to the Kafka cluster:

```
oc cp <log file> iaf-system-kafka-0:/tmp
```

3. Check that the file is copied to the Kafka cluster from the namespace where IBM Cloud Pak for AIOps is installed:

```
oc exec -it iaf-system-kafka-0 -- bash

ls /tmp/<log file>

exit
```

4. Get the `ca.cert` certificate file:

```
oc extract secret/iaf-system-cluster-ca-cert --keys=ca.crt --to=- > ca.crt
```

5. Convert `ca.crt` to Java keystore format (JKS):

```
keytool -import -trustcacerts -alias root -file ca.crt -keystore truststore.jks -storepass password -noprompt
```

6. Copy the truststore file to the Kafka cluster:

```
oc cp truststore.jks iaf-system-kafka-0:/tmp
```

7. Give bash access to the Kafka cluster again:

```
oc exec -it iaf-system-kafka-0 -- bash
```

8. Create a file within the Kafka pod that is called `producer.properties`. The contents of the `producer.properties` file can resemble the following properties. Update the `variable placeholders` before you use this content. The `ssl.truststore.location` value points to the JSK file that you created earlier.

```
security.protocol=SASL_SSL
ssl.truststore.location=/tmp/truststore.jks
ssl.truststore.password=password
sasl.mechanism=SCRAM-SHA-512
sasl.jaas.config=org.apache.Kafka.common.security.scram.ScramLoginModule required \
username="cp4waiops-cartridge-kafka-auth-0" \
password="<kafka password>"
```

9. From the pod, run the Kafka producer script to publish the log data to the corresponding Kafka topic. Update the `variable placeholders`:

```
/opt/Kafka/bin/Kafka-console-producer.sh --broker-list <kafka broker> --producer.config producer.properties --topic
<kafka topic> < <log file>
```

Enabling and disabling Kafka integrations

If you didn't enable your data collection during creation, you can enable your integration after. You can also disable a previously enabled integration the same way. If you selected **Live data for initial AI training** when you created your integration, disable the integration before AI model training. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **Kafka** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting a integration, see Deleting Kafka integrations.

Editing Kafka integrations

After you create your integration, you can edit the integration. For example, if you want to refine the field mapping for a Kafka integration from a Falcon LogScale source, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Kafka** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Kafka integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited. If your application was not previously enabled or disabled, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling Kafka integrations. For more information about deleting a integration, see Deleting Kafka integrations.

Deleting Kafka integrations

If you no longer need your Kafka integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable data collection before you delete your integration. For more information about disabling data collection, see Enabling and disabling Kafka integrations.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Kafka** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Kafka integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Connecting with Metric Manager

To integrate Metric Manager (formerly IBM® Operations Analytics Predictive Insights) with IBM Cloud Pak for AIOps , first ensure that you implement the prerequisites, and then change the rules file in Metric Manager.

Note: IBM Cloud Pak for AIOps comes with built-in metric capabilities. Connect with Metric Manager only if these capabilities are insufficient to your needs.

Prerequisites

To allow the forwarding of event data from Metric Manager to IBM Cloud Pak for AIOps , and the integration of both, do the following:

1. Event forwarding from Metric Manager to IBM Cloud Pak for AIOps is done by forwarding from the on-premises objectserver to IBM Cloud Pak for AIOps by using a uni-directional gateway. To set up this gateway, follow the instructions described here.
2. For Metric Manager 1.3.6 integration with IBM Cloud Pak for AIOps , apply InterimFix005 available here.

Procedure

Change the rules file in Metric Manager. Make the following changes to the rules file (located here: $PI_HOME/probe/omnibus/probes/linux2x86/stdin-tasp.rules) in Metric Manager.

Note: Ensure you backup the rules file before you edit the file.

1. In the file, locate the following two lines:

```
#@Identifier = @Node + " " + @AlertKey + " " + @AlertGroup + " " + @Type + " " + @Agent + " " + @Manager
@Identifier = $Identifier
```

Remove the '#' from the first line and add it to the start of the second, so that it changes to the following:

```
@Identifier = @Node + " " + @AlertKey + " " + @AlertGroup + " " + @Type + " " + @Agent + " " + @Manager
#@Identifier = $Identifier
```

2. Add the following before the last closing bracket "}", replacing "PI_UI_HOST.example.com=" with the fully qualified domain name of the Metric Manager UI host:

```
@URL = "https://PI_UI_HOST.example.com:9998/predictiveinsights/jsp/wlp/wlpAnomalyView.jsp?anomalyKey=" +
$TASPIdentifier
update(@URL)
```

3. Add either of the following just after the last closing bracket "}":

   - IBM Cloud Pak for AIOps

```
#For integration with IBM Cloud Pak for AIOps :

if (regmatch($Class, "89350") || regmatch($Class, "89360")) {
   @AlertGroup = "ANOMALY:" + @TASPAnomalousMetricGroups + ":" + @TASPAnomalousMetrics
   update(@AlertGroup)
   }

   ## Note: Do not forward these consolidated events from Metric Manager, as the downstream system will do this
grouping:

   if (regmatch($TASPIdentifier, "Node") || regmatch($TASPIdentifier, "RelatedEvents") ||
regmatch($TASPIdentifier, "Incident")) {
  discard
  }
```

   - IBM Cloud Pak for AIOps Event Manager

```
#For integration with IBM Cloud Pak for AIOps Event Manager

if (regmatch($Class, "89350") || regmatch($Class, "89360")) {
   @AlertGroup = "ANOMALY:" + @TASPAnomalousMetricGroups + ":" + @TASPAnomalousMetrics
   update(@AlertGroup)
   }
   $Identifier = regreplace($Identifier, "(_\d{13}$)", "" )
   @Identifier = $Identifier
   update(@Identifier)

   ## Note: Do not forward these consolidated events from Metric Manager, as the downstream system will do this
grouping:

   if (regmatch($TASPIdentifier, "Node") || regmatch($TASPIdentifier, "RelatedEvents") ||
regmatch($TASPIdentifier, "Incident")) {
  discard
  }
```

4. To start the gateway, run the following:

```
cd /opt/IBM/tivoli/netcool/omnibus/bin
nohup ./nco_g_objserv_uni -name NCO_GATE -propsfile /opt/IBM/tivoli/netcool/omnibus/gates/aiops/aiops.props &
```

Creating Mezmo integrations

A Mezmo integration provides log data, which is used to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts and events, and published to your ChatOps interface to help you determine the cause and resolution of a problem.

For more information about working with Mezmo integrations, see the following sections:

- Creating Mezmo integrations
- Enabling Mezmo integrations
- Editing Mezmo integrations
- Deleting Mezmo integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating Mezmo integrations

The Mezmo integration type collects data from a Mezmo data source.

About this task

Before you create a integration, gather the following information:

- **Load**: To prevent the integration from placing an inordinate load on your data source and potentially impacting your logging operations, the integration connects to only one API with a default data frequency of 60 seconds. This restriction is controlled by using the **Sampling rate** setting in the Procedure

section.

- **Access:** Mezmo data sources are cloud-based and use a service key approach to enable access.
- **Data volume**: Typical data volumes are as follows:
  - 5.5 MB (15000 lines) for 20 seconds on Development environment.
  - 49 logs (19 KB) in a 10-second interval. Pulled within 30 seconds.
  - 10 logs (3.2 KB) in a 10-second interval. Pulled within 30 seconds.
  - 195 logs (75 KB) in a 30-second interval. Pulled within 60 seconds.

Procedure

To create a Mezmo integration, step through the following sections:

- Adding a Mezmo integration
- Specifying integration parameters
- Specifying field mapping
- Specifying how log data is collected for AI training

Adding a Mezmo integration

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Mezmo** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.

Specifying integration parameters

1. On the *Add integration* page enter the following integration information:
   - **Name**: The display name of your integration.
   - **Description**: An optional description for the integration.
   - **URL**: The hostname or IP address of the Mezmo API. To find the URL, log in to your Mezmo account or check the Mezmo instance in your IBM Cloud account. For more information, see the Log Analysis documentation about Endpoints.
   - **Service key**: The Mezmo service key. For more information, see Service keys.

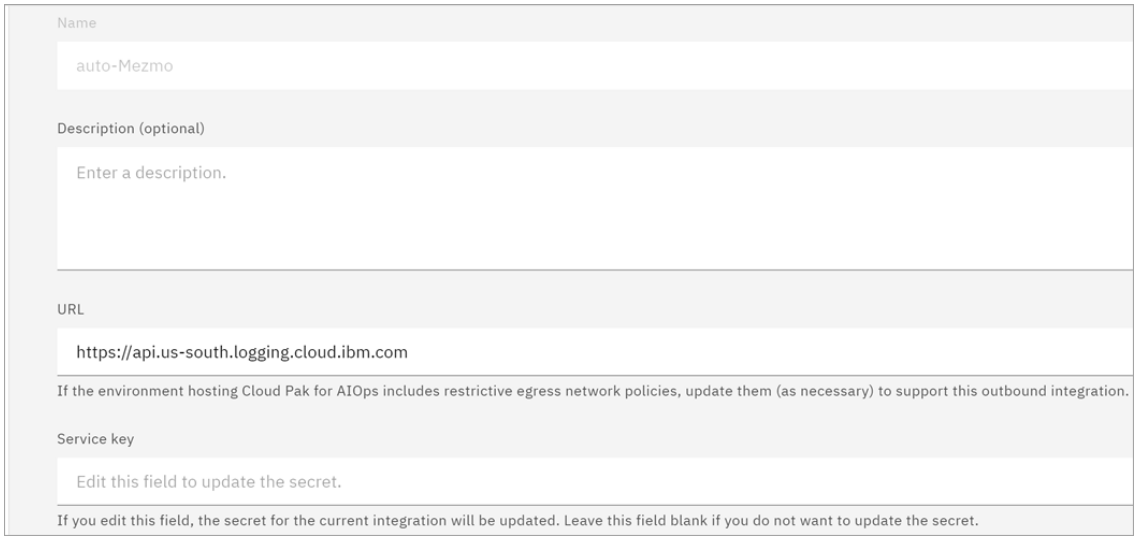


Figure. Create Mezmo integration

   - **Certificate**: An optional certificate used to verify the SSL/TLS connection to the REST service.
   - **Filters**: Defines subsets of data that is pulled from Mezmo. For example, `healthcheck (successful OR ping)` returns all of the log lines with the word `healthcheck` and without the word `successful`, and all of the log lines with the word `healthcheck` and with the word `ping`. For more information about Mezmo filtering, see Search and Filter Log Data documentation.
   - **Base parallelism**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots. In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this integration, you can set this value to be equal to the **source** parallelism.

- **Sampling rate**: The rate at which data is pulled from the live source (in seconds). The default value is **60**.
- **JSON processing option**: Select a JSON processing option.
  - **None**: The default option. The JSON is not processed or modified.
  - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
  - **Filter**: This option extracts the JSON object and replaces it with an empty string.

For more information about the options, see Managing embedded JSON.

2. Click **Test connection**.

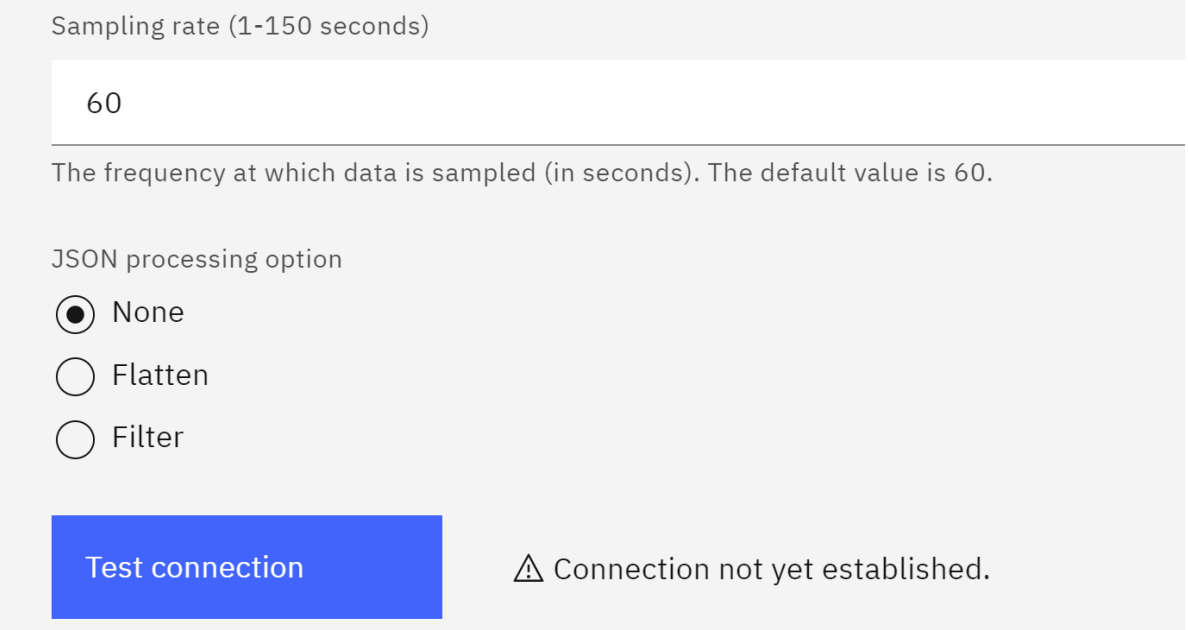


Figure. Test connection

Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see Improving data streaming performance for log anomaly detection.

3. Click **Next** to move to the next page.

Specifying field mapping

1. On the *Field mapping* page you can improve search performance by mapping the fields from your implementation fields to standard fields within IBM Cloud Pak for AIOps.

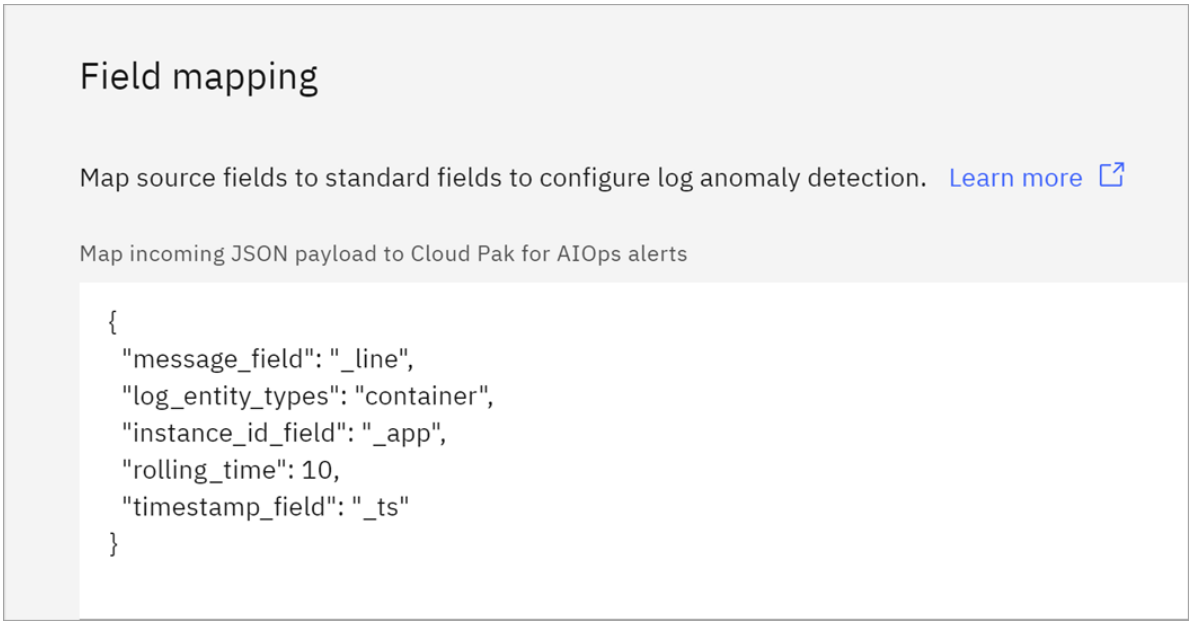


Figure. Field mapping

- For more information about how field mappings are defined, see Mapping data from incoming sources.

- For more information about using mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data using regular expressions.

The following code snippet displays an example of field mapping by using the supported format. When coding your mapping, use this example to help you.

```
{
  "codec": "mezmo",
  "rolling_time": 10,
  "instance_id_field": "_app",
  "log_entity_types": "container",
  "message_field": "_line",
  "timestamp_field": "_ts"
}
```

You can also map from a nested field in the source data to a field in IBM Cloud Pak for AIOps data. For example, in the following code snippet, the line `"message_field": "_line.message",` maps the field `message_field` in IBM Cloud Pak for AIOps data to the nested field `_line.message` in the Mezmo data.

```
{
 "rolling_time": 10,
 "instance_id_field": "_app",
 "log_entity_types": "container",
 "message_field": "_line.message",
 "timestamp_field": "_ts"
}
```

You can also create an OR relationship between multiple nested fields, where the OR relationship is expressed by using a semicolon. For example, in the following code snippet, the line `"message_field": "_line.message1;_line.message2;_line.message3",` maps the field `message_field` in IBM Cloud Pak for AIOps data to one of three possible nested fields in the Mezmo data:

- `_line.message1`, or
- `_line.message2`, or
- `_line.message3`
  - The system first looks for `_line.message1` in the Mezmo data record, and if it finds this field, then it uses it for the mapping.
  - If it can't find `_line.message1` in the Mezmo data record, then it looks for the next item `_line.message2` and if it finds it, then it uses it for the mapping.
  - If it can't find `_line.message2` in the Mezmo data record, then it looks for the next item `_line.message3` and if it finds it, then it uses it for the mapping.

```
{
  "rolling_time": 10,
  "instance_id_field": "_app",
  "log_entity_types": "container",
  "message_field": "_line.message1;_line.message2;_line.message3",
  "timestamp_field": "_ts"
}
```

2. Click **Next** to move to the next page.

Specifying how log data is collected for AI training

1. On the *AI training and log data* page, select how you want to manage collecting data for use in AI training and anomaly detection. Click the *Data flow* toggle to turn on data flow and then select how you want to collect data:

   - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your integration is used to both train AI models and analyze your data for anomalous behavior.

     Note: After an initial installation, there is no data at all in the system. If you select this option, then the two different log anomaly detection algorithms behave in the following ways:

     - Natural language log anomaly detection does not initially detect anomalies as no model has been trained. You can retrieve historical data (select **Historical data for initial AI training**) to speed up the retrieval of data to train on, or you can leave the **Live data for continuous AI training and anomaly detection** setting on. In the latter case, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.
     - Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.

   - **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your integration.

     Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.

   - **Historical data for initial AI training**: A single set of training data used to define your AI model. You must specify a Start and End date, and the parallelism of your source data. Historical data is harvested from existing logs in your integration over a specified time period in the past.
     - **Start date**: Select a start date from the calendar.

Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.

- **End date**: Select an end date from the calendar.

  Note: If you do not specify the end date, then live data collection follows the historical data collection. If you do not want to set an end date, click **Remove end date**.

- **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

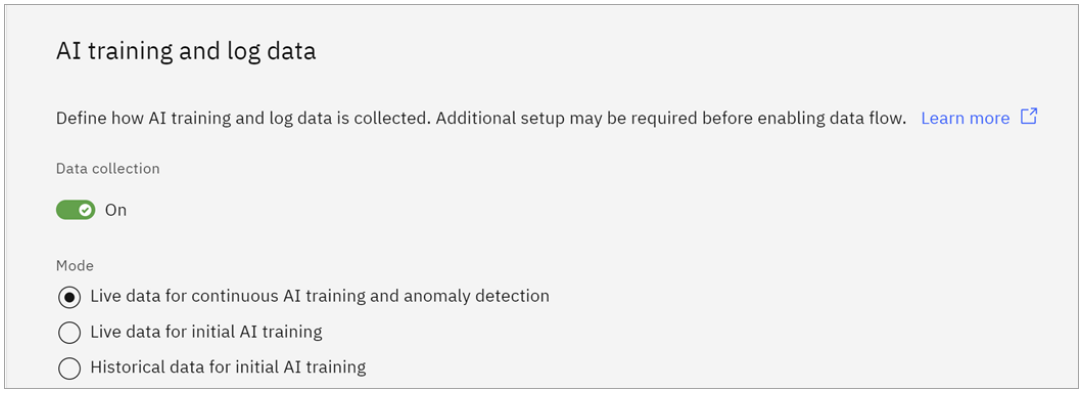


Figure. AI training

Important: Keep in mind the following considerations when you select your data collection type:

- Anomaly detection for your integration occurs if you select *Live data for continuous AI training and anomaly detection*.
- Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

2. Click **Next**.
3. On the *Resource requirements* page, you can review the slot usage for your log integrations to see if there are enough slots to fully support the integration for multizone high availability.

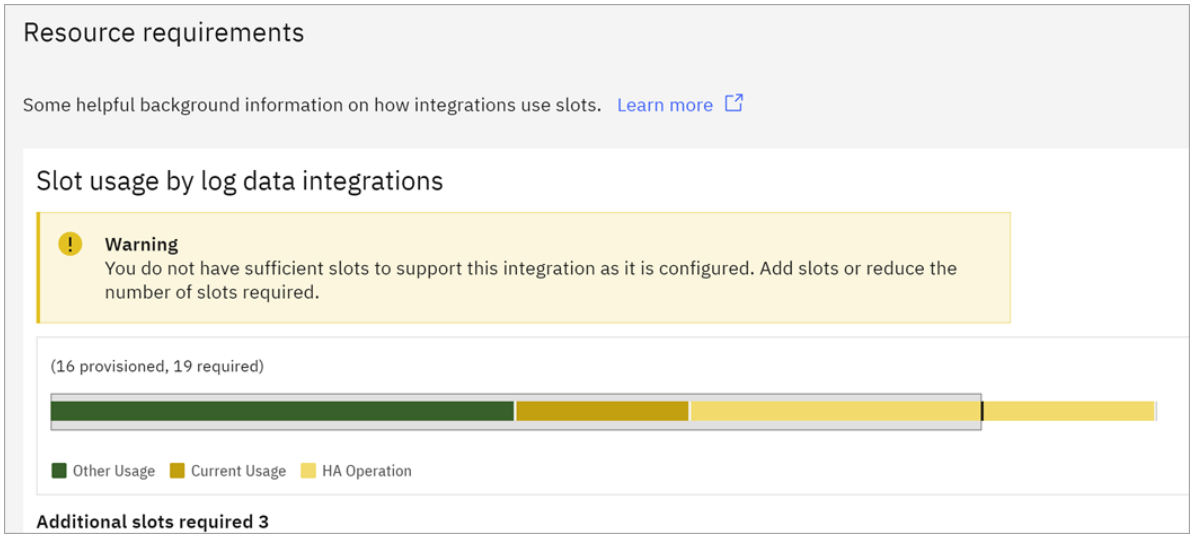


Figure. Resource requirements

If you set the **Data collection** toggle to **On**, you will see the resource management overview.

- If your current usage and other usage are less than the provisioned slots, but the HA slots exceed the provisioned slots, you will be able to create the integration, but will see a warning that you do not have enough slots. The integration will not have multizone high availability.
- If your projected usage exceeds the provisioned slots, you will not be able to create the integration because you do not have enough slots on your system for log data integrations.
- If your total slots, including HA slots, are within the provisioned slots, the integration will have multizone high availability.

  Note: HA operation assumes high availability for three zones.

If you set the **Data collection** toggle to **Off**, you will see a message stating that you need to enable logs data collection to see the resource management overview. When data collection is off, no slots are used by that integration.

4. Click **Save**.

You have created a Mezmo integration in your instance. After you create your integration, you must enable the data collection to connect your integration with the AI of IBM Cloud Pak for AIOps. For more information about enabling your integration, see Enabling Mezmo integrations.

To create more integrations (such as a ChatOps integration), see Configuring Integrations.

For more information about working with the insights provided by your integrations, see ChatOps insight management.

Enabling and disabling Mezmo integrations

If you did not enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. If you selected *Live data for initial AI training* when you created your integration, you must disable the integration before AI model training. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **Mezmo** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting a integration, see Deleting Mezmo integrations.

Editing Mezmo integrations

After you create your integration, your can edit the integration. For example, if you specified *Historical data for initial AI training* but now want your integration to pull in live data for continuous monitoring, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Mezmo** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Mezmo integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

Your integration is now edited. If your application was not previously enabled or disabled, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling Mezmo integrations. For more information about deleting a integration, see Deleting Mezmo integrations.

Deleting Mezmo integrations

If you no longer need your Mezmo integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable data collection before you delete your integration. For more information about disabling data collection, see Enabling and disabling Mezmo integrations.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Mezmo** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Mezmo integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Note: If you manually send Mezmo logs from an archive, the model performs as expected only on other logs that are sent from the archive. Mixing archived data and data from the active service instance results in lower quality models.

Creating New Relic integrations

A New Relic integration collects live and historical metric data from the New Relic server in IBM Cloud Pak® for AIOps. This metric data helps to monitor and analyze application performance. To activate the integration, you must have an endpoint and an Access Token ready. The metric anomaly detection AI algorithm is used to analyze the New Relic metrics for identifying deviations from the normal behavior and detecting anomalies. New Relic v6.5.2 with v2 API is the version that is supported for the New Relic integration.

For more information about working with New Relic integrations, see the following sections:

- Creating New Relic integrations
- Enabling and disabling New Relic integrations
- Editing New Relic integrations
- Deleting New Relic integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating New Relic integrations

To create a New Relic integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **New Relic** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section, or alternatively start typing the integration name in the search bar and the corresponding tile will appear.
5. On the side-panel, review the instructions and select your required options from the following sections:
   - **Select the data types that you need to collect**: Select the relevant checkbox for *Metrics* or *Topology*.
   - **Specify whether you'd like to install the integration**: Select between the *Cluster running Cloud Pak for AIOps* or *Self-hosted with network visability to New Relic* radio buttons.
   - **Select Cloud Pak for AIOps integration**: You can leave this at the default *Use the integration that is recommended by Cloud Pak for AIOps*, or uncheck this and select either *Metrics* or *Topology*.

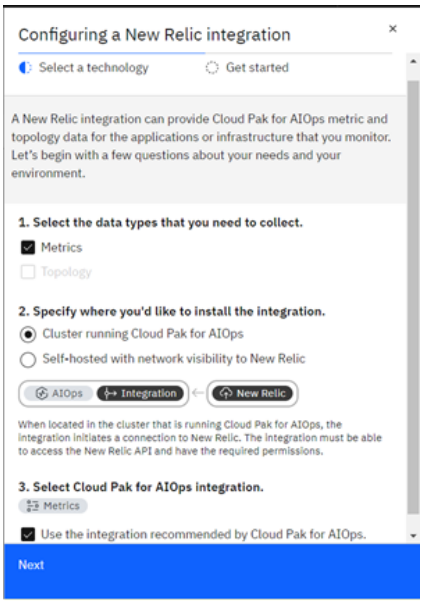
   

   Figure. Configure New Relic integration

   The below steps presuppose you select *Metrics* but if you do select *Topology* then you will be brought to a *New observer job* page. The New Relic observer only supports a load job. When a New Relic job is run, it will gather and read New Relic Infrastructure data from New Relic Insights using New Relic APIs. As a query job, it can be used to load baseline topology data and consequently synchronize it. For more information, see Configuring a New Relic observer job.
6. Then click **Next**.

   You see an overview, further direction such as *what you need to have ready* and also *what algorithms analyze New Relic data*.

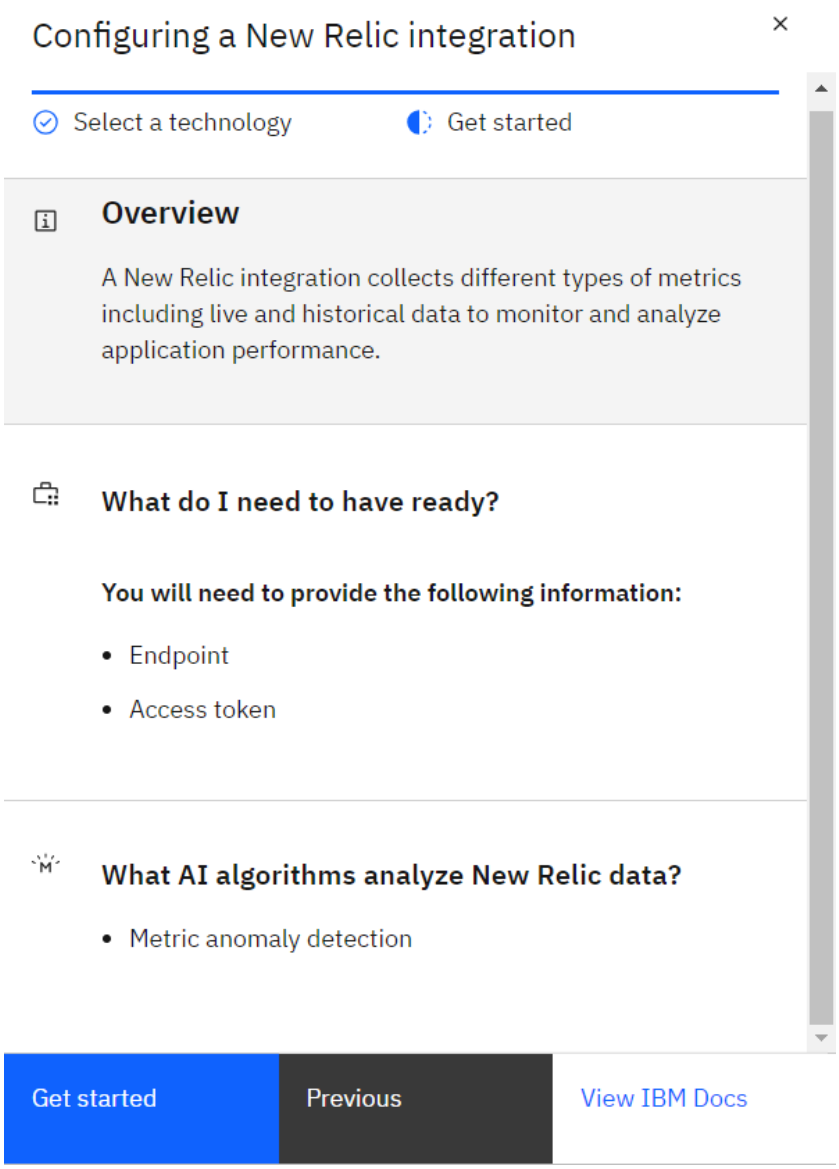
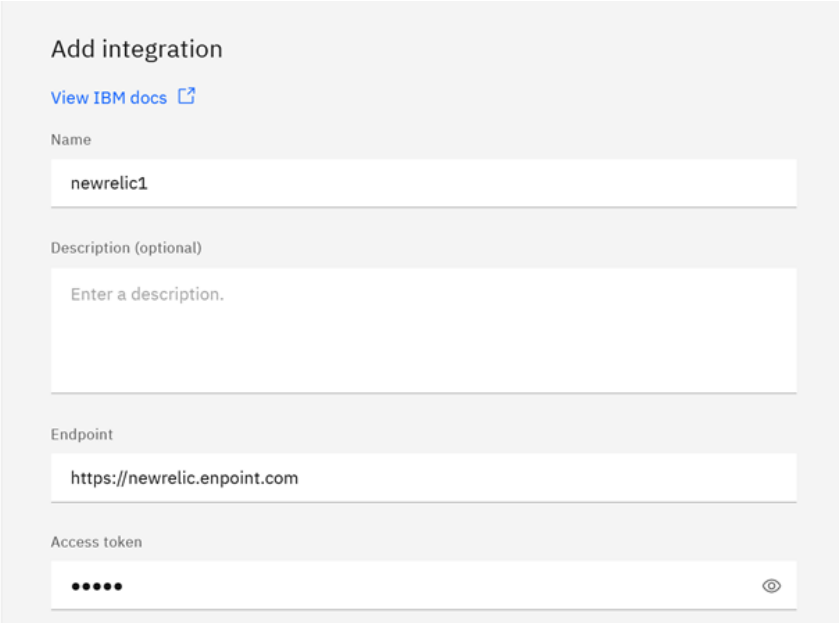


Figure. Configure New Relic integration - get started

7. Click **Get Started**.
8. On the *Add integration* page, define the general integration details:
    - **Name**: Enter the display name of your integration.
    - **Description (optional)**: Type in a description for the integration.
    - **Endpoint**: Enter an endpoint for the integration. For example `https://api.newrelic.com`.
    - **Access token**: Enter an access token for the integration. You can create or obtain the access token from the *API Keys* section in your New Relic account.

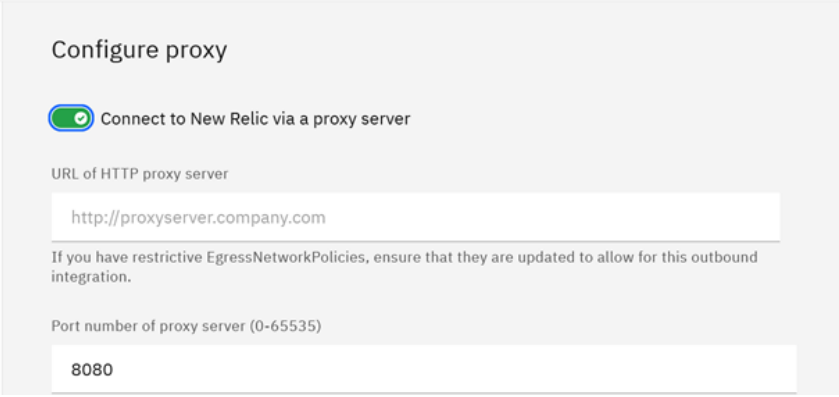


- **Deployment options**: Select one of the options, **Local** or **Remote**. *Local* deploys the integration in the same cluster where IBM Cloud Pak for AIOps is installed. Alternatively, *Remote* deploys it in your chosen location.

Note: If you select *Remote*, after adding the integration, you are provided with a bootstrap command to finish the deployment.

Important: Podman must be installed to execute the remote deployment script with a recommended minimum version of 4.4.1. Read and write access are required for the folder where the remote deployment script will be executed and the script also must have execution permissions in order to be run. No more than one remote deployment script should be executed per host.

9. To verify the SSL/TLS configuration:
    1. Set the **Verify the SSL/TLS certificate for the REST service** toggle to *on*.
    2. Either enter a CA certificate used to verify the SSL/TLS integration to the REST service or click **Fetch SSL certificate** to fetch the certificate automatically.
10. Click **Test Connection**. If successful, a Test succeeded message is displayed.
11. Click **Next**.
12. On the *Configure Proxy* (Optional) page complete the following fields:
    - **Connect to New Relic via a proxy server**: Set the toggle button to *on*.
    - **URL of HTTP proxy server**: Enter a URL for the HTTP proxy server.
    - **Port number of proxy server**: Enter a port. The default is 8080.



Note: Only unauthenticated HTTP proxy is supported.

13. Click **Next**.
14. On the *Collect metric data* page enter your preferences to define how you want to collect metric data:
    - **Enable data collection**: Set *Enable data collection* to on. However, if you disable data collection, a new integration is created, but metrics are not collected.
    - **Select application**: Select one or more applications from the list to see their associated metrics in the filter table. The filter table displays all the associated metrics of the selected applications.

- **Search and select the required metrics from the filter table**: Search through all the metrics and select the metrics from the table. By default, all metrics are selected. To remove a selected metric, search to locate it and clear its corresponding checkbox. You can also click **Cancel** to remove all selected metrics or **Select all** to reselect them.
- **Aggregation interval (minutes)**: Enter a value in minutes. The value represents the time interval for live or historical data collection. The default value is 5 minutes.

  Note: If aggregation interval is less than 60 minutes, IBM Cloud Pak® for AIOps collects only a maximum of past 8 days of historical data. This is because the integration aggregates data in hourly periods, instead of the time you enter, when the data request time range is older than 8 days. For more information about aggregation interval, see the New Relic documentation.
- Select to collect *historical* or *live* data.

  Note: If you choose to collect only live data, you cannot collect historical data after the live data collection starts.
- **Live data collection**: To collect live data, complete the following steps:
  - **Only live data once the integration is established**: Select the checkbox.
  - **Scheduler run time (minutes)**: Enter a value in minutes. The value represents the frequency at which data is collected for live metrics. The default value is 5 minutes.
- **Historical data collection**: To collect historical data, complete the following steps:
  - **Start date and time**: Select the start date from the calendar and enter the time in the *hh:mm* format (hours and minutes).

    Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.
  - **Time zone**: Select your time zone from the dropdown list.
  - **End date and time**: Click **Add end date** and select a date from the calendar. Also, enter the time in the *hh:mm* format (hours and minutes). If you want to remove an end date, click **Remove end date**.

    Note: If you do not set an end date, IBM Cloud Pak for AIOps collects historical data from the specified start date and continues with live data collection. You cannot collect historical data after the live data collection starts.
  - **Scheduler run time (minutes)**: Enter a value in minutes. The value represents the frequency at which data is collected for live metrics. The default value is 5 minutes.

    Note: Scheduler run time is only applicable to live mode. You must configure the scheduler run time only if you do not add an end date as the integration will continue to run in live mode after historical mode.

15. Click **Done**.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

Enabling and disabling New Relic integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **New Relic** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *Collect metric data* section. Set **Enable data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting an integration, see Deleting New Relic integrations.

Editing New Relic integrations

After you create your integration, you can edit it. To edit an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **New Relic** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the New Relic integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration as required. Click **Save** when you are done editing.

You edited your integration.

Deleting New Relic integrations

If you no longer need your New Relic integration and want to delete it entirely, you can delete the integration from the console.

To delete an integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **New Relic** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the New Relic integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating observer jobs

Observers for Instana and ServiceNow are configured as part of the standard integration process on the Integrations page in the UI. These observers can collect data from different types of data sources. Additionally, other topology observers are accessible on the Integrations page in the UI, and link to the Topology Manager UI. To run other observer jobs, you can configure and schedule the jobs through the Topology Manager UI.

Observers provide topology data for integrations, which can help with understanding how resources are connected to each other. In addition, they provide probable cause information for events that are published to integrated collaboration tools like Slack and Microsoft Teams.

For more information about working with observer jobs, see Defining observer jobs.

Creating PagerDuty integrations

A PagerDuty integration provides event data that is analyzed and processed into alerts. An incident is generated, based on an alert severity, and published to your incident viewer interface with insights into potential issues that impact operations.

Contextual alerts can be grouped into an incident by using temporal, topological, and scope correlation methods. For more information, see About events, alerts, and incidents.

The source attribute in the event is used to get the entity information. The entity information is used for the application resolution and the event grouping. For more information, see Application incidents, and Managing incidents and alerts.

For more information about working with PagerDuty integrations, see the following sections:

- Creating PagerDuty integrations
- Enabling PagerDuty integrations
- Editing PagerDuty integrations
- Deleting PagerDuty integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating PagerDuty integrations

To create a PagerDuty integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **PagerDuty** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.

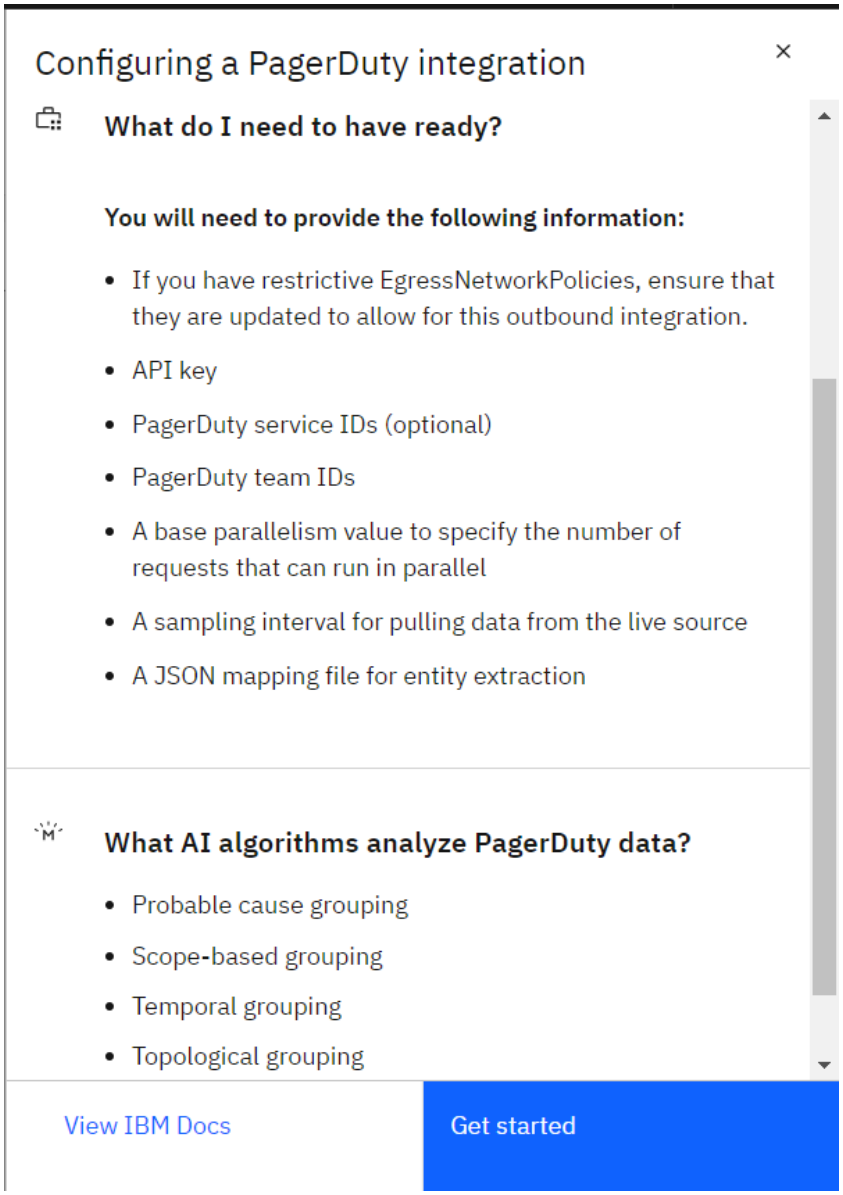


Figure. Configure Pagerduty integration

6. On the *Add integration* page, define the general integration details:

   - **Name**: The display name of your integration.
   - **Description**: An optional description for the integration.
   - **API key**: The integration token for your PagerDuty instance.
   - **Service IDs**: An optional comma-separated list of PagerDuty service IDs. The service ID can be obtained from the PagerDuty URL. If you view a specific service, the PagerDuty URL might be something like the following URL: https://xyz.pagerduty.com/service-directory/P4FY6G2 For this example, the service ID could be `P4FY6G2`.

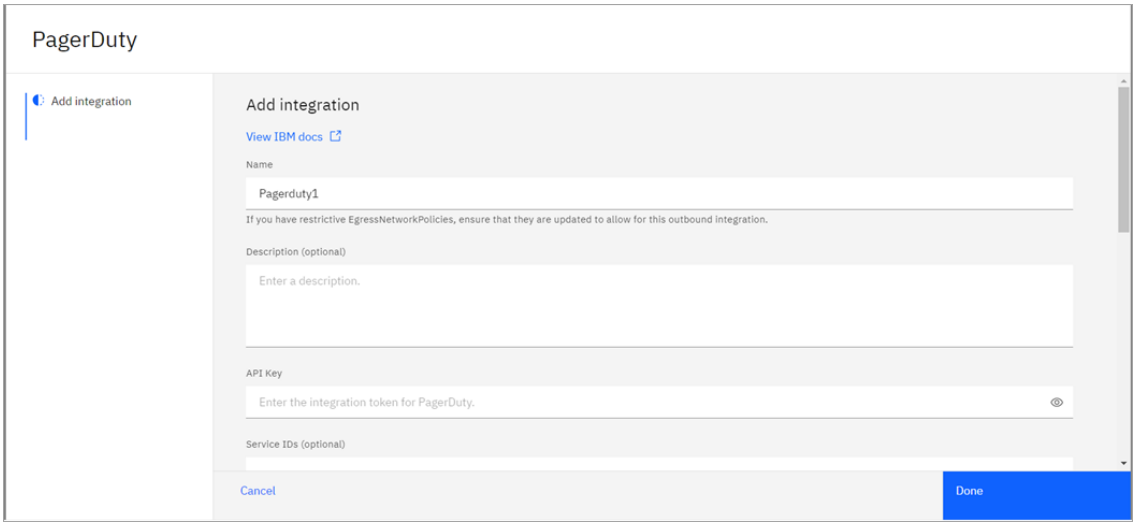


Figure. Create Pagerduty integration

- **Team IDs**: An optional comma-separated list of PagerDuty team IDs. If you view a specific team, the PagerDuty URL might be something like the following URL: https://ibm.pagerduty.com/teams/PV429MG In this example, the team ID is `PV429MG`.
- **Base parallelism**: It is recommended to use a higher value than 1 so that you can process data in parallel. In a small environment the available flinks slots is 16 and for a large environment, the maximum available slots is 32.
- **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is `60`.
- **Data flow**: Set the toggle to 'On' to enable event data from PagerDuty to flow to IBM Cloud Pak for AIOps. For more information about enabling your integration, see Enabling PagerDuty integrations.

  Note: When the `body.cef\_details` attribute is not present in the raw data from PagerDuty, there is no way to extract the entity and assign it to the `resource` attribute in the event input and the event cannot be associated with an application.

You can test your integration by clicking **Test connection**.

Note: PagerDuty mappings are the default mappings for IBM Cloud Pak for AIOps's standard fields for events.

7. Click **Done**.

You created a PagerDuty integration in your instance.

```
![Pagerduty integration](../images/int_pagerduty_running.png){: caption="Figure. Pagerduty integration running" caption-side="bottom"}
```

To create more integrations, see Configuring Integrations.

For more information about working with the insights provided by your integrations, see Managing incidents and alerts.

Enabling and disabling PagerDuty integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **PagerDuty** integration type.
4. Click the integration that you want to enable or disable.
5. Set **Data collection** to **On** or **Off** to enable or disable log data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting a integration, see Deleting PagerDuty integrations.

Editing PagerDuty integrations

After you create your integration, your can edit the integration. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **PagerDuty** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the PagerDuty integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.

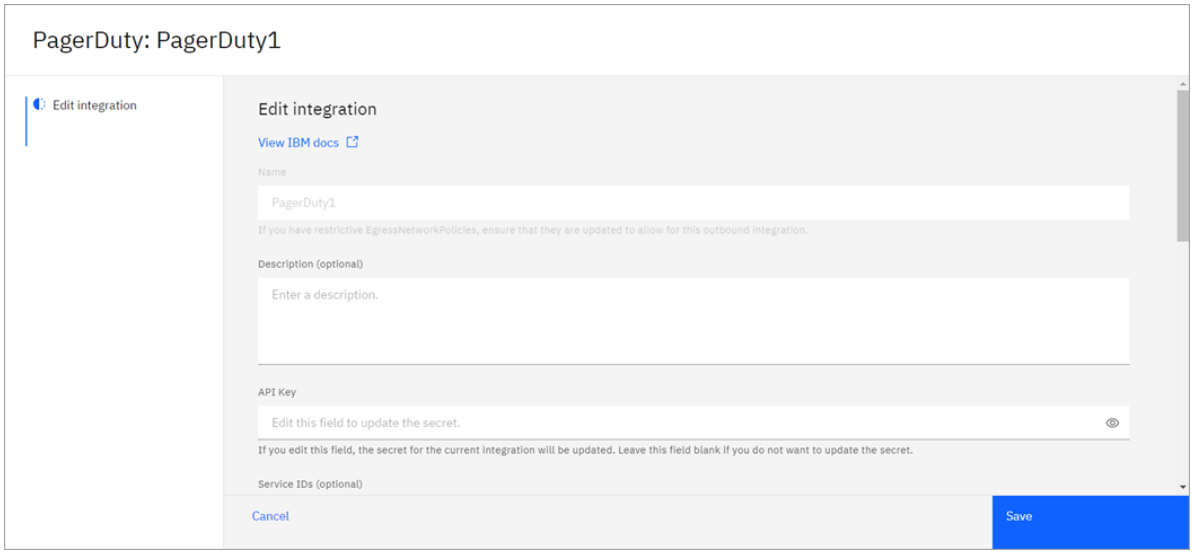


Figure. Edit Pagerduty integration

5. Edit your integration. Click **Next** to go through the integration configurationn pages. Click **Save** when you are done editing.

You edited your integration. If you did not previously enabled or disabled your application, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling PagerDuty integrations. For more information about deleting a integration, see Deleting PagerDuty integrations.

Deleting PagerDuty integrations

If you no longer need your PagerDuty integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable data collection before you delete your integration. For more information about disabling data collection, see Enabling and disabling PagerDuty integrations.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **PagerDuty** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the PagerDuty integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating a ServiceNow integration

A ServiceNow integration allows users to send Cloud Pak for AIOps incidents and alerts to ServiceNow. Optionally, historical and live ServiceNow data can be collected by Cloud Pak for AIOps for more advanced capabilities such as: change request assessments, similar ticket functionality, topology, and bi-directional updates to Cloud Pak for AIOps incidents and alerts.

The ServiceNow integration triggers the following alerts:

- Instance is hibernating
- ServiceNow credentials unauthorized
- Instance does not exist or is deleted
- IBM Cloud Pak for AIOps ServiceNow integration is not installed on ServiceNow

For more information about alerts, see About events, alerts, and incidents.

About this task

Similar incidents, such as from a ServiceNow integration, are called similar tickets. Similar tickets are grouped with other tickets from other integrations as input to IBM Cloud Pak for AIOps AI algorithms.

If your connected ServiceNow application is version 4.4.0 or later, IBM Cloud Pak® for AIOps incidents and alerts are kept up to date in ServiceNow. Specifically the following fields:

- incidents: short description, description, priority, and state
- alerts: alert summary, state, severity, event count, and last occurrence

Prerequisites

- Before you create a ServiceNow integration, you must create users and configure ServiceNow. For more information about configuring ServiceNow, see Configuring the ServiceNow App for IBM Cloud Pak for AIOps.
- Your ServiceNow application must be version 4.4 to connect with IBM Cloud Pak® for AIOps version 4.8.0. This integration is not supported for earlier versions of the ServiceNow application.
- ServiceNow instances with self-signed certificates are not supported for connecting with IBM Cloud Pak for AIOps.
- You can have only one ServiceNow integration per instance.

For more information about working with a ServiceNow integration, see the following sections:

- Creating a ServiceNow integration
- Enabling a ServiceNow integration
- Editing a ServiceNow integration
- Deleting a ServiceNow integration
- Troubleshooting a ServiceNow integration
- Known issue with a ServiceNow integration

Creating a ServiceNow integration

To create a ServiceNow integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **ServiceNow** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Get started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: The display name of your integration.
   - **Description**: An optional description for the integration.
   - **URL**: The URL of your ServiceNow instance. For example `https://myinstance.service-now.com`.

     Important: If you have any restrictive EgressNetworkPolicies in place, ensure that they are updated to allow for this outbound integration. For more information, see Postinstallation tasks.
   - **User ID**: A ServiceNow user with say a `ibm.aiops` user ID and an `x_ibm_waiops.admin` role.

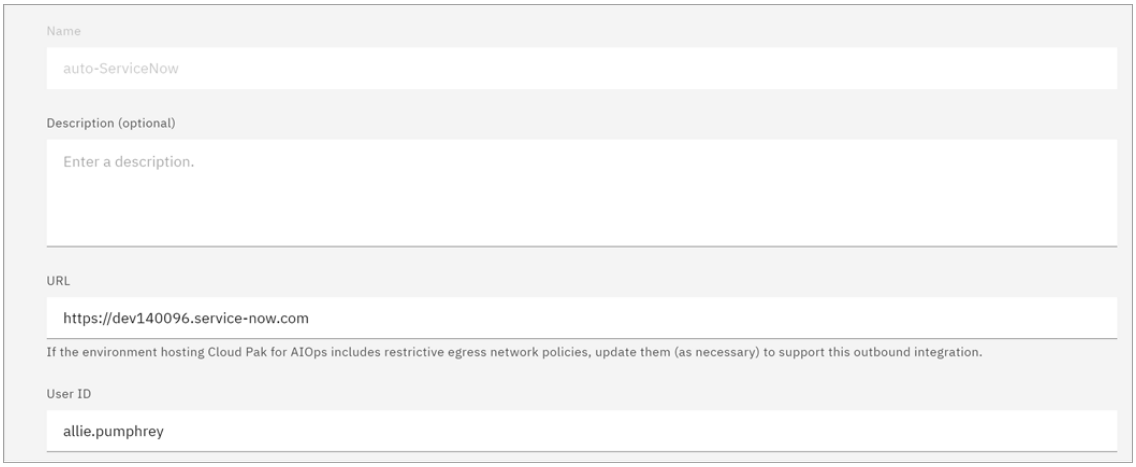


Figure. Enable Service Now integration

   Important: You must create a new, unique user in ServiceNow specifically for your ServiceNow integration before you create your integration. Make sure that you are using the correct user role, the *x_ibm_waiops.admin*, for your integration. You must also make sure that the time zone set for this user matches your system time zone in ServiceNow. For more information about ServiceNow users, see Managing roles.
   - **Password**: Your ServiceNow password for `ibm.aiops` user.
   - Select the orchestration or deployment type as **Local** or **Remote**.

     Note: If you select 'Remote' you will be provided with a bootstrap command to finish the orchestration.

Click **Test Connection** to validate the ServiceNow instance and credentials. If the test fails, verify the ServiceNow instance is stable and that the credentials are correct.

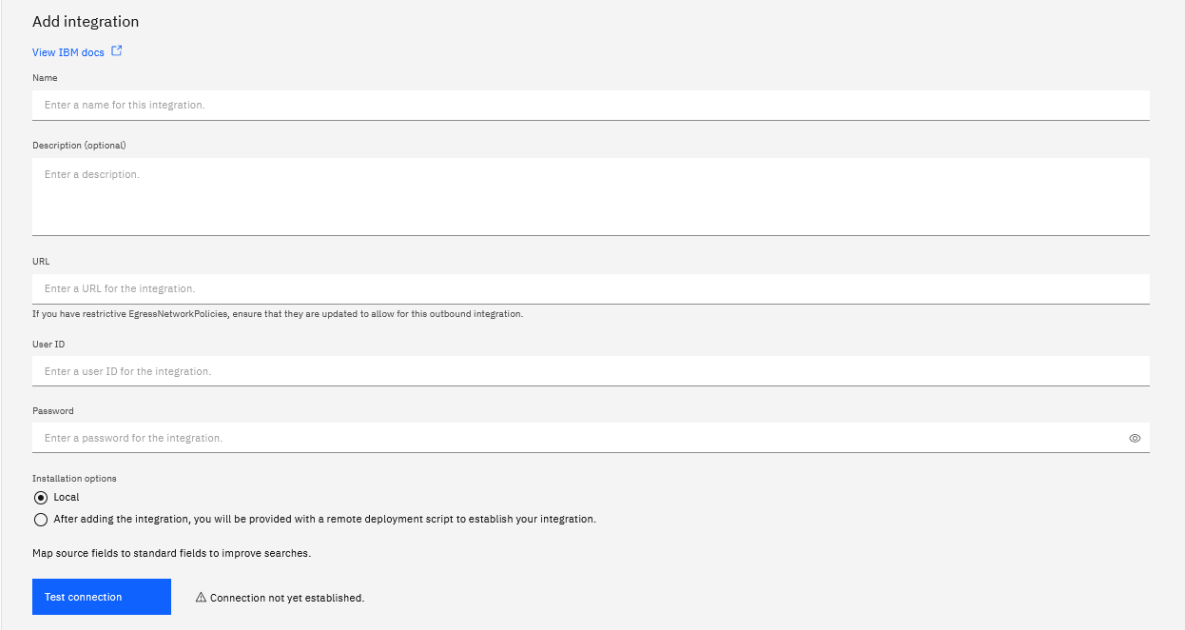


Figure. Test connection button

7. Click **Next**.
8. (Optional) Enter *Field mapping* information.

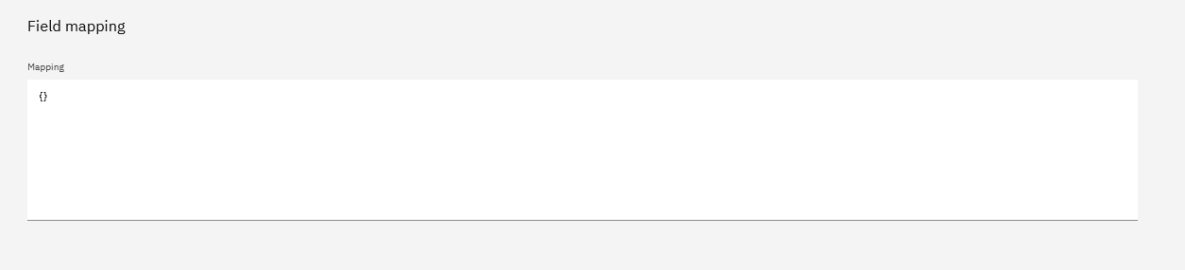


Figure. Field mapping setting

With this field, you can map custom data fields to default fields in ServiceNow. The **change_request** and **incident** tables are queried from ServiceNow. These fields are used for AI training. These tables can be modified to include additional columns.

- The following **change_request** fields are extracted:

  ```
  "u_environment", "u_environment_id", "cmdb_ci", "short_description", "u_purpose", "description",
  "backout_plan", "close_notes", "state", "close_category", "sys_created_by", "assigned_to", "sys_domain",
  "category", "business_service", "priority", "approval", "type", "contact_type", "production_system", "impact",
  "reason", "justification", "start_date", "end_date", "work_start", "work_end", "calendar_duration",
  "close_code","delivery_task", "implementation_plan", "test_plan", "work_notes_list"
  ```

- The following **incident** fields are extracted:

  ```
  "short_description", "description", "u_caused_by_change", "caused_by", "close_code", "u_detection_source",
  "close_notes", "u_environment","cmdb_ci", "comments_and_work_notes", "impact", "problem_id", "priority",
  "severity", "state", "resolved_at", "closed_at", "opened_at", "parent", "made_sla", "watch_list",
  "upon_reject","sys_updated_on", "child_incidents", "hold_reason", "approval_history", "resolved_by",
  "sys_updated_by","opened_by", "user_input", "sys_created_on", "sys_domain", "sys_created_by", "knowledge",
  "order","calendar_stc", "delivery_plan", "contract", "active", "business_service", "sys_domain_path",
  "rfc","time_worked", "expected_start", "business_duration", "group_list", "work_end", "caller_id",
  "reopened_time", "approval_set", "subcategory", "correlation_display", "delivery_task", "work_start",
  "assignment_group", "additional_assignee_list", "business_stc", "calendar_duration", "notify",
  "service_offering", "sys_class_name", "closed_by", "follow_up", "parent_incident", "contact_type",
  "reopened_by", "incident_state", "urgency", "company", "reassignment_count", "activity_due", "assigned_to",
  "comments", "approval", "sla_due", "due_date", "sys_mod_count", "reopen_count", "sys_tags", "escalation",
  "upon_approval", "correlation_id", "location", "category", "work_notes_list"
  ```

Example: If you add a custom column (named **my_custom_field1_name**) into the table, the prefix **u_** is added so the field must be queried as **u_my_custom_field1_name**. If this field contains useful description information for training, you might want the existing **description** field to be updated for both change requests and incidents to use this data. To support this mapping, you need to add the following JSON to the *Field mapping*:

```
{
  "codec": "servicenow",
  "description": "u_my_custom_field1_name"
}
```

With this mapping, the **description** value is replaced with the value from **u_my_custom_field1_name**. For example, assume the default **description** field has the value **Server down for 2 hours due to an upgrade** and the **u_my_custom_field1_name** field had the value **Impact: server down, Duration: 2 hours, Cause: server upgrade**. With the above mapping in place, the **description** now has the value **Impact: server down, Duration: 2 hours, Cause: server upgrade**. This **description** value is also now used for AI training. If the mapping is not in place, the **u_my_custom_field1_name** value is not used for AI training as it is not examined through the integration.

Adding the field mapping can give you flexibility to add your own custom fields into AI training. Also, if an existing field should not be used for training, you can add a new custom column with a default value to use for training instead of the existing field value.

Notes:

- The field mapping is applied to both historic and live data. Once the mapping is set in the integration and training is completed, the mapping should not be changed. Otherwise, if new change requests or incidents are queried, the mapping is not applied properly, which can cause results to be less accurate.
- Currently, the field mapping applies to all tables. If a custom field does not exist in a table, a workaround is to create that custom field with default values to ensure that the mapping does not fail. Otherwise, the data is not be retrieved properly for that table.

9. Click **Next**.
10. (Optional) Select whether to *Collect topology data*, and if selected, set the options for data collection:
    - **Collect inventory and topology data**: Set to **On** for inventory and topology data collection. You can also set advanced options.
    - **Schedule request**: Set to **On** and this allows you set the **Job Schedule** (optional)
        - **Start date**: The start date of when the job is to run.
        - **Time**: The time to run the observer job.
        - **Time interval (period)**: How frequently to run the job (either by hour, minute, days, or weeks).
        - **Interval**: The duration of time between runs based on the *Time interval (period)*. For example, if you wanted the job to run every 2 hours, set the interval to **2** and the *Time interval (period)* to *hours*. Enter **0** for the *Interval* to run the observer job one time (and manually through the interface otherwise).
    - Set advanced options section:
        - **Number of records to fetch per API calls from a table**: The number of records to fetch per API calls from a table. Default value is **10000**. The value should be between **100** and **10000**.
        - **Entity mapping for the sysClassNames parameter to overwrite the entity type**: To use a user-defined entity type, enter a parameter to overwrite the default **sys_class_name**. Separate the values of the class name and the new entity type name with a colon. For example, **cmdb_ci_linux_server:node**.
        - **ServiceNow tables to be discovered by observer**: The supported tables to be discovered are **sys_user** and tables that are either system or user created children of **cmdb_ci**.
        - **Trust all certificates by bypassing certificate verification**: If **On**, the observer allows the integration to the target environment without verification. The default value is *On*.
            - **Options to exclude entity edges**: Select an exclude option to exclude resource edges. You can select the following options. The default value is **none**.
                - **none**
                - **uniqueID**
                - **entityType**
            - **List of entities to exclude edge modelling**: The listed entities will be not have any edge connectivity to any entities.
            - **Use an 'include' or 'exclude' filter in API requests**: Specify that the API filter should use include or exclude in every ServiceNow API call. Enter the filter attributes in the **filter_resources** parameter.
            - **Filter resources based on defined JSON attributes**: To use a filter in API calls to retrieve resources, enter the JSON **key:value** attribute. Use the format **key=value** where value can be more than one, for example **key=value1,value2,value3**. For more than one key, use **|** as a delimiter, for example, **key=value1|key=value1,value2,value3**.
            - **Number of connection retries**: Set the number of times a connection should try again. The default value is **2**.
            - **Delay before a connection retry (milliseconds)**: Specify the period of time in milliseconds to delay before a connection tries again.
            - **Maximum number of tables per cmdb_ci_rel api batch call**: Specify this configurable parameter to split one long call into multiple small calls. This prevents the cmdb_ci_rel (relationship) query from being too long. The default and maximum value is **10**.
            - **Integration timeout**: The amount of time before a connection times out in milliseconds. The default value is **5000** milliseconds.
            - **Read timeout**: The amount of time before a read operation times out in milliseconds. The default value is **5000** milliseconds.
            - **Proxy Host**: The proxy on the network that AIOps is deployed on for making external requests to any target.
            - **Proxy Port**: The proxy port on the network that AIOps is deployed on. The default value is **8080**.
            - **Proxy Username**: Specify the username for connecting to the proxy.
            - **Proxy Password**: Specify the password for connecting to the proxy.

- **Proxy Secure**: Specify whether the proxy server is secure. The default value is `false`
- **Access Scope**: An optional CSV String that lists the values which can be used to provide a scope for the resources. These values can be used to aid the mapping of alerts to resources when resources that are in different scopes share the same matchTokens. Examples of scope can be locations, project names, or namespaces.
- **Generate debug support file**: Set to **On** to generate the debug support file. The default is **Off**.

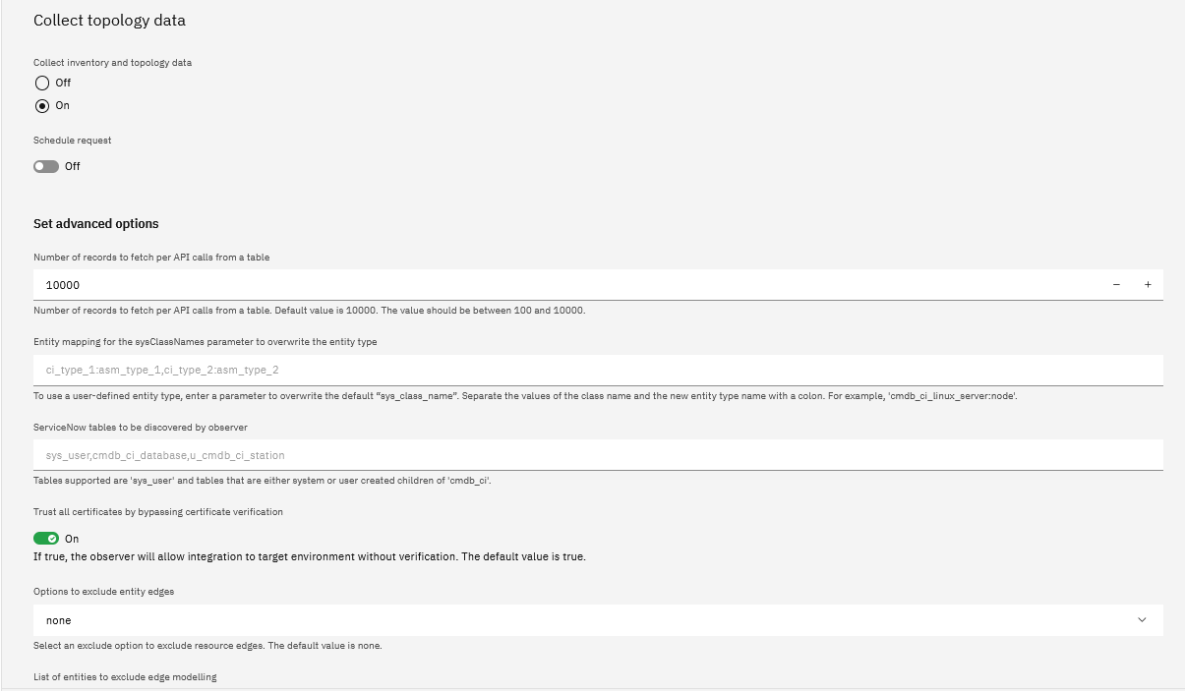


Figure. Topology data collection settings

For information about REST APIs and ServiceNow, see Configuring ServiceNow Observer jobs and Topology REST Observer APIs.

11. Click **Next**.
12. On the *Configure tickets proxy and custom certificate* (optional) page complete the following fields:

    To verify the SSL/TLS configuration:

    1. Set the **Verify the SSL/TLS certificate for the REST service** toggle to *on*.
    2. Either enter a CA certificate used to verify the SSL/TLS integration to the REST service or click **Fetch SSL certificate** to fetch the certificate automatically.

    To connect to ServiceNow via a proxy server:

    1. Set the **Connect to ServiceNow via a proxy server** toggle button to *on*.
    2. Enter a URL for the HTTP proxy server in the **URL of HTTP proxy server** field.
    3. Enter a port number for the proxy server in the **Port number of proxy server** field. The default is 8080.

    Note: Only unauthenticated HTTP proxy is supported.

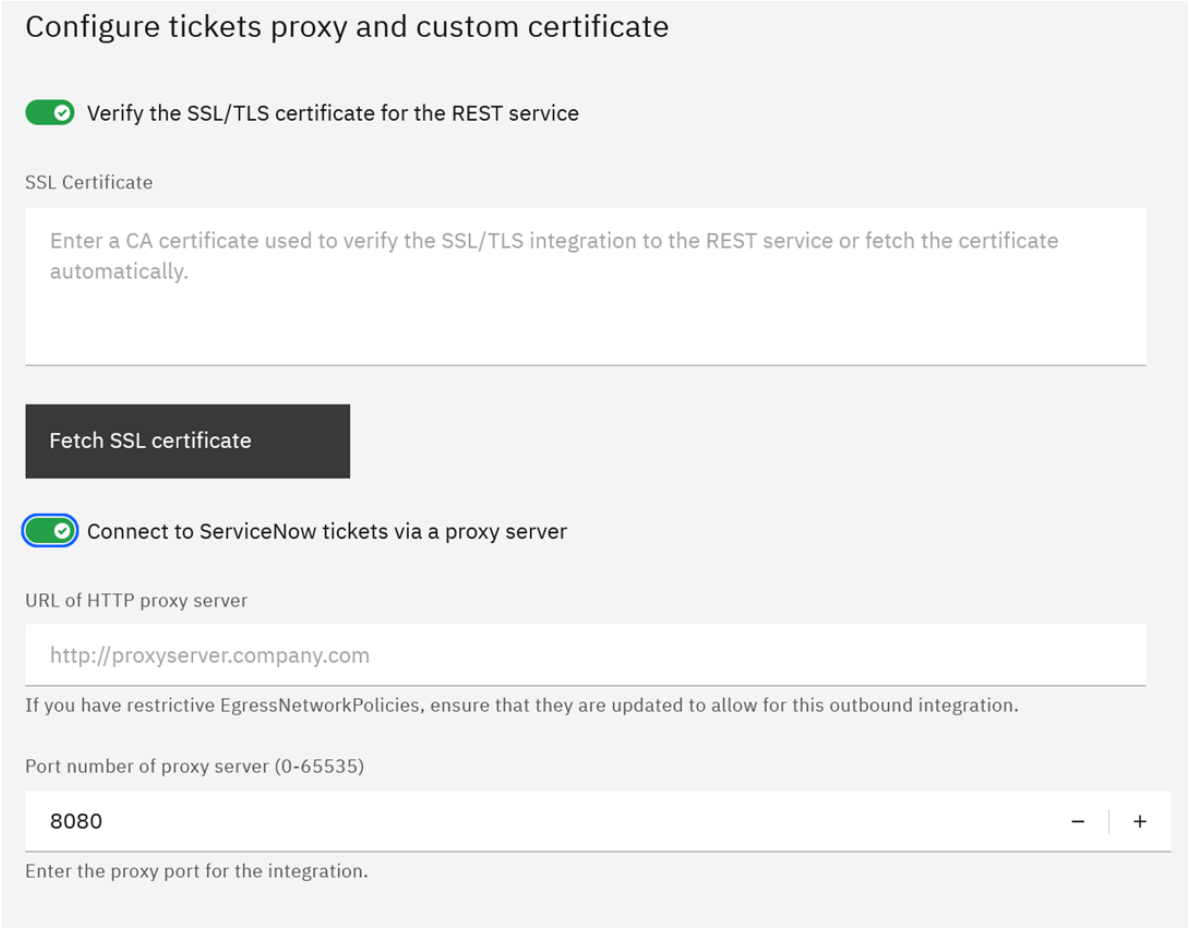


Figure. Configure proxy and custom certificate

13. Click **Next**.
14. (Optional) Enter the *Collect and synchronize operations data* information.

- **Live data - Train and use Change risk and Similar tickets algorithms; synchronize alert and incident data**

  Select the Live data collection mode to enable the following capabilities:

  - Provide risk assessments on change requests (after model is trained and deployed).
  - Update alerts in Cloud Pak for AIOps when corresponding incident-based alerts are updated in ServiceNow.
  - Update incidents in Cloud Pak for AIOps when corresponding incidents are updated in ServiceNow.
  - Collect resolved ServiceNow incidents for Similar tickets training. This option is required for similar tickets functionality.

  Note: If you are interested in utilizing change risk or similar tickets AI algorithms, you must first set **Mode** to **Historical data for ticket data collection** to collect a minimum amount of data, and then run AI training on that data. For more information, see Planning data loading and training.

  - **Configuring polling intervals for live data collection**: Live data collection is done at configurable intervals ranging anywhere from every 10 seconds to once a day. For risk assessments and similar tickets polling, the default is 15 minutes. For alert and incident polling, the default is 1 minute. The default values serve as a starting point but polling can be done at longer or shorter intervals depending on user preference.

    Important: Updates made to Cloud Pak for AIOps data (incidents and alerts) in ServiceNow by the integration user (ibm.aiops) will not be synchronized in Cloud Pak for AIOps. A different ServiceNow user must be used when making updates to Cloud Pak for AIOps data in ServiceNow.

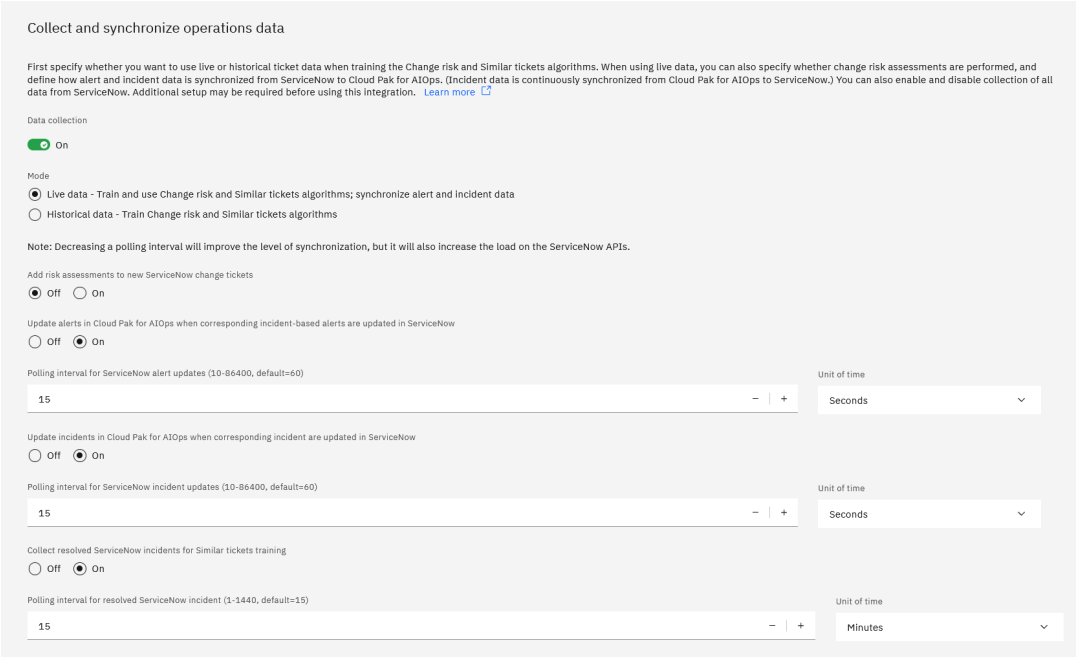


Figure. Live data

- **Historical data - Train Change risk and Similar tickets algorithms**

  To enable the change risk and similar ticket algorithms, a single set of training data is necessary to train the AI model. You complete the 'Start' and 'End' date fields.

  - **Ticket types**: Specify the types of ServiceNow tickets that you want to pull in to the system. By default, change request and incident tickets are included and data is queried from those ServiceNow tables. You can edit this list to remove any types you don't want to sample.

    Important: Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

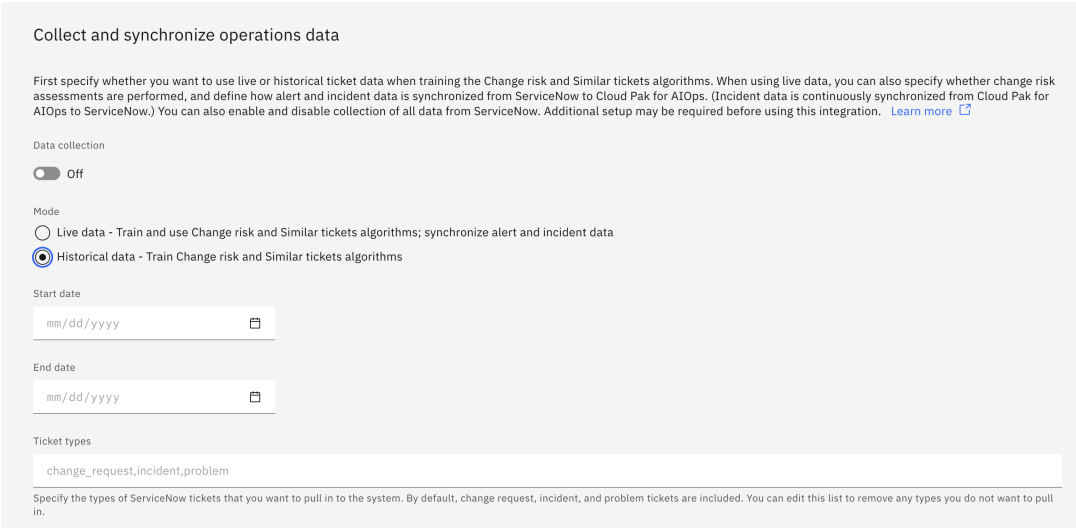


Figure. Historical data

15. Click **Done**. You have created a *ServiceNow* integration in your instance
16. To use the integration that you created, configure it in a policy. For more information, see Configure a policy.

After the integration creation is complete, and if you selected the *Remote* option for integration deployment, you can download the *bootstrap script* by selecting the created integration, and clicking **download** from the options menu (three vertical dots). If a script is regenerated, it invalidates the credentials of previously downloaded scripts for that integration.

Note: You can either copy the script or download as an sh file to run it on a remote cluster.

Prerequisites before running the bootstrap script on remote cluster:

- Podman needs to be installed on the remote cluster.
- Log in to the *Image registry* for the script to pull in the image that uses the Podman login command.

After you create your integration, you must enable the data collection to connect your integration with the AI of IBM Cloud Pak for AIOps. For more information about enabling your integration, see Enabling a ServiceNow integration. To verify that your job completed successfully, view the schedule status of your

ServiceNow integration in the *Schedule integrations* tab.

To create more integrations (such as a ChatOps integration), see Configuring Integrations.

For more information about working with the insights provided by your integrations, see ChatOps insight management. For more information about change risk from ServiceNow, see Change risk.

Training a Change Risk model

For more information about training a change risk model, see Training a Change Risk model.

Enabling and disabling a ServiceNow integration

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. If you selected *Live data for initial AI training* when you created your integration, you must disable the integration before AI model training. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **ServiceNow** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *Collect ticket data* section. Set **Data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for an integration does not delete the integration.

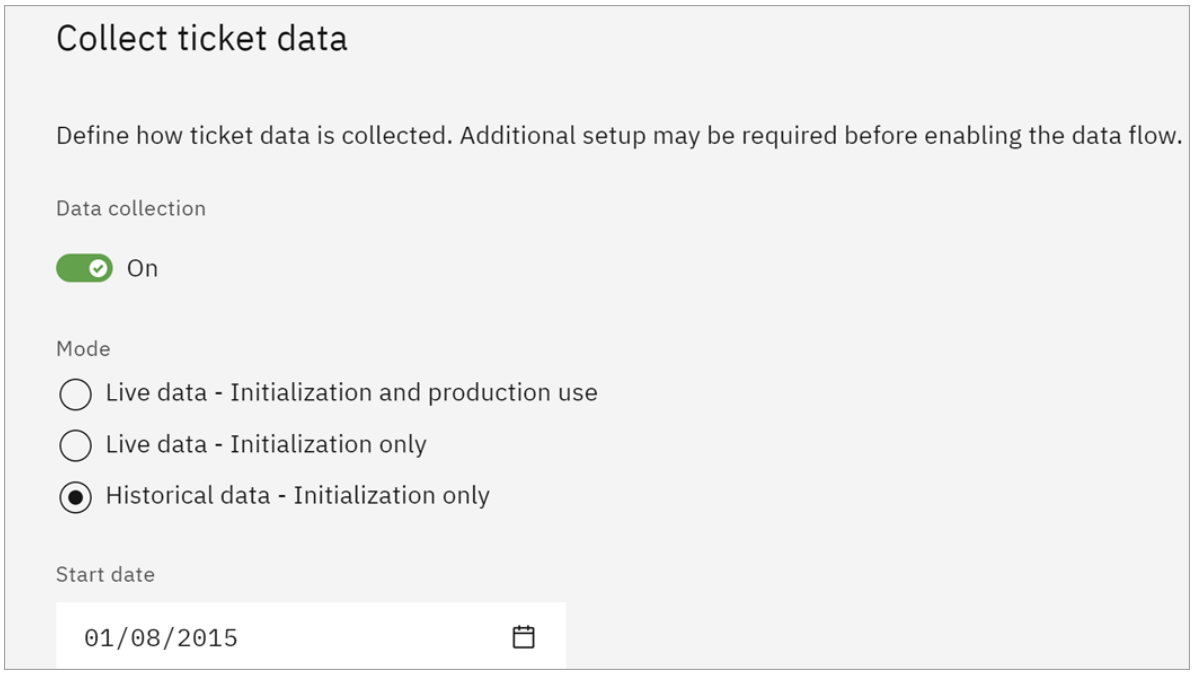


Figure. Enable Service Now integration

You enabled or disabled your integration. For more information about deleting a integration, see Deleting a ServiceNow integration.

Editing a ServiceNow integration

After you create your integration, your can edit the integration. For example, if you specified *Historical data for initial AI training* but you now want your integration to sample live data for continuous monitoring, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **ServiceNow** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the ServiceNow integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration. Click **Next** to go through the integration configurationn pages. Click **Save** when you are done editing.

You edited your integration. If you did not enable ticket data collection when creating your integration, you can enable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling a ServiceNow integration. For more information about deleting a integration, see Deleting a ServiceNow integration.

Note: After you upgrade, you might find that the **Next run** field in the **Scheduled integrations** tab is showing the last run time of the scheduled job, instead of the next run time. If this happens, the next run time should be regarded as the last run time.

Deleting a ServiceNow integration

If you no longer need your ServiceNow integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable ticket data collection before you delete your ServiceNow integration. For more information about disabling ticket data collection, see Enabling and disabling a ServiceNow integration.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **ServiceNow** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the ServiceNow integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Note: Alerts that are generated from the ServiceNow integration are automatically cleared when the integration is deleted.

Mapping similar ticket data from ServiceNow

If you need to use historical incidents from ServiceNow as similar ticket data, you need to map the data to an expected format and insert the data into Elasticsearch.

Historical incidents are required to first be inserted into Elasticsearch for similar ticket data. The data format consists of one incident record per line, in JSON format, with the original raw data. Raw data must be normalized to the IBM Cloud Pak for AIOps target format. This normalization includes mapping fields, converting date-time formatting, and more. Incidents are then inserted in batch. When the insertion is done, IBM Cloud Pak for AIOps tries a simple search to make sure everything runs as expected.

For more information about training similar tickets models in IBM Cloud Pak for AIOps, see Setting up training for similar tickets.

Raw similar ticket data

IBM Cloud Pak for AIOps ingests incidents from standard or custom data sources, for the incidents to be analyzed for similarities and action recommendations. Offline historical raw incidents are first uploaded to the Elastic database to extract indexes and set the stage for finding incidents that are similar to incoming live incident data.

You do not need to transform raw incident data into a normalized format. For more information about normalization, see Normalized output data schema.

Normalization is automatically done in similar incidents model training step. For example, the following raw JSON incident data comes from ServiceNow:

```
{
    "source": "https://dev82395.service-now.com/incident.do?sysparm_query=number=INC0010062",
    "number": "INC0010066",
    "short_description": "Containers being killed in ts-payment-mongo service. Users may notice issues during payment",
    "description": "",
    "opened_at": "2020-01-08T22:48:23Z",
    "sys_created_on": "2020-01-08T22:48:23Z",
    "sys_updated_on": "2020-01-08T22:48:23Z",
    "resolved_at": "2019-12-11T07:36:00Z",
    "closed_at": "2019-12-11T07:36:00Z",
    "business_duration": "",
    "severity": 3,
    "priority": 5,
    "impact": 3,
    "state": "closed",
    "comments_and_work_notes": "",
    "close_notes": "This issue has been resolved by the DBA team, which SQL queries were updated to reboot proxy agent to
prevent the monitoring notification system from failing.",
    "parent_incident": ""
}
```

In this example, you can extract data from ServiceNow. The source and format of your exported data doesn't matter.

For more information about exporting data from ServiceNow, see Exporting data.

For more information about managing incident data in ServiceNow, see Incident management.

Normalized similar ticket data

The following sample illustrates the result of mapping data from an external source for use with IBM Cloud Pak for AIOps:

```
{
  "application_group_id": "1",
  "application_id": "1",
  "timestamp": 1570221600,
  "utc_timestamp": "2019-10-04T20:40:00.038Z",
  "type": "incident",
  "incident": {
    "incident_id": "fed6b7bf-f741-49d1-ab36-5d04e9a5c128",
    "title": "Users are not able to access Discovery and Conversation in EU",
    "description": "Unable to connect to email using my laptop.",
    "created_at": "2019-09-11T06:39:48.000Z",
    "updated_at": "2019-09-11T07:57:58.000Z",
    "resolved_at": "2019-09-11T07:57:58.000Z",
    "closed_at": "2019-09-11T07:57:58.000Z",
    "started_at": "2019-09-11T07:57:58.000Z",
    "business_duration_ms": 15347,
    "severity": 1,
    "priority": 2,
    "impact": 3,
    "state": "new",
    "source": {
      "source_name": "ServiceNow",
      "source_url":"https://ibm.servicenow.com/INC0000060",
      "source_incident_id": "INC0000060",
      "source_application_id": "tokenizer124"
    },
    "comments": [
      {
        "comment_text":"Rebooted it."
      }
    ],
    "related_incidents":[
      {
        "source_incident_id":"INC0000059",
        "relationship":"parent"
      }
    ],
    "resolution": {
      "rca_id": "RCA00013547",
      "resolution_summary": "The problem was resolved by rebooting the laptop."
    },
    "features": [
    ]
  },
  "meta_features": []
}
```


Description of similar ticket normalized attributes

Required attributes expected as input to the pipeline

Attribute	Description
`title`	Short description or title of the incident.
`description`	Long description of the incident.
`timestamp/utc_timestamp`	Epoch timestamp of the event in the log entry and the `utc_timestamp` formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`created_at`	Coordinated Universal Time at which the incident was created formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`updated_at`	Coordinated Universal Time at which the incident was last updated formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`resolved_at`	Coordinated Universal Time at which the incident was resolved formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`closed_at`	Coordinated Universal Time at which the incident was closed formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`started_at`	Coordinated Universal Time at which the incident started formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`severity`	Severity of the issue (integer)
`priority`	Priority of the issue (integer)
`impact`	Impact of the issue (integer).
`source_name`	Data source in which the incident is stored (for example, ServiceNow).
`source_url`	URL of the incident in the source.
`source_incident_id`	`incident_id` from the source (for example, `NUMBER` in ServiceNow).
`source_application_id`	Application from which the incident came (for example, the configuration item in ServiceNow).
`comments`	Comments or closed notes that talk about the list of actions that are taken, or comments that are added by the user who is attending to the incident.
`related_incidents`	Any related incidents, such as parent incidents, child incidents, or CIE documentation that is attached with the incident.
`resolution`	Any RCA ID or resolution that is provided for the incident.

Optional attributes that might add value

Attribute	Description
`business_duration`	Duration for which the incident was alive without resolution.

Static attributes

Attribute	Description
`application_group_id`	The `application_group_id` to support multitenancy. This value is provided as part of the initial configuration.
`application_id`	Application that IBM Cloud Pak for AIOps is monitoring. This value is provided as part of the initial configuration.

Attribute	Description
`type`	This value is hardcoded to the incident.
`timestamp`	Epoch time of the incident normalization, in seconds.
`utc_timestamp`	Normalization time in the Coordinated Universal Time time zone.
`incident_id`	The unique UUID.
`state`	Current state of the incident.

Attributes analyzed and populated in the pipeline

Attribute	Description
`features`	Placeholder for any features.
`meta_features`	Placeholder for service analysis.

Creating Splunk integrations

A Splunk integration collects log data and different types of metrics, including live and historical data, and sends it to the gRPC server as Cloud events at specified intervals.

Log data is used to establish a baseline of normal behavior and detect anomalies. Metric data helps to monitor your environment and process alerts. Anomalies can be correlated with alerts, and published to your ChatOps interface, to help you determine the cause and resolution of a problem. Splunk Enterprise v8.1 is the version that is supported for the Splunk integration.

To activate the integration, you must have an endpoint and a user ID and password credentials or a token. You can also configure settings for performance by providing log or metric data parameters. The metric anomaly detection and Log anomaly detection AI algorithms analyze the Splunk data.

For more information about working with Splunk integrations, see the following sections:

- Creating Splunk integrations
- Specifying a custom query for the Splunk host
- Enabling Splunk integrations
- Editing Splunk integrations
- Deleting Splunk integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating Splunk integrations

About this task

Before you create a Splunk integration, consider the following details:

- **Load**: To prevent the integration from placing an inordinate load on your data source and potentially impacting your logging operations, the integration connects to only one API with a default data frequency of 60 seconds. This frequency is controlled by the **Sampling rate** setting that is configured when you create the integration.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting when you create the integration.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Procedure

To create a Splunk integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Splunk** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section, or alternatively start typing the integration name in the searchbar and the corresponding tile appears.
5. On the side-panel, review the information and when ready to continue, click **Get Started**.
6. On the *Add integration* page, define the general integration details:
   - **Name**: Set a display name for your integration. Use a name that uniquely identifies this integration.
   - **Description (optional)**: Provide a description for the integration.
   - **Splunk service URL**: Enter the URL for your Splunk instance. For example `https://myinstance.mydomain.com:8089`.

     Note: If you have any restrictive EgressNetworkPolicies in place, ensure that they are updated to allow for this integration.
   - **Authentication type**: Select one of the following options:

- **User ID/Password**: The Splunk instance has a user ID and password as authentication. You must enter both in the integration configuration that appears.
- **Token**: The Splunk instance is authenticated with a temporary token field that appears.

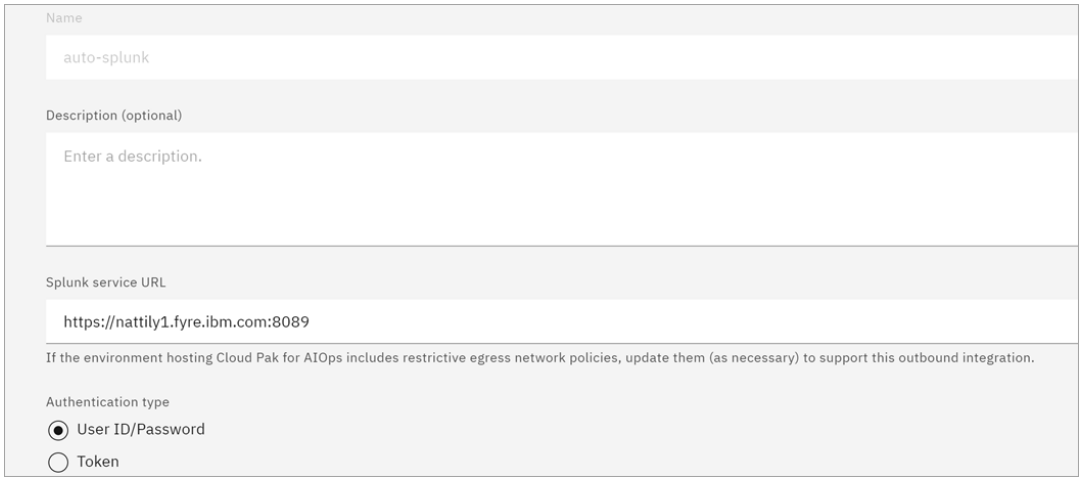


Figure. Create Splunk integration

- **Splunk dashboard URL**: Enter the dashboard URL for your Splunk instance. For example `https://myinstance.mydomain.com:8080`.
- **Certificate (optional)**: Provide a certificate to verify the SSL/TLS connection to the REST service.
- **Deployment options**: Select one of the two options, **Local** or **Remote**. Use **Local** to deploy in the same cluster as IBM Cloud Pak for AIOps console. Select **Remote** to deploy anywhere, such as on a different network region, on SaaS, or on a remote on-premises system (VM, cluster, container).

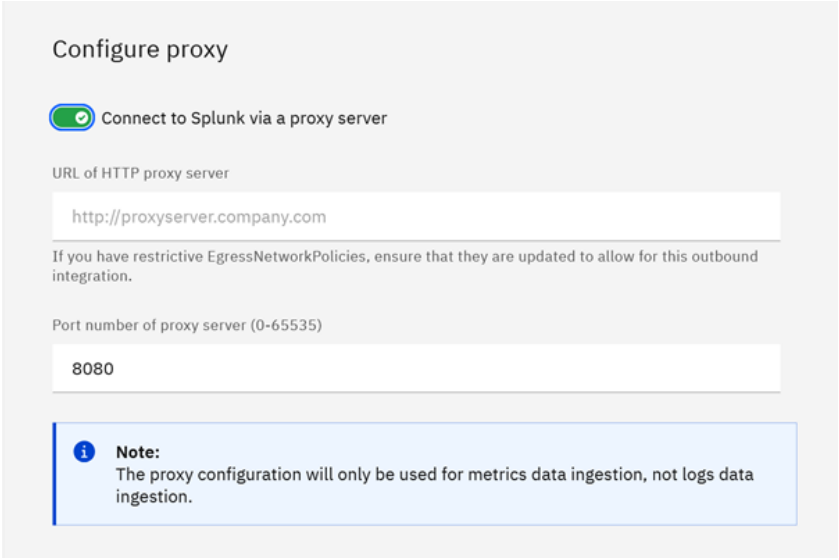
  Note: Remote deployment only supports metric integration. If you need to use the log integration, you must select **Local**. If you select **Remote**, after adding the integration, you are provided with a bootstrap command to finish the deployment.

7. To verify the SSL/TLS configuration:
   1. Set the **Verify the SSL/TLS certificate for the REST service** toggle to *on*.
   2. Either enter a CA certificate used to verify the SSL/TLS integration to the REST service or click **Fetch SSL certificate** to fetch the certificate automatically.



Figure. Test connection

8. Click **Test connection** (required). The test can take some time to complete.
9. Click **Next**.
10. On the *Configure Proxy* (Optional) page complete the following fields:
    - **Connect to Splunk via a proxy server**: Set the toggle button to *on*.
    - **URL of HTTP proxy server**: Enter a URL for the HTTP proxy server. **Note:** The URL for the metrics-only integration should have the protocol in the URL.
    - **Port number of proxy server**: Enter a port. The default is 8080.



Note: Only unauthenticated HTTP proxy is supported.

11. Click **Next**.
12. On the *Collect log data* page, define how you want to collect log data for use in AI training:

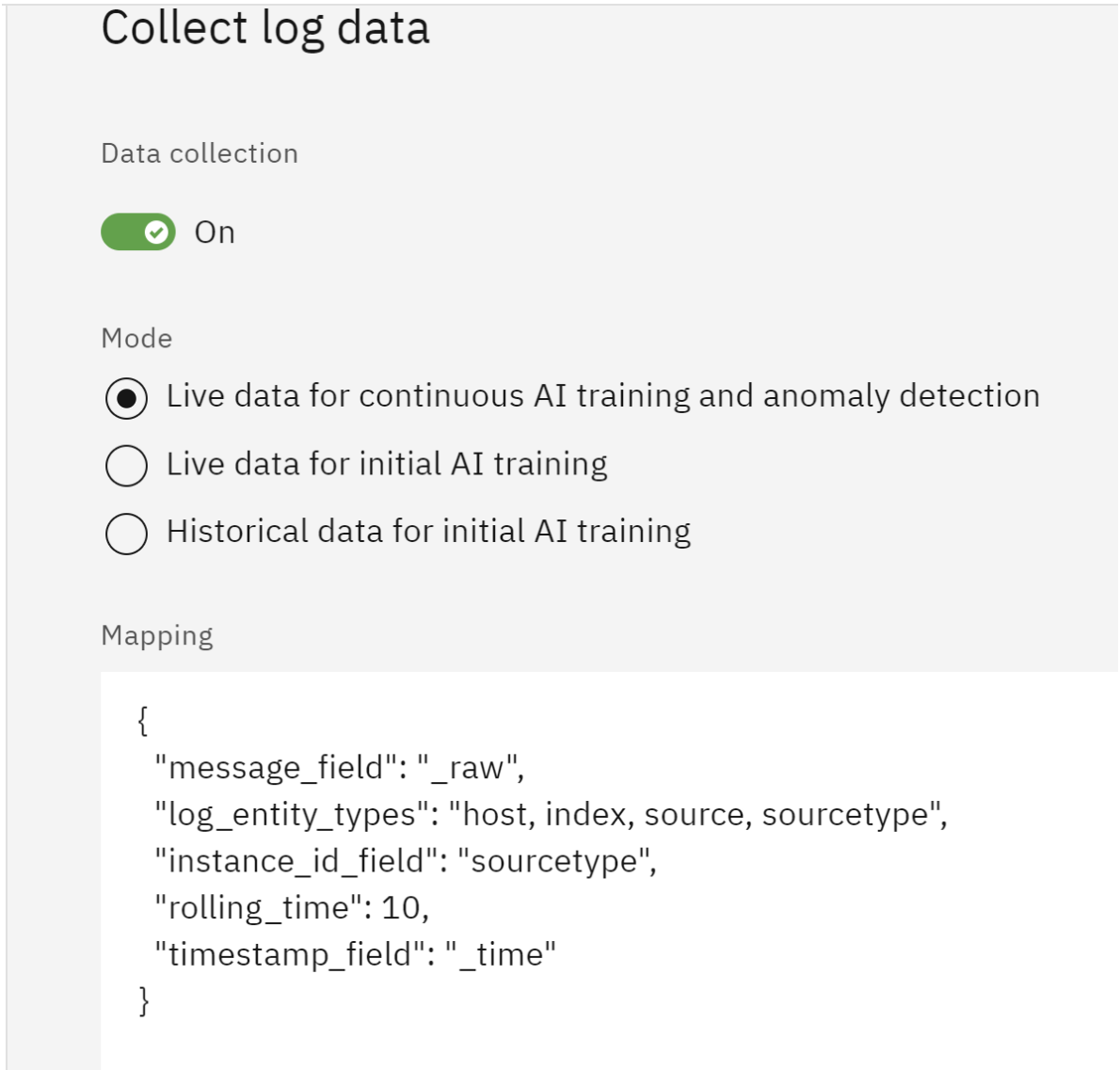


Figure. Collect log data

- **Data collection**: Use the toggle to turn on or off data collection.

- Set the toggle to **Off** (default) if you do not want to collect log data. With this setting, no data is collected through the integration, which results in no anomalies being detected as no AI model is trained. You must turn the toggle to **On** to begin any data flow through the integration.
- Set the toggle to **On** to turn on data collection.

  Note: Log data collection is not supported for a remote deployment. If you want to collect log data, you must deploy a local integration.

- **Mode**: Select how you want data to be collected through the integration for AI training:
  - **Live data for continuous AI training and anomaly detection**: This option enables continuous live data collection through the integration to train AI models and analyze data for anomalous behavior.

    Important: Before you use this option, you must first set **Mode** to **Historical data for initial AI training** and collect a minimum amount of data. Then, you need to run AI training on that data. For more information, see Planning data loading and training.
  - **Live data for initial AI training**: This option collects a single set of training data to define your AI model. Data collection takes place over a specified time period that starts when you create your integration.

    Note: If you select this option, you must disable your integration when you collect enough data for training. If you do not disable the integration, it continues to collect data. For more information about AI model training, including the minimum and ideal data quantities, see Configuring AI training. For more information about disabling and enabling the integration, see Enabling and disabling Splunk integrations.
  - **Historical data for initial AI training**: This option collects a single set of training data to define your AI model. Historical data is harvested from existing logs in your integration over a specified time period in the past. You must specify the start and end dates for this data collection, along with the parallelism of your source data.
    - **Start date**: Select a start date from the calendar. The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.
    - **End date**: Select the end date from the calendar. The recommended time period is two weeks.
    - **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

  Important: Keep in mind the following considerations when you select your data collection type:
  - Anomaly detection for your integration occurs if you select *Live data for continuous AI training and anomaly detection*.
  - Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.
  - **Mapping**: Enter the field-mapping information.

  You can improve search performance by mapping your implementation fields to the standard IBM Cloud Pak for AIOps fields. For more information about this mapping, including how to clean data for use in IBM Cloud Pak for AIOps, see, Mapping data from incoming sources.

  The following code shows the supported data schema for the integration:

```
{
  "rolling_time": 10,
  "instance_id_field": "sourcetype",
  "log_entity_types": "source",
  "message_field": "_raw",
  "timestamp_field": "_time",
  "resource_id": "sourcetype"
}
```

```
* **Filters (optional)**: Enter source, app, or log-level filters to apply to the event data. For more information, see [Search Tutorial ![Opens in a new tab](../images/icons/launch-glyph.svg "Opens in a new tab")](https://docs.splunk.com/Documentation/Splunk/8.1.0/SearchTutorial/Usethesearchlanguage){: external} in the Splunk documentation.

* **Base parallelism (1-50)**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots. In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this integration, you can set this value to be equal to the **source** parallelism.

  **Note:** To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see [Improving data streaming performance for log anomaly detection](../configuring/improving_data_streaming_performance.html).

* **Sampling rate (1-150 seconds)**: Select a rate at which data is sampled from live sources (in seconds). The default value is `60`.

* **JSON processing option**: Select a JSON processing option from the following options:

    * **None**: The default option. The JSON is not processed or modified.
    * **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
    * **Filter**: This option extracts the JSON object and replaces it with an empty string.

 For more information about the options, see [Managing embedded JSON](../configuring/managing_embedded_json_lad.html).
```

13. Click **Next**.
14. On the *Collect metric data* page, define how you want to collect metric data for monitoring.

- **Enable Data collection**: Set the toggle to on to enable the data collection. If data flow is disabled, a new integration is created, but metrics are not collected.
- **Select an index to see all associated source types**: Pick a splunk index from the list to see its related source types.
- **Select a source type to see all associated metrics**: Choose a source type from the list to see its related metrics in the filter table.
- **Search all metrics for the current selection**: Search through all the metrics that are associated with the specified index and source type and select the wanted metrics from the table.
- **Hosts**: Select the hosts you want to get metric data for. All hosts in Splunk of the specifed source type, that are not marked as `too_small` and that have data from within the last 48 hours, will be listed.
- **Aggregation interval (minutes)**: Enter a value in minutes. The value represents the frequency at which data is collected. The default value is 5 minutes.

Complete the settings to use either historical or live data collection:

- To use only live data collection, complete the following settings:

  Select the checkbox **Only collect live data once the integration is established**.

  Note: If you choose to collect only live data, you cannot collect historical data after live data collection starts.

- To use historical data collection, complete the following settings:
  - Do not select the checkbox **Only collect live data once the integration is established**.
  - Set the **Start date**. Select a start date from the calendar and enter the time in hh:mm (hours and minutes) format. Select the time zone from the dropdown list, for example (Localtime or UTC).

    Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.
  - (Optional) Set an *end date* for the data collection. If you do not specify the end date, then live data collection follows the historical data collection. If you do not want to set an end date, click **Remove end date**.

    To add the end date, click **Add end date**. For the **End date**, select an end date from the calendar and enter the time in hh:mm format. The end date uses the same time zone as the start date.

15. Click **Next**.
16. The UI displays the *Resource requirements* page.

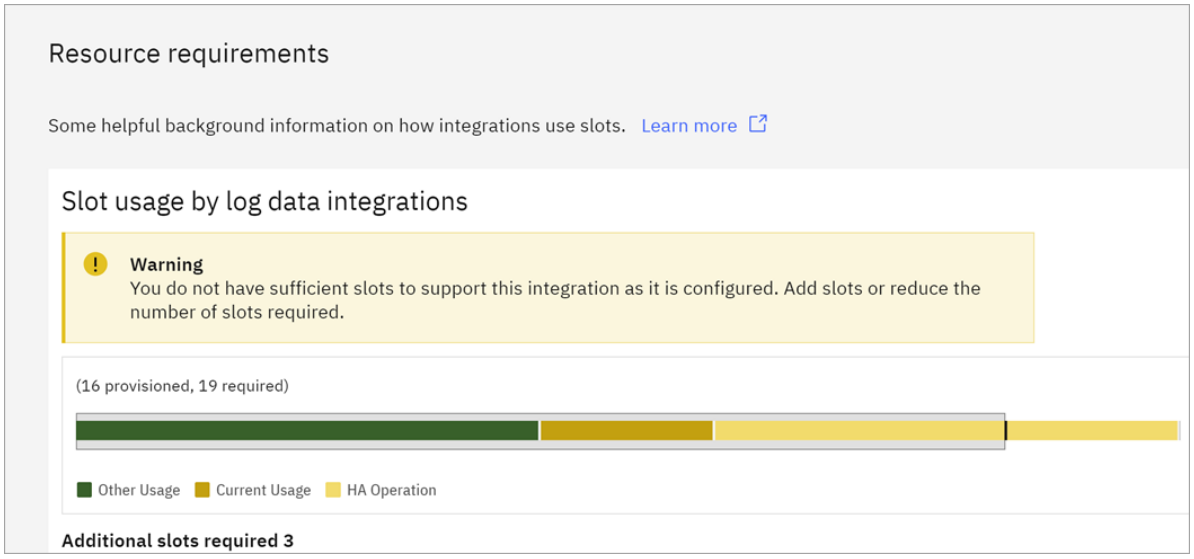


Figure. Resource requirements

Note: If logs data collection is set to **Off** and metrics data collection is set to **On**, you can skip this page.

On the *Resource requirements* page, you can review the slot usage for your log integrations to see if there are enough slots to fully support the integration for multizone high availability.

If you set the **Data collection** toggle to **On**, you will see the resource management overview.

- If your current usage and other usage are less than the provisioned slots, but the high availability slots exceed the provisioned slots, you will be able to create the integration, but will see a warning that you do not have enough slots. The integration will not have multizone high availability.
- If your projected usage exceeds the provisioned slots, you will not be able to create the integration because you do not have enough slots on your system for log data integrations.
- If your total slots, including high availability slots, are within the provisioned slots, the integration will have multizone high availability.

Note: High availability operation assumes high availability for three zones.

If you set the **Data collection** toggle to **Off**, you will see a message stating that you need to enable logs data collection to see the resource management overview. no slots are used by that integration.

17. Click **Done**. You created a Splunk integration in your instance.

After you create your integration, enable data collection to use your integration with the AI of IBM Cloud Pak for AIOps. For more information, see Enabling and disabling data collection.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

To create more integrations (such as a ChatOps integration), see Configuring Integrations.

For more information about working with the insights provided by your integrations, see ChatOps insight management.

Specifying a custom query for the Splunk metrics hosts

If the Splunk hosts are not showing up in the UI, you can specify a custom query. The expected output of the query is a column named **host** which has all the unique hosts for the Splunk system. You should keep performance in mind with a custom host query. Ideally the query should execute in less than 10 seconds. There is a restriction that the custom query must take less than 30 seconds.

Important: If the Splunk hosts are showing up in the UI when using the default host query, there is no need to use a custom host query.

To specify a custom query, use the following steps:

1. Update the configmap from default **SPLUNK_HOST_QUERY** to customize the query: **oc edit configmap aimanager-aio-controller-config**

   Default configmap:

   ```
   data:
     SPLUNK_HOST_QUERY: default
   ```

   Example updated configmap:

   ```
   data:
     SPLUNK_HOST_QUERY: >-
       search index=main sourcetype!=*too_small* earliest=-48h@h | stats count by
       host
   ```

2. Check that the configmap is updated correctly: **oc describe configmap aimanager-aio-controller-config**

   ```
   Data
   ====
   SPLUNK_HOST_QUERY:
   ----
   search index=main sourcetype!=*too_small* earliest=-48h@h | stats count by host
   ```

3. Bring down the controller deployment to 0 and then back up to 1 again.

   ```
   % oc scale deployment aimanager-aio-controller --replicas=0
   % oc get deployment |grep controller
   aimanager-aio-controller                          0/0     0            0           4d10h
   % oc scale deployment aimanager-aio-controller --replicas=1
   % oc rollout status deployment/aimanager-aio-controller
   Waiting for deployment "aimanager-aio-controller" rollout to finish: 0 of 1 updated replicas are available...
   deployment "aimanager-aio-controller" successfully rolled out
   % oc get deployment |grep controller
   aimanager-aio-controller                                  1/1     1            1           7d3h
   ```

4. Check that the env variable is updated in the controller pod:

   ```
   % oc rsh "$(oc get pod |grep aio-controller |awk '{print $1}')" /bin/bash -c 'echo $SPLUNK_HOST_QUERY'
   SPLUNK_HOST_QUERY=search index=main sourcetype!=*too_small* earliest=-48h@h | stats count by host
   ```

Enabling and disabling Splunk integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. If you selected *Live data for initial AI training* when you created your integration, you must disable the integration before AI model training. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **Splunk** integration type.
4. Click the integration that you want to enable or disable.
5. Go to the *AI training and log data* section, and set **Data collection** to **On** or **Off** to enable or disable log data collection.

   Note: Log collection is only supported in local deployment mode. If you need to use the log integration, you must set the deployment option to **Local**.

6. Go to the *Collect metric data* section, and set **Enable Data collection** to on or off to enable or disable metric data collection.

Note: Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting a integration, see Deleting Splunk integrations.

Editing Splunk integrations

After you create your integration, your can edit the integration. For example, if you specified *Historical data for initial AI training* but now want your integration to pull in live data for continuous monitoring, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Splunk** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Splunk integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration. Click **Next** to go through the integration configurationn pages. Click **Save** when you are done editing.

Your integration is now edited. If your application was not previously enabled or disabled, you can enable or disable the integration directly from the interface. For more information about enabling and disabling your integration, see Enabling and disabling Splunk integrations. For more information about deleting a integration, see Deleting Splunk integrations.

Deleting Splunk integrations

If you no longer need your Splunk integration and want to not only disable it, but delete it entirely, you can delete the integration from the console.

Note: You must disable log data collection before you delete your integration. For more information about disabling data collection, see Enabling and disabling Splunk integrations.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Splunk** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Splunk integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Creating Zabbix integrations

A Zabbix integration collects different types of metrics, including live and historical data, to monitor and analyze application performance. It gathers metrics at scheduled intervals from the Zabbix server, and sends data to the gRPC server as cloud events. IBM Cloud Pak for AIOps uses this metric data to identify deviations from the normal behavior and report anomalies. The AI algorithm that analyzes Zabbix data is Metric anomaly detection.

For more information about the Zabbix integrations, see the following sections:

- Before you begin
- Creating Zabbix integrations
- Enabling and disabling Zabbix integrations
- Editing Zabbix integrations
- Deleting Zabbix integrations

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Before you begin

Before you create a Zabbix integration, complete the following steps:

- Install Zabbix v6.0
- Add and configure hosts in the Zabbix server

Note: The Zabbix integration does not support the Zabbix server 6.4 version (or newer).

For more information about installing Zabbix and setting up host, see the Installation and the Quickstart sections in the Zabbix documentation.

To create a Zabbix integration, you must have the following information:

- Endpoint
- Username and Password or API Token

Creating Zabbix integrations

To create a Zabbix integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the **Zabbix** tile.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section, or alternatively start typing the integration name in the searchbar and the corresponding tile appears.
5. On the side-panel, review the instructions and select your required options from the following sections:
   - **Select the data types that you need to collect**: Select the relevant checkbox for *Metrics*, or *Topology*. You can select either Metrics or Topology but not both at the same time.
   - **Specify where you'd like to install the integration**: Select between the *Cluster running Cloud Pak for AIOps* or *Self-hosted with network visibility to Zabbix* radio buttons.
   - **Select Cloud Pak for AIOps integration**: You can leave this at the default *Use the integration recommended by Cloud Pak for AIOps*, or change it to just *Metrics*, or *Topology*.

     The below steps presuppose you select *Metrics* but if you do select *Topology* then you will be brought to a *New observer job* page. The Zabbix observer will read data from a Zabbix instance through its REST APIs and load a topology into Cloud Pak for AIOps. For more information, see Configuring a Zabbix observer job.
6. Click **Next**.

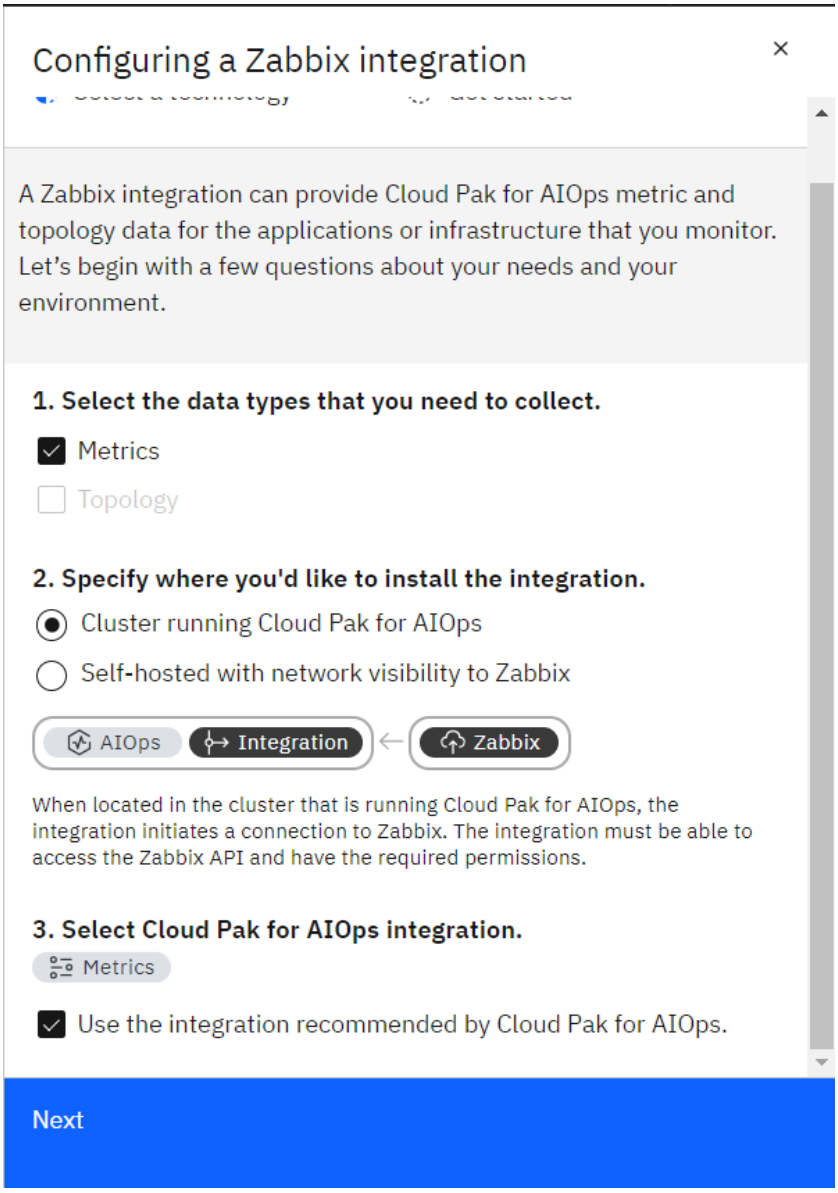


Figure. Configure Zabbix integration

7. On the side-panel, review the information and when ready to continue, click **Get Started**.

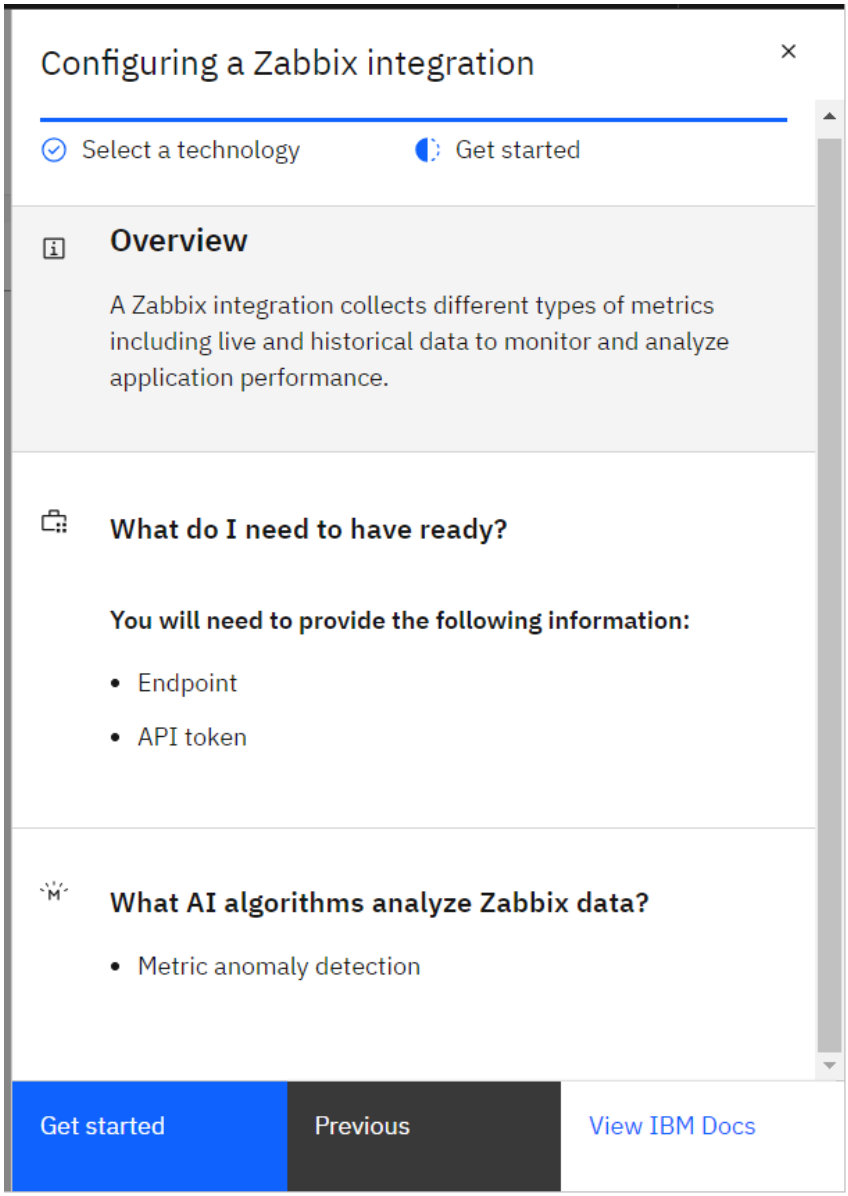


Figure. Get started

8. On the *Add integration* page, define the general integration details:

- **Name**: Create a unique name for your integration.

  Note: Currently, the integration name field only supports alphanumeric characters and excludes spaces and other special characters.

- **Description (optional)**: Provide a description for your integration.
- **Endpoint**: Both http and https urls will be supported by the endpoint (for example, http://<Zabbix-server-hostname>/zabbix/api_jsonrpc.php or https://<Zabbix-server-hostname>/api_jsonrpc.php). In this case you can enter the following endpoint for your integration:

  `<protocol>://<Zabbix-server-hostname>/zabbix/api_jsonrpc.php`

  where **`protocol`** is HTTP or HTTPS and **`zabbix`** is the path to **`api_jsonrpc.php`** file, which is located in the front-end directory.

- **Authentication type**: Select between **Username/Password** or **API Token**.
  - **Username**: Enter a username for the Zabbix integration.
  - **Password**: Enter a password for the Zabbix integration.
  - **API Token**: Enter an API token for the integration.

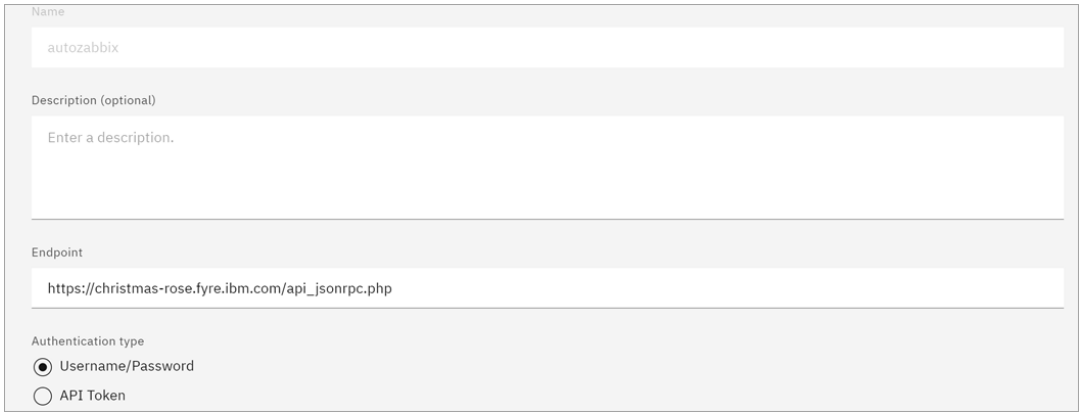


Figure. Create Zabbix integration

Support for Zabbix v5.x

There is a known limitation whereby the **Username/Password** authentication does not work for Zabbix v5.x. You must use the **API Token** authentication method instead.

The following sample steps show how to retrieve the API authentication token **$auth** from the Zabbix server. Ensure that the auto-logout option is not selected for the Zabbix user.

```
% url=http://zabbixserverv5.com/zabbix/api_jsonrpc.php
% auth=$(curl -s -X POST -H 'Content-Type: application/json-rpc' \
-d '
{"jsonrpc":"2.0","method":"user.login","params":
{"user":"myApiUser","password":"myApiPassword"}
,
"id":1,"auth":null}
' $url | \
jq -r .result
)
% echo $auth
3398cacccfa6e8xxx
```

Note: For this example, you must have **curl** and **jq** installed.

For more information see Zabbix documentation.

9. To verify the SSL/TLS configuration:
    1. Set the **Verify the SSL/TLS certificate for the REST service** toggle to *on*.
    2. Either enter a CA certificate used to verify the SSL/TLS integration to the REST service or click **Fetch SSL certificate** to fetch the certificate automatically.
10. Click **Test connection**.

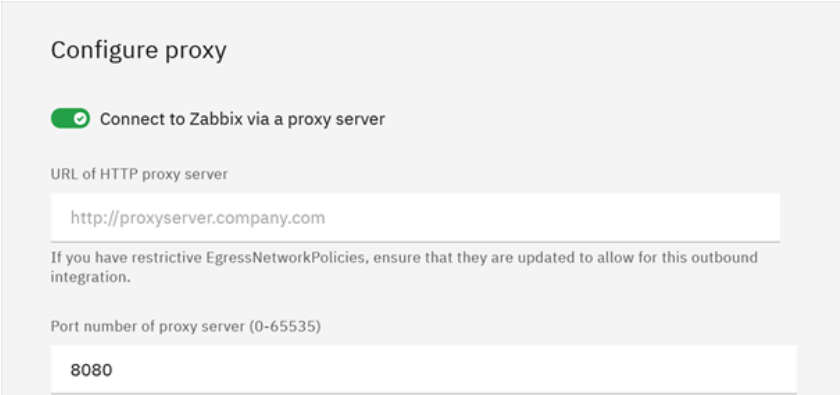
    Note: After you click **Test connection**, wait for the test to succeed before you continue to the next step. It might take some time. If the test fails, verify your credentials, and ensure that the server is up and running. However, you can still move to the metric selection page.

- **Deployment options**: Select one of the options, **Local** or **Remote**.
    - **Local**: This option deploys the integration in the same cluster where IBM Cloud Pak for AIOps is installed.
    - **Remote**: This option deploys it in your chosen location. After you add the integration, you get a bootstrap command to finish the deployment.



Figure. Test connection

11. Click **Next**.
12. On the *Configure Proxy* (Optional) page complete the following fields:
    - **Connect to Zabbix via a proxy server**: Set the toggle button to *on*.
    - **URL of HTTP proxy server**: Enter a URL for the HTTP proxy server.
    - **Port number of proxy server**: Enter a port. The default is 8080.



Note: Only unauthenticated HTTP proxy is supported.

13. Click **Next**.
14. Enter *Collect metric data* information:

- **Enable data collection**: Toggle the switch to the on position. However, if you disable data flow, a new integration is created, but metrics are not collected.
- **Select templates**: You can either select all templates or choose the templates that you want to apply to the hosts. Only templates that have metrics will be shown in the UI.
    - To select all templates, click **Select all templates**.
    - To clear all selection, click **Unselect all templates**.
    - To choose the required templates, select the checkbox of the templates you require from the drop-down list.

  Note: You must select templates appropriate to the hosts. This is because if an item from the template is not available for the host, the integration cannot pull data for that item.

  For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.
- **Search and select the required metrics**: After the templates are loaded, the filter table displays all the associated built-in metrics of the selected templates. These metrics are preselected. You can search through and change the selection, if required.
    - To select a metric, click the checkbox of the required metric.
    - To remove a selected metric, clear its corresponding checkbox.
    - To select or clear all selected metrics, click **Built-in metrics name**. Alternatively, you can click **Select all** or **Unselect all** on the filter table header.
    - To change the size of the page that lists the built-metrics, select a number from the **Items per page** list at the end of the table.
    - To navigate through the pages to see all the metrics, click the next page or previous page arrows.

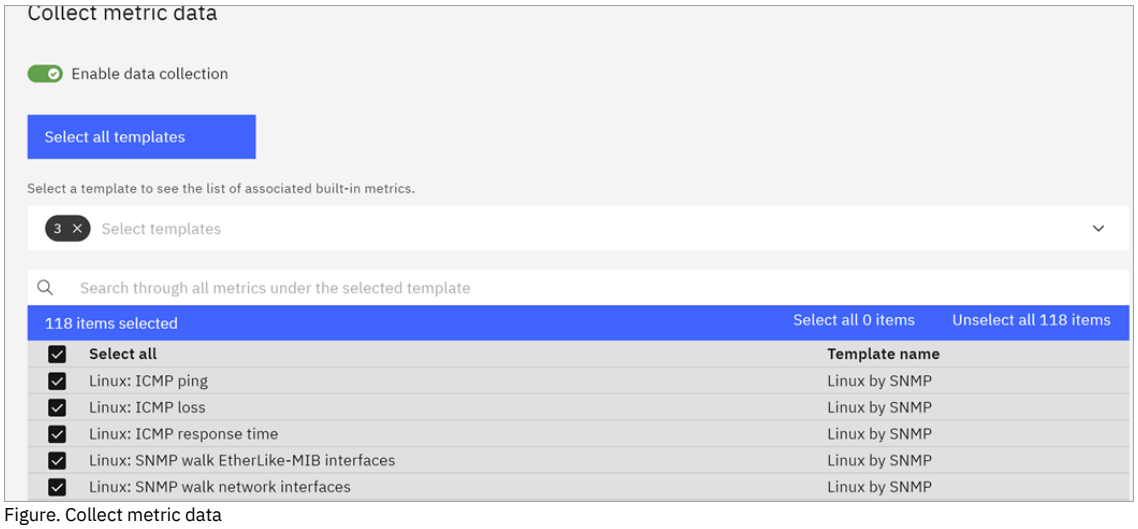


Figure. Collect metric data

- **Hosts**: The hosts are preselected on the load itself. To undo the selection, clear the checkbox of the hostname from the drop-down list.

  Note: If any of the hosts are not available, a notification is displayed. Unselect the hosts that are not available or ensure that all hosts are working in the Zabbix target system before you create your Zabbix integration. If a host goes down after you create the integration, edit and remove the host in your integration.

  Note: Zabbix organizes hosts based on their IP address. Therefore, if two hosts have the same IP address, then selecting one of these hosts results in both hosts getting selected.
- **Base parallelism**: Specify the number of requests that can run in parallel. The default value is 3.

- **History Span (minute)**: Enter a value in minutes. The value represents the time interval for live or historical data collection. The default value is 1.

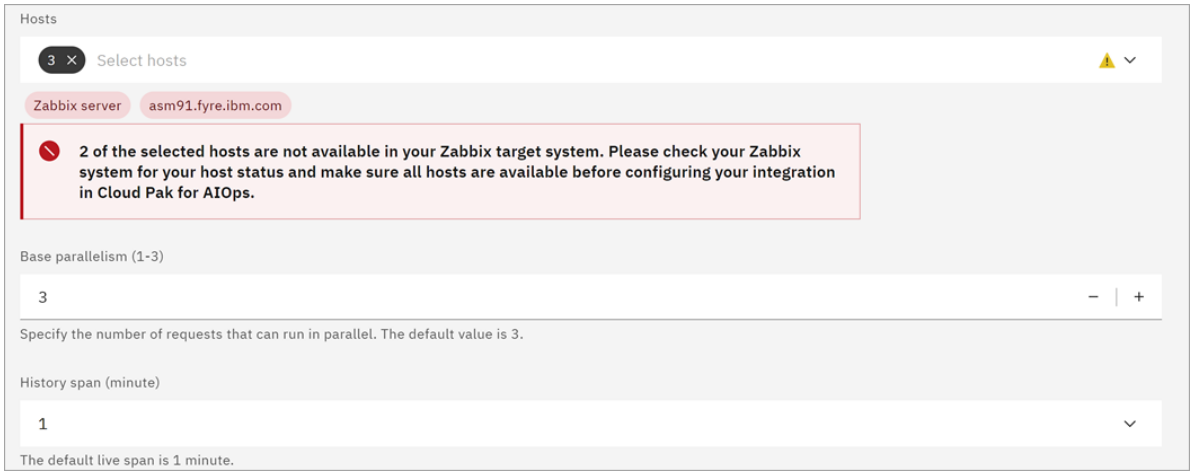


Figure. Create Zabbix integration hosts

- *Select historical or live data collection*:

  Note: If you choose to collect only live data, you cannot collect historical data after the live data collection starts.

- **Live data collection**: To collect live data, complete the following steps: * **Only collect live data once the integration is established**: Select the checkbox. * **Integration interval in minutes for live metrics (1-60)**: Enter a value in minutes. The value represents the frequency at which data is sampled for live metrics. The default value is 5.

- **Historical data collection**: If you do not select live data collection, then historical data metrics are automatically collected. To collect historical data, complete the following steps:

  - **Start date and time**: Select the start date from the calendar and enter the time in the *hh:mm* format (hours and minutes).

  Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.

  - **Time zone**: Select your time zone from the drop-down list.
  - **End date and time**: Click **Add end date** and select a date from the calendar. Also, enter the time in the *hh:mm* format (hours and minutes). If you want to remove an end date, click **Remove end date**.

    Note: If you do not set an end date, IBM Cloud Pak for AIOps collects historical data from the specified start date and continues with live data collection. Therefore, you must specify the integration interval for live metrics.
  - **Integration interval in minutes for live metrics (1-60)**: Enter a value in minutes. The value depicts the frequency at which data is sampled for live metrics. The default value is 5.

15. Click **Done**.

You created a Zabbix integration in your instance.

Enabling and disabling Zabbix integrations

If you didn't enable your data collection during creation, you can enable your integration afterward. You can also disable a previously enabled integration the same way. To enable or disable a created integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Manage integrations* tab of the *Integrations* page, click the **Zabbix** integration type.
4. Click the integration that you want to enable or disable.
5. Set **Enable data collection** to **On** or **Off** to enable or disable log data collection. Disabling data collection for an integration does not delete the integration.

You enabled or disabled your integration. For more information about deleting a integration, see Deleting Zabbix integrations.

Editing Zabbix integrations

After you create your integration, your can edit the integration. For example, if you want to change the time interval of how often your job runs, you can edit it. To edit a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Zabbix** integration type on the *Manage integrations* tab of the *Integrations* page.

4. On the Zabbix integrations page, click the name of the integration that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the integration and click **Edit**. The integration configuration opens.
5. Edit your integration. Click **Next** to go through the integration configuration pages. Click **Save** when you are done editing.

You edited your integration.

Deleting Zabbix integrations

If you no longer need your zabbix integration and want to delete it entirely, you can delete the integration from the console.

To delete a integration, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. Click the **Zabbix** integration type on the *Manage integrations* tab of the *Integrations* page.
4. On the Zabbix integrations page, click the options menu (three vertical dots) for the integration that you want to delete and click **Delete**.
5. Enter the name of the integration to confirm that you want to delete your integration. Then, click **Delete**.

Your integration is deleted.

Delay configuration in data integrations

In live data mode the integration might need a delay to offset the query time windows to provide a time buffer and prevent retrieving partial real-time data.

The following table lists the environment variables for various integrations:

Integration environment variables

Integration	Environment variable for Delay
Custom	CUSTOM_QUERY_DELAY
ELK	ELK_QUERY_DELAY
Falcon LogScale	HUMIO_QUERY_DELAY
Mezmo	MEZMO_QUERY_DELAY
PagerDuty	PAGERDUTY_QUERY_DELAY
Splunk	SPLUNK_QUERY_DELAY

To add delay settings in the Flink, open the configuration by using the following commands:

- For PagerDuty:

  ```
  oc edit flinkdeployment aiops-lad-flink
  ```

- For Falcon LogScale, Custom, Mezmo, ELK, and Splunk:

  ```
  oc edit deployment aimanager-aio-controller
  ```

  In the configuration, edit the `env` section.

  For example, set a 60-seconds delay for Splunk integration:

  ```
  env:
   - name: SPLUNK_QUERY_DELAY
     value: "60"
  ```

Note:

- The values that are set in the configuration must be in seconds.
- For existing integrations to pick up delay values, you must save the integrations again.
- Delays are not applicable to historical mode integrations.

Mapping data from incoming sources

With any integration, you can specify your own mappings to the standard IBM Cloud Pak for AIOps values to optimize search performance for any data type.

For each type of integration, you can edit which fields map to the standard values. By default, each integration type (including custom integrations) uses a default field mapping based on what the source data typically looks like. When you map source fields, you're replacing or supplementing those values with something else in your logs.

Mapping data types

- Mapping log data

- Mapping event data (Only available through a Kafka integration)
- Mapping similar ticket data from ServiceNow

Cleaning mapped data by using regular expressions

While you can use the *Field mapping* section of the integration page to customize field mappings to data values, you can also clean incoming data from your integration here. You can use regular expressions in the mapping field to strip formatting if you know that your data comes formatted in a specific way that affects training. For example, if your log files are always prefixed by the date and time, your training includes that prefix. As a result, IBM Cloud Pak for AIOps flags items that are also prefixed with the same timestamp. The following example illustrates how you can use the *Field mapping* section to clean your message prefixes.

Note: Using regular expressions in the *Field mapping* section is not limited to the following example. You can affect change on any mapped field in the same manner.

In this example, your Mezmo output produces the following timestamp at the beginning of every log entry:

```
Mar  9 03:30:01 TRACE [Controller id=2] Leader imbalance ratio for broker 2 is 0.0 (kafka.controller.KafkaController)
```

You can add `custom_regex` immediately following the mapping that you want to affect to exclude entries that match your proposed regular expression. In this example, you want to exclude timestamps that look like `Mar 9 03:30:01`. You can write a regular expression that follows the formatting of "MMM D HH:MM:SS" to ignore the month value to ensure that your training applies to all months. The following regular expression excludes the timestamp in the example log:

```
^[A-Z][a-z][a-z]\s+([1-9]|[1-2][0-9]|3[0-1])\s+[0-9][0-9]:[0-9][0-9]:[0-9][0-9]\s
```

Note: You can use AND (`&`) and OR operators (`|`) in combination with regular expressions to exclude more patterns. You can implement these operators in the *Field mapping* section by including the following content:

```
{
  "codec": "mezmo",
  "message_field": "message",
  "custom_regex": [
    "^[A-Z][a-z][a-z]\\s+([1-9]|[1-2][0-9]|3[0-1])\\s+[0-9][0-9]:[0-9][0-9]:[0-9][0-9]\\s"
  ],
  "log_entity_types": "container",
  "instance_id_field": "_app",
  "timestamp_field": "_ts"
}
```

Important: If you previously trained your AI models with a different mapping, you must delete and retrain your models with the new mapping in place. For more information about altering models, see Configuring AI models.

Mapping log data from incoming sources

With any integration, you can specify your own mappings to the standard IBM Cloud Pak for AIOps values to optimize search performance for any data type.

For each type of integration (Custom, ELK, Falcon LogScale, Kafka, Mezmo, and Splunk), you can edit which fields map to the standard values. By default, each integration type (including custom integrations) uses a default field mapping based on what the source data typically looks like. When you map source fields, you're replacing or supplementing those values with something else in your logs.

Mapping log anomaly data

The Log Anomaly service detects abnormal activities from system logs, which help Site Reliability Engineers (SREs) to better react to issues. Two types of anomalies are considered: *quantitative anomalies* (where constant linear or quantitative relationships are broken) and *sequential anomalies* (where a log sequence deviates from normal patterns of program flows).

IBM Cloud Pak for AIOps ingests logs from standard or custom data sources. Raw logs from supported standard data sources are transformed to a normalized format, as shown in the following example. Data from custom data sources must be transformed into normalized format first to be used in an integration (or eventually for AI model training).

Log anomaly raw data

To maintain the structure of the log anomaly data in the Elastic database, you must transform raw log anomaly data into a normalized format. The following examples illustrate raw JSON log anomaly data that comes from the ELK Stack, Falcon LogScale, and Splunk:

ELK Stack raw data example


```
{"_index":"syslog-console-example.com-20201018","_type":"_doc","_id":"t1IaPXUBuJIWpBPwb1fG","_score":null,"_source":
{"MESSAGEID":"DSNJ001I","sysplexName":"MCMPLEX1","JOBNAME":"DC33MSTR","TEXT":"DSNJ001I  -DC11 DSNJW307 CURRENT COPY 2
ACTIVE LOG\nDATA SET IS
DSNAME=DSN11.LOGCOPY2.DS02,\nSTARTRBA=00000000000C20358000,ENDRBA=00000000000C2078FFFF\"\n","ROUTECODE":"8000000000000000
0000000000000000","SYSPLEX_SYSTEM_JOB":"MCMPLEX1.ABCLP20.DC33MSTR","SYSPLEX_SYSTEM_CICS":"MCMPLEX1.ABCLP20","path":"SYSLO
G","rcd":"MC","ASID":"0043","systemName":"ABCLP20","JOBNUM":"STC02552","TIMESTAMP":"20292 15.05.38.160
-0400","host":"ABCLP20.svl.ibm.com","@version":"1","FLAGS":"00","seq":
{"c":"14469","w":"4"},"DESCRIPTOR":"00000090","timeZone":"-0400","message":"MC,0043,20292 15.05.38.160 -0400,ABCLP20
,STC02552,         ,80000000000000000000000000000000,00000090,DC33MSTR,00,\" DSNJ001I  -DC11 DSNJW307 CURRENT COPY 2
ACTIVE LOG\n DATA SET IS DSNAME=DSN11.LOGCOPY2.DS02,\n
STARTRBA=00000000000C20358000,ENDRBA=00000000000C2078FFFF\"\n","inputsequence":"20201018190543703:000000","tags":"pipelin
e","SMFID":"ABCLP20 ","@timestamp":"2020-10-18T15:05:38.160-04:00","port":6911,"CONSOLE":"        ","sourceType":"syslog-
console","SYSPLEX_SYSTEM_DB2":"MCMPLEX1.ABCLP20","sourceName":"ABCLP20-
SYSLOG","SYSPLEX_SYSTEM":"MCMPLEX1.ABCLP20"},"sort":[1603047938160]}
```

Falcon LogScale (previously named Humio) raw data example


```
{"kubernetes.namespace_name":"myhumio","@timestamp":1603445464206,"kubernetes.host":"worker.host.example.com","kubernetes.container_hash":"docker.io/value/enterprise-kafka@sha256:bb3d4572b51490b7e9fc2431125ae963bf7cdc4cd30ff4575b1e529e3e53a1e6","kubernetes.docker_id":"39e5cba870732269940daa8378ee2eab52adffa93dd6c5ccf15246e70b115584","kubernetes.labels.app":"cp-kafka","kubernetes.container_name":"cp-kafka-broker","kubernetes.container_image":"docker.io/value/enterprise-kafka:5.2.1","kubernetes.labels.controller-revision-hash":"humio-example-kafka-66c4fb4d64","kubernetes.annotations.k8s_v1_cni_cncf_io/networks-status":"[{\n    \"name\": \"openshift-sdn\",\n    \"interface\": \"eth0\",\n    \"ips\": [\n        \"10.254.20.30\"\n    ],\n    \"dns\": {},\n    \"default-route\": [\n        \"10.254.20.1\"\n    ]\n}]","kubernetes.annotations.prometheus_io/scrape":"true","@timezone":"Z","@rawstring":"2020-10-23T09:31:04.206512759+00:00 stdout F [2020-10-23 09:31:04,206] TRACE [Controller id=2] Leader imbalance ratio for broker 2 is 0.0 (kafka.controller.KafkaController)","kubernetes.labels.statefulset_kubernetes_io/pod-name":"humio-example-kafka-2","@id":"yI1o4C0cV5i8oGU4TkcbavzO_587_147_1603445464","#type":"unparsed","kubernetes.annotations.openshift_io/scc":"restricted","kubernetes.labels.release":"humio","#repo":"kubernetes","kubernetes.pod_name":"humio-example-kafka-2","kubernetes.pod_id":"7a4b0b92-1254-4335-ae29-b1f548c8e21d","#humioAutoShard":"1","kubernetes.annotations.prometheus_io/port":"5556"}
```


Splunk raw data example


```
{"_bkt":"_internal~53~283A637C-D363-4549-A819-E8175BC73650","_cd":"53:2848711","_indextime":"1603220666","_raw":"127.0.0.1 - splunk-system-user [20/Oct/2020:12:04:26.272 -0700] \"GET /services/saved/searches?count=0 HTTP/1.1\" 200 1861412 - - - 162ms","_serial":"31263","_si":["abc.example.com","_internal"],"_sourcetype":"splunkd_access","_subsecond":".272","_time":"2020-10-20T12:04:26.272-07:00","host":"abc.example.com","index":"_internal","linecount":"1","source":"/root/mj/mjsplunk/splunk/var/log/splunk/splunkd_access.log","sourcetype":"splunkd_access","splunk_server":"abc.example.com"}
```


Normalized log data example

The following sample illustrates the result of mapping data from an external log source for use with IBM Cloud Pak for AIOps:


```
{
    "timestamp": 1655991641000,
    "instance_id": "calico-node-jn2d2",
    "application_group_id": "1000",
    "application_id": "1000",
    "features": [],
    "meta_features": [],
    "message": "[64] ipsets.go 254: Resyncing ipsets with dataplane. family=\"inet\"",
    "entities": {
        "pod": "calico-node-jn2d2",
        "cluster": null,
        "container": "calico-node",
        "node": "kube-bmgcm5td0stjujtqv8m0-ai4itsimula-wp16cpu-00002c34"
    },
    "type": "StandardLog"
}
```


Log anomaly normalized attributes

Required attributes expected as input for the pipeline

Attribute	Description
`timestamp`	Epoch Timestamp of the event in the log entry. The epoch time should be in milliseconds (13 digit) and not in seconds (10 digit).
`utc_timestamp`	The `utc_timestamp` of the event in the log entry. The format should be in `yyyy-MM-dd HH:mm:ss.SSS` format. The value of this field should be equivalent for the epoch value of the `timestamp` field.
`instance_id`	Logs can be trained based on certain attributes, like the application to which the log entry belongs, the hosts from which the log is generated, or the pods from which the log is generated. `instance_id` is the value of the attribute that is chosen to be the key upon which the models are built. Attributes chosen to be `instance_id` are specified in the ingest configuration during onboarding.
`application_id`	Application from which the log entry was generated. It can be the same as `instance_id` if the `instance_id` is chosen to be the application
`features`	Contains log templates
`message`	Log message for the log entry. It can be an error message, a normal information message, or a debug message. A good combination of message types yields the best results. For example, if the logs are only error messages or if no error messages appear in the logs, the quality of the anomaly detection suffers.
`entities`	Entities are provided for better event attribution. They provide clues to connect information from multiple data sources such as events, logs, and incidents, and also are used to globalize faulty components. Entities can be infrastructure-related attributes like node name, hostname, IP address, pod name, container name, or cluster name. These attributes are specified as a comma-separated list in the configuration during onboarding. The `entities` attribute must, at least, contain an empty field: `"entities": {}`.

Static attributes

Attribute	Description
`application`	Application that the IBM Cloud Pak for AIOps instance is monitoring. Provided as part of the initial configuration during onboarding
`application_group_id`	Supports multi-tenancy. Provided as part of the initial configuration during onboarding
`type`	Hardcoded to the log

Attributes analyzed and populated in the pipeline

Attribute	Description

Attribute	Description
`meta_features`	Contains windowing details

Examples

Review the following examples to view how to map log data.

Example: Falcon Logscale field mapping

- `message_field` maps to `@rawstring`
- `log_entity_types` maps to `kubernetes.container_name`
- `instance_id_field` maps to `kubernetes.container_name`
- `timestamp_field` maps to `@timestamp`

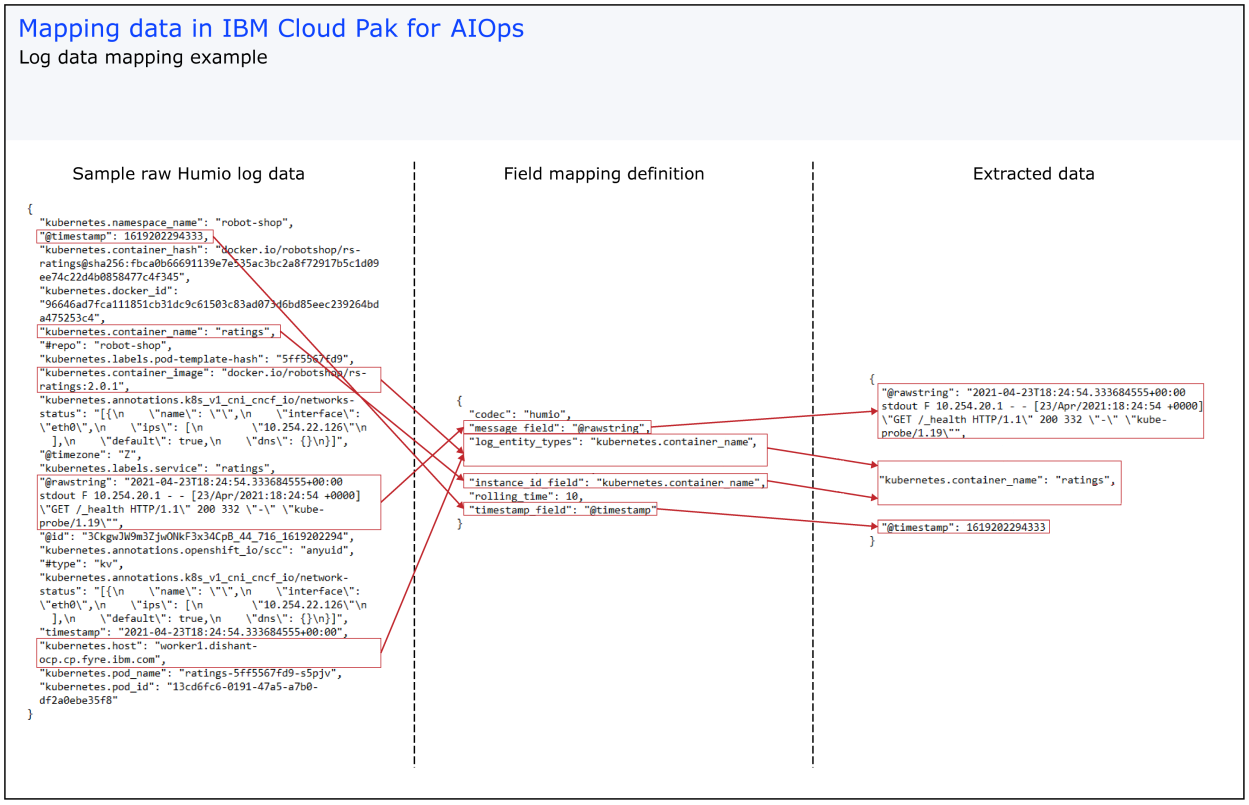


Figure. Falcon Logscale mapping example

All of the values in this Falcon LogScale field mapping correspond to entries in the Falcon LogScale raw data. In this example, the following values map to one another in the Falcon LogScale raw log.

Important: You must specify the corresponding codec in your field mapping when you create your integration. If you do not, the mapping fails. In the preceding example, the mapping specifies `humio` as the codec value.

You can define mappings when you create your integration. The basic building blocks of mappings are similar for most integration types. For logs, the `codec`, `rolling_time`, `instance_id_field`, `log_entity_types`, `message_field`, and `resource_id` mappings typically correspond to values that you might find in your log data. The following table describes typical mapping fields for logs:

Typical mapping values for log integrations

Field Name	Description
`codec`	The type of integration for your incoming data. You must specify the codec value for non-log integrations too. For example, you must specify `humio` for a Falcon Logscale integration mapping, or `servicenow` for a ServiceNow integration.
`rolling_time`	The recurring time interval to capture data from in seconds. The default value is 10 seconds.
`instance_id_field`	The field name in the incoming data that contains the application name. This value is used to identify where your log data is coming from. If you choose an overly specific value, it might lead to flooding the pipeline with irrelevant log data. Ideal candidates for mapping this field include: `kubernetes.namespace_name`, `kubernetes.host`, or `kubernetes.container_name`.
`log_entity_types`	The field name in the incoming data that contains the entity to extract and analyze. The value provides information about the app or service where the log data comes from. Ideal candidates for mapping this field include: `kubernetes.namespace_name`, `kubernetes.host`, or `kubernetes.container_name`.
`message_field`	The field name in the incoming data that contains the log message. For example, `@rawstring`.
`resource_id`	The field name in the incoming data, which contains the specific component that produces the log message. For example, if you deploy your application through a Kubernetes deployment with four replicas, then the `resource_id` field name contains the specific pod name that produces the log message. This information helps to narrow down the component that must be analysed on detection of an anomaly in your application.

Important: Be mindful when you select mapping values for your **instance_id_field** and **log_entity_types** fields. **instance_id_field** identifies the app or service where your messages come from. **instance_id_field** acts almost like the application name in that respect. **log_entity_types** provides information about the app or service, like the name of the pod, the image used in the pod, the node of the pod, and more. If you chose values that are too specific, you might gather too much information from irrelevant pods or projects (namespaces).

Example: Log anomaly output schema


```
{
    "application_group_id": "1",
    "application_id": "1",
    "timestamp":1655991641000,
    "utc_timestamp":"2022-06-23 13:40:41.000"
    "type": "log-anomaly-detection",
    "alert": {
      "alert_id": "6751a428-5827-11ea-9f3a-c4b301ca8457",
      "title": "NETWORK <*> end connection <*> (<*> <*> now open) NETWORK [listener] connection accepted from <*> #<*>
(<*> connections now open)",
      "text": "NETWORK <*> end connection <*> (<*> <*> now open) NETWORK [listener] connection accepted from <*> #<*>
(<*> connections now open)",
      "created_at_utc_timestamp": "2020-02-25T23:34:44.874301",
      "severity": 3,
      "source": {
        "source_name": "anomaly-detector",
        "source_url": "",
        "source_alert_id": "6751a428-5827-11ea-9f3a-c4b301ca8457",
        "source_team": {
          "id": "",
          "name": "",
          "web_url": ""
        },
        "source_application": {
          "id": "ts-assurance-mongo",
          "name": "ts-assurance-mongo",
          "web_url": ""
        }
      },
      "features": [
        {
          "id": "log_anomaly_data",
          "value": {
            "id": "6751a428-5827-11ea-9f3a-c4b301ca8457",
            "start_timestamp": 1578669070000,
            "end_timestamp": 1578669080000,
            "log_anomaly_data": {
              "detected_at": 1582702484874.0,
              "source_application_id": "ts-assurance-mongo",
              "log_anomaly_confidence": 1.0,
              "template_sequence": [
                "NETWORK [listener] connection accepted from <*> #<*> (<*> connections now open)",
                "NETWORK [listener] connection accepted from <*> #<*> (<*> connections now open)",
                "NETWORK <*> end connection <*> (<*> <*> now open)",
                "NETWORK <*> end connection <*> (<*> <*> now open)"
              ],
              "template_list": [
                "[ENTSV:ERRR] <*> <*> <*> <*> <*>",
                "Open Entities Serve Exception received. Max retries: <*>, Current retry <*>",
                "Exception during getNLQuestionClasses() call",
                "Exception invoking mmesh.EntityPredictionService/predictTokenizedDocument method for model ids <*>",
                "Error invoking mmesh.EntityPredictionService/predictTokenizedDocument method on model <*> INTERNAL:
<ENT02360739W> Exception caught: Authentication failed.",
                "applyModel",
                "Error on query to NLU engine.",
                "Error occurred while processing a message v2 request"
              ],
              "count_vector": [
                2,
                2,
                0,
                0
              ]
            },
            "entity": {
              "pod": [
                {
                  "name": "ts-assurance-mongo-54b557bfbc-7txwh",
                  "counts": 4
                }
              ],
              "cluster": [
                {
                  "name": "null",
                  "counts": 4
                }
              ],
              "container": [
                {
                  "name": "ts-assurance-mongo",
                  "counts": 4
                }
              ],
              "node": [
```

```
                {
                  "name": "kube-bmgcm5td0stjujtqv8m0-ai4itsimula-wp16cpu-00002d54",
                  "counts": 4
                }
              ]
            }
          }
        }
      ],
      "resolved_at_utc_timestamp": null
    },
    "meta_features": [
      {}
    ]
  }
```

Example: Normalized output schema

The preceding example adheres to the normalized output schema for IBM Cloud Pak for AIOps, which can resemble the following example output schema.

```
{
    "$id": "incidents-schema.json",
    "$schema": "http://json-schema.org/draft-06/schema",
    "title": "incidents",
    "type": "object",
    "properties": {
        "application_group_id": {
            "description": "Unique identifier for the tenant which owns the application being monitored (defined by IBM
Cloud Pak for AIOps)",
            "type": "string"
        },
        "application_id": {
            "description": "Unique identifier of the application being monitored (defined by IBM Cloud Pak for AIOps)",
            "type": "string"
        },
        "incident_id": {
            "description": "The unique id for this incident created by IBM Cloud Pak for AIOps",
            "type": "string"
        },
        "title": {
            "description": "The short title for the given incident (SN short_description)",
            "type": "string"
        },
        "description": {
            "description": "The description for the given incident (SN description)",
            "type": "string"
        },
        "created_at": {
            "description": "The timestamp the incident was created at (SN sys_created_on).",
            "type": "string"
        },
        "updated_at": {
            "description": "The timestamp the incident was last updated (SN sys_updated_on).",
            "type": "string"
        },
        "resolved_at": {
            "description": "The timestamp the incident was resolved (SN resolved_at).",
            "type": "string"
        },
        "closed_at": {
            "description": "The timestamp the incident was closed (SN closed_at).",
            "type": "string"
        },
        "started_at": {
            "description": "The timestamp the disruption (which caused this incident to be created) was started at (SN
opened_at).",
            "type": "string"
        },
        "business_duration_ms": {
            "description": "The duration in ms the incident. (SN business_duration)",
            "type": "integer"
        },
        "severity": {
            "description": "The severity, as set by the client, of the incident on a scale from 1 (highest) to 5 (lowest)
(SN severity)",
            "type": "integer",
            "enum": [1,2,3,4,5]
        },
        "priority": {
            "description": "The priority, as set by the helpdesk, of the incident on a scale from 1 (highest) to 5
(lowest) (SN priority)",
            "type": "integer",
            "enum": [1,2,3,4,5]
        },
        "impact": {
            "description": "The impact of the incident on a scale from 1 (highest) to 5 (lowest) (SN impact)",
            "type": "integer",
            "enum": [1,2,3,4,5]
        },
        "state": {
            "description": "The current state of the incident (SN state)",
            "type": "string",
```

```
            "enum": ["new", "in_progress", "resolved", "closed"]
        },
        "source": {
            "type": "object",
            "description": "The source that the incident was read from",
            "properties": {
                "source_name": {
                    "description": "The display name for the incident system this incident was read from (such as
ServiceNow).",
                    "type": "string"
                },
                "source_url": {
                    "description": "The url for the incident system this incident was read from (such as https://....).",
                    "type": "string"
                },
                "source_incident_id": {
                    "description": "The id for this incident in the original source (SN number)",
                    "type": "string"
                },
                "source_application_id": {
                    "description": "Id of the application that had the disruption (SN cmdb_ci)",
                    "type": "string"
                }
            },
            "required": [
                "source_name",
                "source_incident_id"
            ]
        },
        "comments": {
            "description": "The set of comments and work notes for the incident",
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "comment_text": {
                        "description": "The text for the given comment (SN comments_and_work_notes).",
                        "type": "string"
                    }
                },
                "additionalProperties": false,
                "required": [
                    "comment_text"
                ]
            },
            "minItems": 0
        },
        "related_incidents": {
            "description": "The set of incidents that are related (SN parent_incident & child_incidents)",
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "source_incident_id": {
                        "description": "The id for the related incident in the source system.",
                        "type": "string"
                    },
                    "relationship": {
                        "description": "The relationship to this incident, namely parent child or similar",
                        "type": "string",
                        "enum": ["parent","child","similar"]
                    }
                },
                "additionalProperties": false,
                "required": [
                    "comment_text"
                ]
            },
            "minItems": 0
        },
        "resolution": {
            "type": "object",
            "description": "The source that the incident was read from",
            "properties": {
                "rca_id": {
                    "description": "The id of the rca (problem) ticket that was created in response to this incident (SN
problem_id).",
                    "type": "string"
                },
                "resolution_summary": {
                    "description": "The summary of what was done to resolve the issue (SN close_notes).",
                    "type": "string"
                }
            },
            "required": [
                "resolution_summary"
            ]
        },
        "features": {
            "description": "The extended set of features related to the incident",
            "type": "array",
            "items": {
```

```
                "type": "object",
                "properties": {
                },
                "additionalProperties": true,
                "required": [
                ]
            },
            "minItems": 0
        },
        "required": [
            "application_group_id",
            "application_id",
            "incident_id",
            "created_at_utc_timestamp",
            "title",
            "description",
            "severity",
            "state",
            "source"
        ],
        "meta_features": {
            "description": "The extended set of features related to the message",
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                },
                "additionalProperties": true,
                "required": [ ]
            },
            "minItems": 0
        }
    }
}
```

Mapping event data for a Kafka integration

With a Kafka integration, you can gather data for both logs and events. If you are gathering event data, you can specify your own mappings to the standard IBM Cloud Pak for AIOps values.

By default, the Kafka integration uses a default field mapping based on what the source data typically looks like. When you map source fields, you're replacing or supplementing those values with something else in your logs.

Important: Values mapped to the `alert.type.classification` field in the Cloud Pak for AIOps data model are especially significant. Inappropriate values assigned to this field at the time of mapping has a negative impact on several downstream activities, such as Golden Signal classification, incident creation, and recommended policies. Classification should contain information summarizing the event type, for example, CPU Utilization, Node Down, or the actual metric name itself. The following example mappings are from an attribute in an Instana event payload:

```
classification = "Garbage Collection Activity High"
or
classification = "Abnormally High backend commit duration"
or
classification = "Container Memory Usage is getting closer to limit"
```

Sample mappings are available when setting up a generic webhook integration in Cloud Pak for AIOps. Click **Load sample mapping** on the *Configure event mapping* screen and select from the available use cases.

Mapping event grouping data

Event grouping receives streams of events from a Kafka integration, and continuously groups events in real time. This grouping process uses various techniques such as deduplication, temporal clustering, and more. The event grouping service interacts with other services to further provide localization, average severity, pattern recognition, diagnosis, and explanation.

The event grouping service takes normalized data through the integration from an event data source, such as PagerDuty. That data is then used to fit into a single structure that the event grouping service uses to derive new information, such as a topology. Event grouping aims to suppress the number of events to help you diagnose problems with relevant events. This grouping increases the efficiency of handling incidents, and associates the event with other information (such as logs or metrics).

Event raw data

To maintain the structure of the event data in the Elasticsearch database, you must map and transform raw event data into a normalized format. This normalized data can then be used in AI training.

The following raw JSON data shows example event data from a PagerDuty data source:

```
{"id": "PBDVRS5", "type": "alert", "summary": "[Kubernetes] Pods Available less than desired", "self":
"https://api.pagerduty.com/alerts/PBDVRS5", "html_url": "https://example.pagerduty.com/alerts/PBDVRS5", "created_at":
"2020-01-17T19:21:24-05:00", "status": "resolved", "resolved_at": "2020-01-17T19:34:22-05:00", "alert_key": "3014911",
"suppressed": false, "service": {"id": "PTOQ55S", "type": "service_reference", "summary": "simulation", "self":
"https://api.pagerduty.com/services/PTOQ55S", "html_url": "https://example.pagerduty.com/services/PTOQ55S"}, "severity":
"critical", "incident": {"id": "P6ZLD21", "type": "incident_reference", "summary": "[#29918643] [Kubernetes] Pods
Available less than desired", "self": "https://api.pagerduty.com/incidents/P6ZLD21", "html_url":
```

```
"https://example.pagerduty.com/incidents/P6ZLD21"}, "first_trigger_log_entry": {"id": "RNSI4KGVTUAA2ZJ8IILWPF52PL",
"type": "trigger_log_entry_reference", "summary": "Triggered through the API", "self":
"https://api.pagerduty.com/log_entries/RNSI4KGVTUAA2ZJ8IILWPF52PL", "html_url":
"https://example.pagerduty.com/alerts/PBDVRS5/log_entries/RNSI4KGVTUAA2ZJ8IILWPF52PL"}, "body": {"contexts": [{"href":
"https://cloud-server.monitoring.cloud.example.com:443/#/alerts/343016", "text": "Link to Alert definition that triggered
the event", "type": "link"}, {"href": "https://cloud-
server.monitoring.cloud.example.com:443/#/events/notifications/1:604800/3014911/details", "text": "Troubleshoot Event",
"type": "link"}], "details": {"Value": "0.95; 1.0", "UTC": "2020-01-18T00:19Z", "Timestamp": 1579306740000, "Subject": "
[Kubernetes] Pods Available less than desired is Triggered on kubernetes.namespace.name = default and
kubernetes.deployment.name = ts-admin-travel-service", "Source": "Sysdig", "Severity": "Low", "Segment":
"kubernetes.deployment.name = 'ts-admin-travel-service' and kubernetes.namespace.name = 'default'", "Scope":
"kubernetes.deployment.name = 'ts-admin-travel-service' and kubernetes.namespace.name = 'default'", "Event ID": 3014911,
"Condition": "avg(timeAvg(kubernetes.deployment.replicas.available)) <
avg(timeAvg(kubernetes.deployment.replicas.desired))", "Body": "\n\n\nEvent Generated:\n\nSeverity:         Low\n
Metric:\n    kubernetes.deployment.replicas.desired = 1\nSegment:\n    kubernetes.deployment.name = 'ts-admin-travel-
service' and kubernetes.namespace.name = 'default'\nScope:\n    Everywhere\n\nTime:             01/18/2020 12:19 AM
UTC\nState:            Triggered\nNotification URL: https://cloud-
server.monitoring.cloud.example.com:443/api/oauth/openid/example/6f42245bn4ab5gc294c55b5785e74ab9/ba4b7qqa-1fe9-47d1-
9355-1i97d23a1855?redirectRoute&#x3D;/events/notifications/1:2419200/3014911/details\n\n------\n\nTriggered by
Alert:\n\nName:         [Kubernetes] Pods Available less than desired\nDescription:  Not enough pods running for
deployment\nTeam:         Monitor Operations\nScope:\n    Everywhere\nSegment by:   kubernetes.namespace.name,
kubernetes.deployment.name\nWhen:         avg(timeAvg(kubernetes.deployment.replicas.available)) <
avg(timeAvg(kubernetes.deployment.replicas.desired))\nFor at least: 10 min\nAlert URL:    https://cloud-
server.monitoring.cloud.example.com:443/api/oauth/openid/example/6f58425be4l14ab295c55b3238e74ab9/ba5n7gga-1fe9-47d1-
9355-lp97d63a1855?redirectRoute&#x3D;/alerts/343016\n\n\n", "Alert name": "[Kubernetes] Pods Available less than
desired", "Alert description": "Not enough pods running for deployment", "Alert ID": 343016}, "cef_details": {"client":
"Sysdig", "client_url": "https://cloud-
server.monitoring.cloud.example.com:443/#/events/notifications/1:604800/3014911/details", "contexts": [{"type": "link",
"text": "Link to Alert definition that triggered the event", "href": "https://cloud-
server.monitoring.cloud.example.com:443/#/alerts/343016"}, {"type": "link", "text": "Troubleshoot Event", "href":
"https://cloud-server.monitoring.cloud.example.com:443/#/events/notifications/1:604800/3014911/details"}], "dedup_key":
"3014911", "description": "[Kubernetes] Pods Available less than desired", "details": {"Value": "0.95; 1.0", "UTC":
"2020-01-18T00:19Z", "Timestamp": 1579306740000, "Subject": "[Kubernetes] Pods Available less than desired is Triggered
on kubernetes.namespace.name = default and kubernetes.deployment.name = ts-admin-travel-service", "Source": "Sysdig",
"Severity": "Low", "Segment": "kubernetes.deployment.name = 'ts-admin-travel-service' and kubernetes.namespace.name =
'default'", "Scope": "kubernetes.deployment.name = 'ts-admin-travel-service' and kubernetes.namespace.name = 'default'",
"Event ID": 3014911, "Condition": "avg(timeAvg(kubernetes.deployment.replicas.available)) <
avg(timeAvg(kubernetes.deployment.replicas.desired))", "Body": "\n\n\nEvent Generated:\n\nSeverity:         Low\n
Metric:\n    kubernetes.deployment.replicas.desired = 1\nSegment:\n    kubernetes.deployment.name = 'ts-admin-travel-
service' and kubernetes.namespace.name = 'default'\nScope:\n    Everywhere\n\nTime:             01/18/2020 12:19 AM
UTC\nState:            Triggered\nNotification URL: https://cloud-
server.monitoring.cloud.example.com:443/api/oauth/openid/example/5gabc35be4ab4cc294c55b5785e74ab9/ba4b7nna-1fe9-47d1-
9355-fm65d63a1855?redirectRoute&#x3D;/events/notifications/1:2419200/3014911/details\n\n------\n\nTriggered by
Alert:\n\nName:         [Kubernetes] Pods Available less than desired\nDescription:  Not enough pods running for
deployment\nTeam:         Monitor Operations\nScope:\n    Everywhere\nSegment by:   kubernetes.namespace.name,
kubernetes.deployment.name\nWhen:         avg(timeAvg(kubernetes.deployment.replicas.available)) <
avg(timeAvg(kubernetes.deployment.replicas.desired))\nFor at least: 10 min\nAlert URL:    https://cloud-
server.monitoring.cloud.example.com:443/api/oauth/openid/example/6f86155be4ab4cc294c55b5785e74ab9/ba4b7bba-1fe9-47d1-
9355-ba97d63a1855?redirectRoute&#x3D;/alerts/343016\n\n\n", "Alert name": "[Kubernetes] Pods Available less than
desired", "Alert description": "Not enough pods running for deployment", "Alert ID": 343016}, "message": "[Kubernetes]
Pods Available less than desired", "mutations": [], "version": "1.0"}, "type": "alert_body"}, "integration": {"id":
"PKKYNGU", "type": "generic_events_api_inbound_integration_reference", "summary": "Sysdig-ai4it", "self":
"https://api.pagerduty.com/services/PTOQ55S/integrations/PKKYNGU", "html_url":
"https://example.pagerduty.com/services/PTOQ55S/integrations/PKKYNGU"}, "privilege": null, "team": {"id": "PWY2HS1",
"type": "team", "summary": "Project Zeno E2E", "self": "https://api.pagerduty.com/teams/PWY2HS1", "html_url":
"https://example.pagerduty.com/teams/PWY2HS1"}}
```


Description of PagerDuty normalized attributes

The following table provides details about the attributes that are expected in the normalized data that results from mapping and transforming the raw PagerDuty data.

Normalized attributes for PagerDuty event data

PagerDuty attribute	Normalized attribute	Description
`created_at`	`timestamp`	Epoch timestamp of the event in the log entry
`created_at`	`utc_timestamp`	Timestamp of the event in the log entry, in the format `"yyyy-mm-ddTHH:MM:SSZ"`
`summary`	`alert.title`	Title of the entry created
`summary`	`alert.text`	Describes the condition and the affected managed object instance
`created_at`	`alert.created_at_utc_timestamp`	Time the entry was created
`resolved_at`	`alert.resolved_at_utc_timestamp`	Time the entry was resolved
`severity`	`alert.severity`	Indicates the severity level from 1 (indeterminate) to 5 (critical)
`html_url`	`alert.source.source_url`	URL of the event in PagerDuty
`id`	`alert.source.source_alert_id`	Internal PagerDuty ID assigned to the event
`team.id`	`alert.source.source_team.id`	Internal PagerDuty ID associated with the team that is assigned to the event. Team data is added to the original PagerDuty payload with an extra API call
`team.summary`	`alert.source.source_team.name`	Name of the team that is assigned to the event
`team.html_url`	`alert.source.source_team.web_url`	URL of the team in PagerDuty
`service.id`	`alert.source.source_application.id`	Internal PagerDuty ID associated with the service that is assigned to the event

PagerDuty attribute	Normalized attribute	Description
`service.summary`	`alert.source.source_application.name`	Name of the service that is assigned to the event
`service.html_url`	`alert.source.source_application.web_url`	URL of the service in PagerDuty
`instance.id`	`alert.features.<id:pagerduty_incident>.value.id`	Internal PagerDuty ID associated with the incident that is assigned to the event `features` is an array. `incident.id` goes into the `value.id` field of the object with `id:pagerduty_incident`
`instance.summary`	`alert.features.<id:pagerduty_incident>.value.name`	Name of the service that is assigned to the alert
`instance.html_url`	`alert.features.<id:pagerduty_incident>.value.web_url`	URL to the incident in PagerDuty
`<entire raw PagerDuty JSON>`	`alert.features.<id:root>.value`	The entire raw PagerDuty JSON goes into the value field of the object with `id:root`

Example of normalized PagerDuty event

```
{
  "application_group_id": "123abc",
  "application_id": "123abc",
  "timestamp": 1581531561.4586372,
  "utc_timestamp": "2020-02-12 18:19:21.458637",
  "type": "alert",
  "alert": {
    "alert_id": "7f83fb17-5a02-8c86-020d23cfe01c",
    "title": "A failure occurred invoking the provision callback on conversation",
    "text": "A failure occurred invoking the provision callback on conversation",
    "created_at_utc_timestamp": "2019-12-09T11:20:46-05:00",
    "resolved_at_utc_timestamp": null,
    "severity": 4,
    "source": {
      "source_name": "PagerDuty",
      "source_url": "https://<pagerdutyserver.com>/alerts/PBCAOOO",
      "source_alert_id": "PBCAOOO",
      "source_team": {
        "id": "P8FWLZZ",
        "name": "SRE - Assistant",
        "web_url": "https://<pagerdutyserver.com>/teams/P8FWLZZ"
      },
      "source_application": {
        "id": "P67C6KK",
        "name": "FirstTest",
        "web_url": "https://<pagerdutyserver.com>/services/P67C6KK"
      }
    },
    "features": [
      {
        "id": "pagerduty_incident",
        "value": {
          "id": "PGAC9NN",
          "name": "[#291477] A failure occurred invoking the provision callback on conversation",
          "web_url": "https://<pagerdutyserver.com>/incidents/PGAC9NN"
        }
      },
      {
        "id": "root",
        "value": {
          "id": "PBCAOOO",
          "type": "alert",
          "summary": "A failure occurred invoking the provision callback on conversation",
          "self": "https://<pagerdutyapi>/alerts/PBCAOOO",
          "html_url": "https://<pagerdutyserver.com>/alerts/PBCAOOO",
          "created_at": "2019-12-09T11:20:46-05:00",
          "status": "triggered",
          "resolved_at": null,
          "alert_key": "mni-dedup-key-2-2019-12-09-17:20:34",
          "suppressed": false,
          "service": {
            "id": "P67C6KK",
            "type": "service_reference",
            "summary": "FirstTest",
            "self": "https://<pagerdutyapi>/services/P67C6KK",
            "html_url": "https://<pagerdutyserver.com>/services/P67C6KK"
          },
          "severity": "info",
          "incident": {
            "id": "PGAC9NN",
            "type": "incident_reference",
            "summary": "[#29147766] A failure occurred invoking the provision callback on conversation",
            "self": "https://<pagerdutyapi>/incidents/PGAC9NN",
            "html_url": "https://<pagerdutyserver.com>/incidents/PGAC9NN"
          },
          "first_trigger_log_entry": {
            "id": "R4Y7P9Y8HTZ9VRPJ",
            "type": "trigger_log_entry_reference",
            "summary": "Triggered through the API",
```

```
            "self": "https://<pagerdutyapi>/log_entries/R4Y7P9Y8HTZ9VRPJ",
            "html_url": "https://<pagerdutyserver.com>/alerts/PBCAOOO/log_entries/R4Y7P9Y8HTZ9VRPJ"
          },
          "body": {
            "contexts": [],
            "details": null,
            "cef_details": {
              "client": "curl",
              "contexts": [],
              "dedup_key": "mni-dedup-key-2-2019-12-09-17:20:34",
              "description": "A failure occurred invoking the provision callback on conversation",
              "event_class": "Test",
              "message": "A failure occurred invoking the provision callback on conversation",
              "mutations": [],
              "service_group": "app-testing",
              "severity": "info",
              "source_component": "manual-test",
              "source_origin": "000.00.000.000",
              "version": "1.0"
            },
            "type": "alert_body"
          },
          "integration": {
            "id": "P02BNYY",
            "type": "events_api_v2_inbound_integration_reference",
            "summary": "AppCluster",
            "self": "https://<pagerdutyapi>/services/P67C6GI/integrations/P02BNYY",
            "html_url": "https://<pagerdutyserver.com>/services/P67C6GI/integrations/P02BNYY"
          },
          "privilege": null,
          "team": {
            "id": "P8FWLZZ",
            "type": "team",
            "summary": "SRE - Assistant",
            "self": "https://<pagerdutyapi>/teams/P8FWLZZ",
            "html_url": "https://<pagerdutyserver.com>/teams/P8FWLZZ"
          }
        }
      }
    ]
  },
  "meta_features": []
}
```


Performance considerations for metric data collection

When configuring an integration in IBM Cloud Pak for AIOps that collects metric data such as AppDynamics, AWS CloudWatch, Dynatrace, Instana, New Relic, Splunk, and Zabbix integrations, it is important to consider the amount of data on the target system when selecting metrics.

If you have an exceptionally large amount of data on the selected target system, there can be relatively higher resource usage from collecting data from that system when compared to a system with less data.

Filtering capabilities table

The types of filtering available for different integrations are indicated here. No target system supports all filtering options, which is why they are different for each integration.

Filtering capabilities matrix

Metric integration name	Metrics filtering	Host filtering	Namespace filtering	Application filtering	Technology filtering	Index filtering
AppDynamics	X			X		
AWS CloudWatch	X		X			
Dynatrace	X				X	
Instana	X				X	
New Relic	X			X		
Splunk	X	X				X
Zabbix	X	X			X	

For target systems with a high amount of data, consider the following actions to ensure your integration works as expected:

- Do not select all metrics. Instead, review the documentation for the target system you are pulling data for and select specific metrics that are useful. Not all metrics are suited to all use cases.
- Do not select all hosts for a particular integration. If you have thousands of hosts, it can be better to create multiple integrations instead of one integration with all your hosts.
- Do not create more than 10 integrations on a small cluster or 20 integrations on a large cluster at any one time. The more integrations that you have running simultaneously, the more computing resources are needed.
- If the Kafka persistent volume claims (PVCs) near capacity, increase each of the PVCs by an extra 60 GB. For more information about adjusting the size of the PVC, see Increasing Kafka PVC.

Note: For production deployments, the default Kafka PVC is 30 GB, but a minimum of 90 GB is recommended. This adjustment must be made after installing the cluster.

- If available within the console for the integration you are configuring, you can reduce the length of time for historical data collection, reduce the base parallelism, or select a smaller span value in order to reduce the resource usage of the integration.

Examples of performance considerations when collecting data:

- A Dynatrace integration taking too long to collect historical data, with unstable and inconsistent time frames, for example taking 4 days to collect 5 minutes of data.
- An AWS CloudWatch integration collecting more data than the capacity of the metric anomaly detection of the system, for example collecting 100,000 KPIs per 5 minutes while the cluster's sizing supports only 30,000 KPIs per 5 minutes with only 5 metrics selected. The solution to this would be to scale the sizing of the cluster to accomodate more metric data.
- A Zabbix integration pod restarts on a small Red Hat OpenShift Container Platform environment collecting historical metric data and shows an "Unknown" status. The Kafka PVC on this cluster has only 30 GB of storage on a production environment and it is at maximum capacity and the iaf-kafka pod is in a restarting status as well. The solution to this would be to increase the size of the Kafka PVC per the preceding recommendations and to restart the connector-bridge, iaf-kafka, and integration pods.

Planning data loading, checking, and training

You can load and validate data for testing for AI training and operational purposes.

Loading AI training data

When you are loading AI training data, such as to correlate events from various sources, or find anomalies in logs, use the following end-to-end guidance for best results.

1. For each AI type, load *all* relevant historical data. For example, if you are training the log anomaly AI type, first load *all* relevant historical log data by using one or more of the following data integrations. Set the integration mode to **Historical data for initial AI training** at the end of the page where you define the data integration.

   - Custom
   - ELK
   - Falcon LogScale
   - Kafka
   - Mezmo
   - Splunk
2. Before you run the training, you can optionally run a data quality precheck on your loaded data to ensure that it is sufficient enough to train on the algorithm selected. Even if you skip this step to run the model training, the data quality precheck still runs as a prerequisite to the model training.
3. Train the AI type, as described in Configuring AI training, and use *all* of the data that you loaded in Step 1. For example, if you are training the log anomaly AI type then follow the instructions to create a log anomaly training definition.
4. Deploy the model for the AI type that you trained in Step 3. Deployment can be automatic or manual.
5. For log anomaly training only, after the model deploys, go back to each data integration that you defined in step 1. Next, change the integration mode to **Live data for continuous AI training and anomaly detection**.

Loading operational data

When you are loading operation data, such as for real-time analysis, use the guidance within the following documentation:

- Incoming integrations

  Incoming integrations enable IBM Cloud Pak for AIOps to collect log, metric, and event data from various sources.
- Connecting probes and gateways

  If you want to load event data, you can also configure probes and gateways form various sources.

Prerequisites for loading data from domain-specific log data sources

If you configure a log integration to load data from domain-specific log data sources, such as IBM MQ or WebSphere, you must meet specific prerequisites that are different from other log systems.

Domain-specific logs must be sourced from a log management system. You must create an incoming log data integration to collect this data. The type of integration to create depends on the following factors:

- Whether you prefer to provide the data in pull or push mode
- Which log management system stores the domain-specific log data

Modes

Consider the following issues when you are deciding whether to provide the data in pull or push mode.

- Pull

    - Positives:
        - If your log management system has a REST API then all you must do is to create the data integration and Cloud Pak for AIOps does the rest.
        - You can also use the Cloud Pak for AIOps built-in integrations, which can cover commonly used options.
    - Negatives: you have no control over data load. When Cloud Pak for AIOps initiates the pull operation, the operation might create excessive load on your log management systems.
- Push
    - Positives: you have full control over data load.
    - Negatives: you must create code to push your log data to the Kafka topic.

Pull mode

For log data from systems that are not Falcon Logscale, Mezmo, or Splunk, you can make use of either the Custom, ELK, or Kafka data integration to ingest your log data.

Pull mode integrations

Domain-specific log management system	Data integration	Link
Falcon LogScale	Falcon LogScale	Falcon LogScale integrations
Mezmo	Mezmo	Mezmo integrations
Splunk	Splunk	Splunk integrations
Log management system that uses an ELK stack	ELK	Custom integrations
Any other log management system	Custom	Custom integrations

Push mode

For log data from systems that are not Falcon Logscale, Mezmo, or Splunk, you can use either the Custom, ELK, or Kafka data integration to ingest your log data.

Push mode integrations

Domain-specific log management system	Data integration	Link
Any system	Kafka	Custom integrations

Other requirements

Log entries must be in JSON format, and not in plain text.

When you are preparing domain-specific log data for loading, ensure that the **`ibm_messageId`** and **`loglevel`** fields are at the top level. The **`module`** field is optional. These fields are used by the statistical baseline log anomaly detection algorithm to perform statistical analysis and detect log anomalies.

Deduplication

Occurrences of the same event are deduplicated into a single alert instance.

The default correlation policy constructs a deduplication key for events from the following elements:

```
{{ event.resource }}-{{ event.type.classification }}-{{ event.type.condition }}
```

When the same event occurs multiple times (the resource, classification, and condition fields are the same), then deduplication increments the event count of the current alert instance to show how many times the event occurred.

Problem events (**`type.eventType = problem`**) create an alert (if one does not exist). Meanwhile, resolution events (**`type.eventType = resolution`**) do not become alerts and are not visible in the Alert Viewer. They clear an alert if one exists and ignore if none exists. For a resolution event to clear a problem alert instance, it must have the same deduplication key as the problem event that the alert was created from. Also, it must have occurred after the last problem event that contributed to the alert.

Type fields and alert correlation

Since the same alert arises for all events with same type fields, these fields must clearly indicate the specific fault condition and have separate values for different conditions. The resource fields are important for both alert correlation and topology lookups. Therefore, they must accurately describe the resource that exhibits the fault condition. By default, alerts with the same **`resource.name`** field that occur within 15 minutes of each other are correlated together.

Additionally, all resource fields are used to look up each alert in the topology. Therefore, where possible one or more of these fields must be in the same form as the match tokens for the corresponding topology resource.

Alerts are also correlated by using temporal correlation. Temporal correlation looks for historic patterns of co-occurrence and uses this information to predict future correlation. This correlation relies on consistent value of the type fields over time for the same conditions against the same resources. Therefore, the values of these fields must be deterministic.

HTTP header credential types

The following table shows the different HTTP headers for each credential type.

Note: Different integrations use different credentials, which are outlined in each of the relevant integration topics.

Credential type headers

Credential Type	HTTP Header
API Key	Authorization: `api-key-value-pair` For example, X-API-Key: `abcdef12345`
API Token	Authorization: Bearer `token`
UserID/Password	Authorization: Basic `username:password`

Note: The final value of `username:password` in the Basic header is a base64-encoded value.

Remote deployment of integrations

You can deploy an integration locally or remotely. The local deployment option deploys the integration in the same cluster where IBM Cloud Pak for AIOps is installed. Alternatively, the remote option deploys it in your chosen location, for example, on a different network region, on Saas, or remotely on premises (VM, cluster, or container).

You can choose the remote option for deployment if the integration source is not reachable from the IBM Cloud Pak for AIOps cluster. On selecting the remote option, you will be provided with a bootstrap script after you add the integration to finish the deployment. Run this script on the remote server where you want to deploy the integration.

The bootstrap command creates a secure tunnel between the IBM Cloud Pak for AIOps cluster and the remote cluster on which you are running the integration source. This tunnel feeds the data from the remote server to your IBM Cloud Pak for AIOps cluster.

Prerequisites

Before you run the bootstrap script on a remote cluster, complete the following tasks:

- Install Podman on the remote cluster.
- Log in to Image registry for the script to pull in the integration image by using the Podman login command.

Note: You do not need Red Hat OpenShift running on the remote cluster for deploying the integration remotely.

Procedure

To deploy an integration remotely, complete the following steps:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. From the list of available integrations, find and click the tile for the integration that you want to deploy remotely.

   Note: If you do not immediately see the integration that you want to create, you can filter the tiles by type of integration. Click the type of integration that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add integration* page, go to the *Deployment options* and select **Remote**.
7. Continue with procedure for creating the integration.
8. After the integration is created, select the created integration on the *Integrations* page.
9. Download the bootstrap script by clicking **Download** from the options menu (three vertical dots) for the integration. Copy the script or download the code as an `sh` file.
10. Run the script on the remote cluster where you want to deploy the integration. When the script completes running, the integration is successfully deployed on the remote server.

Defining observer jobs for application and topology data

You can configure observer jobs for collecting application and topology data by using the Observer Configuration UI.

Note: The restriction on the number of concurrent observer jobs has been lifted. Any jobs that are started via the UI or the Observer Service Swagger page that exceed the set threshold of concurrently running jobs (ten) will not be discarded, but instead queued and run as soon as previous jobs complete. Should a scheduled job be prevented from running in such a scenario, it will not fail, though it might run later than scheduled.

Before you begin

For more information on configuring certificates and passwords, see the Configuring observer jobs security topic.

Note: The following observers are enabled by default:

- ServiceNow observer*
- Kubernetes observer*
- VMware vCenter observer*
- File observer
- REST observer

*If you have any restrictive EgressNetworkPolicies in place when connecting to these observer target systems, ensure that they are updated to allow for this inbound connection. For more information, see Postinstallation tasks.

To add other observers

1. In the Operator Lifecycle Management console, go to **Operators > Installed Operators**.
2. Choose the project where IBM Cloud Pak® for AIOps is installed from the **Project** drop-down list.
3. Click the IBM Netcool Agile Service Manager operator.
4. Navigate to the Agile Service Manager tab and select the existing Agile Service Manager instance (such as aiops-topology).
5. Click the YAML tab and edit the file. To enable an observer, set it to `true`, then click **Save**. (**Tip:** To disable it, set it to `false`.)

To enable observers

1. Change to the namespace (project) where IBM Cloud Pak for AIOps was installed:

   ```
   oc project
   ```

2. Find the ASM release name using the following command:

   ```
   oc get asm
   ```

   Example:

   ```
   oc get asm aiops-topology
   ```

   ```
   NAME             AGE
   aiops-topology   20h
   ```

3. Edit `aiops-topology`:

   ```
   oc edit asm aiops-topology
   ```

   Change the value to `true` for the observers that you want to enable. For example:

   ```
   asm:
     almObserver: false
     ansibleawxObserver: true
     appdynamicsObserver: true
     awsObserver: true
     azureObserver: true
   ```

4. To check all observers have been enabled run the following command:

   ```
   oc get pod | grep aiops-topology
   ```

Tips:

- To disable an observer, set the associated observer parameter to `false` in the ASM release name.
- To verify that the ASM release name has the status of `OK` after the change:

  ```
  oc get asm aiops-topology --no-headers -o custom-columns=":status.phase"
  ```

- To verify that the ASMformation release name has the status of `OK`:

  ```
  oc get asmformation aiops-topology --no-headers -o custom-columns=":status.phase"
  ```

- To manually delete the kubernetes deployment for that observer to free up resources:

  ```
  oc delete deployment -l app.kubernetes.io/name=<observer_name>
  ```

Example

```
oc delete deployment -l app.kubernetes.io/name=alm-observer
```

```
deployment.apps "aiops-topology-alm-observer" deleted
```

To access the Observer Configuration UI

1. Expand **Define**, click **Integrations**, and then click the **Manage observer jobs** tab.
2. Click **Configure, schedule, and manage observer jobs**. The Observer jobs page is displayed.
3. Here, you can add a new job, or view, search for, or sort existing jobs by job name, observer name, job status or job type.

Observer jobs

The Observer jobs panel displays all existing jobs in table format. From here, you can interact with existing jobs, or create new ones.

If you have a large number of jobs configured, you can use **Search** to find it.

All existing jobs are listed in a table by name, observer name, job status or job type columns, which are sortable. For each job its state is indicated. Hover over the state to see a tooltip with more information.

If a job has been scheduled, you can hover over the Schedule icon to view more details. The Schedule icon is either a clock icon if scheduled to run at set intervals, or an hourglass icon if it is scheduled to run only once at a future time. When you cancel a scheduled job (Load jobs only) using **Remove Schedule**, this reverts the job to a default, on-demand job.

You can run a job manually, or switch it On or Off, depending on the job type.

You can also use the **List of options** drop-down to **View & Edit** or **Delete** a job, or **View History** for it (View History is available for Load jobs only).

When you **Delete** a running job, it will not complete its current run, but any previously loaded data will remain. You cannot delete a job that is in the 'Finishing' state.

When you select **View History** for a job, the Observer job history window opens listing the last 10 jobs, when they were run, their observation count, duration, and status. You can refine the list using the **All**, **Complete**, or **Incomplete** tabs.

Click the **Add a new job +** button to open the *Add a new job* panel, which displays all jobs that can be configured in tile format. Click a specific observer tile to open its job configuration window. The Observer job configuration UI lists each job parameter that you must configure. See the topic specific to the observer job you are configuring for more information.

New 'Generate debug support file' parameter: Set the optional *Generate debug support file* parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer (from **`/var/log/itsm/<obs>-observer`**). **Note** that this functionality is available only from the Observer job configuration UI.

Tip: If an observer stops or is removed, you cannot run existing jobs, or add new jobs. Stopped or removed observers and jobs that are listed in the Observer Configuration UI will be disabled or removed in progressive (housekeeping) steps. If you are reinstalling or reactivating an observer, the jobs and the observer will again become available.

1. **Up to 5 minutes** after removal, observers and jobs still appear as normal until the housekeeping process runs, but cannot be used.
2. **Up to 60 minutes** after removal, the observer is still listed, but jobs are unavailable and marked offline until the next housekeeping process runs. You can delete existing jobs, but cannot view, add or edit jobs.
3. **After 60 minutes** the removed observer is no longer listed, **but jobs remain**, though they are disabled and marked offline. You can delete existing jobs, but cannot view, add or edit jobs.
4. If **at any time** you reinstall or reactivate the observer, it reappears in the UI, and existing (previously active) jobs are no longer disable. You can delete, view or edit existing jobs, or add new jobs.

Configuring ALM Observer jobs

You can define jobs that dynamically load data associated with intent for analysis.

Configuring Ansible AWX Observer jobs

Using the Ansible AWX Observer, you can define load jobs that will gather and read data from a defined Ansible AWX instance. When a load job is run, it will load a baseline topology data view from Ansible AWX through its REST APIs.

Configuring AppDynamics Observer jobs

Using the AppDynamics Observer, you can define a full load job that will read data from the AppDynamics Controller through the REST API. This job can provide, for example, business applications, nodes, and tiers.

Configuring AWS Observer jobs

Using the AWS Observer, you can define jobs that read services data from the Amazon Web Services (AWS) through AWS SDK and generate a topology. It is installed as part of the core installation procedure.

Configuring Azure Observer jobs

Using the Azure Observer, you can define a full load job that will read data from Azure cloud services through its REST APIs and generate a topology.

Configuring Big Cloud Fabric Observer jobs

Using the Big Cloud Fabric Observer, you can define a full load job that will load data from the Big Cloud Fabric REST interface..

Configuring BigFix Inventory Observer jobs

You configure BigFix Inventory Observer jobs to read data from a BigFix Inventory instance through its REST APIs, and generate a topology.

Configuring Ciena Blue Planet Observer jobs

Using the Ciena Blue Planet Observer, you can define jobs that will gather and read all topology data from the Blue Planet MCP instance by REST API and generate a topology.

Configuring Cisco ACI Observer jobs

You use the Cisco ACI Observer when you have a Cisco ACI environment with Cisco Application Policy Infrastructure Controller (APIC) in your environment. The Observer interfaces with Cisco APIC and makes active REST calls to Cisco APIC in the Cisco ACI environment. You configure observer jobs that dynamically load Cisco ACI data for analysis from the Observer Configuration UI.

Configuring Contrail Observer jobs

Using the Contrail Observer, you can retrieve topology data from Juniper Network Contrail Release 4.1 through REST APIs exposed by the Contrail apiserver. This observer is developed against Juniper Network Contrail that integrates with OpenStack orchestration platform (Ubuntu 18.04 + Contrail Cloud - Ocata).

Configuring DNS Observer jobs

Using the DNS Observer, you can query internal DNS server performance, and use the returned information on response times and service addresses to create topologies within the topology service. The DNS Observer supports forward and reverse job types, with 'recurse' or 'no recurse' options.

Configuring Docker Observer jobs

Using the Docker Observer, you can discover Docker network resources, including Docker Swarm clusters, and then visualize (or model) this data as a topology view in the UI. You configure observer jobs from the Observer Configuration UI.

Configuring Dynatrace Observer jobs

Using the Dynatrace Observer, you can query a specified Dynatrace environment for information about its applications, services, process groups, and infrastructure entities.

Configuring File Observer jobs

Using the File Observer functionality, you can write bespoke data to a file in a specific format. You can upload this data to the topology service, and then visualize this data as a topology view in the UI.

Configuring GitLab Observer jobs

Using the GitLab Observer functionality, you dynamically load GitLab projects and their child resources to the topology service, and then visualize this data as a topology view in the UI.

Configuring GoogleCloud Observer jobs

Using the GoogleCloud Observer, you can define a full load job that will read services data from the Google Cloud Platform Compute Services through Google's Compute Services SDK, and then generate a topology.

Configuring HP NFVD Observer jobs

Using the HP NFVD Observer, you can dynamically load data from the HP NFVD REST interface.

Configuring IBM Cloud Observer jobs

Use the IBM Cloud Observer when you install IBM Cloud in your environment to run jobs that read data from an IBM cloud instance. These jobs retrieve Cloud Foundry Apps information and services, and then dynamically load the retrieved data for analysis.

Configuring Jenkins Observer jobs

Using Jenkins Observer, you can define listen jobs that receive build information generated by the plug-in for Jenkins.

Configuring Juniper CSO Observer jobs

Using the Juniper CSO Observer, you can define a full load job that will gather and read data about topology data from Juniper CSO. The observer is installed as part of the core installation procedure.

Configuring Kubernetes Observer jobs

Using this observer, you can configure jobs that discover the structure of your Kubernetes clusters, including pods, worker nodes and containers.

Configuring Network Manager Observer jobs

Using the IBM Tivoli Network Manager Observer, you can define jobs that dynamically load data discovered by Network Manager for analysis.

Configuring New Relic Observer jobs

Use New Relic Observer when you have a New Relic account with a New Relic Infrastructure subscription. Using New Relic Observer, you can configure jobs that dynamically load New Relic Infrastructure resource data through New Relic for analysis.

Configuring OpenStack Observer jobs

Using the OpenStack Observer, you can configure jobs that dynamically load OpenStack data for analysis.

Configuring Rancher Observer jobs

Using the Rancher Observer, you can define a full load job that will read data from Rancher through the REST API and generate a topology.

Configuring REST Observer jobs

Use the REST Observer for obtaining topology data through REST endpoints. This observer is a counterpart to the File Observer.

Configuring SDC ONAP Observer jobs

You configure SDC (Service Design and Creation) ONAP (Open Network Automation Platform) Observer jobs to dynamically load data from the SDC REST interface.

Configuring ServiceNow Observer jobs

Using the ServiceNow Observer job, you can retrieve the configuration management database (CMDB) data from ServiceNow through REST API, using basic authentication credentials.

Configuring SevOne Observer jobs

Using the SevOne Observer functionality, you can load monitored servers and their relationships, and then visualize this data as a topology view.

Configuring Tivoli Application Dependency Discovery Manager Observer jobs

IBM Tivoli Application Dependency Discovery Manager Observer jobs retrieve network topology data, including discovered applications, their components, configurations and dependencies, from the Tivoli Application Dependency Discovery Manager database server (running either an IBM Db2 or an Oracle database), and use this data to create topology views.

Configuring Viptela Observer jobs

You configure Viptela Observer jobs to load managed Viptela SD-WAN control plane, vEdge and their associated network resources.

Configuring VMware NSX Observer jobs

You configure VMware NSX Observer jobs to dynamically load data from the VMware NSX REST interface.

Configuring VMware vCenter Observer jobs

You configure VMware vCenter Observer jobs to dynamically load data from the VMware vCenter REST interface.

Configuring Zabbix Observer jobs

Using the Zabbix Observer functionality, you can load monitored servers and their associated network resources, and then visualize this data as a topology view in the UI. It is installed as part of the core installation procedure.

Configuring ALM Observer jobs

Using the Agile LifeCycle Manager Observer, you can define jobs that dynamically load data associated with intent from the Agile LifeCycle Manager for analysis.

Before you start

Important: The ALM Observer supports IBM Agile Lifecycle Manager 2.2.

Attention: The ALM observer supports the discovery of Cloud Pak for Network Automation (CP4NA) and has been tested against versions 2.3.x and 2.4.x. The ALM observer was first developed to integrate with the legacy Telco Network Cloud Manager (TNCM) Orchestrator (TNCO/ALM). A re-branding from ALM (Agile Lifecycle Manager) to CP4NA happened, while the observer retained its ALM name. The observer has some level of backward compatibility with the legacy ALM.

Ensure you have the Agile LifeCycle Manager Kafka server host and topics to hand, such as the Agile Lifecycle Manager server, the Kafka port, and the topics used for lifecycle events.

Important: To access Agile LifeCycle Manager remotely, you must ensure that the Agile LifeCycle Manager installation is configured with the KAFKA_ADVERTISED_HOST_NAME so as to allow remote connections. For more information, see the Configuration reference topic in the Agile LifeCycle Manager documentation: Configuration reference

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The Agile LifeCycle Manager Observer jobs listen to the Kafka 'state change' topics of Agile Lifecycle Manager, as well as the Agile Lifecycle Manager Resource Manager. Information is extracted from Agile Lifecycle Manager about Assemblies and Resources and a topology is created.

After installation, you define and start the following two jobs.

Listen for Agile Lifecycle Manager events ('alm' job)

The **alm** job is a long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the observer is stopped.

Listen for Resource Manager events ('rm' job)

The **rm** job is a long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the observer is stopped.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Important: The value of the IBM Agile Lifecycle Manager instance name parameter needs to be the same for both jobs to allow the topology to be combined.

Listen for lifecycle events ('alm' job) parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
IBM Agile LifeCycle Manager instance name	Use this to identify the Agile LifeCycle Manager installation and any associated Resource Managers	Required

Parameter	Action	Details
Group ID	Use this to identify the Kafka group ID to be used	Required
Connection	Use this to specify the Kafka Host and Port to be used	Required
ALM State Change Topic	Use this to identify the Agile LifeCycle Manager Kafka topic name to be consumed.	Required
ALM Service State Transition Topic	Use this to identify the Agile LifeCycle Manager State Transition Kafka topic name to be consumed.	Required
Secured or unsecured Kafka connection	Select false or true to make secured or unsecured connection to the CP4NA server. The default is 'false'.	Required. When set to 'true', you must also provide the credentials and target system certificate parameters. See the following topic for more information on obtaining connection and credential details: IBM Cloud Pak for Network Automation: Configuring Kafka connections
Kafka username	Enter the username of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'.
Kafka password	Enter the password of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'.
Certificate	Enter the certificate of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'. For more information, see Configuring observer jobs security
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Listen for Resource Manager events ('rm' job) parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
IBM Agile Lifecycle Manager instance name	Use this to identify the Agile LifeCycle Manager installation and any associated Resource Managers	Required
Group ID	Use this to identify the Kafka group ID to be used	Required
Connection	Use this to specify the Kafka Host and Port to be used	Required
Topic	Use this to identify the Agile Lifecycle Manager **Resource Manager** Kafka topic	Required
Secured or unsecured Kafka connection	Select false or true to make secured or unsecured connection to the CP4NA server. The default is 'false'.	Required. When set to 'true', you must also provide the credentials and target system certificate parameters. See the following topic for more information on obtaining connection and credential details: IBM Cloud Pak for Network Automation: Configuring Kafka connections
Kafka username	Enter the username of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'.
Kafka password	Enter the password of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'.
Certificate	Enter the certificate of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'. For more information, see Configuring observer jobs security
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Ansible AWX Observer jobs

Using the Ansible AWX Observer, you can define load jobs that will gather and read data from a defined Ansible AWX instance. When a load job is run, it will load a baseline topology data view from Ansible AWX through its REST APIs.

Before you begin

Important: The Ansible AWX Observer supports the on-premises Ansible AWX version 21.14.0 and Tower API version 3.8.6.

Ensure you have the Ansible AWX details to hand, such as the host, port, user and token before defining and running the observer job.

Required: Before defining the observer load job, you must obtain a personal access token (PAT) from AWX. The token authentication is recommended for the usage of Tower's API.

1. Access the AWX UI.
2. From the menu bar, navigate to the **Users** page that is under the **Access** section.
3. Select a username, then click the **Token** option.
4. Click the **+** (plus) icon.
5. Leave the application field blank, and choose **read** for the token's access.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Full Topology Upload job (through restapi)

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job.	Required
Ansible AWX hostname	Specify the Ansible AWX hostname	Required
Ansible AWX host port	Specify the Ansible AWX host port	Required
Username (owner) of the personal access token (PAT)	Specify the username (owner) of the personal access token (PAT)	Required
Personal access token (PAT)	Specify the personal access token (PAT)	Required.
HTTPS truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `Ansible.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
Target system certificate	Specify a certificate by name to automatically create and load into the truststore	Required
SSL verification	Choose whether SSL validation is true or false. Turning SSL validation to `false` will use HTTPS without host verification.	Optional
Number of jobs to display for each template	Specify the number of recent jobs that will be modeled on the topology for each of the job template. Max value is 50. Larger value might result in the topology being less perceptible	Optional
Connection timeout (milliseconds)	Enter a connection timeout in milliseconds	Optional. Must be a value greater than 0 (zero), and the default is `5000` (5 seconds).
Read timeout (milliseconds)	Enter a read timeout in milliseconds	Optional

Parameter	Action	Details
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring AppDynamics Observer jobs

Using the AppDynamics Observer, you can define a full load job that will read data from the AppDynamics Controller through the REST API. This job can provide, for example, business applications, nodes, and tiers.

Before you start

Important: The AppDynamics Observer supports the cloud and SaaS AppDynamics version 4.5.12 and API version 4.5.x.

Ensure you have the AppDynamics details to hand, such as the instance, account, username and password before running the observer job.

Tip: Before defining the observer load job, you must create an AppDynamics user with the correct permissions. This is required for REST API authentication.

1. On the AppDynamics Administration page, click the gear icon (upper right).
2. Under the Admin group, select **Administration**, then select the **Users** tab.
3. Select the **AppDynamics** option from the **Display users** drop-down list.
4. Assign all available roles to the user by selecting **Add from Roles** > **Select All** > **Done**, then click **Save**.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The AppDynamics Observer imports ITSM Resource Topology Service data.

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
AppDynamics Controller instance	Specify the AppDynamics Controller instance	Required
Tenant account name	Specify the Tenant account name	Required
Username	Specify the name of the user of the specified tenant account	Required
Password	Specify the password for the specified user	Required
HTTPS instance target system	Set to `true` to discover HTTPS instance, and to `false` for HTTP instance.	Optional
SSL Verification	Choose to use SSL validation or not. The options are `true` or `false`.	Optional. The default is 'false', which uses HTTPS validation without hostname validation.

Parameter	Action	Details
Connection timeout	Specify the connection timeout in milliseconds.	Optional. Default is 5000 (5 seconds).
Read timeout	Specify the read timeout in milliseconds.	Optional. Default is 5000 (5 seconds).
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring AWS Observer jobs

Using the AWS Observer, you can define jobs that read services data from the Amazon Web Services (AWS) through AWS SDK and generate a topology. The AWS Observer supports multiple Amazon web services such as EC2 for its 'elastic compute' services.

Before you begin

Important: The AWS Observer supports the cloud and SaaS AWS version 1.11.

The AWS Observer supports EC2 service and currently processes EC2 resources in the following order. When the account in use has insufficient privileges to access a resource, the observer skips the resource(s) and proceed to the next one.

- Instance
- AvailabilityZones
- NetworkInterfaces
- Volumes
- SecurityGroups
- VPCs
- Subnets
- RouteTables

Important: As a minimum, the account in use must have the DescribeInstances policy, or the observer job fails. For more information, see Identity and access management for Amazon EC2.

About this task

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Job data tenant	Specify the unique data tenant name to track the discovered data for each job.	Required
AWS Access Key ID	Specify the AWS access key. **Tip:** You can find your access key ID using AWS's IAM console, the AWS CLI, or the AWS API. See the AWS Managing access keys for IAM users website for more information.	Required
AWS Secret Access Key	Specify the AWS secret key. For more information, see the AWS Managing access keys for IAM users website	Required
AWS Region	Specify the AWS region or multiple regions to discover	Required

Parameter	Action	Details
Proxy	Specify the proxy (`true` or `false`)	Required. If set to `true`, you must supply the proxy host, port and protocol.
Proxy Host	Specify the proxy host	Optional (unless proxy is set to `true`)
Proxy Port	Specify the proxy port	Optional (unless proxy is set to `true`)
Proxy Protocol	Specify the proxy protocol	Optional (unless proxy is set to `true`)
Property to exclude	Single or multiple properties to exclude	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Azure Observer jobs

Using the Azure Observer, you can define a full load job that will read data from Azure cloud services through its REST APIs and generate a topology.

Before you begin

Important: The Azure Observer supports the cloud and SaaS Azure version.

Ensure you have the Azure details to hand, such as the Tenant ID, Client ID, and client password before running the observer job.

Tip: To configure access to Azure resources, see the following information: Create an Azure Active Directory application and service principal that can access resources

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The Azure Observer retrieves topology data from Azure cloud services through REST APIs exposed by the Azure API server.

In addition to the standard resources, the following resources are now also supported:

- Load Balancers
- Virtual Machine Scale Set
- Virtual Network Gateway (including VPN Gateway and Express Route Gateway)
- ExpressRoute

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required

Parameter	Action	Details
Data center	Enter the data center of the Azure service	Required
Tenant ID	Specify the Azure account tenant ID	Required
Client ID	Specify the client ID to access the registered app	Required
Client password	Enter the client password to access the registered app	Required
Subscriptions to be discovered	Specify the subscription Ids to be discovered in comma-separated lists. To discover all subscriptions, use `*`	Optional
Connection timeout (milliseconds)	Define the Azure connection timeout in milliseconds. The default value is 5000.	Optional
Read timeout (milliseconds)	Specify the read timeout in milliseconds. The default value is 5000.	Optional
Proxy Host	Specify the proxy host	Optional
Proxy Port	Specify the proxy port; defaults to 8080	Optional
Proxy Username	Specify the proxy username.	Optional
Trust all certificates by bypassing certificate verification	Set to true to allow connection to target environment without verification.	Optional. The default is 'false'.
Proxy Password	Specify the proxy password.	Optional
Proxy Secure	Specify whether the proxy server is secure.	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Big Cloud Fabric Observer jobs

You configure Big Cloud Fabric Observer jobs to dynamically load data from the Big Cloud Fabric REST interface.

Before you begin

Ensure you have the Big Cloud Fabric service details to hand, such as username, password, hostname and port.

About this task

The Big Cloud Fabric Observer job extracts Big Cloud Fabric resource information through REST and loads and updates the resources and their relationships within the topology service.

You define and start the following job:

LDAP job

Run this job whenever you need Big Cloud Fabric topology data refreshed.

By default, these jobs are one-off, transient jobs that perform a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

The Big Cloud Fabric Observer loads the following resources and their relationship:

Big Cloud Fabric resources and relationships loaded by the Big Cloud Fabric Observer LDAP job

Vertex	Description
Access Leaf	Access Leaf Switch part of DC fabric
Access Port	Access Leaf Switch port connected to Compute Node and Spine Switch port
Spine Switch	Spine Switch part of DC fabric
Spine Switch Port	Spine Switch port connected to Access Leaf Switch and L3 Router Port

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

LDAP job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Proxy hostname	Specify the proxy server name	Required
Proxy Port	Specify the proxy port number for the REST API	Required
Proxy username	Specify the proxy username	Required
Proxy password	Specify the proxy password	Required. Plain text password can be entered.
BCF authentication URL	Specify the BCF authentication URL.	Required
BCF Controller Host Name	Specify the BCF controller hostname in string format. The hostname is taken from the job parameter.	Required. **Note:** The BCF controller hostname value (together with the proxy hostname and username) identifies the source of data for a Big Cloud Fabric Observer job instance. If your BCF controller hostname changes after a job is created, you must create more than one job instance for each BCF controller hostname. If the BCF controller hostname **does not** change after the job is created, then create a single job instance with all BCF controller hostnames specified.
HTTPS or HTTP connection to the BCF authentication server	Specify whether the connection to the BCF authentication server should be HTTPS (secured) or HTTP.	Required
HTTPS or HTTP connection to the BCF proxy server	Specify whether the connection to the BCF proxy server should be HTTPS (secured) or HTTP.	Required
Trust all certificates Verification	Choose whether Trust all certificates Verification is true or false.	Required. When set to true, the observer will trust all certificates.
Delay between two api calls (in seconds)	Specify the delay between retries	Optional
Concurrency Level	Specify the Concurrency Level, with a minimum of 3 and a maximum of 5.	Optional
Authentication system certificate	Specify the BCF Authentication system certificate to be used (if any).	Optional. Required if the HTTPS or HTTP connection to the BCF authentication server parameter is true. To obtain authentication certificates using OpenSSL and store them as secrets, see the Configuring observer jobs security topic.
Proxy system certificate	Specify the BCF proxy system certificate to be used (if any)	Optional. Required if the HTTPS or HTTP connection to the BCF proxy server parameter is true. To obtain proxy certificates using OpenSSL and store them as secrets, see the Configuring observer jobs security topic.
SSL Host Name Verification	Choose whether SSL validation is true or false.	Optional. When SSL validation is set to false, the observer will make HTTPS connections without validating the proxy hostname of the provided certificate.
Connection timeout (milliseconds)	Enter the time at which the connection actions time out.	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Read timeout (milliseconds)	Enter the time at which the read actions time out.	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring BigFix Inventory Observer jobs

You configure BigFix Inventory Observer jobs to read data from a BigFix Inventory instance through its REST APIs, and generate a topology.

Before you begin

Important: The BigFix Inventory Observer supports BigFix Inventory 9.5.0.

Before configuring a BigFix Inventory job, ensure you have the BigFix Inventory details to hand such as the BigFix Inventory URL, API token and SSL truststore.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

BigFix Inventory Observer job (full topology load)

Run this job whenever you need BigFix Inventory topology data refreshed.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data Center	Specify the data centers in which the BigFix Inventory instance runs	Required.
BigFix Inventory instance URL	Specify the API URL of the BigFix Inventory endpoint (including port)	Required. Usually in the following format: https://<hostname or IP address>:<port>
BigFix Inventory API token	Enter the BigFix token for authentication	Required. Must be encrypted.
HTTPS truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `BigFixInventory.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
Truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
BigFix Inventory resources	Specify the resources to be discovered	Optional. Lists supported values such as software, hardware or `*`. If left blank, all available resources are discovered.
BigFix Inventory certificate	Specify the name of the certificate to be loaded into the trust store	Required
SSL Validation	Choose whether SSL validation is on or off. Turning off SSL validation will use HTTPS without host verification.	Optional
BigFix Inventory connection timeout (ms)	Enter the time at which the connection times out	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Troubleshooting: If you configure a BigFix Inventory Observer job, the pod for the job might not start. After you configure the job, if you view the list of all observers you might not see the BigFix Inventory Observer job (**`bigfixinventory-observer`**) pod listed with a status of running.

If you encounter this issue, you can manually enable the job by changing the value of **`asm.bigfixinventoryObserver.enabled`** to true:

```
helmValuesASM:
  asm.bigfixinventoryObserver.enabled: true
```

To change this value, you can run the following command:

```
oc patch asm.asm.ibm.com/aiops-topology --type=json -p
'[{"op":"add","path":"/spec/helmValuesASM/asm.bigfixinventoryObserver.enabled","value":true}]'
```

To verify that the pod is now running, run the following command:

```
oc get pods | grep -i big
aiops-topology-bigfixinventory-observer-7f8c465698-gtm5p          1/1     Running             0          5m
```

Configuring Ciena Blue Planet Observer jobs

Using the Ciena Blue Planet Observer, you can define jobs that will gather and read all topology data from the Blue Planet MCP instance by REST API and generate a topology.

Before you begin

Important: The Ciena Blue Planet Observer supports MCP 4.0.

Ensure you have the Ciena Blue Planet details to hand, such as API username, API password, MCP URL, MCP certificate, truststore file and truststore password.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The Ciena Blue Planet Observer has two jobs, the restapi load and WebSocket listen jobs.

- When a **load** job is run, it loads baseline topology data through Blue Planet MCP APIs: Network Elements (constructs), EquipmentHolder, Equipment, TPE (Terminating Point Encapsulation), and FRE (Forwarding Relationship Encapsulation).
- When a **listening** job is run, the observer listens to changes in resource's status from the BluePlanet MCP instance through a WebSocket connection.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Full Topology Upload job (through restapi)

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
SSL truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (**`<observer>.jks`**), for example **`VMVCenter.jks`**.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
SSL truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
Data Center	The data center the MCP instance is running in	Required
Tenant Name	The tenant to use	Required
Blue Planet MCP server URL	Specify the MCP server endpoint including port and version (such as **`AUTH_URL`**)	Required
Username	MCP API username	Required
Password	MCP API password.	Required
Blue Planet MCP certificate	Certificate name to load into the truststore	Optional
SSL Validation	Choose whether SSL validation is on or off. Turning off SSL validation will use HTTPS without host verification.	Optional

Parameter	Action	Details
Connection timeout	Sets the connection and read timeout in milliseconds	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

WebSocket Listen job parameters

A long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the observer is stopped.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
SSL truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `CienaBluePlanet.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
SSL truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
Data Center	The data center the MCP instance is running in	Required
Tenant Name	The tenant to use	Required
Blue Planet MCP server URL	The URL of the MCP server instance	Required
Username	MCP API username	Required
Password	MCP API password	Required
WebSocket URL	The MCP WebSocket URL to connect to	Required
WebSocket Topics Subscriptions	The MCP topics to subscribe to	Required
Blue Planet MCP Certificate	Certificate name to load into the truststore	Optional
SSL Validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
Connection timeout	Sets the connection and read timeout in milliseconds	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Cisco ACI Observer jobs

You use the Cisco ACI Observer when you have a Cisco ACI environment with Cisco Application Policy Infrastructure Controller (APIC) in your environment. The Observer interfaces with Cisco APIC and makes active REST calls to Cisco APIC in the Cisco ACI environment.

Before you begin

Important: The Cisco ACI Observer supports the on-premises Cisco ACI version 4.1.

Ensure you have the Cisco ACI service details to hand, such as the Cisco APIC username, Cisco APIC password, Cisco APIC SSL truststore and Cisco APIC URL.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

Optional: Install and configure a proxy

The Cisco ACI Observer can (optionally) connect to Cisco ACI through a proxy. See the Proxy Host and Proxy Port parameters listed in the table. A proxy can be used to create a tunnel to the target system.

Example: Squid proxy

1. Install the squid proxy:

   ```
   sudo yum install squid
   ```

2. Edit the /etc/squid/squid.conf file to add ACL for the source and target system. Example of a line added:

   ```
   acl localnet src 1.2.3.0/19 # ASM Host
   ```

3. Initialize the squid directories:

   ```
   sudo squid -z
   ```

4. Configure the squid proxy for auto-start:

   ```
   sudo systemctl start squid
   sudo systemctl enable squid
   ```

For more information about the squid proxy, see http://www.squid-cache.org.

About this task

A Cisco ACI Observer job extracts Cisco ACI resources from Cisco APIC through REST, and then the observer loads and updates the resources and their relationships within the topology view.

You define and start the following jobs.

Restapi Load job

A transient (one-off) job that loads all requested topology data using Cisco APIC REST APIs to build a tenant logical construct topology or a fabric topology, and then exits.

A 'restapi' job loads initial topology data, and can resynchronize topology data from Cisco ACI when required.

You assign 'restapi' as the job type for /jobs/restapi observer endpoint.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

WebSocket Listen job

A long-running job that listens for notifications from Cisco APIC to build the topology and runs until it is explicitly stopped, or until the observer is stopped.

A 'websocket' job monitors changes from Cisco APIC object notification and updates the topology.

You always run a 'websocket' job **after** running a 'restapi' job type.

You assign 'websocket' as the job type for /jobs/websocket observer endpoint.

Cisco ACI objects and relationships loaded

The Cisco ACI Observer loads the following Cisco ACI objects and their relationships into the topology service:

Tenant Logical construct

- fvTenant
- fvAp - A policy owner in the virtual fabric
- fvAEPg - A set of requirements for the application-level EPG instance
- fvAEpP - Abstract representation of an endpoint profile
- fvEpP - An endpoint profile
- fvBD - A bridge domain is a unique layer 2 forwarding domain that contains one or more subnets
- fvCtx - The private layer 3 network context that belongs to a specific tenant or is shared
- vzBrCP - A contract is a logical container for the subjects, which relate to the filters that govern the rules for communication between endpoint groups (EPGs)

- vzOOBBrCP - An out-of-band binary contract profile can only be provided by an out-of-band endpoint group and can only be used by the external prefix set
- vzSubj - A subject is a subapplication running behind an endpoint group (for example, an Exchange server). A subject is parented by the contract, which can encapsulate multiple subjects
- vzFilter - A filter policy is a group of resolvable filter entries
- fvSubnet - A subnet defines the IP address range that can be used within the bridge domain
- fvRsCons -The Consumer contract profile information and on Cisco ACI console the option to create this object is through Consumed Contract. Used to build relationship between fvAEPg and vzBrCP
- fvRsBd - A source relation to the bridge domain associated to this endpoint group. Used to build relationship between fvBD and fvAEPg
- fvRsCtx - A source relation to a private layer 3 network context that either belongs to a specific tenant or is shared. Used to build relationship between fvBD and fvCtx
- vzRsSubjFiltAtt - The filter for the subject of a service contract. Used to build relationship between vzSubj and vzFilter

Fabric Topology

- fabricInst - A container object for fabric policies
- fabricNode - The root node for the APIC
- polUni - Represents policy definition or resolution universe
- firmwareRunning - Information about leaf or spine switch firmware running on a node
- firmwareCtrlrRunning - Information about each controller firmware that is running
- eqptLCSlot - The slot for the module card
- eqptLC - A line card (IO card) contains IO ports
- eqptPsuSlot - The power supply slot
- eqptPsu - The power supply unit
- eqptFtSlot - A fan tray slot
- eqptFan - The fan in a fan tray
- topSystem - Used to retrieve fabric node Operational State
- cnwPhysIf - The physical interface that is assigned to the node cluster
- l1PhysIf - The object that represents the Layer 1 physical Ethernet interface information object
- mgmtMgmtIf - The management interface
- lldpAdjEp - The LLDP neighbors, which contains the information regarding the neighbors
- eqptRsIoPhysConf - A source relation to an L1 Ethernet interface. Used to build relationship between l1PhysIf and eqptLC
- mgmtRsOoBStNode - An object that contains management ip address of fabric spine switches and fabric leaf switches

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see the Configuring observer jobs security topic.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Tenant name	Use this to identify the tenant	Required. Set to `admin` if no specific tenant is referenced. Set to '' to load Fabric Topology resources.
Cisco APIC endpoint	Specify the API URL of the Cisco APIC endpoint	Required. Usually in the following format: https://[hostname or IP address]/api
Cisco APIC username	Specify the username to connect as, or listen to	Required
Cisco APIC password	Enter the password for Cisco APIC authentication	Required
HTTPS truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `CiscoACI.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
Truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
Cisco APIC certificate	Specify a certificate by name to load into the truststore	Optional.
Read timeout	Specify the read timeout in ms (default is 2000)	Optional
SSL certificate validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
Connection timeout	Specify the connection timeout in ms (default is 5000)	Optional
Proxy Host	Specify the proxy host through which to connect	Optional
Proxy Port	Specify the proxy port, defaults to 8080	Optional

Parameter	Action	Details
Trust all certificates by bypassing certificate verification	Set to true to allow connection to target environment without verification.	Optional. The default is 'false'.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Contrail Observer jobs

Using the Contrail Observer, you can retrieve topology data from Juniper Network Contrail Release 4.1 through REST APIs exposed by the Contrail apiserver. This observer is developed against Juniper Network Contrail that integrates with OpenStack orchestration platform (Ubuntu 18.04 + Contrail Cloud - Ocata).

Before you begin

Important: The Contrail Observer supports the on-premises Contrail version 4.1.0.

Ensure you have the Contrail API Server and OpenStack credentials details to hand. For rabbitmq jobs, you must also specify the location of the RabbitMQ queue and its authentication details.

About this task

You define and start the following jobs.

Rabbitmq Listen job

A long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the Observer is stopped.

This job loads all supported resources during startup, and listens to RabbitMQ messages from 'vnc_config.object-update' fanout exchange.

The restapi job does not have to be run before running the rabbitmq job because the rabbitmq job performs a restapi job during initialization.

Restapi Load job

This job loads all supported resources.

Run this job whenever you need the Contrail topology data refreshed.

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Rabbitmq Listen job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Contrail API URL	Specify the URL for the Contrail apiserver	Required
OpenStack Authentication URL	Enter the authentication URL for the identity service	Required

Parameter	Action	Details
OpenStack username	Specify the OpenStack username to connect as (or to)	Required
OpenStack password	Specify the OpenStack password with which to authenticate	Required
RabbitMQ hostname or IP address	Enter a hostname or IP address for the RabbitMQ server	Required
RabbitMQ port	Specify the port for connection	Required
RabbitMQ username	Specify the username for authentication with RabbitMQ	Required
RabbitMQ password	Enter the password for authentication with RabbitMQ	Required
Authentication type	Specify the authentication type used.	Optional. The default is **Keystone**. The other option is **None**.
OpenStack project domain name	Specify the OpenStack project domain name	Optional
OpenStack user domain name	Specify the OpenStack project user domain name	Optional
OpenStack project name	Enter the OpenStack project name for version 3 authentication	Optional
OpenStack tenant name	Enter the OpenStack tenant name for version 2 authentication	Optional
OpenStack identity API version	Select an option from the drop-down list. Defaults to 3.	Optional
Connection and read timeout (milliseconds)	Choose the time at which the connection and read action times out	Optional. Must be a value greater than 0 (zero). The default is 5000 (5 seconds).
RabbitMQ virtual host	Enter the RabbitMQ virtual hostname	Optional. Default is /.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Restapi Load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Contrail API URL	Specify the URL for the Contrail API server	Required
OpenStack Authentication URL	Enter the authentication URL for the identity service	Required
OpenStack username	Specify the OpenStack username to connect as (or to)	Required
OpenStack password	Specify the OpenStack password with which to authenticate	Required
Authentication type	Specify the authentication type used	Optional. The default is **Keystone**, the other option is **None**.
OpenStack project domain name	Specify the OpenStack project domain name	Optional
OpenStack user domain name	Specify the OpenStack project user domain name	Optional
OpenStack project name	Enter the OpenStack project name for version 3 authentication	Optional
OpenStack tenant name	Enter the OpenStack tenant name for version 2 authentication	Optional

Parameter	Action	Details
OpenStack identity API version	Select an option from the drop-down list	Optional
Connection and read timeout (milliseconds)	Choose the time at which the connection and read action times out	Optional. Must be a value greater than 0 (zero). The default is 5000 (5 seconds).
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Datadog Observer jobs

Using the Datadog Observer functionality, you can load monitored hosts and their associated processes, and then visualize this data as a topology view in the IBM Cloud Pak for AIOps UI.

Before you start

Important: The Datadog Observer supports the cloud/SaaS Datadog version.

Ensure you have the Datadog server details to hand, such as the data tenant, API server URL, Auth API key and App key.

About this task

A Datadog Observer job extracts host and its associated processes from Datadog via REST API. The Observer loads and updates the resources and their relationships within the IBM Cloud Pak for AIOps core topology service.

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Table. Parameter details

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data Tenant	Enter the data tenant to track the Datadog topology	Required
API server URL	Enter the Datadog API base URL	Required
Auth API key	Specify the authorized API key to authorize with Datadog	Required
App key	Specify the application key to access the Datadog resources	Required
Configure Host API query	Specify the host API query. Each query parameters needs to be separated by ";" and its value needs to be inside "()". For parameters that support list, the values need to be separated by "," and API will treat them as AND operation	Optional
Configure Process API query	Specify the process API query. Each query parameters needs to be separated by ";" and its value needs to be inside "()". For parameters that support list, the values need to be separated by "," and API will treat them as AND operation	Optional
Datadog certificate	Specify the Datadog certificate to enable secure connection to the target system	Optional. For more information, see Configuring observer jobs security.

Parameter	Action	Details
Certificate validation	Choose whether to use SSL validation ('true' or 'false'). Turning SSL validation off will use HTTPS without host verification.	Optional
Connection and read timeout (milliseconds)	Set the connection and read timeout value.	Optional
Trust all certificates	Trust all certificates without validating the certificate's contents. Choose 'true' or 'false'.	Optional
Number of connection retries	Set the number of times a connection should try again. The default value is 3.	Optional
Delay before a connection retry (milliseconds)	Specify the period of time in milliseconds to delay before a connection tries again.	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring DNS Observer jobs

Using the DNS Observer, you can query internal DNS server performance, and use the returned information on response times and service addresses to create topologies within the topology service. The DNS Observer supports forward and reverse job types, with 'recurse' or 'no recurse' options.

Before you begin

Ensure you have the DNS access details to hand, such as DNS server, address types and port numbers.

Tip: The relationship types can be customized with line color, width and pattern functions. For more information, see the Creating custom relationship type styles topic.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The DNS Observer provides DNS query services and topological insight into how a specified DNS server is performing forward (name-to-IP address) or reverse (IP address-to-name) lookups. Query results include a list of addresses, information on how long it takes the DNS server to resolve a lookup, and, optionally (with the maximum number of recursive calls set at 200) how the DNS server is recursively resolving a given name or IP address.

Note: The DNS observer models the resources based on the DNS lookup record. A separate set of resources will be created for IPv4 and IPv6 discovery of the same domain name.

You define and start the following jobs.

Reverse lookup job

A transient (one-off) Load job that loads all requested DNS reverse (IP address-to-name) lookup topology data.

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Forward lookup job

A transient (one-off) Load job that loads all requested DNS forward (name-to-IP address) lookup topology data.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see the Configuring observer jobs security topic.

Reverse lookup job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
IP address	Enter the host or internet name to look up	Required
Server	Specify the DNS server	Required
Port	Specify the UDP DNS port	Optional
Address type	Specify either **`IPV4`** or **`IPV6`**.	Optional
Recursive query	Toggle **True** or **False**	Optional. If set to True, the maximum number of calls is set at 200.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Forward lookup job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Domain name	Specify a domain	Required
Address Types	Specify the address types to be observed	Required. Select either IPv4 or IPv6.
Server	Specify the DNS server	Required
Port	Specify the UDP DNS port	Optional
Address type	Specify either **`IPV4`** or **`IPV6`**.	Optional
Recursive query	Toggle **True** or **False**	Optional. If set to True, the maximum number of calls is set at 200.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Docker Observer jobs

Using the Docker Observer, you can discover Docker network resources, including Docker Swarm clusters, and then visualize (or model) this data as a topology view.

Before you begin

Important: The Docker Observer supports Docker 3.1.0.

Note: Docker UCP 3.1.0 supports only TLS 1.2 for SSL negotiation and no longer supports TLS 1 and TLS 1.1.

Ensure you have the details for your Docker job to hand, specifically your Docker system's Unix socket, and or host and port number.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

Using the Observer Configuration UI you configure observer jobs that query the Docker REST API to retrieve data and display it as a topology in the Topology Viewer. The Docker Observer can model external Docker systems, and it can also provide a System health view of the Docker system on which IBM Cloud Pak® for AIOps runs.

The job parameters connect to a remote Docker using the host and port parameters.

Remote Docker

The 'host' and 'port' parameters of the job can be used to identify the TCP port that Docker can be reached on. The **unix_socket** parameter must not be supplied.

Docker is not accessible through TCP by default. To enable it, edit the 'docker.service' file. On Red Hat® OpenShift® Container Platform, this is available in **`/usr/lib/systemd/system`**. Amend the **ExecStart** option under the **Service** section to include a **`-H`** option. For example, to make it available externally on port 2375, you might add **`-H tcp://0.0.0.0:2375`**.

Note: If you want to continue to be able to access Docker through the default socket, for example if the Docker Observer container needs access, or if you want to be able to perform **`docker ps -a`** rather than **`docker -H tcp://0.0.0.0:2375 ps -a`**, then you need to also list it in the same line, as on the following example:

```
-H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sockCopy
```

You must reload the configuration:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

Tip: If this fails to start Docker, and a Unix socket (or no socket at all) was specified, check that no directory with that name exists. If you start Docker with just a TCP socket and no Unix socket, this creates a **`/var/run/docker.sock`** directory, which you must delete after Docker is stopped, so that you can restart with access through that Unix socket.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Host	Use this to identify the TCP host socket (HTTP or HTTPS) on which to access the remote Docker system	Required for **remote** Docker access only
Port	Use this to identify the TCP port (HTTP or HTTPS) on which to access the remote Docker system	Required for **remote** Docker access only
Unix Socket	Use this to access local docker environments using the complete path	Required for **local** Docker access only. Host and port parameters **must** be empty.
Username	Specify the username of the remote Docker environment with HTTPS	Required for remote Docker with HTTPS access only
Docker password	Specify the password of the remote Docker environment with HTTPS	Required for remote Docker with HTTPS access only.
Docker SSL Certificate	Specify the certificate file name	Optional
Docker SSL truststore File	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (**`<observer>.jks`**), for example **`Docker.jks`**.	Required for remote Docker with HTTPs access only.The supported format is JKS and the file is relative to $ASM_HOME/security.
SSL truststore File Password	Specify the truststore password the observer will use to decrypt the truststore file. Use a password that conforms to your internal security requirements.	Required for remote Docker with HTTPs access only.
View	Use this to select which resources are modeled in the topology view	Optional. The Default displays running resources only. Options are **Container** (all running containers), **Image** (images used by running containers), and **Task** (running tasks only)
Containers to exclude	List container you want to exclude	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional

Parameter	Action	Details
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Dynatrace Observer jobs

Using the Dynatrace Observer, you can query a specified Dynatrace environment for information about its applications, services, process groups, and infrastructure entities.

Before you begin

Important: The Dynatrace Observer supports Dynatrace cloud, SaaS and Dynatrace onPrem v1.182.102.

Ensure you generate a Dynatrace token to access your Dynatrace environment. You also need topology access scope to access the Dynatrace resources.

Ensure you have the Dynatrace access details to hand, such as Dynatrace API URL and API token.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Dynatrace Load job

A transient (one-off) job that loads all requested Dynatrace resource data.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Notice: Dynatrace v1 API endpoints are no longer supported.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Dynatrace Observer job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data tenant	Data tenant is used to track your data. It needs to be unique for each job or else it will override the previously discovered data.	Required
Base URL	Specify the API URL of the Dynatrace endpoint (including version). An example of a Dynatrace **cloud-based URL** is: `https://< your-environment-id >.live.dynatrace.com/api/v2/`, while an example of a Dynatrace managed or **onPrem URL** is: `https://< your-domain >/e/< your-environment-id >/api/v2/`	Required
Dynatrace API Token	Enter the Dynatrace token for authentication	Required
Enable/Disable Secure Connection to CSO's host	Select to either enable or disable a secure connection to the CSO's host. Set to true to secure the connection, or false to bypass it.	Required
The entitySelectors for the Dynatrace v2 API	The format is `{"taskname" : "entitySelector"}`, and multiple tasks can be added to the entitySelector list. The **taskname (string, required)** is a user defined value that identifies each task uniquely. The **entitySelector** defines the scope of the query. Only entities matching the scope and criteria will be discovered. Separate each filter of entitySelector with ';' and place its value inside parentheses '( )'. For a filter that supports a list, separate the values using a comma ','. For more entitySelector information, see more details and example below this table.	Required
Edge Capture Direction	Choose to display either a one-directional or bi-directional edge topology. A (default) bi-directional direction uses both `FromRelationships` and `ToRelationships` edges when displaying the Dynatrace relationship, while a one-directional direction uses only `FromRelationships` edges. **Tip:** If the Dynatrace relationship is bi-directional, the observer will use both `FromRelationships` and `ToRelationships` between A and B, and also between B and A, and therefore both edges will be displayed in a bi-directional edge topology. If the Dynatrace relationship is one-directional, the observer will use `FromRelationships` from host A to host B, and also `FromRelationships` from host B to host A.	Optional
Number of records to be fetched per page	Number of records to be fetched per page. Maximum value is `4000` and minimum is `1`.	Optional. Default value is `4000`.

Parameter	Action	Details
Relationship mapping for dynatrace with ASM	Observer relationship to be used for Dynatrace relationship. The user can use edge mapping values to also override the default values	Optional. Observer will use a default mapping for edges.
Entity mapping for dynatrace with ASM	Observer entity types to be used for Dynatrace entity type. Observer will use a default mapping for entity. The user can use entity mapping values to also override the default values.	Optional. If there is no default value then Dynatrace entity type will be used as ASM entity type.
Grouping vertex preference	The observer creates the defined topology for the discovered vertices based on the preference in this field.	Optional. Value can be **host** or **none**. - Default value is **none**, which indicates that no resource groups are created. - When **host** is selected, resource groups are created with the host as **`seed`** and its connected entities, except other hosts, as members.
Certificate validation	Option for SSL validation. Turning SSL validation off will use HTTPS without host verification.	Optional
Trust all certificates by bypassing certificate verification	Trust all certificates without validating the certificate's contents.	Required. Default value is true.
Connection timeout (milliseconds)	Set the connection timeout in milliseconds.	Optional. The default is 5000ms.
Read timeout (milliseconds)	Set the read timeout in milliseconds.	Optional. The default is 5000ms.
Proxy Host	Specify the proxy host to connect to.	Optional
Proxy Port	Specify the proxy port, defaults to 8080.	Optional
Target system certificate	Specify a certificate by name to load into the trustStore.	Optional. To obtain authentication certificates using OpenSSL and store them as secrets, see Configuring observer job security.
Proxy Username	Specify the proxy user name to connect. Set for basic auth proxy.	Optional
Proxy Password	Specify the password for the proxy user (if **`Proxy Username`** has been specified).	Optional. Use plain text.
Proxy Secure	Specify whether to connect to the proxy in secure mode. Set to True for HTTPS proxy.	Optional. The default is 'false'.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Note: For the **Grouping vertex preference** job parameter, when **host** is selected for creating resource groups, the host unique ID is set as the seed during the processing of the host API response. Connected entities from the response (excluding connected hosts) are used as the members of the created resource group. In Dynatrace, entities like application, service, process and even custom data can be part of several hosts.

Dynatrace API entitySelector parameters and examples:

Examples:

Single task example:

```
{
"HostTask": "type(host);entityId(HOST-E9B8C2EF21BA5217,HOST-B343896535B69B84,HOST-F4A80CBC1027416A);timeframe(now-10m);fields(properties.osType,properties.hypervisorType)"
}
```

Multiple tasks example:

```
{
"HostTask":"type(host);timeframe(now-10m);fields(properties)",
"AppTask":"type(application);timeframe(now-10m);fields(properties)",
"ServiceTask":"type(service_instance);timeframe(now-10m);fields(properties)",
"ProcessTask":"type(process_group_instance);timeframe(now-10m);fields(properties)"
}
```

Parameters:

- **!** (negation): For example `!entityName(eur01.ibm.com)`
- **type (string, required)**: Each task can only accept one type. You must set this criteria in the entitySelector. Negation is not supported by type.
- **mzName (string, list, optional)**: Management zone name of the entity
- **mzId (string, list, optional)**: Management zone id of the entity
- **entityName (string, list, optional)**: Name of the entity
- **✱** (wildcards): For example the string `"type(host);entityName(eur0*);..."` will fetch all of the hosts that start with an entityName of 'eur0', such as 'eur01.ibm.com,eur02.ibm.com', etc; while the string `"type(host);entityName(*eur0*);..."` will fetch all hosts that contain 'eur0' as part of their entityName parameters, such as `ibm.eur0.com,ibm.eur02.com`, etc.
- **entityId (string, list, optional)**: Id of the entity
- **tag (string, list, optional)**: Tags of the entity
- **state (string, list, optional, case-sensitive):** State of the entity
- **excludeRelationsWith**: Excludes the connectivity with the listed entity type. Negation is not supported.
- **timeframe (string, optional)**: Time frame of the entity. The time frame should be a minimum of 5m and a maximum of 24h. The format is timeframe-NU, where 'N' is the amount of time, and 'U' is the unit of time. The default is '(now-5m)'. Supported time units for the relative time frame are 'm' (minutes), and 'h' (hours). For a setting of `timeframe-1hr`, the observer filters out if the lastSeenTms is within one hour of the observer job's start time, and only model those resources.
- **fields (string, list, optional)**: Entity fields that will be added to the entity vertex

Configuring File Observer jobs

Using the File Observer functionality, you can write bespoke data to a file in a specific format, upload this data to the topology service, and then visualize this data as a topology view.

Before you begin

The File Observer reads topology data from specified files, and uploads it. You must create these files manually.

About this task

Topology data in a file is composed of vertices (nodes) and edges. A vertex represents an object (resource), while an edge represents the relationship between two objects.

Each line of the file you create should be in one of the following formats, loading a single resource vertex (including optional relationships in the '_references' field) or a single edge, deleting a single vertex, or pausing execution.

Lines starting with `V:` (vertex), `E:` (edge), `D:` (delete) or `W:` (wait) are treated as instruction lines to be processed. Other lines, for example lines that are empty or commented out, are ignored.

Line format

V - Load a resource vertex, with a JSON representation as documented for the body of the topology service API method: `POST /resources`

If specifying the `_status` element, acceptable state values are `open`, `closed`, or `clear`, and acceptable severity values are `clear`, `indeterminate`, `warning`, `minor`, `major`, or `critical`. The following **geolocation example** adds `point` geolocation data as a GeoJSON object inside the vertex JSON, with the coordinates being longitude first, then latitude.

```
V:{"name":"Smallbrook Junction", "uniqueId":"gb-nr-SAB", "entityTypes":["railStation"], "geolocation":{"geometry":{"coordinates":[-1.15511,50.71141],"type":"Point"}}}
```

E - Load an edge, with a JSON representation as documented for the `_references` section of the body of the topology service API method `POST /resources`.

D - Delete a resource vertex, identified by it's `uniqueId`.

W - Pause for the duration (for testing purposes only). Takes an integer period followed by a string specifying the units.

G - To load a group via the File Observer, see the following example syntax:

```
G:{"uniqueId":"EA","entityTypes":["group"],"name":"Event Analytics"}
```

Note: When creating a group's edge, it must be based on `uniqueId`.

Important: Ensure that the file is structured correctly. For each line of the file, information included after the closing **}** that matches an opening **{** is ignored, and no error is recorded.

Mapping alerts to topologies: When using the File Observer to correlate events from Logs Anomaly, ensure you use the `matchTokens` property when creating topology templates. The `matchTokens` property is a list of values you can use to look up a resource from an alert. If a unique `matchToken` is found that matches an alert's resource fields, such as `resource.name` or `resource.sourceId`, the alert will be assigned to that resource.

File Observer syntax

When uploaded to the file observer in a file, say `example.txt`, the following small topology can be generated by running a File Observer job for that file. The figure also shows an alert matched to a resource.

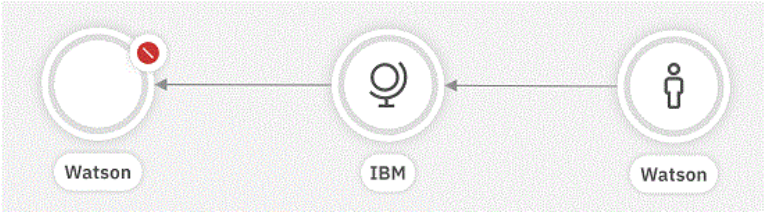


Figure. File Observer topology

```
V:{"uniqueId":"Thomas J. Watson","entityTypes":["person"],"name":"Watson"}
V:{"uniqueId":"IBM","entityTypes":["company","domain"],"name":"IBM","_references":
[{"_edgeType":"owns","_toUniqueId":"Cloud Pak for AIOps"},{"_edgeType":"contributedTo", "_fromUniqueId":"Thomas J.
Watson"}]}
V:{"uniqueId":"Cloud Pak for AIOps","entityTypes":["ai"],"name":"Watson","platform":"OCP","matchTokens":["AIOps","Cloud
Pak for AIOps"]}
```

Line 1: This line has the minimal set of properties for a resource vertex:

- `uniqueId` - Uniquely identifies the resource within the provider namespace (the provider is supplied when running the job).
- `entityTypes` - Determines the icon displayed in the UI. Some types have a pre-defined mapping to icons, as described in the Entity types topic.
- `name` - The label displayed in the UI.

Line 2: This line shows multiple entityTypes. There is no default icon mapped to type company, so the icon is determined from the domain type. In addition to the properties of the resource, this adds an edge to another resource.

- The edgeType must exist already. Either one of the default Edge types or one created by the topology service's `POST /types/edge` API (see Accessing APIs)
- The referenced uniqueId does not have to exist yet.
- Multiple edges can be added in the `_references` list.

Line 3: This line shows that more data or properties can be added. It adds a custom `platform` property (might be any additional property required). See the Properties topic for a list of existing properties. This line also illustrates that any `entityType` can be supplied, and that the `name` does not have to be unique. The UI does not show an icon for this resource because no default icon mapping has been created. You can add custom mapping through the UI as described in the Defining custom icons topic. And finally, this line also adds a `matchToken` property, which is a list of values that can be used to link a resource to an alert.

File Load job

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Save** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required.
File mode	Select either **Upload local file**, then drag and drop (or browse) to the input file; or **Use server file**, then specify the name of the file previously uploaded through the `POST /files` API.	Required. **Note:** If the same file name is used in a second job, or the file is replaced through `PUT /files`, any resources missing from the second load will be deleted.
Provider	Enter the job's provider, that is, a namespace for the resources' uniqueIds. Resources can be loaded from multiple files with the same provider, and linked as if they were all loaded from the same file. To link to resources from another provider (for example, from another observer), the cross-provider semantics are required.	Optional. Defaults to `FILE.OBSERVER:{fileName}` if not supplied.
Include delays	Select **true** (default) to accept any delays included in the files, or **false** to skip such delays.	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional

Parameter	Action	Details
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Swagger API documentation

For the Swagger API documentation, see Topology File Observer API. The documentation explains how to use the File Observer APIs.

File Observer API access

Two public APIs are available, and both APIs can be used to retrieve data.

- **API name**

  topology-file-observer

- **Public API URLs**

  `/aiops/api/application-manager/topology-file-observer/v1/*`

  `/aiops/api/app/topology-file-observer/v1/*`

For information on how to obtain route and token information, see Accessing the console.

For information on how to obtain your token, see Accessing APIs.

Example API queries

GET /health API

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
https://$ROUTE/aiops/api/application-manager/topology-file-observer/v1/healthcheck
{"date":"2021-02-05T19:24:00.974+00:00","jobs":0,"threads":10,"items":[{"type":"load"}]}
```

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
https://$ROUTE/aiops/api/app/topology-file-observer/v1/healthcheck
{"date":"2021-02-05T19:24:00.974+00:00","jobs":0,"threads":10,"items":[{"type":"load"}]}
```

GET /service/info API

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
"https://$ROUTE/aiops/api/application-manager/topology-file-observer/v1/service/info"
{"startDate":"2021-02-04T10:30:56.991+00:00","id":"example-noi-topology-file-observer-6f88fb6fc4-n9xm2","name":"file-
observer","version":"1.1.0.17 66dd27a","adminConnector":"https://example-noi-topology-file-observer-6f88fb6fc4-
n9xm2:9099","appConnector":"https://example-noi-topology-file-observer-6f88fb6fc4-n9xm2:9098","uptimeMillis":118575702}
```

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
"https://$ROUTE/aiops/api/app/topology-file-observer/v1/service/info"
{"startDate":"2021-02-04T10:30:56.991+00:00","id":"example-noi-topology-file-observer-6f88fb6fc4-n9xm2","name":"file-
observer","version":"1.1.0.17 66dd27a","adminConnector":"https://example-noi-topology-file-observer-6f88fb6fc4-
n9xm2:9099","appConnector":"https://example-noi-topology-file-observer-6f88fb6fc4-n9xm2:9098","uptimeMillis":118575702}
```

Upload a file

```
curl -k -X POST --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
https://$ROUTE/aiops/api/app/topology-file-observer/v1/files --form 'job_file=@/tmp/file.json.txt' --header 'X-TenantID:
cfd95b7e-3bc7-4006-a4a8-a73a79c71255'
```

Creating cross-data origin relationships

Using the File Observer functionality, you can create edges between resources from different observations or providers.

It requires both the `uniqueID` and the provider of the resources at each end of the link (so typically applies to previously observed resources).

In the following syntax examples, note the `_toProvider` and `_fromProvider` edge properties.

You can define an edge from a resource in the current observation to one owned by a different provider:

```
V:{"uniqueId":"host1.ibm.com","entityTypes":["host"],"name":"host1","_references":
[{"_edgeType":"connectedTo","_toProvider":"someOtherProvider","_toUniqueId":"host2.ibm.com"}]}
```

You can define an edge from a resource between two different providers:

```
E:{"_fromProvider":"providerOne","_fromUniqueId":"host1-
interface1","_edgeType":"connectedTo","_toProvider":"providerTwo","_toUniqueId":"host2-interface1"}
```

File Observer sample files

You can upload sample file data to the topology service on File Observer, and then view this data in the topology. To download the sample files, go to Sample files.

Configuring GoogleCloud Observer jobs

Using the GoogleCloud Observer, you can define a full load job that will read services data from the Google Cloud Platform Compute Services through Google's Compute Services SDK, and then generate a topology.

Before you begin

Important: The Google Cloud Observer supports the cloud and SaaS Google Cloud version.

The GoogleCloud Observer supports GoogleCloud's compute services. Ensure you have the GoogleCloud details in hand, such as the Project ID, Service Account Key File and Zone, before running the observer job.

The GoogleCloud Observer supports a transient (one-off) Load job that loads all requested topology data through Google's Compute Services SDK to build the topology, and then exits.

To create a service account key file

1. From the Google Cloud Platform dashboard, under your 'Project ID', go to APIs and Services and then choose **Credentials**. The Credentials page is displayed listing a number of authentication methods.
2. Select the **Service account** authentication service
3. From **Create Credentials**, choose **Service account**.
4. Complete the service account details, then click **Create** to create a new service account.
5. Select the **Compute Engine > Compute Admin** role, then click **Continue** and **Done**.
6. To download the JSON file from the Credentials page, click **Service account > Manage Service Accounts**.
7. Select the new service account, then choose **Create key** from the overflow menu (the three dots under the **Actions** column).
8. Choose a key type of JSON, then click **Create**. A JSON file will be downloaded. For more information on storing the service account key file as a secret, see the Configuring observer jobs security topic. The file name will be used in the observer parameter (service_account_key_file) for the full load job.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job. You must edit the parameters in the configuration file before running this job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see the Configuring observer jobs security topic.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Project ID	Enter the Google Cloud Platform Project ID	Required
Service Account Key File	Supply the Google Cloud Platform Service Account Key File	Required
Zones	Specify the Google Cloud Platform zones	Required.
Connection timeout	Specify the connection timeout in ms (default is 5000)	Optional
Read timeout (milliseconds)	Specify the read timeout in milliseconds. The default value is 5000.	Optional
Proxy Host	Specify the proxy host via which to connect.	Optional
Proxy Port	Specify the proxy port. Set as per HTTP/HTTPS Proxy Host.	Optional. Defaults to 8080.
Proxy Username	Specify the proxy username. Set for basic auth proxy.	Optional
Trust all certificates by bypassing certificate verification	Set to true to allow connection to target environment without verification.	Optional. The default is 'false'.
Proxy Password	Specify the proxy password. Set if Proxy Username has been specified.	Optional. Use plain text.
Proxy Secure	Specify whether the proxy server is secure. Set to 'true' for HTTPS proxy.	Optional. Default is 'false'.

Parameter	Action	Details
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Note: You must create a service account key file or use an existing one to allow the GoogleCloud Observer to discover resources from GoogleCloud.

Troubleshooting: While the job is running, the status of discovered resources may appear as 'indeterminate' in the topology until the full upload is complete.

Configuring HP NFVD Observer jobs

You configure HP NFVD Observer jobs to dynamically load data from the HP NFVD REST interface.

Before you begin

Important: The HP NFVD Observer supports the HPE NFV Director API release 6.0.3.

Ensure you have the HP NFVD service details to hand, such as username, password, hostname and port.

About this task

The HP NFVD Observer job extracts HP NFVD resource information through REST. The observer loads and updates the resources and their relationships and then displays them as a topology view.

You define and start the following jobs.

Load job

Run this job whenever you need HP NFVD topology data refreshed.

By default, these jobs are one-off, transient jobs that perform a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Fileload job

Run this job to consumes topology data from a file.

The fileload job loads a baseline topology data view from a provided data file.

The fileload job is a transient (one-off) job that consumes topology data from the file, sends the data for processing, and then stops.

The fileload job provides an alternative to loading topology data through a file.

Listen job

A long-running job that reads LCM events published by HP NFVD on CGMB (Kafka) topic continually until it is explicitly stopped, or until the observer is stopped.

LCM messages in Kafka are typically encoded in Avro and are decoded by the observer using an HP NFVD LCM notification Avro schema file. Refer to the HP NFVD documentation on how to obtain the schema file. The additional steps required when using an Avro schema file are documented in the following subsection of the procedure.

The job should only be run *after* a load job has been completed.

HP NFVD resources and relationships loaded by the HP NFVD Observer job

The HP NFVD Observer loads the following resources and their relationship into the Agile Service Manager core topology service:

Vertex	Description
Organization	Organization (for example, LOB) within Domain
VDC	Tenant within Organization
VNF Group	Group of VNF / CNF
VNF	Virtual Network Function
CNF	Containerized Network Function
VNFM	Manager of VNF/CNF

Vertex	Description
VM	Virtual Machine attached to VNF
Network Interface / Port	Network Interface or Port attached to VNF
Connection Point	Internal Connection Point
VNF-C	VNF Component
CNF-C	CNF Component
Virtual Link	Virtual Network attached to VNF
Virtual Storage	Storage attached to VNF
Deployment	K8s resources attached to CNF
Statefulset	K8s resources attached to CNF

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see the Configuring observer jobs security topic.

Load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
HP NFVD data center name	The name of the data center on which the HP NFVD is hosted.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD hostname	Specify the hostname of the HP NFVD.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD host port	Specify the port of the HP NFVD.	Required
HP NFVD username	Specify the username.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
Password	Enter the password with which to authenticate.	Required. Plain text password can be entered.
HTTPS or HTTP connection to the HP NFVD server	Choose to make either a HTTPS (secured) or HTTP connection to the HP NFVD server.	Required. When set to true, provide the target system certificate to the certificate parameter.
True for a CNF job and false for a VNF job	The VNF job is expected to run with fewer threads and will take longer to complete. If this filter parameter is set to `false`, cnf_group_filters defines group names to exclude. If this filter parameter is set to `true`, cnf_group_filters defines group names to include.	Required
Target system certificate	Specify a certificate by name to load into the trustStore.	Required if the `HTTPS connection to the HP NFVD server` parameter is set to true. To obtain authentication certificates using OpenSSL and store them as secrets, see Configuring observer jobs security.
SSL Verification	Choose whether SSL validation is true or false. Turning SSL validation false will use HTTPS without hostname validation.	Optional
Connection timeout	Enter the time at which the connection actions time out.	Optional. Must be a value greater than 0 (zero). The default is 5000 (5 seconds).
Read timeout	Enter the time at which the read actions time out.	Optional. Must be a value greater than 0 (zero). The default is 5000 (5 seconds).
First retry after delay (milliseconds)	Set the initial delay for a retry.	Optional
Correlate	Set this to true to enable Event Analytics correlation on the topology grouping based on VNF Groups created by this job.	Optional
CNF group name for filters	If specified, this parameter selects CNF groups by partial matching on group name. For example, if `CNF1,CNF2` is specified, then a CNF job will only process matching groups such as 'MyCNF1Yeh' and 'YourCNF2No', and a VNF job will process groups that do not match.	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional

Parameter	Action	Details
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Fileload job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
HP NFVD data center name	The name of the data center on which the HP NFVD is hosted.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD hostname	Specify the hostname of the HP NFVD.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD username	Specify the username.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
Name of the VNF or CNF data file	Specify the name of the VNF or CNF data file. Wildcards (`*`) are supported. Use separate data files for CNF and VNF resource types. **Tip:** If the filename contains the token `nfvd-cnf-data-dump/nfvd_cnf_data_dump`, it is treated as a **CNF** resource. If the filename contains the token `nfvd-vnf-data-dump/nfvd_vnf_data_dump` then it is treated as a **VNF** resource.	Required. The data file must be located in the $/opt/ibm/netcool/asm/data/hpnfvd-observer directory.
HP NFVD archive file retention days	Specify the number of days that the HP NFVD archive files are retained.	Optional. Default is 5 days.
Correlate	Set this to true to enable Event Analytics correlation on the topology grouping based on VNF Groups created by this job.	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Listen job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
HP NFVD data center name	The name of the data center on which the HP NFVD is hosted.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD Kafka broker host	Specify the hostname of the HP NFVD.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD Kafka broker port	Specify the port number of the Kafka broker.	Required
The HP NFVD Kafka topic to listen for LCM notifications	Specify the topic name for LCM notifications.	Required
HP NFVD username	Specify the username.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.

Parameter	Action	Details
True for a CNF job and false for a VNF job	Specify `True` for a CNF job, or `False` for a VNF job. The VNF job is expected to run with fewer threads and will take longer to complete.	Required. **Important:** To observe the complete lifecycle of resources, this must be the same as the option used in the Load job.
SSL	Choose to turn SSL on (or off) for Kafka connection. Turn this option on if the Kafka connection is secured using SSL.	Optional
Avro Decoding	Choose whether Avro decoding is on or off (True or False). Set to `True` if incoming LCM messages are Avro encoded. If so, the Avro schema file is required.	Optional. Default is false.
Avro schema file	The name of the Avro schema file to be used in decoding binary Avro Kafka messages.	Required if Avro Decoding is set to true. Complete the Avro schema file setup.
Kafka TrustStore file	Specify the truststore file used to establish a secured connection to the Kafka broker	Required if SSL has been enabled. For TrustStore and KeyStore configuration, see Avro schema file.
Kafka trustStore password	Enter the password of the Kafka truststore	Required if a Kafka TrustStore file is specified. Use plain text.
Kafka KeyStore file	Specify the keyStore file used to establish a secured connection to the Kafka broker	Required if SSL has been enabled. For TrustStore and KeyStore configuration, see Avro schema file.
Kafka keyStore password	Enter the password of the Kafka keyStore	Required if a Kafka KeyStore file has been specified. Use plain text.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

To use the Avro schema file

Avro schema file setup is required if the Avro decoding parameter is set to `True`.

To use the LCM Avro schema file, create a ConfigMap for the schema file to be mounted to the existing volume. If Kafka is in secured mode (SSL), additionally create two ConfigMaps (that is, truststore and keystore to be mounted to the existing volume).

Assumption: The Kafka keystore and truststore files are already set up together with the LCM Avro schema file.

1. Create a configMaps to mount the schema files to the existing volume.

   a. Create ConfigMaps to mount schema files to the existing volume.

   The example below creates the ConfigMap Avro schema file named 'lcm-notification-schema.avsc':

   ```
   oc create configmap hpnfvd-lcm-notification-schema --from-file=lcm-notification-schema.avsc
   ```

   If Kafka is secured (SSL), keystore/truststore is most likely required. The example below creates ConfigMaps using the keystore file named 'keystore', and the truststore file named 'truststore':

   ```
   oc create configmap hpnfvd-kafka-truststore --from-file=truststore
   ```

   ```
   oc create configmap hpnfvd-kafka-keystore --from-file=keystore
   ```

   After creating the ConfigMap, you can verify the results using one of the following commands:

   ```
   oc describe <configMap name>
   ```

   or

   ```
   oc get <configMap name>
   ```

   b. Map the ConfigMap to a volume in the pod's deployment.

   Add to the existing volume, (asm-security), under 'volume Mounts' and 'volumes':

   ```
   volumeMounts:
     - mountPath: /opt/ibm/netcool/asm/security         <--- existing mount
       name: asm-security

   volumes:
     - name: asm-security
   ```

```
  - configMap:
    name: hpnfvd-lcm-notification-schema
    optional: false
    items:
    - key: lcm-notification-schema.avsc
      path: lcm-notification-schema.avsc
```

You can verify the results using the following command:

```
oc exec -it <pod-name> -- ls /opt/ibm/netcool/asm/security
```

The schema files should be listed.

Configuring IBM Cloud Observer jobs

Use the IBM Cloud Observer when you have IBM Cloud installed in your environment to run jobs that read data from an IBM cloud instance. These jobs retrieve Cloud Foundry Apps information and services, and then dynamically load the retrieved data.

Before you begin

Important: The IBM Cloud Observer supports the cloud and SaaS IBM Cloud version.

Important: The IBM Cloud Observer supports Cloud Foundry API version 2.92.

Ensure you have the IBM Cloud access details to hand to specify and access the cloud instance, such as the instance ID, credentials, and region.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Full Topology Upload job

The IBM Cloud Observer imports ITSM Resource Topology Service data.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Mapping IBM Cloud model objects to entity types

IBM Cloud resource object	Entity types
stacks	`operatingsystem`
apps	`application`
routes	`path`
service bindings	`hub`
service instance	`service`
user provided service instance	`service`
spaces	`group`
organization	`organization`
buildpacks	`component`

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Password	Enter the password or API key used to access the instance.	Required
Instance	Enter the name of the IBM cloud instance.	Required
IBM Cloud Region	Choose the cloud instance region from the drop-down list (US, UK, EU, AP)	Required. Each region has its own URI, and only a single region is discovered in a full load job. To discover different regions, a full load job needs to be triggered for each region.
User email	The email address used to access the instance, or enter `apikey` if IBM Cloud API Key is used.	Required. See Creating an API key on steps to create an IBM Cloud API key.

Parameter	Action	Details
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Instana Observer jobs

Using the Instana Observer job, you can retrieve Instana topology information.

Important: The Instana Observer supports the following Instana version: SAAS build 286.

Before you begin

Note: The Instana Observer only supports running an instance within the Cluster hosting AIOps and cannot be deployed remotely behind a firewall.

Ensure that you have installed and enabled the Instana Observer as described here.

Ensure that you have following information to hand:

- Endpoint
- API Token
- Filters to define subsets of Instana data

Important when migrating from the Instana Connector

When integrating with Instana, the Instana Observer provides an efficient alternative to the Instana Collector. You can use either the oberver or the collector to collect topology information, **but not both**.

If you have an existing integration with Instana via the Instana Connector and want to switch to using the observer instead, disable the previous integration and delete previously retrieved data **before** defining and running the Instana Observer. This will prevent the creation of duplicate resources.

Note: After migrating, the Instana Observer collects *topology* data, while the Instana Collector continues to collect *event* and *metric* data.

See Migrating from the Instana Connector to the Observer for the steps to migrate from the Instana Connector to the Observer.

About this task

The list of Instana technologies retrieved by the observer is determined dynamically and is based on the actual resources available in the Instana instance from which the observer is retrieving data.

Instana job: By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered. You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them. Run this job whenever you want to refresh the Instana topology data.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Table. Parameter action details

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Base URL	Specify the Instana tenant unit URL	Required
Instana API token	Specify the Instana API token. **Note** that the API Token can be obtained from the Instana web UI.	Required
Page size configuration to query the Instana API	Set the page size for Instana APIs that support pagination. The default is `5000`.	Required
Size of time window in seconds to query the Instana API	Set the window size for Instana APIs that support time windows. The default is `86400` (seconds), which is 1 day.	Required

Parameter	Action	Details
Trust all certificates by bypassing certificate verification	Specify whether to connect to the target environment without certificate verification. The default is `true`.	Required
Import Instana applications	Specify whether Instana application perspectives should be imported as waiopsApplication groups. The default is `true`.	Optional
White list regular expression pattern	Provide the white list pattern used to filter applications	Optional
CSV list of regular expressions for infrastructure data properties to filter out	Enter the regular expressions to filter out properties. The defaults are `.*conditions.*,.*containers.data.*,.*metrics.*,.*lastProbeTime.*,.*lastHeartbeatTime.*,.*requirements.*,.*containerState.*,.*itemIds.*,.*annotations.*,env,clusterUuid,dependencies,directDependencies,filesystems,jvm.*,initContainers.data.cri-o://.*,table_info.shared.*,volumes_.*,creationTimestamp`	Optional
Exclude empty applications and services	Specify whether applications and services with no members should be excluded. The default is `true`.	Optional
Connection timeout (milliseconds)	Set the connection timeout in milliseconds.	Optional. The default is `5000ms`.
Read timeout (milliseconds)	Set the read timeout in milliseconds.	Optional. The default is `5000ms`.
Proxy Host	Specify the proxy host to connect to.	Optional
Proxy Port	Specify the proxy port. The default is `8080`.	Optional
Proxy Username	Specify the proxy user name to connect as. Set for basic auth proxy.	Optional
Target system certificate	Specify the certificate to load into the trustStore. **Note** that the file must be located in the `/opt/ibm/netcool/asm/security` directory.	Optional
Proxy Password	Specify the password for the proxy user (if `Proxy Username` has been specified).	Optional
SSL certificate validation	Select whether to use SSL validation. The default is `false`, which uses HTTPS without host verification.	Optional
HTTPS trustStore file name	Specify the HTTPS trustStore file name.	Optional. **Note** that if no certificate is being used, a valid truststore file **must** be provided.
Proxy Secure	Specify whether to connect to the proxy in secure mode. The default is `false`. Set to `true` for HTTPS proxy.	Optional
TrustStore file password	Enter the password needed to decrypt the HTTPS trustStore file. You must supply the password_ssl_truststore_file in encrypted form. See the Configuring observer jobs security topic for more information.	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Migrating from the Instana Connector to the Observer

If you have defined the Instana Connector to integrate with Instana and intend to switch to the Instana Observer, you must disable the previous integration and delete previously retrieved data before defining and running the Instana Observer. This will prevent the creation of duplicate resources.

When migrating from the Instana Connector to the Instana Observer, you first edit the existing Instana Connector integration to disable the topology, then delete the provider resources to remove data from the old integration, and finally scale the instana-topology-integrator pod to zero. You then define and run the Instana Observer.

Remember: The Instana Observer provides a more efficient alternative to the Instana Collector, which uses more system resources. You can use either to collect topology data, **but not both**.

About this task

The topology model created by the AIOps Instana Topology Integrator differs slightly from that created by the Instana Observer. Key differences are:

- The data properties from the Instana APIs are no longer stored as first level properties in the resources, but are instead nested under a parent property named `InstanaData`.
- The Observer creates a History Exclusion Rule, which means that property changes to `instanaData` will no longer drive topology history.
- The Instana end point resources are now members of the same Instana resource group as the Infrastructure resources.
- A single topology resource is created for each Instana service rather than a copy for each application (as before).
- Where possible the Instana end point resources will be connected to the Infrastructure resources via the technologies attributed to the service.
- As with all observers you cannot schedule jobs with a period that is shorter than the length of time it takes the topology to consume the data. If the next job is scheduled to start before the previous one has completed then it will be skipped.
- The `name` attribute of a resource is now the Instana label, which is more readable when looking at a topology view. If the name was previously used in a rule, such as a File Enrichment Rule, then the rule will need to be amended.

Procedure

1. Disable the Topology Integrator to prevent the Instana connector pod from downloading topology data from Instana.

   Important: Make a note of the topology configuration parameters which were used as you will need them later when creating a new job for the Instana Observer.

   The instana-topology-integrator pod registers itself with the Topology Configuration UI using the name of `instana-observer`; to avoid clashes between it and the Instana Observer it should be scaled to zero.

   a. Log in to IBM Cloud Pak for AIOps console.

   b. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.

   c. On the *Manage integrations* tab of the *Integrations* page, click the **Instana** integration type.

   d. Click the integration that you want to disable.

   e. Go to the *Collect topology data* section and use the toggle to disable the integration.

   Note: Disabling topology data collection for an integration does not delete the integration. Also, the Collector will continue to collect event and metric data.

   For more information about Instana integrations, see the Creating Instana integrations topic.

2. Backup your UI configuration and existing Observer jobs, and then delete the old topology data using one of the following methods (substeps).

   - Clear out all the topology data as documented in in Clearing out topology data without reinstalling.
   - Remove the old topology using the `delete provider` resources crawler, which can be run from the Data administration routines UI. **Remember** that this crawler has a limit (of `10,000`) on the number of resources it can delete, which means that if there are more than this number of resources in your existing Instana topology, either use the alternative method (Clearing out topology data without reinstalling), or contact IBM Support for further assistance.

What's next

Next, define and run Instana Observer jobs using the same parameters as the AIOps Instana Topology Integrator, as described in Configuring Instana Observer jobs.

Remember: When you use the Instana Observer instead of the Connector, the `name` attribute of a resource is now the Instana label, which means that if the name was previously used in a rule (for example a File Enrichment Rule), you must amend the rule accordingly.

Configuring Jenkins Observer jobs

Using the Jenkins Observer, you can define listen jobs that receive build information generated by the Jenkins plug-in.

Before you begin

Ensure you have all required Jenkins details to hand.

Before you configure the Jenkins Observer, you must install and configure the Jenkins plug-in provided with the installation images. The plug-in sends your instrumented build data to the Jenkins Observer in order to generate the appropriate topology information: Configuring the Jenkins plug-in.

For the Jenkins plug-in to be able to communicate and send information to a Jenkins Observer job, you must enable a URL route for the Jenkins Observer, which is not exposed in your installation by default.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Listen job: The stand-alone listen job receives Jenkins build notification data for a specified namespace and processes it as a topology.

The listen job is long-running, and runs until it is explicitly stopped or until the observer is stopped.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Provider namespace	Specify a provider namespace with which to associate all the topology data generated by this job	Required
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring the Jenkins plug-in

The IBM Cloud Pak® for AIOps software includes the Jenkins plug-in, which you install on your Jenkins server using the Jenkins Plugin Manager Advanced installation web UI. From the Jenkins server, the plug-in gathers and sends information to the Jenkins Observer.

Ensure you have all the required administrator or management permissions to deploy the IBM Cloud Pak for AIOps Jenkins component onto the Jenkins server.

Plug-in	Version	Description
Jenkins core	2.159	Jenkins server minimum version
apache-httpcomponents-client-4-api	4.5.10-2.0	Provided by Jenkins core
workflow-step-api	2.22	
workflow-cps	1.9	
workflow-multibranch	2.17	
credentials	2.3.1	Compatible version usually provided by Jenkins core
Git	2.6.5	
artifactory	3.15.0	Optional (for using the Artifactory integration only)

Obtain the Jenkins plug-in

You can find the Jenkins plug-in included with the Jenkins Observer image. For example:

```
JENKINS_OBSERVER_POD=(oc get po -l app.kubernetes.io/name=jenkins-observer -o jsonpath='{.items[0].metadata.name}')

echo {JENKINS_OBSERVER_POD}
evtmanager-topology-jenkins-observer-84c456f58c-jxlrj
```

To copy the plug-in from the image:

```
oc cp JENKINS_OBSERVER_POD:/opt/ibm/asm-observer-plugin-2.0.0-SNAPSHOT.hpi asm-observer-plugin-2.0.0-SNAPSHOT.hpi

ls asm-observer-plugin-2.0.0-SNAPSHOT.hpi
asm-observer-plugin-2.0.0-SNAPSHOT.hpi
```

Obtain credentials

The credentials for the API are dynamically created. If IBM Cloud Pak for AIOps is installed with a release name of **asm**, then:

```
CREDS_USER=(oc get secret evtmanager-topology-asm-credentials -o jsonpath={.data.username} | base64 -d)
CREDS_PASS=(oc get secret evtmanager-topology-asm-credentials -o jsonpath={.data.password} | base64 -d)
```

Use the following command to print the UI API credentials:

```
echo CREDS_USER CREDS_PASS
```

Example user and password credentials returned:

```
aimanager-topology-aiops-user N8gJJGEfmmDnF6Q/1zg8NyAGKgQ9PmZQLhUSKd9/j54=
```

Obtain certificate

The following sample steps apply to the default router deployed on Red Hat OpenShift Container Platform. If more routers are used, adjust these steps. You perform these steps as **cluster administrator**.

Obtain the certificate:

```
oc get secret -n openshift-ingress router-certs-default -o jsonpath='{.data.tls\.crt}' | base64 -d > ocp-default-router.crt
```

Obtain the certificate key:

```
oc get secret -n openshift-ingress router-certs-default -o jsonpath='{.data.tls\.key}' | base64 -d > ocp-default-router.key
```

Convert to PKCS12:

```
openssl pkcs12 -export -out ocp-default-router.p12 -inkey ocp-default-router.key -in ocp-default-router.crt
Enter Export Password: enter-suitably-secure-password
Verifying - Enter Export Password: enter-suitably-secure-password
```

Ensure that the files you have extracted from Red Hat OpenShift Container Platform are securely deleted.

Obtain observer URL

Obtain the hostname as in the following example:

```
oc get routes -l app.kubernetes.io/name=jenkins-observer -o jsonpath='{.items[*].spec.host}'
```

The Jenkins plug-in defines Jenkins pipeline DSL steps that you use to instruct the Jenkins builds to generate and then send topology data to the Jenkins Observer. Use the DSL steps to send the notification to include Artifactory BuildInfo modules information, in order for the build products to be modeled. You create an observer instance, which then triggers the sending of the notification to the observer.

For information on configuring Jenkins Observer jobs, see the related tasks.

Important: You perform the following steps as **Jenkins administrator**.

Install the Jenkins plug-in

1. Log in to your Jenkins server with a user ID that has administration (management) permissions.
2. Navigate to **Manage Jenkins** > **Manage Plugins** and select the **Advanced** tab.
3. In the Upload Plugin section, upload and install the provided plug-in (asm-observer-plugin-2.0.0-SNAPSHOT.hpi)

Configure the Jenkins plug-in

1. Navigate to **Manage Jenkins** > **Configure System** and select the **Agile Service Manager plugin**.
2. Enter the Jenkins Observer URL obtained earlier in the **Observer URL** field.
3. Create credentials for IBM Cloud Pak for AIOps using the **Manage Jenkins** > **Credentials** configuration page.

   Enter the credentials and the certificate details obtained earlier from the (or as the) cluster administrator.

   Auth Credentials Enter the username and password to authenticate with the Jenkins Observer at API level.

   Click **ADD**.

   Certificate Enter the PKCS12 format certificate to secure the communication between the Jenkins plug-in and the observer using SSL.

   Click **ADD**.

   For more detailed information on using the Jenkins Credentials feature, see the Jenkins site: https://jenkins.io/doc/book/using/using-credentials/
4. Enter a **Default Tenant ID**.

   The tenant to be used as default when sending information to IBM Cloud Pak for AIOps (if it is not overridden in the build).
5. Enter a **Default Job ID**.

   The default job ID to be used when sending information to IBM Cloud Pak for AIOps (if it is not overridden in the build). This represents the ID of the job the observer is running to process your information.
6. Click **TEST CONNECTION** to attempt to establish a connection with the observer using the configured credentials.

   If successful, a Connection successful! message is displayed. If unsuccessful, an error is returned.

   Note: The connection test does not verify the default job and tenant IDs.

Configuring your Jenkins build to notify IBM Cloud Pak for AIOps

After configuring the installed plug-in in Jenkins, you integrate the provided functionality into your existing Jenkins build. The following two steps describe how to use scripted pipeline syntax to send your build information to the observer.

Remember: You perform the following two steps on your Jenkins build (and **not** the Jenkins Observer).

1. Create an instance of the IBM Cloud Pak for AIOps observer communication object using the default credentials set during the configuration steps.

   ```
   def <my observer> = ASM.newObserver()
   ```

   Tip: Override credentials: to override any configured credentials (that is, the Jenkins credentials ID rather than the actual credentials themselves), you can use the following syntax:

   ```
   def <my observer> = ASM.newObserver
   authCredsId: 'auth-credentials-id', certCredsId: 'cert-credentials-id'
   ```

2. Send a build notification using the following syntax:

   ```
   ASM.notifyASM asmObserver: <my observer>
   ```

   Tip: You can override the default values for the tenant and job IDs:

   ```
   ASM.notifyASM tenantId: '<tenantId>', jobId: '<jobId>', asmObserver: <my observer>
   ```

   You can integrate the Jenkins plug-in with the Artifactory component using the modules information to model any artifacts the build has published to the repository:

   ```
   ASM.notifyASM asmObserver: <my observer>, artModules: <buildInfoObject>.getModules()
   ```

The following Jenkins file code sample shows the integration with IBM Cloud Pak for AIOps. It starts with the declaration of the observer instance, and ends with the notification to IBM Cloud Pak for AIOps. This notification should be added as a **`post / always`** block, so that the information is sent as the last operation of the build, and is sent regardless of the result of the build.

The optional **`try / catch`** around the notification to IBM Cloud Pak for AIOps is included in this sample to avoid the build failing if the notification to IBM Cloud Pak for AIOps fails.

The following sample also depicts the use of Artifactory to publish the build products, and how that information can be included in the notification to IBM Cloud Pak for AIOps using the artModules parameter.

```
**def asm = ASM.newObserver()**
def buildInfo = Artifactory.newBuildInfo()
buildInfo.env.capture = true

… your build logic here …
**
post {
    always {
      script {
            try {
                ASM.notifyASM asmObserver: asm, artModules: buildInfo.getModules()
            } catch (err) {
                echo An error ocurred sending a notification to ASM:  {err}"
            }
          }
      }
   }
}**
```

Tip: You can further refine your integration and visualization.

After you install and configure the Jenkins plug-in, configure the Jenkins Observer listen job.

You can further refine your IBM Cloud Pak for AIOps system to take advantage of its topology and other customization functionality. For more information, see Customizing the Jenkins topologies.

You can extend your Jenkins integration with rules to merge data from different sources, custom topology display conventions, and the use of templates for the automated generation of topologies. For more information, see Additional Jenkins plug-in customization tasks.

More Jenkins plug-in customization tasks

You can extend your Jenkins integration with rules to merge data from different sources, custom topology display conventions, and the use of templates for the automated generation of topologies.

You first install and configure the Jenkins plug-in, before extending its functionality as described here.

Trouble: For useful Jenkins troubleshooting information, see troubleshooting.

Customizing the Jenkins topologies

You can customize how your Jenkins topologies are displayed in a number of ways, either by using the Jenkins customizations supplied with IBM Cloud Pak® for AIOps, or by creating your own custom display elements.

Creating Jenkins topology templates

Over time topologies generated through a specific Jenkins Observer job may grow, including the information from different Jenkins builds. You can use the IBM Cloud Pak for AIOps topology template functions to create meaningful, scoped views per build.

Example

The information obtained from Jenkins through the Red Hat OpenShift Container Platform plug-in for Jenkins and the Jenkins Observer includes URLs for several Jenkins elements. You can create topology tools that use these URLs to provide in-context navigation to Jenkins or your Git repositories from within the topologies displayed in Red Hat OpenShift Container Platform. The code snippets in this section demonstrate how to create such tools.

If required, you can customize how your Jenkins topologies are displayed further using the Configuring topology elements topics.

Opens a GitHub repository commits log

```
function url2Http(url) {
  if(url.indexOf("http") != -1) {
      return url;
  }
  var res = url.split(":");
  var ghUrl = res[0].replace("git@", "https://");
  var repoUrl = res[1].replace(".git", "");
  return ghUrl + "/" + repoUrl;
}

if (asmProperties.repositoryUrl && asmProperties.revision) {
    var normUrl = url2Http(asmProperties.repositoryUrl);
    window.open(normUrl + '/ commits/'
        + asmProperties.revision);
}
```

Relationship type is **builds**

Opens the specific commit

```
function url2Http(url) {
  if(url.indexOf("http") != -1) {
      return url;
  }
  var res = url.split(":");
  var ghUrl = res[0].replace("git@", "https://");
  var repoUrl = res[1].replace(".git", "");
  return ghUrl + "/" + repoUrl;
}

if (asmProperties.repositoryUrl && asmProperties.revision) {
    var normUrl = url2Http(asmProperties.repositoryUrl);
    window.open(normUrl + '/commit/'
        + asmProperties.revision);
}
```

Relationship type is **builds**

Opens the Jenkins resources records

```
if (asmProperties.url) {
    window.open(asmProperties.url);
}
```

Resource types are **person**, **build**, **job**

Opens a GitHub repository tree view from the commit

```
function url2Http(url) {
  if(url.indexOf("http") != -1) {
      return url;
  }
  var res = url.split(":");
  var ghUrl = res[0].replace("git@", "https://");
  var repoUrl = res[1].replace(".git", "");
  return ghUrl + "/" + repoUrl;
}

if (asmProperties.repositoryUrl && asmProperties.revision) {
    var normUrl = url2Http(asmProperties.repositoryUrl);
    window.open(normUrl + '/tree/'
        + asmProperties.revision);
}
```

Relationship type is **builds**

Opens the build log console view in Jenkins

```
if (asmProperties.buildConsoleLog) {
    window.open(asmProperties.buildConsoleLog);
}
```

Resource type is `build`

Creating Jenkins topology templates

Over time topologies generated by a specific Jenkins Observer job may grow, including the information from different Jenkins builds. You can use the IBM Cloud Pak® for AIOps group templates functionality to create meaningful, scoped groups per build. This can then be included as part of a broader application definition.

Create 'per build' topology templates

The following template generates a group for each build, which includes the build and its associated pipeline resources:

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Application management**.
3. Click the **Group templates** icon on the upper right corner of the page.
4. From the **Group templates** page, click **Create a new template**.
5. Select **Dynamic template** to ensure that all matching builds will generate topologies like this.
6. Click **Start** to begin building your template.
7. On the Preview pane, search for the build resource and select **Follow Relationship**. Repeat this for all of the connected resources in your build topology.

   Important: Do not expand the 'lastRunOf' relationship.
8. Set the Group type as 'cicd'.
9. (Optional) Define tags for the template, then click **Save template & generate groups**.

Create `current pipeline status` group templates.

The following template generates a group that shows the status of your pipeline that is the latest build for which a notification was received by the observer and associated resources:

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Application management**.
3. Click the **Group templates** icon on the upper right corner of the page.
4. From the **Group templates** page, click **Create a new template**.
5. Select **Dynamic template** to ensure that all matching builds will generate topologies like this.
6. Click **Start** to begin building your template.
7. On the Preview pane, search for the job resource and select **Follow Relationship**. Then click 'lastRunOf'. The latest build resource for which a topology was generated is displayed. Expand any relationships you want to include in this template.

Important: Do not expand the 'RunOf' relationship.

6. Set the Group type as 'cicd'.
7. (Optional) Define tags for the template, then click **Save template & generate groups**.

Configuring Juniper CSO Observer jobs

Using the Juniper CSO Observer, you can define a full load job that will gather and read data about topology data from Juniper CSO.

Before you begin

Important: The Juniper CSO observer supports the on-premises Juniper CSO version 4.1.0.

Ensure you have the Juniper CSO details to hand, such as details of the Juniper CSO apiserver and its credentials.

The Juniper CSO Observer retrieves topology data from Juniper CSO Release 4.1 through REST APIs exposed by CSO API server.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Run this job whenever you need the Juniper CSO topology data refreshed.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Juniper CSO Central MS URL	Specify the Juniper CSO Central micro-service URL	Required
CSO Keystone Authentication URL	Enter the authentication URL for the identity service	Required
CSO user domain name	Enter the CSO user domain name	Required
CSO domain or project or tenant name	Enter the CSO domain or project or tenant name	Required
CSO Authentication username	Specify the CSO authentication username	Required
CSO Authentication password	Specify the CSO authentication password	Required
Enable/Disable Secure Connection to CSO Host	Set to 'true' to secure the connection, otherwise set to 'false' to bypass	Required
Target system certificate	Specify the certificate file name.	Required if **`Enable/Disable Secure Connection to CSO Host`** is set to 'true'. To obtain authentication certificates using OpenSSL and store them as secrets, see Configuring observer jobs security.
SSL truststore File	Specify the HTTPS truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (**`<observer>.jks`**), for example **`JuniperCSO.jks`**.	Required if **`Enable/Disable Secure Connection to CSO Host`** is set to 'true'. The supported format is JKS and the file is relative to $ASM_HOME/security.
SSL truststore Password	Specify a password to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required if **`Enable/Disable Secure Connection to CSO Host`** is set to 'true'.
SSL Validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Kubernetes Observer jobs

Using this observer, you can configure jobs that discover the structure of your Kubernetes clusters, including pods, worker nodes and containers.

Before you begin

Important: The Kubernetes Observer has been tested against OpenShift version 4.12.53 with Kubernetes 1.25.8, and GKE version 1.27.8-gke.1067004.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The observer reads topology data from Kubernetes through its REST APIs, or Weave Scope.

You can run the following jobs:

Load

A transient (one-off) job that loads all requested topology data from a Kubernetes environment.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Tip: When observing different Kubernetes clusters, make sure that the name of the data center is unique across all of the jobs.

Listen

A long-running job that monitors the Kubernetes REST API for updated resources, and loads them in the topology service.

Important: A Load job must be run before a Listen job can run.

Local

Note: The Kubernetes Observer Local job has been deprecated as of IBM Cloud Pak for AIOps 3.4.0. Use the Load job instead to performs local observations of the Kubernetes REST API for available resources.

Weave_scope Listen

Note: The Kubernetes Observer Weave Scope listen job has been deprecated as of IBM Cloud Pak for AIOps 4.5.0.

The Weave Scope listen job is a stand-alone job that listens to the Weave Scope agent and continues to stream topology and state data to IBM Cloud Pak® for AIOps.

It provides visibility of your Kubernetes services, pods, containers, deployments, stateful sets, Cron Jobs and processes for a specified namespace.

It is a long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the Observer is stopped.

Kubeconfig Load

A stand-alone load job that accepts kubeconfig data in JSON format to perform a one-off discovery.

Kubernetes Observer Load job prerequisites

Before defining a Kubernetes Observer Load job, you must create a service account in the Kubernetes environment and obtain its token. The following example uses the default namespace.

1. Create either cluster role access or role access.

   For clusterole access, create a configuration file called **`asm-k8s-observer.yaml`** with the custom cluster role **`asm:kubernetes-observer`**.

   Use the following sample content:

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  name: asm:kubernetes-observer
rules:
- apiGroups: [""]
  resources: [ "pods", "namespaces", "nodes", "services", "endpoints", "persistentvolumes",
"persistentvolumeclaims"]
  verbs: ["get", "list"]
- apiGroups: ["apps"]
  resources: ["replicasets", "deployments", "statefulsets", "daemonsets"]
  verbs: ["get", "list"]
```

   For role access:

Use the following sample content:

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: {namespace} <-- Specify which namespace to create the Role access
  name: asm:kubernetes-observer
rules:
- apiGroups: [""]
  resources: [ "pods", "services", "endpoints"]
  verbs: ["get", "list"]
- apiGroups: ["apps"]
  resources: ["replicasets", "deployments", "statefulsets", "daemonsets"]
  verbs: ["get", "list"]
```

Observing Custom Resources:

To model custom resource definitions (CRDs) and custom resources (CRs) in the topology, identify the names of the CRDs that are present in the environment by running the **`oc get crds`** command. Then, update the job parameters to include the names of CRDs to be discovered. When a CRD is being included in the job, the respective CRs will also be discovered.

Example (discovering ibm.com CRDs):

```
oc get crds | grep ibm.com
```

Sample output:

```
asmformations.asm.ibm.com                                 2020-12-03T01:53:38Z
asms.asm.ibm.com                                          2020-12-03T01:53:38Z
cemformations.cem.ibm.com                                 2020-12-03T01:53:38Z
cemserviceinstances.cem.ibm.com                           2020-12-03T01:53:38Z
noiconnectionlayers.noi.ibm.com                           2020-12-03T01:47:19Z
noiformations.noi.ibm.com                                 2020-12-03T01:47:19Z
noihybrids.noi.ibm.com                                    2020-12-03T01:47:19Z
nois.noi.ibm.com                                          2020-12-03T01:47:20Z
```

Also, change the existing **`ClusterRole`** in the target environment, to include the following resources:

```
- apiGroups: ["apiextensions.k8s.io"]
  resources: ["customresourcedefinitions"]
  verbs: ["get", "list"]
- apiGroups: ["noi.ibm.com"]
  resources: ["nois", "noihybrids", "noiformations", "noiconnectionlayers"]
  verbs: ["get", "list"]
- apiGroups: ["asm.ibm.com"]
  resources: ["asms", "asmformations"]
  verbs: ["get", "list"]
- apiGroups: ["cem.ibm.com"]
  resources: ["cemformations"]
  verbs: ["get", "list"]
```

The **`apiextensions.k8s.io`** segment of the cluster role provides your permission required to access all the custom resource definitions resources.

```
- apiGroups: ["apiextensions.k8s.io"]
  resources: ["customresourcedefinitions"]
  verbs: ["get", "list"]
```

The **`asm.ibm.com`** segment provides access to **`asms`** and **`asmformations`** custom resources. The names of the resources listed are the plural for each of the resources. They are usually derived from the first word of the CRD.

```
- apiGroups: ["asm.ibm.com"]
  resources: ["asms", "asmformations"]
  verbs: ["get", "list"]
```

For a fresh discovery:

Create a configuration file called **`asm-k8s-observer.yaml`** in the target environment using the yaml resource provided in the sample content, then run the following command before moving to the next step:

```
oc create -f asm-k8s-observer.yaml
```

For an existing discovery:

Create a configuration file called asm-k8s-observer.yaml using the yaml resource provided in the sample, then run the following command to patch the clusterole:

```
oc patch clusterrole asm:kubernetes-observer --patch "$(cat asm-k8s-observer.yaml )"
```

2. Create the service account in the target environment.

```
oc create serviceaccount asm-k8s-account -n <namespace>
```

To verify that the service account exists:

```
oc get serviceaccount
```

Important: When running the following command, ensure you know the namespace in which it was created:

```
oc create serviceaccount asm-k8s-account
```

3. Bind the **asm:kubernetes-observer** ClusterRole or role to the **asm-k8s-account** service account.

   Important: The service account and the cluster role binding need to be created in the same namespace.

   Example command for clusterole binding:

   ```
   oc create clusterrolebinding asm-k8s --clusterrole=asm:kubernetes-observer --serviceaccount={namespace}:asm-k8s-account
   ```

   Example command for role binding:

   ```
   oc create rolebinding asm-k8s --role=asm:kubernetes-observer --serviceaccount={namespace}:asm-k8s-account
   ```

   Example role binding output:

   ```
   rolebinding.rbac.authorization.k8s.io/asm-k8s created
   ```

4. Obtain the Kubernetes service account token. Complete the following substeps depending on your version of Kubernetes.

   For Kubernetes on OpenShift:

   1. Look for the mounted service account token secret from the secrets section:

      ```
      oc describe serviceaccount asm-k8s-account
      ```

      Example:

      ```
      oc describe serviceaccount asm-k8s-account
      Name: asm-k8s-account
      Namespace: default
      Labels: <none>
      Annotations: <none>
      Image pull secrets: asm-k8s-account-dockercfg-wwzx2
      Mountable secrets: asm-k8s-account-dockercfg-wwzx2
      Tokens: asm-k8s-account-token-tzl52
      Events: <none>
      ```

   2. Describe the asm-k8s-account-token-******* (which in this example is tzl52) to obtain the token's value:

      ```
      oc describe secret asm-k8s-account-token-tzl52
      ```

      Alternatively, use the following method to obtain the token's value (applicable to both OpenShift and GKE environments):

      ```
      oc describe serviceaccount asm-k8s-account | grep Tokens | awk '{ print $2} ' | xargs -I{} oc get secret {} --template='{{.data.token}}' | base64 -d
      ```

      For Kubernetes load jobs, ensure you have the Kubernetes service details to hand, such as the Kubernetes host IP and SSL Certificate details.

   For Kubernetes on GKE:

   1. Verify that the token has been created using the **kubectl describe serviceaccount <service-account>** command.

      Example of a missing token:

      ```
      oc kubectl describe serviceaccount asm-k8s-account
      Name:                asm-k8s-account
      Namespace:           default
      Labels:              <none>
      Annotations:         <none>
      Image pull secrets:  <none>
      Mountable secrets:   <none>
      Tokens:              <none>
      Events:              <none>
      ```

   2. If missing, create the token manually.

      Example:

      ```
      kubectl apply -f - <<EOF
      apiVersion: v1
      kind: Secret
      metadata:
         name: asm-k8s-account-secret
         annotations:
            kubernetes.io/service-account.name: asm-k8s-account
      type: kubernetes.io/service-account-token
      EOF
      secret/asm-k8s-account-secret created
      ```

   3. Verify that the token exists.

      Example of a token that is present:

      ```
      oc kubectl describe serviceaccount asm-k8s-account
      Name:                asm-k8s-account
      Namespace:           default
      Labels:              <none>
      Annotations:         <none>
      Image pull secrets:  <none>
      Mountable secrets:   <none>
      ```

```
Tokens:              asm-k8s-account-secret
Events:              <none>
```

4. Describe the secret to obtain the token's value.

   Example:

```
oc kubectl describe secret asm-k8s-account-secret
Name:         asm-k8s-account-secret
Namespace:    default
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: asm-k8s-account
              kubernetes.io/service-account.uid: d82e86c1-2258-4072-8a53-f517f776a149

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1509 bytes
namespace:  7 bytes
token:      eyJhbGciO
```

Kubernetes Observer Listen job prerequisites

Important: Before defining a Kubernetes Observer Listen job, you must have configured and run a Load job.

Before defining a Listen job, update the ClusterRole or Role configuration.

1. Edit the **asm-k8s-observer.yaml** configuration file and add the following information:

   For clusterole access, add **watch** as in the following example.

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  name: asm:kubernetes-observer
rules:
- apiGroups: [""]
  resources: [ "pods", "namespaces", "nodes", "services", "endpoints", "persistentvolumes",
"persistentvolumeclaims"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
  resources: ["replicasets", "deployments", "statefulsets", "daemonsets"]
  verbs: ["get", "list", "watch"]
```

   For role access, add **watch** as in the following example.

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: asm:kubernetes-observer
rules:
- apiGroups: [""]
resources: [ "pods", "services", "endpoints"]
verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
resources: ["replicasets", "deployments", "statefulsets", "daemonsets"]
verbs: ["get", "list", "watch"]
```

2. Run the following command:

```
oc apply -f asm-k8s-observer.yaml
```

Kubernetes Observer Weave Scope listen job prerequisites

For Weave Scope listen jobs, do a Weave Scope installation, and then configure a job using the Weave Scope IP and port parameters. You must install Weave Scope and then use the Weave Scope Master IP and Node port parameters. For more information on Weave Scope, see the following location: https://www.weave.works/docs/scope/latest/introducing/

1. Create namespace 'weave' with 'ibm-privileged-psp'.

oc create namespace weave oc -n weave create rolebinding weave-clusterrole-rolebinding --clusterrole=ibm-privileged-clusterrole --group=system:serviceaccounts:weave

```
2. Install Weave Scope using the following command:

  ```
oc apply -f "https://cloud.weave.works/k8s/scope.yaml?k8s-service-type=NodePort&k8s-version=$(kubectl version | base64 |
tr -d '\n')"

This will result in a port being opened that the observer can use.
```

3. You can discover the NodePort using the following command:

oc -n weave describe service weave-scope-app
```

4. Launch the Weave Scope UI using the following URL:

```
https://<master ip>:<NodePort>
```

5. Create a new route using the Red Hat OpenShift Container Platform console to expose the Red Hat OpenShift Container Platform Weave Scope service under the weave namespace. In the left panel of the Red Hat OpenShift Container Platform console, click **Networking** > **Routes**.
6. Click **Create Route** to create a new route.
7. Complete the required fields and then click **Create**.
8. A new route has been created. Launch the Weave Scope UI using the location link.

Kubernetes Observer kubeconfig load job prerequisites

For a kubeconfig load job, ensure your kubeconfig.yml file is in the correct format. If it is not, generate a new file.

Important: The kubeconfig should meet the following criteria:

- Single cluster
- User type of 'token'
- Base64 encoded JSON

View the kubeconfig.yml file:

```
oc config view --raw -o json
```

Important: Make sure the token and certificate are not omitted or redacted.

If it meets all required conditions, encode it using the following command:

```
oc config view --raw -o json | base64 -w 0
```

If it does not meet all conditions, generate a new kubeconfig.yml file using the following steps:

1. Create a cluster-role or role, service account, cluster-role/role binding. From the service account, get the secret, and from the secret get the token. Refer to "Kubernetes Observer Load job prerequisites" for detailed steps.
2. Get the current context details:

   ```
   oc config current-context
   ```

3. Get the current cluster, cluster ca (certificate-authority-data), and cluster server details:

   ```
   oc config view --raw -o json
   ```

4. Create a kubeconfig.yml using the obtained data. Sample kubeconfig.yml file:

   ```
   apiVersion: v1
   kind: Config
   current-context: <CURRENT_CONTEXT>
   contexts:
   - name: <CURRENT_CONTEXT>
     context:
       cluster: <CURRENT_CONTEXT>
       user: <USER>
       namespace: <NAMESPACE>
   clusters:
   - name: <CURRENT_CONTEXT>
     cluster:
       certificate-authority-data: <CLUSTER_CA>
       server: <CLUSTER_SERVER>
   users:
   - name: <USER>
     user:
       token: <TOKEN_VALUE>
   ```

   Where **USER** is the user-defined value, and **NAMESPACE** is the namespace of the service account.

5. Test the kubeconfig validity and access:

   ```
   oc --kubeconfig <config>.yml get all
   ```

6. Run the following command to convert the file to base64 encoded json:

   ```
   oc --kubeconfig <config>.yml config view --raw -o json | base64 -w 0
   ```

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Load job parameters

Parameter	Action	Details

Parameter	Action	Details
Unique ID	Enter a unique name for the job. **Note:** The name cannot be the same as any of the following observer job names: 'local', 'listen', 'weave_scope', 'load', 'kubeconfig'.	Required
data_center	Specify the name of the data center in which the Kubernetes instance is running. The data center name is treated as the provider of the data. **Note:** If you change the data center name, the observation is treated as a separate observation under the different provider, and a duplicate set of resources and groups will be created under the new provider.	Required
Kubernetes Master IP address	Enter the Kubernetes Master IP address	Required. Get the Kubernetes master IP and its API port using the 'oc cluster-info' command.
Kubernetes API port	Enter the Kubernetes API port number	Required
Kubernetes token	The service account token for Kubernetes	Required
Trust all certificate by bypassing certificate verification	Enter **true** if you want to connect to Kubernetes without certificate	Required
Exact HTTPS certificate file name	Enter the exact name of the SSL/HTTPS certificate	Optional. If 'Trust all certificate' is set to **false**, then this parameter is Required.
SSl Validation	Choose whether SSL validation is **true** or **false**. If **false**, HTTPS without hostname validation is used.	Optional
Terminated pods	Choose whether terminated pods should be hidden (**true** or **false**)	Optional. The default is false.
API query timeout (ms)	Specify the Kubernetes REST API query timeout	Optional. The default is 5000 ms (that is, 5 seconds)
Correlate	If 'true', enables the Event Analytics correlation on the namespace groups	Optional
Names of the custom resource definitions	Specify the custom resource definitions (CRDs) queryable from the Kubernetes API	Optional
Role permission token	Determine whether the provided token is a role token. Defaults to `false` for cluster role permissions.	Optional
Namespace	Specify the Kubernetes namespace	Optional. Specify the Kubernetes namespace to observe, leave blank or enter `*` to observe all.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Listen job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job. **Note:** The name cannot be the same as any of the following observer job names: 'local', 'listen', 'weave_scope', 'load', 'kubeconfig'.	Required
data_center	Specify the name of the data center in which the Kubernetes instance is running. The data center name is treated as the provider of the data. **Note:** If you change the data center name, the observation is treated as a separate observation under the different provider, and a duplicate set of resources and groups will be created under the new provider.	Required
Kubernetes Master IP address	Enter the Kubernetes Master IP address	Required. Get the Kubernetes master IP and its API port using the 'oc cluster-info' command.

Parameter	Action	Details
Kubernetes API port	Enter the Kubernetes API port number	Required
Kubernetes token	The service account token for Kubernetes	Required
Trust all certificate by bypassing certificate verification	Enter **true** if you want to connect to Kubernetes without certificate	Required
Exact HTTPS certificate file name	Enter the exact name of the SSL/HTTPS certificate	Optional. If 'Trust all certificate' is set to **false**, then this parameter is Required.
SSl Validation	Choose whether SSL validation is **true** or **false**. If **false**, HTTPS without hostname validation is used.	Optional
Terminated pods	Choose whether terminated pods should be hidden (**true** or **false**)	Optional. The default is false.
API query timeout (ms)	Specify the Kubernetes REST API query timeout	Optional. The default is 5000 ms (that is, 5 seconds)
Correlate	If 'true', enables the Event Analytics correlation on the namespace groups	Optional
Names of the custom resource definitions	A number of custom resource definitions (CRDs) that can be queried from the Kubernetes API	Optional
Role permission token	Determine whether the provided token is a role token. Default to false for cluster role permissions.	Optional
Namespace	Specify the Kubernetes namespace	Optional. Specify the Kubernetes namespace to observe, leave blank or enter `*` to observe all.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Weave_scope job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job. **Note:** The name cannot be the same as any of the following observer job names: 'local', 'weave_scope', 'load', 'kubeconfig'.	Required
Host	Enter the Weave Scope hostname (or IP address) of the web socket to be observed	Required
Port	Enter the Weave Scope port number of the web socket to be observed	Required. The Port can be obtained using the 'oc -n weave describe service weave-scope-app' command.
Cluster Name	Enter the name of the cluster or data center to be observed	Required
Namespaces	Enter a list of namespaces to be observed	Optional. If left empty, all namespaces will be observed.
Resource types	Select the Weave Scope resource types to observe	Optional.
Resources to exclude	List resources to be excluded by ID, label, rank or namespace	Optional. Containers named 'pod' are excluded by default.
Refresh interval	Specify the refresh interval in seconds between updates. The minimum allowed value is 60s, and the default value is 120s.	Optional

Parameter	Action	Details
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Kubeconfig load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job. **Note:** The name cannot be the same as any of the following observer job names: 'local', 'weave_scope', 'load', 'kubeconfig'.	Required
data_center	Specify the name of the data center in which the Kubernetes instance is running. The data center name is treated as the provider of the data. **Note:** If you change the data center name, the observation is treated as a separate observation under the different provider, and a duplicate set of resources and groups will be created under the new provider.	Required
kubeconfig	Provide the contents of the kubeconfig in json format. The kubeconfig needs to be a json with a single cluster and its user needs to be of token type.	Required. The json needs to be base64 encoded.
Terminated pods	Choose whether terminated pods should be hidden (**true** or **false**)	Optional. The default is false.
API query timeout (ms)	Specify the Kubernetes REST API query timeout	Optional. The default is 5000 ms (that is, 5 seconds)
Correlate	If 'true', enables the Event Analytics correlation on the namespace groups	Optional
Names of the custom resource definitions	A number of custom resource definitions (CRDs) that can be queried from the Kubernetes API	Optional
Role permission token	Determine whether the provided token is a role token. Default to false for cluster role permissions.	Optional
Namespace	Specify the Kubernetes namespace	Optional. Specify the Kubernetes namespace to observe, or * to observe all. Leaving this field blank will observe the namespace in which IBM Cloud Pak for AIOps is installed. When using a role permission token, specify the single namespace where the role was created.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Network Manager Observer jobs

Using the IBM Tivoli Network Manager Observer, you can define jobs that dynamically load data discovered by Network Manager for analysis.

Before you begin

Important: The IBM Tivoli Network Manager Observer supports the on-premises IBM Tivoli Network Manager version 4.2.

Ensure you have the IBM Tivoli Network Manager service details to hand, such as the domain, host and port number.

The IBM Tivoli Network Manager Observer jobs extract Network Manager resources using an Object Query Language JDBC driver, and the observer loads and updates the resources and their relationships within the topology.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You configure the following two jobs.

Load

A transient (one-off) job that queries Network Manager for topology data, and performs a complete upload for a single IBM Tivoli Network Manager domain.

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

For cross-domain connections: Each ITNM domain must read existing cross-domain connections from the topology service, which means that you may need to run two observations of all domains contributing cross-domain connections (that is, run the full load job twice) before all cross-domain edges are created.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Listen

A long-running job that monitors the Network Manager message bus for changes and updates the topology service. When the job is started, the observer creates an OQL connection that listens for changes in the IBM Tivoli Network Manager network. Any resources added, changed or deleted are passed on by the OQL connection and the topology is updated.

The listen job runs until it is explicitly stopped, or until the observer is stopped.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
ITNM instance name	Specify the IBM Tivoli Network Manager instance name	Required. Specify the same instance name for all jobs to enable connectivity across domains.
ITNM domain port	Specify the IBM Tivoli Network Manager port for the specified domain	Required. The value of `port` will vary if multiple domains exist. To determine which port number to use for a Network Manager domain, look for the domain-specific ncp_config entry in the `$NCHOME/etc/precision/ServiceData.cfg` file.
ITNM domain	Specify the IBM Tivoli Network Manager domain	Required
Hostname or Server IP	Specify the IBM Tivoli Network Manager hostname or server IP on which the domain is running	Required
OQL connection timeout (ms)	Specify the OQL Connection timeout value	Optional. The default is 3000 (30 seconds).
Edge Type Map	Map IBM Tivoli Network Manager layers to topology relationship types	Optional. If blank, IBM Cloud Pak® for AIOps will auto-map. The value of topology type can be found in IBM Tivoli Network Manager in '$NCHOME/precision/disco/stitchers/DNCIM/PopulateDNCIMTopologies.stch'; alternatively, run the following OQL statement against the model service to list the available topology type: `select ENTITYNAME from ncimCache.entityData where METACLASS='Topology'`
Exclude resources without connections	Select whether to display disconnected resources	Optional. The choices are **true** and **false**. The default is **true**.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring a New Relic Observer job

Use the New Relic Observer when you have a New Relic account with a New Relic Infrastructure subscription. Using this observer you can configure jobs that dynamically load New Relic Infrastructure resource data through New Relic for analysis.

There is a known issue that is currently impacting the New Relic Observer job. For more information, see New Relic observer job fails with an authorization error.

Before you begin

Important: The New Relic Observer supports the cloud/SaaS New Relic version.

Ensure you have the New Relic account and New Relic Infrastructure subscription details to hand, such as the account name, account ID, and New Relic Insights API query key.

Restriction: New Relic applies a 1000 results limit on all New Relic Query Language (NRQL) queries. To accommodate this limit when retrieving data from the SystemSample, StorageSample, ProcessSample and NetworkSample event tables, the New Relic Observer uses the following NRQL query time clause:

```
"SINCE 4 hours ago LIMIT 1000"
```

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The Observer uses the New Relic Infrastructure subscription and makes active New Relic Query Language (NRQL) calls over REST to New Relic Insights to download New Relic Infrastructure resource data.

The New Relic Observer loads the following New Relic Infrastructure resources and their relationships:

- Host
- Storage
- OS
- Network Interfaces
- Processes

The New Relic Observer job extracts New Relic Infrastructure resources from New Relic using New Relic Query Language (NRQL) over REST, and the observer loads and updates the resources and their relationships.

You configure the following job.

Load: By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see Configuring observer job security

Table. Parameter details

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
New Relic Account Name	Specify the New Relic account name or tenant name	Required
New Relic account ID in numeric	Specify the New Relic numeric account ID	Required.
New Relic Insights Query API key	Specify the New Relic Insights Query API key	Required.
New Relic Events table filter criteria	Extend the result set returned to IBM Cloud Pak® for AIOps	Optional. The default value is `SINCE 4 hours ago LIMIT 1000`. For more information, see the documentation for New Relic Query Language.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Tips:

- **New Relic account ID**: To obtain the account ID log in to New Relic and then go to the URL https://rpm.newrelic.com/accounts/%3CaccountId%3E

- **New Relic Insights Query API Key**: A New Relic user with a New Relic Infrastructure subscription is required to generate a New Relic Insights Qeury API Key. For more information, see https://docs.newrelic.com/docs/insights/insights-api/get-data/query-insights-event-data-api.

Configuring OpenStack Observer jobs

Using the OpenStack Observer, you can configure jobs that dynamically load OpenStack data for analysis.

Before you begin

Important: The OpenStack Observer supports the on-premises OpenStack up to versions Ocata, Pike, Rocky, Queens, Stein, Train, Zed and Antelope.

Ensure you have the OpenStack service details to hand, such as the parameters for its APIs or RabbitMQ message bus. If you are configuring a query job, have OpenStack location and authorization details to hand. If you are configuring a rabbitmq job, you must also identify and provide access to the RabbitMQ message bus.

OpenStack installation requirements

If you previously installed OpenStack with DevStack, you must add the configuration that is specified here to the end of the local.conf file, and reinstall it. If you previously installed OpenStack with another installation method, you must add the configuration that is specified here to the nova.conf file, and then restart the Nova (compute) service.

- If you **have already** installed OpenStack using DevStack, add the following configuration to the end of the local.conf file, and **then reinstall** OpenStack.
- If you are **planning to** install OpenStack using DevStack, add the following configuration to the end of the local.conf file **before installation**.

```
[[post-config|$NOVA_CONF]]
[DEFAULT]
notification_topics = notifications,com.ibm.asm.obs.nova.notify
notification_driver=messagingv2
notify_on_state_change=vm_and_task_state
notify_on_any_change=True
```

For standard (or any other) OpenStack installations, add the following configuration under the `[DEFAULT]` section of the nova.conf file, and then you restart the Nova (compute) service.

```
notification_topics = notifications,com.ibm.asm.obs.nova.notify
notification_driver=messagingv2
notify_on_state_change=vm_and_task_state
notify_on_any_change=True
```

Note: OpenStack uses RBAC-based protection of its API by defining policy rules based on an RBAC approach. Availability of resources retrieved by the observer is also governed by this same policy. For example, a VM created in project A by users with the admin role may be available to other users with the same admin role. This can be configured or modified according to user requirements in the OpenStack's policy configuration.

OpenStack port numbers

OpenStack assigns standard port numbers to each of its service. To communicate with the OpenStack instance, you must enable the following default ports.

Tip: Check with your OpenStack administrator if custom port numbers have been configured.

Load jobs default ports:

```
- KeyStone: 5000
- Nova: 8774
- Neutron: 9696
- Glance: 9292
- Cinder: 8776
- Heat: 8004
```

Listen jobs default port:

```
- RabbitMQ: 5672
```

Troubleshooting

A **Certificate Chaining Error** can occur when launching an OpenStack Observer job.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The OpenStack Observer jobs extract OpenStack resources through REST or RabbitMQ, and the observer loads and updates the resources and their relationships within the topology service.

You configure and run the following two jobs.

Restapi Load job

A transient (one-off) job that loads all requested topology data from the OpenStack instance by REST API.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

The job loads baseline topology data through the following OpenStack's APIs:

- Keystone (identity)
- Cinder (block storage)
- Glance (image)
- Heat (orchestration)
- Neutron (network)
- Nova (compute)

Restriction: An OpenStack environment that has a list of endpoints whereby the 'heat-cfn' service comes first (before the 'heat' service) will encounter a JSON parsing error recorded in the logs due to a known issue in the openstack4j library. When this happens, the full load for the heat service will be skipped entirely. Other services will run as normal.

Rabbitmq Listen job A long-running job that reads messages on OpenStack's RabbitMQ message bus for activity from the Cinder (block storage), Heat (orchestration), Neutron (network) and Nova (compute) components continually, until it is explicitly stopped, or until the Observer is stopped.

Note: The rabbitmq job should only be run after an initial restapi job has been completed.

Restriction: Only one rabbitmq job should be listening to one queue (or sets of queues) at any one time. If you need to listen to multiple projects, then separate queues must be set up in OpenStack, with appropriate names, before separate listening jobs are submitted for each. For example, for Nova through the rmq_nova_notify attribute, for Neutron through the rmq_neutron_notify attribute.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Restapi Load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data center name	Specify the name of the data center in which the OpenStack instance is running	Required. If more than one OpenStack instance is run, and duplicate project or tenant names exist, you must disambiguate them here.
OpenStack Project/Tenant name	Specify the OpenStack project name(s) to use, for example the value of `OS_PROJECT_NAME`. Multi project names should be separated by comma.	Required
OpenStack authentication type	Specify the OpenStack connection authentication technique to use	Required. Choose either `v2_Tenant`, `v3_Unscoped`, v3_Project, v3_Domain, `or` v3_ProjectDomain`.
OpenStack identity endpoint	Specify the OpenStack Identity Endpoint to connect to inclusive of port and version, e.g. the value of `OS_AUTH_URL`.	Required
OpenStack username	Specify the OpenStack username to connect as (or to)	Required
OpenStack password	Specify the OpenStack password with which to authenticate	Required
OpenStack host certificate	Specify a certificate name to load into the truststore	Required
SSL truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `OpenStack.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
SSL truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
OpenStack user domain name	Specify the user domain name to use, for example the value of `OS_USER_DOMAIN_NAME`.	Optional
OpenStack project domain name	Specify the project domain name to use, for example the value of `OS_PROJECT_DOMAIN_NAME`.	Optional
OpenStack region name	Specify the OpenStack region to use, for example the value of `OS_REGION_NAME` The value is case-sensitive. If it is not set, the default value is *regionOne*.	Required for v3 authentication.
OpenStack perspective	Select the URL perspective the API accesses data from	Optional. Choose from `Admin`, `Public`, and `Internal`.
SSL Verification	Choose whether to use SSL verification (true or false). If false, HTTPS is used, but **without** hostname validation.	Optional
Optionally load hypervisors	Choose whether to turn the loading of hypervisors on or off. **Note:** To load hypervisors, admin privileges must be assigned to the user.	Optional. The default is `false` (off).

Parameter	Action	Details
Optionally load Host Aggregates	Choose whether to turn the loading of Host Aggregates on or off. **Note:** To load host aggregates data, admin privileges must be assigned to the user.	Optional. The default is `false` (off).
Connection timeout (ms)	Choose the timeout setting for the connection actions	Optional. The default is 5000 (5 seconds).
Read timeout (ms)	Choose the timeout setting for the read actions	Optional. The default is 5000 (5 seconds).
Proxy host	Specify the proxy host to connect via	Optional
Proxy username	Specify the proxy username.	Optional
Proxy password	Specify the proxy password	Optional
Proxy port	Specify the proxy port to connect via	Optional. Default is `8080`.
Trust all certificates	Set to `true` to connect to target environment without verification	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Rabbitmq Listen job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data center name	Specify the name of the data center in which the OpenStack instance is running	Required. If more than one OpenStack instance is run, and duplicate project or tenant names exist, you must disambiguate them here.
OpenStack Project/Tenant name	Specify the project or tenant name(s) to use, such as the value of `OS_PROJECT_NAME. Multi-project or tenant names should be separated by comma.	Optional
OpenStack username	Specify the OpenStack username to connect as (or to)	Required
RabbitMQ username	Specify the AMQP username to connect to the broker	Required
RabbitMQ hosts	Enter a (comma-separated) list of hosts in the RabbitMQ cluster	Required. Add a comma-separated host, or list of hosts, (for example, `host1:5672,host2:5672`) to connect to in the RabbitMQ cluster. The first host connected to successfully is used. **Note:** Use IP addresses instead of hostnames.
RabbitMQ password	Specify the password to use to connect to the broker	Required
RabbitMQ virtual host name	Specify the virtual host to connect to the broker	Optional
Use SSL	Choose whether to use an SSL connection	Optional. Choose `true` or `false`. For RabbitMQ, you must choose `true`.
Nova v2 Oslo message queue	Specify the Nova v2 Oslo message queue	Optional
Neutron v2 Oslo message queue	Specify the Neutron v2 Oslo message queue	Optional
Cinder v2 Oslo message queue	Specify the Cinder v2 Oslo message queue	Optional
Heat v2 Oslo message queue	Specify the Heat v2 Oslo message queue	Optional
Number of consumer instances	Specify the number of consumer instances to create for each API queue type	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional

Parameter	Action	Details
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Important: You **must** specify the following properties consistently for both the restapi and rabbitmq jobs:

- Data center name
- OpenStack project name
- OpenStack username

Configuring Rancher Observer jobs

Using the Rancher Observer, you can define a full load job that will read data from Rancher through the REST API and generate a topology.

Before you begin

Important: The Rancher Observer supports the on-premises Rancher version 2.3.2 and API version 3.0.

Ensure you have the Rancher details to hand, such as the Rancher host IP, accessKey, secretKey and clusterId, before running the observer job.

The Rancher Observer supports a transient (one-off) Load job that loads all requested topology data through Rancher REST API to build the topology, and then exit.

You define and start the following job. You must edit the parameters in the configuration file before running this job.

Full Topology Upload job By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Rancher Host IP	Enter the Rancher host IP address	Required
Rancher Access Key ID	Supply the Rancher access key	Required
Rancher Secret Access Key	Supply the Rancher secret key	Required
Rancher Cluster ID	Enter the cluster ID to be discovered	Required
Enable/Disable Secure Host Connection to Rancher's Host	Set to 'true' to secure the connection. Alternatively, set to 'false' to bypass secure host connection.	Required
Rancher certificate	If enable_secure_host_connection is set to 'true', update the global secret with encrypted certificate information. See the relevant step here.	Optional
SSL TrustStore File	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `rancher.jks`.	Required if `enable_secure_host_connection` is set to 'true'. The supported format is JKS and the file is relative to $ASM_HOME/security. You **must** also supply the SSL truststore password to decrypt the HTTPS truststore file.
SSL TrustStore Password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required if `enable_secure_host_connection` is set to 'true'
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional

Parameter	Action	Details
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring REST Observer jobs

Use the REST Observer for obtaining topology data through REST endpoints. This observer is a counterpart to the File Observer.

Before you begin

For the Swagger API documentation, see REST Observer for Agile Service Manager.

About this task

The REST Observer passes topology data using a RESTful set of interfaces, which provide REST APIs that enable the following functionality:

- Management of Listen and bulk-replace job types.
- The insert-update (HTTP POST) of resources.
- The insert-update (HTTP POST) of relationships.
- The insert-replacement (HTTP PUT) of resources.
- The deletion (HTTP DELETE) of resources.
- A REST API that supports the deletion (HTTP DELETE) of all relationships of a particular type from a specified resource.
- A REST API that supports the deletion (HTTP DELETE) of a specific relationship.

Restriction: Resources created through REST can have a provider, but not an observer.

Benefits: Using the REST Observer rather than the File Observer or Topology Service APIs includes the following benefits:

- The ability to provide data through HTTP REST Endpoints instead of files.
- The processing performed by all observers in their framework ensures that metadata about observations from observers is managed correctly.
- A simple way of deleting all edges of a given type on a resource or a specific edge instance.

To use the REST Observer, a job request must be issued (HTTP POST) to the Observer instance job management APIs before sending data to the Resource and Relationship APIs.

Listen

A long-running listen job capable of consuming topology data over a long period. A listen job is designed to support scenarios where the input data stream is unpredictable, or little or no consistency or versioning of resources exists within the data stream.

Synchronize (bulk replace)

A long-running job with the same resource replace semantics as the File Observer. Bulk-replace jobs are designed to support scenarios where a known set of resources are subject to updates or versioning, and a prior observation about resources is to be replaced with a new one. This job provides a new set of resources and relationships and synchronizes them to the topology service, causing any previous data provided by the observer to be deleted and replaced with the new data.

Note: In both cases, a provider job parameter is required to identify the origin of the data being provided to the observer job.

Once a job request has been successfully submitted, you can start to provide data to the Resource and Relationship APIs on behalf of a particular job instance.

The Resource and Relationship APIs may respond with an HTTP 503 Service Unavailable response with a Retry-After: 10 seconds in the header. This indicates that even though the request against those APIs is valid, the observer has not been able to ascertain that metadata about the job is held in IBM Cloud Pak® for AIOps yet; this may be due to, for example, any prevailing conditions in the network that support the micro-services.

Tip: If such a response is received, try the request again later.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Important: Ensure that the body is structured correctly. When posting the body, information included in the body after the closing **}** that matches an opening **{** is ignored, and no error is recorded.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Provider	Specify the name of the program or system to provide data	Required
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.

Parameter	Action	Details
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

REST Observer API access

API name: topology-rest-observer

Public API URLs: `/aiops/api/application-manager/topology-rest-observer/v1/*`

For information on how to obtain route and token information, see Accessing the console.

For information on how to obtain your token, see Accessing APIs.

Swagger API documentation

For the Swagger API documentation, see Topology REST Observer API. The documentation explains how to use the REST Observer APIs.

Creating cross-data origin relationships

Using the REST Observer functionality, you can create edges between resources from different observations or providers.

It requires both the **`uniqueID`** and the provider of the resources at each end of the link (so typically applies to previously observed resources).

In the following syntax examples, note the **`_toProvider`** and **`_fromProvider`** edge properties.

You can define an edge from a resource in the current observation to one owned by a different provider. This example needs to be POSTed to the REST observer's **`/rest/resources`** endpoint:

```
{
   "uniqueId":"host1.ibm.com",
   "entityTypes":["host"],
   "name":"host1",
   "_references":
   [
       {
          "_edgeType":"connectedTo",
          "_toProvider":"someOtherProvider",
          "_toUniqueId":"host2.ibm.com"
       }
   ]
}
```

You can define an edge from a resource between two different providers. This example needs to be POSTed to the REST observer's **`/rest/references`** endpoint:

```
{
   "_fromProvider":"providerOne",
   "_fromUniqueId":"host1-interface1",
   "_edgeType":"connectedTo",
   "_toProvider":"providerTwo",
   "_toUniqueId":"host2-interface1"
}
```

Example listen job

The following examples show how to use the REST Observer listen job to create and adjust a small topology.

Start the Listen job

Use the following example as a model.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -d '{
  "unique_id": "my job",
  "type": "listen",
  "parameters": {
    "provider": "MyListenJob"
  }
}' 'https://localhost/1.0/rest-observer/jobs/listen'
```

Verify that the listen job is running

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X GET --header 'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-
a4a8-a73a79c71255' 'https://localhost/1.0/rest-observer/jobs/my%20job'
```

Create a 'person' resource

This example creates a person resource called 'Thomas Watson'.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my job' -d '{
  "name": "Thomas Watson",
  "uniqueId": "Thomas Watson",
  "entityTypes": [
    "person"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Create an 'organization' resource

This example creates an 'organization' resource of 'IBM'.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my job' -d '{
  "name": "IBM",
  "uniqueId": "IBM",
  "entityTypes": [
    "organization"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Create a 'manages' relationship

This example creates a 'manages' releationship between Thomas Watson and IBM.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my job' -d '{
  "_fromUniqueId": "Thomas Watson",
  "_edgeType": "manages",
  "_toUniqueId": "IBM"
}' 'https://localhost/1.0/rest-observer/rest/references'
```

Create a new 'location' resource

This example creates a 'location' resource of 'Campbell, New York' for Thomas Watson, and a location relationship (an edgeType of locatedAt).

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my job' -d '{
  "name": "Campbell, New York",
  "uniqueId": "Campbell, New York",
  "entityTypes": [
    "location"
  ],
  "_references": [
    {
      "_fromUniqueId": "Thomas Watson",
      "_edgeType": "locatedAt"
    }
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Replace the location resource with one having latitude and longitude properties

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my job' -d '{
  "name": "Campbell, New York",
  "uniqueId": "Campbell, New York",
  "entityTypes": [
    "location"
  ],
  "longitude": -77.196918,
  "latitude": 42.232909
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Delete all 'locatedAt' relationships

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X DELETE --header 'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-
4006-a4a8-a73a79c71255' --header 'JobId: my job' 'https://localhost/1.0/rest-
observer/rest/resources/Thomas%20Watson/references/both/locatedAt'
```

Delete the location

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X DELETE --header 'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-
4006-a4a8-a73a79c71255' --header 'JobId: my job' 'https://localhost/1.0/rest-
observer/rest/resources/Campbell%2C%20New%20York'
```

Delete the manages relationship

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X DELETE --header 'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-
4006-a4a8-a73a79c71255' --header 'JobId: my job' 'https://localhost/1.0/rest-
```

```
observer/rest/resources/Thomas%20Watson/references/both/manages/IBM'
```

Example bulk-replace job

The following examples show how to use the REST Observer bulk replace job to create and adjust a small topology.

Note: These examples use a mixture of the provided scripts in $ASM_HOME/bin and the cURL command.

Submit a bulk-replace job request with the unique ID of 'my bulk replace'

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -d '{
  "unique_id": "my bulk replace",
  "type": "bulk replace",
  "parameters": {
    "provider": "Me"
  }
}' 'https://localhost/1.0/rest-observer/jobs/bulk_replace'
```

Verify that the bulk-replace job is running

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X GET --header 'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-
a4a8-a73a79c71255' 'https://localhost/1.0/rest-observer/jobs/my%20bulk%20replace'
```

Submit a location resource for the city of Coventry

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my bulk replace' -d '{
  "name": "Coventry",
  "uniqueId": "Coventry",
  "entityTypes": [
    "location"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Submit a location resource for the town of Rugby

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my bulk replace' -d '{
  "name": "Rugby",
  "uniqueId": "Rugby",
  "entityTypes": [
    "location"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Submit a location resource for the Town of Leamington Spa with relationships to the existing resources of Coventry and Rugby

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my bulk replace' -d '{
  "name": "Leamington Spa",
  "uniqueId": "Leamington Spa",
  "entityTypes": [
    "location"
  ],
  "_references": [
     {
        "_toUniqueId": "Coventry",
        "_edgeType": "routesVia"
     },
     {
        "_toUniqueId": "Rugby",
        "_edgeType": "routesVia"
     }
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Check your progress by rendering the topology

Having completed this set of observations, initiate a synchronize request for 'my bulk replace' job.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -d '' 'https://localhost/1.0/rest-
observer/jobs/my%20bulk%20replace/synchronize'
```

Provide a new topology for the 'my bulk replace' job

This example submits a location resource for the town of Leamington Spa with relationships to the towns of Warwick and Stratford-Upon-Avon (resource placeholders).

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my bulk replace' -d '{
  "name": "Leamington Spa",
  "uniqueId": "Leamington Spa",
  "entityTypes": [
```

```
    "location"
  ],
  "_references": [
     {
        "_toUniqueId": "Warwick",
        "_edgeType": "routesVia"
     },
     {
        "_toUniqueId": "Stratford-upon-Avon",
        "_edgeType": "routesVia"
     }
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Provide the resource data for the Town of Warwick resource placeholder, thus fully creating the resource

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my bulk replace' -d '{
  "name": "Warwick",
  "uniqueId": "Warwick",
  "entityTypes": [
    "location"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Provide the resource data for the town of Stratford-upon-Avon resource placeholder, thus fully creating the resource

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my bulk replace' -d '{
  "name": "Stratford-upon-Avon",
  "uniqueId": "Stratford-upon-Avon",
  "entityTypes": [
    "location"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Initiate a synchronize request for the 'my bulk replace' job. This signifies to the Observer that it should instruct the system to replace the previous set of observations with the new ones.

Note: The new data is available immediately as it is provided. The synchronize request simply deletes any resources previously observed that were not observed this time. In the current example, Coventry and Rugby were not observed, and therefore they are deleted.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
--header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -d '' 'https://localhost/1.0/rest-
observer/jobs/my%20bulk%20replace/synchronize'
```

Configuring SDC ONAP Observer jobs

You configure SDC (Service Design and Creation) ONAP (Open Network Automation Platform) Observer jobs to dynamically load data from the SDC REST interface.

Before you begin

Important: The SDC ONAP Observer supports the SDC release 8.0.0.

Ensure you have the SDC ONAP service details to hand, such as username, password, hostname and port.

To obtain SDC ONAP system details:

1. To get port

   ```
   kubectl describe svc --selector=app.kubernetes.io/name=sdc-be | grep NodePort
   ```

2. To get message bus port

   ```
   kubectl describe svc --selector=app.kubernetes.io/name=message-router | grep NodePort
   ```

3. To copy trust store

   ```
   kubectl cp <onap-sdc-be-*>:/var/lib/jetty/chef-solo/environments/org.onap.sdc.trust.jks ./org.onap.sdc.trust.jks
   ```

   Or

   ```
   kubectl cp <onap-sdc-be-*>:/app/jetty/chef-solo/environments/org.onap.sdc.trust.jks ./org.onap.sdc.trust.jks
   ```

4. To get trust store password

   ```
   kubectl exec -it <onap-sdc-be-*> -- sh -c "cat /var/lib/jetty/chef-solo/environments/AUTO.json | grep
   truststore_pwd"
   ```

   Or

   ```
   kubectl exec -it <onap-sdc-be-*> -- sh -c "cat /app/jetty/chef-solo/environments/AUTO.json | grep truststore_pwd"
   ```

About this task

The SDC ONAP Observer job retrieves the service design data from the SDC server through REST. The observer loads and updates the resources and their relationships within the Agile Service Manager core topology service.

SDC manages two levels of assets:

Resource

A fundamental capability, implemented either entirely in software, or as software that interacts with a hardware device. Each Resource is a combination of one or more Virtual Function Components (VFCs), along with all the information necessary to instantiate, update, delete, and manage the Resource. A Resource also includes license-related information.

There are three kinds of resource:

- Infrastructure: Cloud resources, such as Compute, Storage, etc
- Network: Network connectivity functions & elements, for example a Virtual Network Function (VNF)
- Application: Features and capabilities of a software application, for example a load-balancing function=

Service

A well-formed object comprising one or more resources.

Service designers create services from resources, and include all of the information about the service needed to instantiate, update, delete, and manage the service.

You define and start the following jobs:

Load job

Run this job whenever you need SDC ONAP topology data refreshed.

By default, these jobs are one-off, transient jobs that perform a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Listen job

A long-running job that consumes relevant messages from the SDC DMaaP broker continually until it is explicitly stopped, or until the observer is stopped.

You should run this job after a load job has been completed.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
SDC Distribution Engine host	Specify the host name of the SDC Distribution Engine	Required
SDC Distribution Engine port	Specify the port number of the SDC Distribution Engine	Required
SDC username	Specify the user name to use when connecting to the SDC	Required
SDC password	Specify the password to use when connecting to the SDC	Required. Plain text password can be entered.
Target system Key Store file name	Specify the key store name that contains either the CA certificate or the SDC's public key	Required. For information on encrypting the contents of the key store file, see step 2 of the Configuring observer jobs security topic.
Target system Key Store password	Specify the password to access the key store file	Required. Plain text password can be entered.
Optional provider name	Auto-generated if not specified	Optional. **Warning:** Do not change to avoid resource duplication.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.

Parameter	Action	Details
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Listen job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
SDC Distribution Engine host	Specify the host name of the SDC Distribution Engine	Required
SDC Distribution Engine port	Specify the port number of the SDC Distribution Engine	Required
SDC username	Specify the user name to use when connecting to the SDC	Required
SDC password	Specify the password to use when connecting to the SDC	Required. Plain text password can be entered.
SDC environment name	Specify the environment name configured on the SDC, for example `TEST`, `PROD`, `AUTO`, etc	Required
DMaaP message-router address	Specify the DMaaP message bus addresses on which to listen for distribution events	Required
DMaaP message-router port	Specify the DMaaP message bus port on which to listen for distribution events	Required
Target system Key Store file name	Specify the key store name that contains either the CA certificate or the SDC's public key	Required. For information on encrypting the contents of the service account key file, see step 2 of the Configuring observer jobs security topic.
Target system Key Store password	Specify the password to access the key store file	Required. Plain text password can be entered.
DMaaP polling interval (in seconds)	Specify how often (in seconds) to poll the DMaaP cluster for new model events	Optional. The default is 20s.
DMaaP polling timeout (in seconds)	Specify the timeout value (in seconds) when polling DMaaP for new model events	Optional. The default is 20s.
Optional provider name	Auto-generated if not specified	Optional. **Warning:** Do not change to avoid resource duplication.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring ServiceNow Observer jobs

Using the ServiceNow Observer job, you can retrieve the configuration management database (CMDB) data from ServiceNow through REST API, using basic authentication credentials.

Important: The ServiceNow Observer supports the cloud and SaaS ServiceNow versions `New York`, `Quebec`, `Paris`, `Tokyo`.

Before you begin

- **Important**: If you have any restrictive EgressFirewall in place, ensure that they are updated to allow for this inbound connection. For more information, see Create an EgressFirewall.

- **Important**: Ensure that your ServiceNow user account has the required roles to configure this observer and to access resources from ServiceNow. You need the following roles:
  - `rest_api_explorer`
  - `web_service_admin`
- Ensure that you have the ServiceNow service details to hand to complete this observer configuration, such as username, password, and URL.

To add a role to your user account, complete the following steps:

1. Log in to your **ServiceNow** instance using an admin account.
2. Go to **User Administration** and select **Users** from the menu.
3. Select to create or edit a user:
   - To create a user, click **New**.
   - To edit an existing user, search for, and then select, the user.
4. From the user's information tab, select the **Roles** tab, then click **Edit**.
5. Assign the required roles to the user.
6. Click **Save**, then **Update**.

For more detailed descriptions of available roles, go to **User Administration** and select the **Roles** page. You can also assign other roles with similar privileges to the non-admin account. To find other roles with similar privileges, navigate to **System Security** and select **Access Control (ACL)**.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

ServiceNow jobs retrieve configuration management database (CMDB) data from ServiceNow through REST API. The Observer loads and updates the resources and their relationships within the topology service.

By default, the observer discovers the following relationship tables:

- /api/now/table/cmdb_ci_service_discovered (services data)
- /api/now/table/cmdb_rel_ci (connectivity information for inventory data)
- /api/now/table/cmdb_ci_service_calculated (calculated service data)
- /api/now/table/cmdb_ci_query_based_service (query-based service data)

The observer supports all system-created or user-created child tables of cmdb_ci, as well as the following resource:

- /api/now/table/sys_user (user data)

ServiceNow job: By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered. You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them. Run this job whenever you want to refresh the ServiceNow topology data.

Mapping of ServiceNow object types to topology entity types

Table. ServiceNow object and entity types

ServiceNow object types	Entity types
cmdb_ci, cmdb_ci_service_discovered	based on sys_class_name attribute
sys_user	person

Tip

Important: Specify `cmdb_ci_{name}` tables to discover using the `ServiceNow tables to be discovered` parameter.

Follow these steps to run the ServiceNow glide script that lists availabe `cmdb_ci` tables.

Note: You must have admin access to run the script.

1. Login to the ServiceNow instance as the admin user.
2. Go to **System Definition > Scripts - Background**.
3. Paste the following script into the text box:

```
var g = new GlideAggregate('cmdb_ci');
g.addAggregate('COUNT', 'sys_class_name')
g.orderByAggregate('COUNT', 'sys_class_name');
g.query();
while (g.next()) {
gs.info(g.sys_class_name + ': ' + g.getAggregate('COUNT', 'sys_class_name'));
}
```

4. Run the script.
5. ServiceNow will list all `cmdb_ci` tables in the environment. Sample output:

```
*** Script: cmdb_ci_spkg: 1767
*** Script: cmdb_ci_computer: 814
*** Script: cmdb_ci_service: 41
*** Script: cmdb_ci_printer: 28
*** Script: cmdb_ci_server: 19
*** Script: cmdb_ci_win_cluster_node: 14
*** Script: cmdb_ci_database: 13
*** Script: cmdb_ci_web_server: 9
*** Script: cmdb_ci_win_server: 8
*** Script: cmdb_ci_unix_server: 7
*** Script: cmdb_ci_ups: 6
*** Script: cmdb_ci_email_server: 5
*** Script: cmdb_ci_netgear: 5
*** Script: cmdb_ci_rack: 5
*** Script: cmdb_ci_zone: 5
*** Script: cmdb_ci_linux_server: 4
*** Script: cmdb_ci_win_cluster: 4
*** Script: cmdb_ci_cluster_node: 3
*** Script: cmdb_ci_ip_router: 3
*** Script: cmdb_ci_msd: 3
*** Script: cmdb_ci_network_adapter: 3
*** Script: cmdb_ci_aix_server: 2
*** Script: cmdb_ci_computer_room: 2
*** Script: cmdb_ci_disk: 2
*** Script: cmdb_ci_peripheral: 2
*** Script: cmdb_ci: 1
*** Script: cmdb_ci_appl: 1
*** Script: cmdb_ci_app_server_java: 1
*** Script: cmdb_ci_cluster: 1
*** Script: cmdb_ci_datacenter: 1
*** Script: cmdb_ci_db_mysql_catalog: 1
*** Script: cmdb_ci_db_ora_catalog: 1
*** Script: cmdb_ci_ip_switch: 1
*** Script: cmdb_ci_service_group: 1
*** Script: cmdb_ci_storage_switch: 1
```

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Remember: To improve performance and to focus your topology, **do not** include all ServiceNow tables. Instead (**before** running an observer job) identify the ServiceNow tables that are most useful to your requirements, and exclude the ones you do not require. For example, you may want to exclude the vast number of records of SIM cards from being displayed in the topology, and include only servers, services or applications.

***Tip:** The **Device group path** parameter cannot be configured.

Table. Parameter action details

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
ServiceNow instance	Specify the ServiceNow instance, which should be `NMS`	Required
ServiceNow username	Specify the ServiceNow username	Required
ServiceNow password	Specify the ServiceNow password	Required
ServiceNow tables to be discovered	The supported tables are `sys_user`, and all system-created or user-created child tables of `cmdb_ci`. Using a comma as delimiter for multiple entries, specify either `sys_user` or tables that are either system-created or user-created children of the `cmdb_ci` table, such as `cmdb_ci_database` (the system table), `u_ibm_database` (the global scope custom table), or `sn_app_ws_u_ibm_database` (the application scope custom table). **Note** that the following relationship tables are added by default: `cmdb_rel_ci`, `cmdb_ci_service_discovered`, `cmdb_ci_service_calculated`, and `cmdb_ci_query_based_service`.	Required
Number of records to fetch per API call from a table	Enter the number of records to fetch per API call from a table. The default value is 10000. The value you enter should be in the range of 100 to 10000.	Optional
Entity mapping for sysClassNames	This parameter will override the default observer mapping for the sys_class_name entity type with a user-defined entity type. Enter the class name and the new entity type name, where the values of the class and the new entity type name are separated by a colon. Example: `cmdb_ci_linux_server:node`.	Optional
Options to exclude entity edges	Select an exclude option to exclude resource edges	Optional
List of entities to exclude edge modelling	Enter the entities to exclude resource edges. The listed entities will be not have any edge connectivity to any entities	Optional

Parameter	Action	Details
Use `include` or `exclude` filter	Specify the type of filter to use when retrieving resources. The options are `include` (default), or `exclude`. Specify whether to include or exclude an API filter in every ServiceNow API call.	Optional. If you select `include`, you specify the filter next.
Filter resources based on JSON attributes	Specify a filter using the JSON `key:value` attribute. The format used is `key=value`, where value can be more than one: `key=value1,value2,value3` For more than one key, use the vertical bar symbol (|) as delimiter: `key=value1|key2=value1,value2,value3`	Optional. Required only if the `include` option has been set.
Number of connection retry times	Set the number of connection retries.	Optional. The default is `2`.
Delay in (milliseconds) before a connection retry	Set the delay in milliseconds before a connection retry.	Optional. The default is `5000ms`.
Maximum number of tables	Enter the number of batch calls into which you want to split each API query. Defining a number greater than one (up to a maximum of 10) can avoid the query from being too long, thereby avoiding `query too long` ServiceNow API errors.	Optional. The default and maximum is 10.
Connection timeout (milliseconds)	Set the connection timeout in milliseconds.	Optional. The default is `5000ms`.
Read timeout (milliseconds)	Set the read timeout in milliseconds.	Optional. The default is `5000ms`.
Proxy Host	Specify the proxy host to connect to.	Optional
Proxy Port	Specify the proxy port; defaults to `8080`.	Optional
Proxy Username	Specify the proxy user name to connect as. Set for basic auth proxy.	Optional
Trust all certificates by bypassing certificate verification	Set to true to allow connection to target environment without verification.	Optional. The default is 'false'.
Proxy Password	Specify the password for the proxy user (if `Proxy Username` has been specified).	Optional
Proxy Secure	Specify whether to connect to the proxy in secure mode. Set to True for HTTPS proxy.	Optional. The default is `false`.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring SevOne Observer jobs

Using the SevOne Observer functionality, you can load monitored servers and their relationships, and then visualize this data as a topology view.

Before you begin

Important: The SevOne Observer supports SevOne version 6.2.0 and API version 2.20.0.

Ensure you have the SevOne server details to hand, such as the username and password.

About this task

A SevOne Observer job extracts SevOne managed servers and their relationships via REST API calls.

You define and start the following job.

SevOne Load job

Run this job whenever you need SevOne topology data refreshed.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

SevOne resources and relationships loaded by the SevOne Observer job

Vertex	Description
Server	SevOne managed server
Ipaddress	IP address associated with the server managed by SevOne

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

SevOne Observer job parameters for Load job

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Hostname	Enter the SevOne hostname	Required
Data Tenant	Specify the data tenant used for the device group or groups.	Required
Username	Specify the SevOne username	Required
Password	Specify the SevOne password	Required. Use plain text.
Device group path	Specify the device group path to be discovered. To discover more than one group, enter a comma-separated list.	Required
Number of SevOne hops	Choose the number of hops to be discovered	Required. Default is 2, maximum hops is 3.
HTTPS connection to SevOne server	Select true to enable HTTPS connection. If `true`, you must also provide the target system certificate.	Required
Topology relationship type	Select one of the following topology relationship types: `host` (creates a direct host-to-host topology relationship), or `network interface` (creates a relationship between the hosts' network interfaces)	Required
Target system certificate	Specify a certificate by name to load into the trustStore.	Required
SSL certificate hostname verification	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring TADDM Observer jobs

IBM Tivoli Application Dependency Discovery Manager (TADDM) Observer jobs retrieve network topology data, including discovered applications, their components, configurations and dependencies, from the TADDM database server (running either an IBM Db2 or an Oracle database), and use this data to create topology views.

Before you begin

Important: The TADDM Observer supports the on-premise TADDM version.

To connect to a TADDM Oracle database, you must place the Oracle JDBC Driver into the $ASM_HOME/lib folder, and then restart the observer for it to take effect. You can download the driver from the Oracle website, or copy it from the Oracle server (**not** the Oracle client) from the following location: ../app/oracle/product/Oracle_version/dbhome/jdbc/lib/ojdbc6.jar

Ensure you have the TADDM Rest API login access details in hand, such as the TADDM API URL, username and password.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The TADDM Observer is built on the Observer framework:

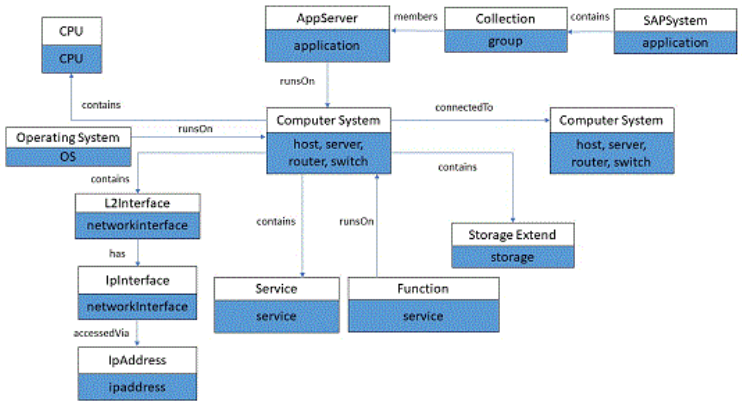


Figure. Observer framework

- A computer system can be a host, server, router, or switch
- A computer system contains CPU, L2Interface, and storage
- Operating system, application server and service run on a computer system
- A computer system can connect to another computer system
- A SAPSystem contains collection
- An application server can be a member of a collection

Mapping TADDM model objects to topology entity types

TADDM model object	Entity types
AppServer	`application`
ComputerSystem	`host`, `server`, `router`, `switch`
CPU	`cpu`
L2Interface	`networkinterface`
IpInterface	`networkinterface`
IpAddress	`ipaddress`
OperatingSystem	`os`
Service	`service`
StorageExtent	`storage`
Function	`service`
SAPSystem	`application`
Collection	`group`

TADDM Observer jobs retrieve topology data using the TADDM REST API, and the observer loads and updates the resources and their relationships within the topology service.

You define and start the following jobs.

Load job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
TADDM password	Specify the password for the TADDM user	Required

Parameter	Action	Details
TADDM API URL	Specify the TADDM endpoint to connect to	Required.
TADDM username	Specify the TADDM username	Required
TADDM objects to observe	Select one or more options from the drop-down list	Optional. If none are selected, all supported model objects are retrieved.
TADDM certificate	Specify the certificate file name.	Optional
Certificate validation	Choose whether SSL validation is `on` or `off`. Turning SSL validation off will use HTTPS without host verification.	Optional
Connection and read timeout (ms)	Enter the time at which the connection and read actions time out	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Trust all certificates verification	Set to true to allow connection to target environment without verification.	Optional. The default is 'false'.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Viptela Observer jobs

You use the Viptela Observer to load managed Viptela SD-WAN control plane, vEdge and their associated network resources, and then visualize this data as a topology view.

Before you begin

Important: The Viptela Observer supports version 18.4.302 of Cisco SD-WAN.

Ensure you have the server details to hand, such as the username and password.

About this task

The Viptela Observer targets the Cisco SD-WAN (or 'Cisco Viptela') system. A Viptela Observer job extracts server information and its associated network resources from Viptela through REST API. The observer loads and updates the resources and their relationships and then displays them as a topologgy view.

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required

Parameter	Action	Details
Data center name	Specify the data center the vEdge runs in.	Required. If you want to discover multiple vEdge locations or sets, then provide a different data center, for example `EURDC01`,`USDC01`.
Cisco Viptela vManage host IP	Enter the Cisco SD-Wan Viptela vManage host IP.	Required
Cisco Viptela API port.	Specify the Cisco SD-Wan Viptela vManage API port.	Required. The default value is 8443.
Username to login to Cisco Viptela service	Specify the Cisco SD-Wan Viptela vManage username.	Required
Password for Cisco Viptela user	Specify the Cisco SD-Wan Viptela vManage password.	Required. A plain text password can be entered.
Include the control plane data	Select true to load the control plane data.	Optional
vEdge grouping preference for defined topology	Specify the vEdge grouping preference for your topology.	Optional. Supported grouping: `device-group (default)`, `site-id Data Tenant`
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring VMware NSX Observer jobs

You configure VMware NSX Observer jobs to dynamically load data from the VMware NSX REST interface. You can use the VMware NSX Observer when you have a VMware NSX appliance in your environment.

Before you begin

Important: The VMware NSX Observer supports the on-premises VMware NSX version 6.3.

Ensure you have the VMware NSX REST API details to hand, such as the VMware NSX URL, username, password, and SSL truststore.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The VMware NSX Observer job extracts VMware NSX resource information through REST, and the observer loads and updates the resources and their relationships within the topology service.

You define and start the following job.

VMware NSX Observer job (full topology load) This job loads a baseline of topology data from an environment, which contains a VMware NSX appliance.

Run this job whenever you need VMware NSX topology data refreshed.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

The VMware NSX Observer loads the following resources and their relationship into the IBM Cloud Pak® for AIOps core topology service:

- NSX Appliance
- vCenter Appliance
- NSX Controller
- Edge Router - Logical (Distributed) Router, Edge Service Gateway
- Virtual machines

- Host
- VNIC

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data center name	Specify the data center in which the VMware NSX instance runs	Required.
Network Virtualization and Security Platform API URL	Specify the API URL of the VMware NSX endpoint	Required. Usually in the following format: https://[hostname or IP address]/api
Network Virtualization and Security Platform username	Specify the username to connect as, or listen to	Required
Network Virtualization and Security Platform password	Enter the password to authenticate with	Required
Network Virtualization and Security Platform tenant name	Specify the tenant	Optional.
Network Virtualization and Security Platform certificate	Specify a certificate by name to load into the truststore	Required.
SSL Validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Required
SSL truststore file	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `VMWareNSX.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
SSL truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
Connection and read timeout (ms)	Enter the time at which the connection and read actions time out	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring VMware vCenter Observer jobs

You configure VMware vCenter Observer jobs to dynamically load data from the VMware vCenter REST interface.

Before you begin

Important: The VMware vCenter Observer supports the on-premise VMware vCenter versions 6.5, 6.7 and 7.0.3.

Ensure you have the VMware vCenter service details to hand, such as username, password, SSL truststore and URL.

Important: If you have any restrictive EgressNetworkPolicies in place, ensure that they are updated to allow for this inbound connection. For more information, see Postinstallation tasks.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The VMware vCenter Observer job extracts VMware vCenter resource information through REST, and the observer loads and updates the resources and their relationships within the topology service.

You define and start the following job.

VMware vCenter Observer job (full topology load)

Run this job whenever you need VMware vCenter topology data refreshed.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

The VMware vCenter Observer loads the following resources and their relationship into the topology service:

- ESXi / ESX Hosts
- Virtual machines
- VNICs
- Storage

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data center name	Specify the data center in which the VMware vCenter instance runs	Required. If more than one, list them (comma-separated).
vCenter API URL	Specify the API URL of the VMware vCenter endpoint (including port and version)	Required. Usually in the following format: https://[hostname or IP address]/rest
vCenter username	Specify the username to connect as, or listen to	Required
vCenter password	Enter the password to authenticate with	Required
vCenter certificate	Specify a certificate by name to load into the truststore	Required. For more information, see the Configuring observer jobs security topic.
Certificate validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
HTTPS truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `VMVCenter.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
Connection and read timeout	Enter the time at which the connection and read actions time out	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
vCenter host's name regex to discover	Specify an exact match or a regular expression match for a host's name in order to discover all its virtual machines	Optional

Parameter	Action	Details
Connection retry times	Set the connection retry times	Optional. The default value is 5.
Retry after delay (milliseconds)	Set the time delay before trying to reconnect (in milliseconds)	Optional. The default value is 1000.
Connection timeout	Specify the connection timeout in ms (default is 5000)	Optional
Read timeout	Specify the read timeout in ms (default is 5000)	Optional
Proxy Host	Specify the proxy host to connect to	Optional
Proxy Port	Specify the proxy port, defaults to 8080	Optional
Proxy Username	Specify the proxy user name to connect to. Set for basic auth proxy.	Optional
Trust all certificates	Select `true` to connect to target environment without verification, or 'false' to validate the certificate before connecting.	Optional. Default is `true`.
Proxy Password	Specify the password for the proxy user (if `Proxy Username` has been specified)	Optional. Use plain text.
Proxy Secure	Specify whether to connect to the proxy in secure mode. Set to True for HTTPS proxy.	Optional. The default is 'false'.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring Zabbix Observer jobs

Using the Zabbix Observer functionality, you can load monitored servers and their associated network resources, and then visualize this data as a topology view.

Before you begin

The Zabbix Observer supports Zabbix 4.0.3, 6.0.3 and 6.4.11.

Ensure you have the Zabbix server details to hand, such as the username, password and SSL truststore.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

A Zabbix Observer job extracts server information and its associated network resources from Zabbix through REST RPC, and the observer loads and updates the resources and their relationships within the topology service.

You define and start the following job.

Full Topology Upload job By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
The datacenter of zabbix server	Specify the data center in which the Zabbix instance runs	Required. If more than one, list them (comma-separated).
Zabbix server	Enter the Zabbix virtual hostname	Required
Zabbix username	Specify the Zabbix username	Required

Parameter	Action	Details
Zabbix password	Specify the Zabbix user password	Required
Zabbix SSL certificate	Specify a certificate file name	Required
HTTPS truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `Zabbix.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
TrustStore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
Zabbix connection timeout (milliseconds)	Timeout, in ms, when querying the Zabbix REST API	Optional. Default is 5000 (5s).
Resources to exclude	List resources by name/regex (seperated by "	"), to be excluded from the topology. By default, virtual network interfaces (veth) are not observed.
SSL Validation	Choose whether SSL validation is on or off. Turning off SSL validation will use HTTPS without host verification.	Optional
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

Configuring observer job security

Some observers require authentication credentials such as certificates, keystores, or both. This topic describes such configuration tasks.

Note: Information on specific observers is located in the subtopics under Defining observer jobs.

To configure the security, complete the following steps:

- Obtain an authentication certificate.
- Store that certificate as a secret.

Important: Only update `topology-custom-secrets` with the Red Hat OpenShift Container Platform command-line interface (CLI). If you update custom secrets in the console, the crypto key is corrupted and the command to encrypt passwords does not work.

Truststore tip: The supported file format for truststore files is JKS and the file is relative to $ASM_HOME/security. You can use the observer name as file name (`<observer>.jks`), for example `VMVCenter.jks`, and use any password that conforms to your internal security requirements.

Managing authentication certificates and storing them as secrets (Red Hat OpenShift Container Platform)

1. Obtain the authentication certificate by using OpenSSL.

   ```
   echo -n | openssl s_client -connect {ipAddress}:{port} | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' | base64 -w 0 > target_system.crt
   ```

   Where `target_system.crt` contains the encoded certificate, and `{ipAddress}` is the IP address of any of the following target systems:

   - Ansible
   - Big Cloud fabric
   - BigFix Inventory
   - Ciena Blue Planet
   - Cisco ACI
   - Datadog
   - Dynatrace
   - Juniper CSO
   - HPNFVD

- GitLab
- Kubernetes master node
- OpenStack
- Rancher
- SDC ONAP
- VMware NSX
- VMware vCenter
- Zabbix

Example `target_system.crt`:

```
[root@localhost ~]# cat target_system.crt
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN3RENDQWFnQ0NRRGRuMENqU3BXZXhUQU5CZ2txaGtpRzl3MEJBUVVGQURBaU1RMHdDd11lEVlFR
RERBUkIKVUVsRE1SRXdEd11lKS29aSWh2Y05BUWtCRmdKVlV6QWVGdzB4TmpBeE1qRXdOekV5TWpWYUZ3MHlOakF4TVRndwpOekV5TWpWYU1DSXhEVEFM
QmdOVkJBTU1CRUZRU1VNeEVUQVBCZ2txaGtpRzl3MEJDUUVXQWxWVE1JSUJJakFOCkJna3Foa2lHOXcwQkFRUZBQU9DQVE4QU1JSUJDZ0tDQVFFQW1O
b0dxdFI0RlFPWkdoUWFtand1YmxRYjRobU0KTzJwOGtjbGUwL2NuUno3cSttWGYzWlRQYTZsWEk2MG9BbmVPSGowZEVaMkhwRWFFblBUbWJmWUF6Y0ZQ
djNVWApMWjM3VWVoMDZXTjMxS29tSSs2czJtSk1IWWM0MW44M1RiUU5uWUNjYjZjdlZLcWV5NVhhaFBtdkZDbDBtM3Y3Cisxa1lFMFRNVlBnTk56R0ZS
UXU1RVlGc3FZWHZGbFZhZ0lUalF6cks3YnE0RkJiMW1kVjFsYnVOMWhISzd2SFEKS3ZUNHBGbGx1NTRHU0JhZ2RSbUdad0dta0tNZHRGUkEvc3pBWEMr
ejQ0cHN3T05ydTJnbDR3bG5MZTVvM2NWZwpFQUx1THM4UDgrOUx0eFN3YWJvb0VMcHRjb3pKdEpUb2E4QS9zZXRaSi81RUJQNmhjNk1yUWxHQktRSURB
UUFCCk1BMEdDU3FHU01iM0RRRUJCUVVBQTRJQkFRQkJuZzlJK2pBdjhNUjBYemM1SUUxdTBkK0JweW9OZGVRbkl4T2sKZWFsNzNUbmkzWmh4QUQzdlQz
enZNSElSUEc0d3lxMWJqQ05LY3BZOGVCbVJuVzhOSnlNdG9vcU9hNlJMWGNPTAoyeVZublVna092THRPVjM5eFNFQ1BOMzV4YXJJdGYydE9NZWJRWlc1
ZC9Hc1lPZUFLTlNrTlQwRmtreDE0UzJFC1pBVi9IUUVHaVpUR0tQNkx1czYzLzJiTEJVNHdGUjg3bjNkdFJFVUp5eGQ4ZDJDTFA4MkE2UTNOT2lOZEdk
am0KSnFQZXNEaWxXWE5Gd09xUk1XOWFGWTVUSUtOL25PQzhqczI0cVFmZTJZc1lnZ242N0crLytBQy9kV21JSVQ2dgpBWTVMejhwOWQwSzZUaGxLeVpN
ZkdYVkNnMFlvTms1ajQ4ckJlZ2J5cFhTM1J2SnIKLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=

[root@locahost ~]#
```

2. **For the SDC ONAP Observer**, encrypt the contents of the key store file using base64.

   For example:

   ```
   cat {onap_sdc_file_name}.jks | base64 -w 0 > org.onap.sdc.trust.txt
   ```

   Where `org.onap.sdc.trust.txt` contains the encoded key store file, and `{onap_sdc_file_name}.jks` is the key store file downloaded from the target ONAP SDC system.

   Next, store the encoded file as a secret.

3. **For the GoogleCloud Observer**, encrypt the contents of the service account key file by using base64.

   For example:

   ```
   cat {project_id_file_name}.json | base64 -w 0 > googlecloud.json
   ```

   Where `googlecloud.json` contains the encoded service account key file, and `{project_id_file_name}.json` is the service account key file you downloaded from **Credentials** under **API & Services** in the Google Cloud Platform dashboard.

   Next, store the encoded service account key file as a secret.

4. Store the encoded certificate that was generated in the previous step as a secret.

   Each installed IBM Cloud Pak for AIOps release has a single special secrets file. Data added to that is made available to the appropriate observer containers.

   - Find the correct secrets file by using the following command:

     ```
     oc get secret <releasename>-custom-secrets
     ```

     Where the value of `<releasename>` and the secret can be obtained by using the following example:

     ```
     oc get asm
     ```

     Sample output:

     ```
     NAME            AGE
     aiops-topology  12d
     ```

     Find the secret for `aiops-topology`:

     ```
     oc get secret aiops-topology-custom-secrets
     ```

     Sample output:

     ```
     NAME                           TYPE     DATA   AGE
     aiops-topology-custom-secrets   Opaque   4      12d
     ```

   - Edit the appropriate file for your release.

     ```
     oc edit secret <releasename>-custom-secrets
     ```

   - Add a name and value pair to the data section.

     The value is the certificate generated earlier. The name is the value that you entered as the certificate file name to run the observer job.

     Example: The following sample shows the expected content in the secret file after adding `vcenter.crt`. The data section is between the `apiVersion` and `kind` sections.

     ```
     data:
     {name}:{value}
     ```

Note: This VMware vCenter Observer example registers the `vcenter.crt` SSL certificate in Red Hat OpenShift Container Platform Secret. `vcenter.crt` is the job parameter value for the VMware vCenter Observer. Define a new `{name}` parameter in the same file for other observers that require SSL certificates. You provide the certificate in Red Hat OpenShift Container Platform Secret and the settings you provide for truststore/truststore password is used to generate a new truststore automatically.

```
apiVersion: v1
data:
vcenter.crt:LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN3RENDQWFnQ0NRRGRuMENqU3BXZXhUQU5CZ2txaGtpRzl3MEJBUVVGQUR
BaU1RMHdDd1lEVlFRERBUkIKVUVsRE1SRXdEd1lKS29aSWh2Y05BUWtCRmdKVlV6QWVGdzB4TmpBeE1qRXdOekV5TWpWYUZ3MHlOakF4TVRndw
pOekV5TWpWYU1DSXhEVEFMQmdOVkJBTU1CRUZRU1VNeEVUQVBCZ2txaGtpRzl3MEJDUUVXQWxWVE1JSUJJakFOCkJna3Foa2lHOXcwQkFRUZBQ
U9DQVE4QU1JSUJDZ0tDQVFFQW1Ob0dxdFI0RlFPWkdoUWFtand1YmxRYjRobU0KTzJwOGtjbGUwL2NuUno3cSttWGYzWlRQYTZsWEk2MG9BbmVP
SGowZEVaMkhwRWFFblBUbWJmWUF6Y0ZQdjNVWApMWjM3VWVoMDZXTjMxS29tSSs2czJtSk1IWWM0MW44M1RiUU5uWUNjYjZjdlZLcWV5NVhhaFB
tdkZDbDBtM3Y3Cisxa1lFMFRNVlBnTk56R0ZSUXU1RVlGc3FZWHZGbFZhZ0lUalF6cks3YnE0RkJiMW1kVjFsYnVOMWhISzd2SFEKS3ZUNHBGbG
x1NTRHU0JhZ2RSbUdad0dta0tNZHRGUkEvc3pBWEMrejQ0cHN3T05ydTJnbDR3bG5MZTVvM2NWZwpFQUx1THM4UDgrOUx0eFN3YWJvb0VMcHRjb
3pKdEpUb2E4QS9zZXRaSi81RUJQNmhjNk1yUWxHQktRSURBUUFCCk1BMEdDU3FHU01iM0RRRUJCUVVBQTRJQkFRQkJuZzlJK2pBdjhNUjBYemM1
SUUxdTBkK0JweW9OZGVRbkl4T2sKZWFsNzNUbmkzWmh4QUQzdlQzenZNSE1SUEc0d31xMWJqQ05LY3BZOGVCbVJuVzhOSnlNdG9vcU9hNlJMWGN
PTAoyeVZublVna092THRPVjM5eFNFQ1BOMzV4YXJJdGYydE9NZWJRWlc1ZC9Hc1lPZUFLTlNrTlQwRmtreDE0UzJFClpBVi9IUUVHaVpUR0tQNk
x1czYzLzJiTEJVNHdGUjg3bjNkdFJFVUp5eGQ4ZDJDTFA4MkE2UTNOT2lOZEdkam0KSnFQZXNEaWxXWE5Gd09xUk1XOWFGWTVUSUtOL25PQzhqc
zI0cVFmZTJZcllnZ242N0crLytBQy9kV21JSVQ2dgpBWTVMejhwOWQwSzZUaGxLeVpNZkdYVkNnMFlvTms1ajQ4ckJlZ2J5cFhTM1J2SnIKLS0t
LS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
kind:Secret
```

If the edit is successful, the following message will be displayed:

```
secret "<release-name>-custom-secrets" edited
```

5. In the Red Hat OpenShift Container Platform web console, you can view the configured secret under the **Menu** > **Workload** > **Secrets** option, where the **Name** is `'<releasename>-custom-secrets'`. Within `'<releasename>-custom-secrets'`, all data configured earlier displays.

Event Manager integrations

Probes connect to an event source, detect and acquire event data, and forward the data as events. They use the logic specified in a rules file to manipulate the event elements before converting them into fields of an event in the ObjectServer alerts.status table.

If a built-in integration does not exist, you can use a probe connection from the Netcool Probe library. The Netcool Probe library contains more than 140 Probes to retrieve or receive events from many different sources. For example, any trap event sources can be consumed by an SNMP Probe. By integrating with event manager probes, you can acquire data from different data sources, including devices, databases, and log files. Some probes are uniquely designed to acquire event data from a specific source, while other more generic probes (for example, the Message Bus Probe) can be configured to integrate with a variety of sources.

Important: You can connect IBM Netcool Operations Insight probes directly to IBM Cloud Pak for AIOps without needing to install the Netcool OMNIbus ObjectServer. If you do not have an existing version of Netcool or IBM Netcool Operations Insight deployed, install the probe on either a Linux VM or Red Hat OpenShift (if supported), and follow the instructions in Installing the IBM Netcool Event Integrations Operator and common probe configuration steps to get the information of the AIOps ObjectServer and point the probe to the AIOps ObjectServer.

Note: Netcool probes are still supported in an IBM Cloud Pak for AIOps deployment. For more information, see Probes and probe integrations in the IBM Tivoli Netcool/OMNIbus documentation.

- IBM® Netcool® Operations Insight® (on-premises) probe integration
- IBM SevOne NPM probe integration
- IBM Turbonomic probe integration
- Metric Manager probe integration
- Probe Common Tasks

Resource requirements for running probes

This section describes the resources that each probe require per pod.

The operator controller pod resource requirements are shown below:

- CPU:
  - Requested 100m (100 millicpu)
  - Limit 200m (200 millicpu)
- Memory:
  - Requested 64Mi (~ 67 MB)
  - Limit 256Mi (~ 267 MB)

Typically each probe pod requires at least the following level of resources per pod:

- CPU Requested: 100m (100 millicpu)
- Memory Requested: 128Mi (~ 134 MB)

You can increase or decrease the resource request and limit by modifying the resources settings (resources.limits and resources.requests) in the customer resource YAML.

Cluster configuration

The minimum cluster configuration required is 1 master node and 3 worker nodes. The recommended cluster configuration is 3 master nodes and 5 worker nodes. This operator requires worker nodes that support amd64 architecture.

Connecting with on-premises probes

You can connect IBM® Netcool® Operations Insight® probes directly to IBM Cloud Pak for AIOps using an ObjectServer pair running within the Cloud Pak. By default, the ObjectServers are not started up. If you plan to connect IBM® Netcool® Operations Insight® probes, you must update the installation resource. This will change the number of replicas of the **aiops-ir-core-ncoprimary** and **aiops-ir-core-ncobackup** stateful sets and the **aiops-ir-core-ncodl-if** deployment to one.

The update that you need to make is to the **Installation** custom resource. After installing IBM Cloud Pak for AIOps, use the following steps:

1. Find your installation name and set an environment variable for it.
    1. Find the name of your IBM Cloud Pak for AIOps installation.

        ```
        oc get installation -n <project>
        ```

        Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.

        Example output, where **ibm-cp-aiops** is the name of the IBM Cloud Pak for AIOps installation:

        ```
        NAME          PHASE    LICENSE    STORAGECLASS  STORAGECLASSLARGEBLOCK AGE
        ibm-cp-aiops  Running  Accepted   rook-cephfs   rook-ceph-rbd          34h
        ```

    2. Export environment variables.

        ```
        export INSTALLATION_NAME=<name>
        export PROJECT_CP4AIOPS=<project>
        ```

        Where:

        - **<name>** is the name returned by the previous command, for example **ibm-cp-aiops**.
        - **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.
2. Run the following command to edit the installation.

    ```
    oc edit installation ${INSTALLATION_NAME} -n ${PROJECT_CP4AIOPS}
    ```

3. Set the values as follows to scale up **nocprimary** and **ncobackup**:

    ```
    spec:
      pakModules:
        - enabled: true
          name: aiManager
        - enabled: true
          name: aiopsFoundation
        - enabled: true
          name: applicationManager
          config:
           - name: ir-core-operator
             spec:
               issueresolutioncore:
                 customSizing:
                   deployments:
                   - name: ncodl-if
                     replicas: 1
                   statefulSets:
                   - name: ncoprimary
                     replicas: 1
                   - name: ncobackup
                     replicas: 1
    ```

These pods will start **aiops-ir-core-ncoprimary-0** and **aiops-ir-core-ncobackup-0**

Shutting down `nocprimary` and `ncobackup`

1. Find your installation name and set an environment variable for it.
    1. Find the name of your IBM Cloud Pak for AIOps installation.

        ```
        oc get installation -n <project>
        ```

        Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.

        Example output, where **ibm-cp-aiops** is the name of the IBM Cloud Pak for AIOps installation:

        ```
        NAME          PHASE    LICENSE    STORAGECLASS  STORAGECLASSLARGEBLOCK AGE
        ibm-cp-aiops  Running  Accepted   rook-cephfs   rook-ceph-rbd          34h
        ```

2. Export environment variables.

   ```
   export INSTALLATION_NAME=<name>
   export PROJECT_CP4AIOPS=<project>
   ```

   Where:

   - `<name>` is the name returned by the previous command, for example `ibm-cp-aiops`.
   - `<project>` is the project that your IBM Cloud Pak for AIOps installation is deployed in.

2. Run the following command to edit the installation.

   ```
   oc edit installation ${INSTALLATION_NAME} -n ${PROJECT_CP4AIOPS}
   ```

3. To shut down **nocprimary** and **ncobackup**, scale them down to 0.

   ```
   spec:
     pakModules:
       - enabled: true
         name: aiManager
       - enabled: true
         name: aiopsFoundation
       - enabled: true
         name: applicationManager
         config:
          - name: ir-core-operator
            spec:
              issueresolutioncore:
                customSizing:
                  deployments:
                  - name: ncodl-if
                    replicas: 0
                  statefulSets:
                  - name: ncoprimary
                    replicas: 0
                  - name: ncobackup
                    replicas: 0
   ```

Monitoring the processes

ncoprimary and **ncobackup** are not managed by deployment. They are managed by statefulsets (sts).

If you run the following command to grep the pods for **ncoprimary** and **ncobackup**, you will not see anything:

```
oc get pod | grep aiops-ir-core-ncoprimary-0
```

If you then check the statefulset using the following command:

```
oc get sts | grep nco
```

You can see you can see that the pods are listed under the statefulset, but they are not running:

```
aiops-ir-core-ncobackup                                  0/0     49d
aiops-ir-core-ncoprimary                                 0/0     49d
```

To get these pods up and running, complete the following steps:

1. Scale up the statefulsets.

   Scale up **aiops-ir-core-ncobackup**:

   ```
   oc scale --replicas=1 sts/aiops-ir-core-ncobackup
   ```

   Expected output:

   ```
   statefulset.apps/aiops-ir-core-ncobackup scaled
   ```

   Scale up **aiops-ir-core-ncoprimary**:

   ```
   oc scale --replicas=1 sts/aiops-ir-core-ncoprimary
   ```

   Expected output:

   ```
   statefulset.apps/aiops-ir-core-ncoprimary scaled
   ```

2. Check the pods for **ncoprimary** and **ncobackup**.

   Check for **ncoprimary**:

   ```
   oc get pod | grep aiops-ir-core-ncoprimary-0
   ```

   Expected output:

   ```
   aiops-ir-core-ncoprimary-0                                        1/1     Running     0               5m24s
   ```

   Check for **ncobackup**:

   ```
   oc get pod | grep aiops-ir-core-ncobackup-0
   ```

   Expected output:

```
aiops-ir-core-ncobackup-0                                         2/2     Running     0               5m54s
```

3. Grep **pvc** for **nco**.

```
oc get pvc | grep nco
```

Expected output:

```
database-aiops-ir-core-ncobackup-0                                Bound    pvc-534c0182-0ce9-47d1-a134-20168781965c
5Gi        RWO            ocs-storagecluster-cephfs     18m
database-aiops-ir-core-ncoprimary-0                               Bound    pvc-4507add6-9d6f-432b-9a53-644c742c61cd
5Gi        RWO            ocs-storagecluster-cephfs     17m
```

Integrating stand-alone on-premises probes to IBM Cloud Pak for AIOps

If you want to connect stand-alone on-premises probes to IBM Cloud Pak for AIOps, you can use NodePorts to securely connect to the in-cluster ObjectServer.

These instructions can be used to connect an on-premises deployment of IBM® Netcool® Operations Insight® to IBM Cloud Pak for AIOps.

Complete the steps in both of the following topics.

- Configuring the Cloud system
- Configuring the on-premises system

Note: IBM Cloud Pak for AIOps does not support connecting with a uni-directional gateway.

Configuring the Cloud system

1. Run the following commands to expose the ObjectServer services as node ports.

   Ensure that you replace **<name>** with the name of your IBM Cloud Pak® for AIOps instance.

   For more information about the configured IBM Cloud Pak for AIOps ports, see Ports.

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: ncoprimary
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ir-core-operator
    app.kubernetes.io/name: IssueResolutionCore
  name: aiops-ir-core-ncoprimary-np
spec:
  ports:
  - name: tds
    port: 4100
    protocol: TCP
    targetPort: 4100
  selector:
    app.kubernetes.io/component: ncoprimary
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ir-core-operator
    app.kubernetes.io/name: IssueResolutionCore
  sessionAffinity: None
  type: NodePort
EOF

cat << EOF | oc apply -f -
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: ncobackup
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ir-core-operator
    app.kubernetes.io/name: IssueResolutionCore
  name: aiops-ir-core-ncobackup-np
spec:
  ports:
  - name: tds
    port: 4100
    protocol: TCP
    targetPort: 4100
  selector:
    app.kubernetes.io/component: ncobackup
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ir-core-operator
    app.kubernetes.io/name: IssueResolutionCore
  sessionAffinity: None
  type: NodePort
EOF
```

2. Get the service node port for the primary:

```
NCOPRIMARY_NODEPORT=$(oc get svc aiops-ir-core-ncoprimary-np -o jsonpath='{.spec.ports[?(@.port==4100)].nodePort}')
```

3. Get the service node port for the backup:

```
NCOBACKUP_NODEPORT=$(oc get svc aiops-ir-core-ncobackup-np -o jsonpath='{.spec.ports[?(@.port==4100)].nodePort}')
```

4. Get the node internal IP addresses:

```
oc get node -o jsonpath='{.items[*].status.addresses[?(@.type=="InternalIP")].address}'
```

The output of the command returns the master node IP addresses and worker node IP addresses. You can run the following command to confirm which IP address belongs to which node.

```
oc get nodes -o wide | grep <NAME_OF_MASTER_OR_WORKER_NODE>
```

Where **<NAME_OF_MASTER_OR_WORKER_NODE>** is the name of the master node or worker node. For example, the name can be **master1** or **worker1**.

5. Depending on the configuration of your cluster, you might need to add network policies to allow network traffic from outside the cluster. To test this, use the commands below. If you are using a proxy such as HA proxy (see below) you should run the test on the host running the proxy. Otherwise, run the test on the host running your on-premise system. Each of the following commands should output **CONNECTED**, followed by a **Certificate chain** with the first subject containing **ncoprimary** or **ncobackup**.

```
openssl s_client -connect <WORKER0_IP>:${NCOPRIMARY_NODEPORT} < /dev/null
openssl s_client -connect <WORKER0_IP>:${NCOBACKUP_NODEPORT} < /dev/null
```

Where

- **<WORKER0_IP>** is the value of worker IP address from the node internal IP addresses command.
- **${NCOPRIMARY_NODEPORT}** is the service node port for the primary ObjectServer.
- **${NCOBACKUP_NODEPORT}** is the service node port for the backup ObjectServer.

6. If the test in step **5** fails, you might need to add network policies. The ingress "from" rules are specific to your cluster and the clients or proxies that you want to connect to the ObjectServers. The rest of the policies should look as follows:

```
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ncoprimary-external-ingress
  namespace: aiops_namespace
spec:
  ingress:
  - from:
       # Add ingress rules required for your cluster and on-premises system here
    ports:
    - port: 4100
      protocol: TCP
  podSelector:
    matchLabels:
      app.kubernetes.io/component: ncoprimary
      app.kubernetes.io/instance: aiops
  policyTypes:
  - Ingress
...
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ncobackup-external-ingress
spec:
  ingress:
  - from:
       # Add ingress rules required for your cluster and on-premises system here
    ports:
    - port: 4100
      protocol: TCP
  podSelector:
    matchLabels:
      app.kubernetes.io/component: ncobackup
      app.kubernetes.io/instance: aiops
  policyTypes:
  - Ingress
...
```

7. Next, you must update any proxies and firewalls to ensure that the exposed node ports are reachable. For example, if you have a HA proxy in front of your Red Hat OpenShift Container Platform cluster, you only need to update the HA proxy configuration that is running on the main control node. This routes that port to the master and worker nodes in the cluster. When you are logged on to that cluster, perform the following steps:

    1. Run the command:

    ```
    systemctl stop haproxy.service
    ```

    2. Add the following entries to **/etc/haproxy/haproxy.cfg**:

    ```
    frontend ingress-ncoprimary
        bind *:<NCOPRIMARY_NODEPORT>
        default_backend ingress-ncoprimary
        mode tcp
        option tcplog

    backend ingress-ncoprimary
    ```

```
    balance source
    mode tcp
    server master0 <MASTER0_IP>:<NCOPRIMARY_NODEPORT> check
    server master1 <MASTER1_IP>:<NCOPRIMARY_NODEPORT> check
    server master2 <MASTER2_IP>:<NCOPRIMARY_NODEPORT> check
    server worker0 <WORKER0_IP>:<NCOPRIMARY_NODEPORT> check
    server worker1 <WORKER1_IP>:<NCOPRIMARY_NODEPORT> check
    server worker2 <WORKER2_IP>:<NCOPRIMARY_NODEPORT> check

frontend ingress-ncobackup
    bind *:<NCOBACKUP_NODEPORT>
    default_backend ingress-ncobackup
    mode tcp
    option tcplog

backend ingress-ncobackup
    balance source
    mode tcp
    server master0 <MASTER0_IP>:<NCOBACKUP_NODEPORT> check
    server master1 <MASTER1_IP>:<NCOBACKUP_NODEPORT> check
    server master2 <MASTER2_IP>:<NCOBACKUP_NODEPORT> check
    server worker0 <WORKER0_IP>:<NCOBACKUP_NODEPORT> check
    server worker1 <WORKER1_IP>:<NCOBACKUP_NODEPORT> check
    server worker2 <WORKER2_IP>:<NCOBACKUP_NODEPORT> check
```

Where

- **<MASTER0_IP>**, **<MASTER1_IP>**, and **<MASTER2_IP>** are the values of master IP addresses from node internal IP addresses command.
- **<WORKER0_IP>**, **<WORKER1_IP>**, and **<WORKER2_IP>** are the values of worker IP addresses from node internal IP addresses command.
- **<NCOPRIMARY_NODEPORT>** is the value of service node port for the primary ObjectServer.
- **<NCOBACKUP_NODEPORT>** is the value of service node port for the backup ObjectServer.

3. Run the command:

```
systemctl start haproxy.service
```

8. Run the following command to retrieve the challenge credentials for the probe:

```
oc get secret aiops-ir-core-omni-secret -o=jsonpath='{.data.OMNIBUS_PROBE_PASSWORD}' | base64 -d
```

Configuring the on-premises system

1. Get the service CA that signed the ObjectServer certificates:

   Note: You must run this step on the cloud side and copy over the **service-ca.crt** to the on-premises server.

   ```
   oc get configmap <name>-tls-ca -o jsonpath='{.data.ca\.crt}' > service-ca.crt
   ```

   Where **<name>** is the name of your installation resource.

2. Import the CA certificate into the IBM Tivoli Netcool/OMNIbus keystore:

   ```
   $NCHOME/bin/nc_gskcmd -keydb -create -db $NCHOME/etc/security/keys/omni.kdb  -pw "$KEYSTORE_PW" -stash

   $NCHOME/bin/nc_gskcmd -cert -add -db $NCHOME/etc/security/keys/omni.kdb -pw "$KEYSTORE_PW" -label "CA" -file
   service-ca.crt
   ```

   Note: The default for **$NCHOME is /opt/IBM/tivoli/netcool/**.

3. Add the local host resolution for service name that is used in the CN of certificates to the IP of the cluster control node. This is so that the client can validate the certificate chain by the hostname used.

   ```
   /etc/hosts:
   <cluster_control_node_ip> aiops-ir-core-ncoprimary.<namespace>.svc
   <cluster_control_node_ip> aiops-ir-core-ncobackup.<namespace>.svc
   ```

4. Update the **omni.dat** (**$NCHOME/etc/omni.dat**) with the necessary connection details, for example:

   ```
   [AIOPS_AGGV]
   {
           Primary: aiops-ir-core-ncoprimary.<namespace>.svc ssl ${NCOPRIMARY_NODEPORT}
           Backup: aiops-ir-core-ncobackup.<namespace>.svc ssl ${NCOBACKUP_NODEPORT}
   }
   ```

   Then, regenerate the interfaces file through **$NCHOME/bin/nco_igen**.

5. Try pinging the server by name:

   ```
   $NCHOME/omnibus/bin/nco_ping AIOPS_AGGV
   ```

6. Specify the challenge credentials in the probe property files:

   ```
   AuthUsername: "aiopsprobe"
   AuthPassword: "<password from step 6 of on cloud side>"
   ```

Probe for IBM SevOne Network Performance Management (NPM) integration

You can configure the Probe integration for connecting IBM SevOne Network Performance Management to IBM Cloud Pak for AIOps.

Note: Since the release of IBM Cloud Pak AIOps 4.1, you can now use a Generic Webhook integration to integrate IBM Cloud Pak for AIOps with IBM SevOne Network Performance Management (NPM) to digest events. Using a Webhook integration is preferred over installing a probe to connect with IBM SevOne as it simplifies the setup process. For details see Creating Generic Webhook integrations.

- Overview
- Deployment Scenarios
- Installation

Overview

The Probe integration for IBM SevOne NPM creates a webhook endpoint to receive policy based alerts from IBM SevOne NPM. This integration is provided by the IBM Netcool Operations Insight Event Integrations Operator 3.4. Depending on your deployment you can configure the probe integration for IBM SevOne NPM connection to the IBM Cloud Pak for AIOps. If it is the IBM Cloud Pak for AIOps, the Probe integration for IBM SevOne NPM with IBM Cloud Pak for AIOps consists of the following steps. As a prerequisite you need the following systems: IBM Cloud Pak for AIOps in openshift container platform, and IBM SevOne Network Management System (NMS) 6.1. You then need to install the IBM Netcool event integrations operator, gather IBM Cloud Pak for AIOps ObjectServer connection information, configure and install the probe for SevOne integration, obtain the probe Webhook URL, and then lastly, configure IBM SevOne Network Management System (NMS) to forward events to the probe for SevOne.

Deployment Scenarios

Depending on your deployment scenario, you can configure the Probe integration for IBM SevOne NPM connecting to the IBM Cloud Pak for AIOps.

Installation

Note: Using the Generic Webhook integration is preferred over installing a probe to integrate with IBM SevOne, so you should consider migrating to this method. See Creating Generic Webhook integrations.

The instructions to configure the probe with IBM Cloud Pak for AIOps are provided in the following link. Follow the instructions according to your deployment architecture.

IBM SevOne NPM Integration with IBM Cloud Pak for AIOps

Install the Probe integration for IBM SevOne NPM connecting to IBM Cloud Pak for AIOps.

Probe integration for IBM SevOne Network Performance Management (NPM) with IBM Cloud Pak for AIOps

You can use a probe to integrate IBM Cloud Pak for AIOps with IBM SevOne Network Performance Management (NPM).

When this integration is created, alert changes from SevOne are reported within IBM Cloud Pak for AIOps. Changes in IBM Cloud Pak for AIOps are not reported in SevOne.

The properties of the alerts include fields that are based on the IBM Netcool Operations Insight **`alert.status`** table columns (**`Identifier`**, **`AlertGroup`**, **`AlertKey`**, **`EventId`**, and more).

The following table shows comparative severity mapping:

SevOne Severity	IBM Netcool Operations Insight Severity	IBM Cloud Pak for AIOps Severity
Emergency, Alert, Critical	Critical	Critical
Error	Major	Major
Warning	Warning	Warning
Notice, Info, Debug	Informational	Informational
Indeterminate	Indeterminate	Indeterminate
Cleared (from clear condition)	(Creates a resolution event)	(Clears alert)

Prerequisites

You need to install the IBM Netcool Event Integrations Operator and gather the ObjectServer connection information before proceeding with defining the probe integration. For more information about installing the operator, see Installing the IBM Netcool Event Integrations Operator and common probe configuration steps

Configure and Install Probe For SevOne Integration

1. Set the probe namespace variable to the namespace where you want to install the probe. Usually this namespace is the same namespace as the IBM Netcool Operations Insight Event Integrations Operator.

```
PROBE_NAMESPACE=<probe-namespace>
```

2. Create a secret with the ObjectServer credentials for the probe to authenticate with the ObjectServer.

```
PROBE_OMNI_SECRET=noi-probe-secret
oc create secret generic $PROBE_OMNI_SECRET --namespace $PROBE_NAMESPACE \
 --from-literal=AuthUserName=$IRC_OMNI_USERNAME \
 --from-literal=AuthPassword=$IRC_OMNI_PASSWORD
```

(Optional) If you want to import the ObjectServer TLS Certificate, use the following command to specify the **tls.crt** file in the secret.

```
PROBE_OMNI_SECRET=noi-probe-secret
oc create secret generic $PROBE_OMNI_SECRET --namespace $PROBE_NAMESPACE \
 --from-literal=AuthUserName=$IRC_OMNI_USERNAME \
 --from-literal=AuthPassword=$IRC_OMNI_PASSWORD \
 --from-file=tls.crt=tls.crt
```

3. Create a secret for the probe to use for basic authentication.

```
PROBE_AUTH_SECRET=sevone-probe-client-basic-auth
oc create secret generic --namespace $PROBE_NAMESPACE $PROBE_AUTH_SECRET --from-
literal=serverBasicAuthenticationUsername=<username>  --from-literal=serverBasicAuthenticationPassword=<password>
```

where **<username>** is a user name and **<password>** is the password for SevOne NMS to use as basic authentication.

4. Create a Network Policy in the IBM Cloud Pak for AIOps namespace.

Note: Review any other Network Policy that may be denying access to the ObjectServer pod and update the policy to allow ingress connection to the pods.

```
cat << EOF | tee >(oc apply -f -) | cat
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: probe-allow-objectserver
  namespace: ${IRC_NAMESPACE}
spec:
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          kubernetes.io/metadata.name: noi-integrations
    ports:
      - protocol: TCP
        port: 4100
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: ${IRC_INSTANCE}
      app.kubernetes.io/managed-by: ir-core-operator
      app.kubernetes.io/name: IssueResolutionCore
  policyTypes:
  - Ingress
EOF
```

5. Create a Probe for SevOne Integrations with the **WebhookProbe** custom resource by running the following commands.

```
PROBE_SEVONE_INSTANCE=sevone-probe

cat << EOF | tee >(oc apply -f -) | cat
apiVersion: probes.integrations.noi.ibm.com/v1
kind: WebhookProbe
metadata:
  name: ${PROBE_SEVONE_INSTANCE}
  labels:
    app.kubernetes.io/name: ${PROBE_SEVONE_INSTANCE}
    app.kubernetes.io/managed-by: netcool-integrations-operator
    app.kubernetes.io/instance: ${PROBE_SEVONE_INSTANCE}
  namespace: ${PROBE_NAMESPACE}
spec:
  helmValues:
    netcool:
      backupHost: '${IRC_BACKUP_OBJECTSERVER_SVC}.${IRC_NAMESPACE}.svc'
      backupPort: ${IRC_BACKUP_OBJECTSERVER_PORT}
      backupServer: 'AGGB'
      connectionMode: SSLAndAuth
      primaryHost: '${IRC_PRIMARY_OBJECTSERVER_SVC}.${IRC_NAMESPACE}.svc'
      primaryPort: ${IRC_PRIMARY_OBJECTSERVER_PORT}
      primaryServer: 'AGGP'
      secretName: '${PROBE_OMNI_SECRET}'
    probe:
      jsonParserConfig:
        notification:
          jsonNestedHeader: ''
          jsonNestedPayload: ''
          messageDepth: 3
          messageHeader: ''
          messagePayload: json
      integration: sevone
      enableTransportDebugLog: false
      messageLevel: debug
    ingress:
      enabled: true
      host: ''
```

```
    arch: amd64
    webhook:
      uri: /probe/sevone
      serverBasicAuthenticationCredentialsSecretName: '${PROBE_AUTH_SECRET}'
      tls:
        enabled: true
        secretName: ''
  license:
    accept: true
  version: 3.1.0
EOF
```

6. Verify that the probe pod is running.

```
oc get pods -l app.kubernetes.io/instance=$PROBE_SEVONE_INSTANCE
```

Obtain the Probe Webhook URL

Use the following command to get the probe webhook URL, which is used to configure a Webhook destination in IBM SevOne NMS.

```
PROBE_HOSTNAME=$(oc get route $PROBE_SEVONE_INSTANCE-mb-webhook -o jsonpath='{.spec.host}')
PROBE_URI=$(oc get route $PROBE_SEVONE_INSTANCE-mb-webhook -o jsonpath='{.spec.path}')
PROBE_WEBHOOK_URL=https://$PROBE_HOSTNAME$PROBE_URI
echo "$PROBE_WEBHOOK_URL"
```

Configure IBM SevOne Network Management System (NMS) to forward events to the Probe for SevOne

IBM SevOne NMS must be configured with a new webhook configuration to send alerts to the Probe for SevOne NMS.

Note: Instructions to configure the webhook with SevOne NMS can differ depending on your SevOne NMS version. For more information about using the SevOne Webhook Definition Manager and configuring alerts in SevOne NMS, see NMS 6.6 Guides and the SevOne NMS User Guide.

Note IBM SevOne requires the webhook destination to use a TLS certificate signed by a trusted Certificate Authority (CA).

1. Log in to the IBM SevOne console.
2. On the top menu, go to **Events > Configuration > Webhook Definition Manager**.
3. On the Policy Browser page, click **Configure Webhook Destination** on the Policies panel.
4. Complete the following steps from the Add Webhook Destination panel:
    1. Set a Webhook Destination Name, for example: IBM Cloud Pak for AIOps Probe for SevOne
    2. Set a Description, for example: IBM Cloud Pak for AIOps Probe for SevOne NMS
    3. Set the URL with the `$PROBE_WEBHOOK_URL` value from the previous Obtaining the Probe Webhook URL section.
    4. Select the **Allow insecure webhook connection** option if the probe uses a self-signed certificate.
    5. Expand the Headers section and select **Use Basic Auth**. Then, set the **Username** and **Password** fields with the credentials set in the `PROBE_AUTH_SECRET` from the Configure and install probe for SevOne integration section.
    6. Click **Test Definition** to send a test HTTP event to the probe. Test both the triggering condition and clear condition webhooks to ensure that the recieved response code is `200` or `204`.
    - A `204` status code indicates that no payload is included in the HTTP response, but the connection is successful.
    - If a response code of `400 Bad request` is received, check the JSON template that is configured. Ensure that the JSON does not contain any syntax errors.
    7. Click **Save**.
5. Configure or create policies to send events to the probe by selecting the probe as the webhook destination. For each of the policies in the **Webhooks** section, select the webhook that was created in the previous step. Complete the selection for the **Trigger Conditions** and **Clear Conditions** tabs and select the **One Webhook per Alert** option.

Probe for IBM Turbonomic integration

You can configure the Probe integration for connecting IBM Turbonomic to IBM Cloud Pak for AIOps.

Note: Since the release of IBM Cloud Pak AIOps 4.1, you can now use a Generic Webhook integration to integrate IBM Cloud Pak for AIOps with IBM Turbonomic to digest events. Using a Webhook integration is preferred over installing a probe to connect with IBM Turbonomic as it simplifies the setup process. For details see Creating Generic Webhook integrations.

The Probe integration for IBM Turbonomic is provided by the IBM Netcool Operations Insight Event Integrations Operator.

- Prerequisites
- Common pre-installation tasks
- Configure and install probe For Turbonomic integration
- Obtain the probe Webhook URL
- Configure IBM Turbonomic to forward events to the probe for Turbonomic

Prerequisites

The following systems are prerequisites for this integration.

1. IBM Cloud Pak for AIOps in Openshift Container Platform
2. IBM Turbonomic (IBM Turbonomic 8.6)

Common pre-installation tasks

Follow the instructions in Installing the IBM Netcool Event Integrations Operator and common probe configuration steps to install the operator and gather the ObjectServer connection information before proceeding with the following sections.

Configure and install probe For Turbonomic integration

Note: Using the Generic Webhook integration is preferred over installing a probe to connect with IBM Turbonomic, so you should consider migrating to this method. See Creating Generic Webhook integrations.

Use the following steps to create the required secrets, configure the WebhookProbe custom resource and install the probe in the namespace.

1. Create a secret with the ObjectServer credentials for the probe to authenticate with the ObjectServer.

```
PROBE_OMNI_SECRET=noi-probe-secret
oc create secret generic $PROBE_OMNI_SECRET \
 --from-literal=AuthUserName=$IRC_OMNI_USERNAME \
 --from-literal=AuthPassword=$IRC_OMNI_PASSWORD
```

   If you want to import the ObjectServer TLS Certificate (optional), use the following command to specify the **tls.crt** file in the secret.

```
PROBE_OMNI_SECRET=noi-probe-secret
oc create secret generic $PROBE_OMNI_SECRET \
 --from-literal=AuthUserName=$IRC_OMNI_USERNAME \
 --from-literal=AuthPassword=$IRC_OMNI_PASSWORD \
 --from-file=tls.crt=tls.crt
```

2. Create a secret for the probe to use for basic authentication.

```
PROBE_AUTH_SECRET=turbonomic-probe-client-basic-auth
oc create secret generic $PROBE_AUTH_SECRET --from-literal=serverBasicAuthenticationUsername=<username>  --from-
literal=serverBasicAuthenticationPassword=<password>
```

   where **<username>** is a user name and **<password>** is the password for Turbonomic to use as basic authentication.

3. Create a Network Policy in the IBM Cloud Pak for AIOps namespace.

   Note: Review any other network policy that may be denying access to the ObjectServer pod(s) and update the policy to allow ingress connection to the pods.

```
cat << EOF | tee >(oc apply -f -) | cat
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: probe-allow-objectserver
  namespace: ${IRC_NAMESPACE}
spec:
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          kubernetes.io/metadata.name: noi-integrations
    ports:
      - protocol: TCP
        port: 4100
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: ${IRC_INSTANCE}
      app.kubernetes.io/managed-by: ir-core-operator
      app.kubernetes.io/name: IssueResolutionCore
  policyTypes:
  - Ingress
EOF
```

4. Create a probe for Turbonomic integrations with the **WebhookProbe** custom resource by running the following commands:

```
PROBE_TURBONOMIC_INSTANCE=turbonomic-probe

cat << EOF | tee >(oc apply -f -) | cat
apiVersion: probes.integrations.noi.ibm.com/v1
kind: WebhookProbe
metadata:
  name: ${PROBE_TURBONOMIC_INSTANCE}
  labels:
    app.kubernetes.io/name: ${PROBE_TURBONOMIC_INSTANCE}
    app.kubernetes.io/managed-by: netcool-integrations-operator
    app.kubernetes.io/instance: ${PROBE_TURBONOMIC_INSTANCE}
  namespace: ${NAMESPACE}
spec:
```

```
  helmValues:
    netcool:
      backupHost: '${IRC_BACKUP_OBJECTSERVER_SVC}.${IRC_NAMESPACE}.svc'
      backupPort: ${IRC_BACKUP_OBJECTSERVER_PORT}
      backupServer: 'AGGB'
      connectionMode: SSLAndAuth
      primaryHost: '${IRC_PRIMARY_OBJECTSERVER_SVC}.${IRC_NAMESPACE}.svc'
      primaryPort: ${IRC_PRIMARY_OBJECTSERVER_PORT}
      primaryServer: 'AGGP'
      secretName: '${PROBE_OMNI_SECRET}'
    probe:
      jsonParserConfig:
        notification:
          jsonNestedHeader: ''
          jsonNestedPayload: ''
          messageDepth: 3
          messageHeader: ''
          messagePayload: json
      integration: Turbonomic
      enableTransportDebugLog: false
      messageLevel: debug
    ingress:
      enabled: true
      host: ''
    arch: amd64
    webhook:
      uri: /probe/turbonomic
      serverBasicAuthenticationCredentialsSecretName: '${PROBE_AUTH_SECRET}'
      tls:
        enabled: true
        secretName: ''
  license:
    accept: true
  version: 3.1.0
EOF
```

5. Verify that the probe pod is running.

```
oc get pods -l app.kubernetes.io/instance=$PROBE_TURBONOMIC_INSTANCE
```

Obtain the probe Webhook URL

Use the following command to get the probe webhook URL, which is used to configure a Webhook destination in IBM Turbonomic.

```
PROBE_HOSTNAME=$(oc get route $PROBE_TURBONOMIC_INSTANCE-mb-webhook -o jsonpath='{.spec.host}')
PROBE_URI=$(oc get route $PROBE_TURBONOMIC_INSTANCE-mb-webhook -o jsonpath='{.spec.path}')
PROBE_WEBHOOK_URL=https://$PROBE_HOSTNAME$PROBE_URI
echo "$PROBE_WEBHOOK_URL"
```

Configure IBM Turbonomic to forward events to the probe for Turbonomic

IBM Turbonomic must be configured with a new webhook configuration to send alerts to the probe for Turbonomic.

Creating a workflow

1. Set the following variables:

```
TURBONOMIC_HOSTNAME=<api endpoint hostname>
TURBONOMIC_ADMIN_USER=<turbonomic username>
TURBONOMIC_ADMIN_PASSWORD=<turbonomic password>
BASIC_AUTH_USERNAME=<probe basic authentication username>
BASIC_AUTH_PASSWORD=<probe basic authentication password>
```

2. Get the JSESSIONID token

```
JSESSIONID=$(curl \
--silent \
--cookie-jar - \
--insecure \
https://${TURBONOMIC_HOSTNAME}/api/v3/login?hateoas=true \
--data "username=${TURBONOMIC_ADMIN_USER}&password=${TURBONOMIC_ADMIN_PASSWORD}" \
| awk '/JSESSIONID/{print $7}')
```

3. Create a workflow

```
curl \
"https://${TURBONOMIC_HOSTNAME}/api/v3/workflows" \
--insecure \
--compressed \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "cookie: JSESSIONID=${JSESSIONID}" \
```

```
--request POST \
-vv \
--data @- <<EOF
{
"displayName": "Event Probe",
"className": "Workflow,
"description": "Event webhook probe",
"discoveredBy": {
    "readonly": false
},
"type": "WEBHOOK",
"typeSpecificDetails": {
  "url": "${PROBE_WEBHOOK_URL}",
  "method": "POST",
  "authenticationMethod": "BASIC",
  "username": "${BASIC_AUTH_USERNAME}",
  "password": "${BASIC_AUTH_PASSWORD}",
  "template": "\$converter.toJson(\$action)",
  "trustSelfSignedCertificates": true,
  "headers": [
     {
      "name": "Content-type",
      "value": "application/json"
     }
   ],
  "type": "WebhookApiDTO"
  }
}
EOF
```

4. Get the workflow UID for later use.

```
WORKFLOW_UUID=$(curl \
"https://${TURBONOMIC_HOSTNAME}/api/v3/workflows" \
--insecure \
--compressed \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "cookie: JSESSIONID=${JSESSIONID}" \
--request GET | jq -c ".[] | select(.typeSpecificDetails .
url | contains(\"${PROBE_WEBHOOK_URL}\")) | .uuid" | tr -d '"')
```

Create a policy

1. Log in to the Turbonomic console.
2. Navigate to the *Settings* page and then choose **Policies**.
3. Click **New Automation Policy** and then select the policy type, **Virtual Machine** in this case. This sets the type of entity that your policy will affect. Refer to Working with policies for more information.

   Name the policy:

   - **Add Scope** The scope determines which entities this policy will affect. Choose one or more groups, or create new groups and add them to the policy scope. These groups match the type of entity you have set for the policy.
   - **Policy Schedule** The *Select Schedule* fly-out lists all the schedules that are currently defined for your instance of Turbonomic. Expand a schedule entry to see further details, or skip it if you don't need the schedule.
   - **Automation and Orchestration** You can define automation and orchestration settings for different action types within the same policy.
     - **Action Type** See a list of actions that are viable for the policy, and then make your selections.
     - **Action Generation and Acceptance**:
       - Do not generate actions as Turbonomic doesn't considers your selected actions in its calculations. For example, if you do not want to generate resize actions for VMs in the policy, analysis will still drive toward the desired state, but will do so without considering resizes.
       - Generate Actions Turbonomic generates your selected actions to address or prevent problems. Choose from the following *Action Acceptance* modes to indicate how you would like the actions to run:
         - **Recommend** — Recommend the action so a user can run it via the given hypervisor or by other means
         - **Manual** — Recommend the action, and provide the option to run that action through the Turbonomic user interface
         - **Automatic** — Run the action automatically For automated resize or move actions on the same entity, Turbonomic waits five minutes between each action to avoid failures associated with trying to run all actions at once. Any action awaiting execution stays in queue.
     - **Before Execution and Action Execution:** By default, generated actions run without the need for orchestration. Turbonomic gives you the ability to set up orchestration to affect the execution of actions.
     - **After Execution**: The default is to 'do nothing'. Other options include:
       - **Create Record in Orchestration** Turbonomic registers the action in the Orchestration log, showing that the given action has been run. For example: A final ActionID could be created and with the final ActionState, the Type and Severity would be updated accordingly in

IBM Cloud Pak for AIOps. Those actions would also be updated to IBM Cloud Pak for AIOps, and it would handle with the alerts with updated state: closed or cleared or keep open.

- **Use Action Script** Run an action script that is set up for the POST entrypoint. The action script name must match the entity type and action type.

4. Save the policy.
5. Action events will be sent to the probe when an action is run.

Triggering a test event

To trigger a test event to verify that the webhook integration is successful:

1. Get a list of actions by market to test out the workflow. Select an action from the list and copy the UUID, which will be used to trigger an action event manually using an API.

```
curl \
"https://${TURBONOMIC_HOSTNAME}/api/v3/markets/Market/actions?ascending=true" \
--insecure \
--compressed \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "cookie: JSESSIONID=${JSESSIONID}" \
--request GET
```

2. To test triggering an action event based on the UUID in the workflow, use one of the action UUIDs from the previous command (in the actionId attribute value) as in the following command:

```
curl --location --request POST "https://${TURBONOMIC_HOSTNAME}/api/v3/workflows/${WORKFLOW_UUID}" \
--header 'Content-Type: application/json' \
--header "Cookie: JSESSIONID=$JSESSIONID" \
--insecure \
--data-raw '{
    "operation": "TEST",
    "actionId": "637553640602793"
}'
```

Datadog integration probe

An IBM Tivoli Netcool/OMNIbus probe for integrating with Datadog is available. This probe integration sends HTTP GET requests to Datadog to retrieve events based on the time frame set by start and end mandatory query parameters in the URI.

Prerequisites

The following systems are prerequisites for this integration.

- IBM Cloud Pak for AIOps version 4.8.0
- Datadog: Cloud Monitoring as a Service

Note: This probe only supports the `https://api.datadoghq.com/api/v1/events` v1 API to get events from Datadog.

Note: The IBM Netcool Operations Insight Event Integrations Operator must be deployed before deploying this integration. See Installing the IBM Netcool Operations Insight Event Integrations Operator

To deploy a new instance of the Generic Webhook Probe, review the custom resource YAML template supplied with the operator package, update the parameters if required, and save the configurable parameters in a file, for example `probes_v1_webhookprobe_datadog_integration.yaml`.

Then deploy the custom resource using the updated file:

```
kubectl apply -f deploy/crs/probes_v1_webhookprobe_datadog_integration.yaml --namespace <NAMESPACE>
```

Note: If you are installing using the CLI, you can get the custom resource Yaml file from the `ibm-netcool-integrations/inventory/netcoolIntegrationsOperatorSetup/files/op-cli/deploy/crs` directory in our CASE archive. If you are installing using the Operator Lifecycle Manager Web console, the custom resource is provided in the YAML view.

Pre-installation tasks

The Probe for SevOne, Probe for Turbonomics, and Probe for Datadog have the same pre-installation requirements. See Probe Common Tasks before proceeding with the following sections.

Configuring and installing the Probe for Datadog integration

To create the required secrets, complete the following steps, which configure the `WebhookProbe` custom resource and install the probe in the namespace.

1. Create a secret with the ObjectServer credentials for the probe to authenticate with the ObjectServer.

```
PROBE_OMNI_SECRET=noi-probe-secret
kubectl create secret generic $PROBE_OMNI_SECRET --from-literal=AuthUserName=$NOI_OMNI_USERNAME --from-
literal=AuthPassword=$NOI_OMNI_PASSWORD --from-file=tls.crt=tls.crt
```

2. Create a secret for the Datadog certificate.

```
echo -n | openssl s_client -connect api.datadoghq.com:443 | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p'  >
datadog.pem
PROBE_CLIENT_TLS_SECRET=datadog-probe-client-tls-cert
kubectl create secret generic $PROBE_CLIENT_TLS_SECRET --from-file=server.crt=datadog.pem
```

3. Create a secret for the transport properties.

```
TRANSPORT_PROPS=restMultiChannelHttpTransport.json
cat << EOF | tee > $TRANSPORT_PROPS
{
  "GLOBAL":
  {
    "httpVersion":"1.1",
    "httpHeaders":"",
    "responseTimeout":"60",
    "securityProtocol":"TLSv1.2",
    "keepTokens":"",
    "autoReconnect":"OFF",
    "gatherSubsTopicInfo":"false"
  },
  "RESYNC":
  {
      "RESYNC_CHILD_BY_FDN":
      {
          "uri":"https://api.datadoghq.com:443/api/v1/events?start=++date_happened:<initial-start-epoch-
time>++&end=++now++&page=0",
          "method":"GET",
          "headers":"Accept=application/json,DD-API-KEY=<datadog-api-key>,DD-APPLICATION-KEY=<datadog-application-
key>",
          "content":"{}",
          "contentFile" : "",
          "interval":"60",
          "attempts":"0",
          "requireSSL":"true",
          "asEventStream" : "false",
          "enablePagination" : "true"
      }
  }
}
EOF

PROBE_HTTP_REQUEST=http-requests
kubectl create secret generic $PROBE_HTTP_REQUEST --from-file=httpRequests=$TRANSPORT_PROPS
```

Where:

- **++date_happened:<initial-start-epoch-time>++** allows the probe to assign the value of RecordData.date_happened in the probe data backup file to the start query parameter to continue from previous resynchronization if RecordData.date_happened is available. Otherwise, the probe assigns **<initial-start-epoch-time>** to the start query parameter.
- **<datadog-api-key>** is the Datadog API key.
- **<datadog-application-key>** is Datadog Application key.

Note: The end query parameter can be set to an epoch time or ++now++ which represents the current epoch time.

4. Create a Network Policy in the IBM Tivoli Netcool/OMNIbus namespace to allow the probe to access the TLS Proxy service and port in IBM Tivoli Netcool/OMNIbus namespace. Note: Review any other Network Policy that may be denying access to the ObjectServer TLS Proxy pod and update the policy to allow ingress connection to the TLS Proxy pod.

```
cat << EOF | tee >(kubectl apply -f -) | cat
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: noi-allow-proxy
  namespace: ${NOI_NAMESPACE}
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: ${NOI_INSTANCE}
      app.kubernetes.io/name: proxy
  ingress:
    - ports:
        - protocol: TCP
          port: ${NOI_PROXY_PRIMARY_PORT}
        - protocol: TCP
          port: ${NOI_PROXY_BACKUP_PORT}
  policyTypes:
    - Ingress
EOF
```

5. Create a Probe for Datadog Integrations with the WebhookProbe custom resource by running the following commands.

```
PROBE_DATADOG_INSTANCE=datadog-probe
```

```
cat << EOF | tee >(kubectl apply -f -) | cat
kind: WebhookProbe
apiVersion: probes.integrations.noi.ibm.com/v1
metadata:
  name: webhookprobe
  labels:
    app.kubernetes.io/name: ${PROBE_DATADOG_INSTANCE}
    app.kubernetes.io/managed-by: netcool-integrations-operator
    app.kubernetes.io/instance: ${PROBE_DATADOG_INSTANCE}
  namespace: ${NAMESPACE}
spec:
  license:
    accept: true
  version: 3.3.0
  helmValues:
    global:
      image:
        secretName: ''
      serviceAccountName: ''
      enableNetworkPolicy: true
    image:
      useDefaultOperandImages: true
      probe: >-
        cp.icr.io/cp/noi-int/netcool-probe-
messagebus@sha256:745a7dae41f1a0c8f1517a9dc2368d227e39be4798b74175f903bc5ac940b95b
      utility: >-
        cp.icr.io/cp/noi-int/netcool-integration-
util@sha256:6e0b9883a34810c468b5386b4b6d9ed216f68df015f6622b83656d400d01dcd2
      pullPolicy: IfNotPresent
    netcool:
      connectionMode: SSLAndAuth
      primaryServer: 'AGGP'
      primaryHost: '${NOI_PROXY_SVC}.${NOI_NAMESPACE}.svc'
      primaryPort: 4100
      backupServer: 'AGGB'
      backupHost: '${NOI_PROXY_SVC}.${NOI_NAMESPACE}.svc'
      backupPort: 4100
      secretName: '${PROBE_OMNI_SECRET}'
    probe:
      integration: datadog
      messageLevel: warn
      setUIDandGID: false
      sslServerCommonName: ''
      locale: en_US.utf8
      enableTransportDebugLog: false
      heartbeatInterval: 10
      recordData: 'date_happened'
      configPVC:
        name: ''
        rulesFile: ''
        parserConfigFile: ''
        dataBackupFile: ''
      rulesConfigmap: ''
      jsonParserConfig:
        notification:
          messagePayload: json
          messageHeader: ''
          jsonNestedPayload: ''
          jsonNestedHeader: ''
          messageDepth: 3
        resync:
          messagePayload: 'json.events'
          messageHeader: ''
          jsonNestedPayload: ''
          jsonNestedHeader: ''
          messageDepth: 3
      httpClientCredentialsSecretName: ''
      initialResync: true
      resyncInterval: 120
      poddisruptionbudget:
        enabled: false
        minAvailable: 1
      selfMonitoring:
        discardHeartbeatEvent: false
        populateMasterProbeStat: false
        logProbeStat: false
        generateThresholdEvents: false
      configs: 'ProbeWatchHeartbeatInterval:0 EventLoadProfiling:''true'''
    ingress:
      enabled: false
      host: ''
    webhook:
      uri: /probe
      httpVersion: '1.1'
      respondWithContent: 'OFF'
      validateBodySyntax: 'ON'
      validateRequestURI: 'ON'
      idleTimeout: 180
      tls:
        enabled: true
        secretName: ''
      serverBasicAuthenticationCredentialsSecretName: ''
```

```
      replicaCount: 1
      autoscaling:
        enabled: false
        minReplicas: 1
        maxReplicas: 3
        cpuUtil: 60
    httpClient:
      enabled: true
      host: ''
      port: 80
      sslSecretName: '$PROBE_CLIENT_TLS_SECRET'
      httpRequestsSecretName: '$PROBE_HTTP_REQUEST'
    resources:
      requests:
        cpu: 100m
        memory: 128Mi
        ephemeral-storage: 100Mi
      limits:
        cpu: 200m
        memory: 256Mi
        ephemeral-storage: 200Mi
    arch: amd64
EOF
```

6. Verify that the probe pod is running.

```
kubectl get pods -l app.kubernetes.io/instance=$PROBE_DATADOG_INSTANCE
```

7. If you want to store the probe data backup file in Persistent Volume Claim (PVC), follow the instructions in Providing custom Message Bus Probe configuration files in a persistent volume. Then set configPVC.name to a pre-created PVC name and configPVC.dataBackupFile to the data backup filename.

Installing the IBM Netcool Event Integrations Operator and common probe configuration steps

This topic describes how to install the IBM Netcool Event Integrations Operator which contains the probes you will need to connect to, and acquire event data from, event sources. It also describes how to gather the IBM Cloud Pak for AIOps ObjectServer connection information that the individual probe installations require.

- 1. Install the IBM Netcool Event Integrations Operator
- 2. Gather IBM Cloud Pak for AIOps ObjectServer connection information

1. Install the IBM Netcool Event Integrations Operator

Install the IBM® Netcool® Operations Insight® Event Integrations Operator.

- For more information, see Installing the IBM Netcool Operations Insight Event Integrations Operator but skip the Install Netcool Operations Insight step.
- Or see Installing the IBM Netcool Operations Insight Event Integrations Operator to a private container registry.

2. Gather IBM Cloud Pak for AIOps ObjectServer connection information

Before you configure the probe and connect to the ObjectServer, you need to gather the following information:

- Primary and backup ObjectServer service name and port
- ObjectServer client credentials
- TLS certificate used by the ObjectServer

In the IBM Cloud Pak for AIOps, the ObjectServer is deployed by the issue resolution core component. Complete the following instructions to get the ObjectServer connection details.

1. Set the IBM Cloud Pak for AIOps namespace variable to the namespace where IBM Cloud Pak for AIOps is installed.

```
IRC_NAMESPACE=<namespace>
```

2. Determine the issue resolution core instance name.

```
IRC_INSTANCE=$(oc get issueresolutioncore -n $IRC_NAMESPACE -o custom-columns=name:metadata.name --no-headers)
```

3. Run the following commands to gather service information and credentials.

```
 IRC_PRIMARY_OBJECTSERVER_SVC=$IRC_INSTANCE-ir-core-ncoprimary
 IRC_BACKUP_OBJECTSERVER_SVC=$IRC_INSTANCE-ir-core-ncobackup
 IRC_PRIMARY_OBJECTSERVER_PORT=$(oc get svc -n $IRC_NAMESPACE $IRC_PRIMARY_OBJECTSERVER_SVC -o
jsonpath='{.spec.ports[?(@.name=="primary-tds-port")].port}')
 IRC_BACKUP_OBJECTSERVER_PORT=$(oc get svc -n $IRC_NAMESPACE $IRC_BACKUP_OBJECTSERVER_SVC -o
jsonpath='{.spec.ports[?(@.name=="backup-tds-port")].port}')
 IRC_OMNI_USERNAME=aiopsprobe
 IRC_OMNI_PASSWORD=$(oc get secret -n $IRC_NAMESPACE $IRC_INSTANCE-ir-core-omni-secret -o
jsonpath='{.data.OMNIBUS_PROBE_PASSWORD}' | base64 --decode && echo)
```

4. Extract the ObjectServer TLS certificate from the secret into your current directory. This file is used when creating a secret for the probe to connect with the ObjectServer in the following sections.

```
oc extract secret/$IRC_INSTANCE-ir-core-ncoprimary-tls -n $IRC_NAMESPACE --to=. --keys=tls.crt
```

5. Proceed to install the probe.

Individual probe integration PDF documentation

To view the offline PDF documentation for individual probe integrations, download the appropriate PDF documentation.

For the full list of PDF documentation for each probe integration, view the online IBM documentation IBM Cloud Pak for AIOps offline documentation.

Managing AIOps Insights

IBM Cloud Pak® for AIOps includes an AIOps Insights dashboard in the IBM Cloud Pak for AIOps console. This dashboard provides a set of predefined interactive charts that show you, at a glance, what is going on in your environment. For more information about using the dashboard, see AIOps Insights dashboard.

Enabling AIOps Insights

During installation of IBM Cloud Pak® for AIOps, AIOps Insights is enabled by default and is accessible to all users.

Disabling AIOps Insights

Access to AIOps Insights can be restricted by changing the permissions of a user. For more information, see Roles and permissions.

Scopes and filtering

Once you enable data to flow into the AIOps Insights dashboard, you can refine your insights by using filters and scopes to segment that data. For more information, see AIOps Insights Scopes and Filter.

AIOps Insights dashboard

Use the AIOps Insights dashboard to view key insights provided by IBM Cloud Pak® for AIOps. View insights for a 24-hour, 7-day, or 30-day period. This information can show the value of using IBM Cloud Pak for AIOps for your IT operations.

The AIOps Insights dashboard can be accessed from the administration page of IBM Cloud Pak for AIOps, under **Overview** > **Quick navigation** > **AIOps insights**. Alternatively, click the navigation icon (four horizontal bars) in the upper-left corner of the screen to go to the main navigation menu. Then click **Operate** > **AIOps insights**.



Figure. AIOps Insights dashboard

Prerequisites

All relevant data integrations and activity (incident opened, active runbook, and so on) must be operational to allow live data to flow into the chart.

By default, all users have access to AIOps Insights. However, Administrators can restrict access, by changing a user's permissions. For more information, see Roles and permissions.

Note: If you have not yet setup integrations or implemented activity (or there is limited information) you can view examples of each AIOps Insights. chart, containing sample data, by taking the AIOps Insights. product tour. Click the signpost icon beside your avatar, then **Product Tours** > **Discover AIOps insights**.

Summary bar

The summary bar of the dashboard displays key indicators and shows how IBM Cloud Pak for AIOps is improving the efficiency of your IT operations teams. It summarizes the charts currently showing on the dashboard. An arrow indicates an increase or decrease in activity.



Figure. Summary bar

For more detailed information on metrics in the summary bar, view the individual charts.

- Mean time to restore
- Incident activity
- Noise reduction
- Runbook automation
- Chart tooltips

Mean time to restore

The *Mean time to restore* chart displays the average time that is needed to isolate, repair, and fully resolve IT incidents.

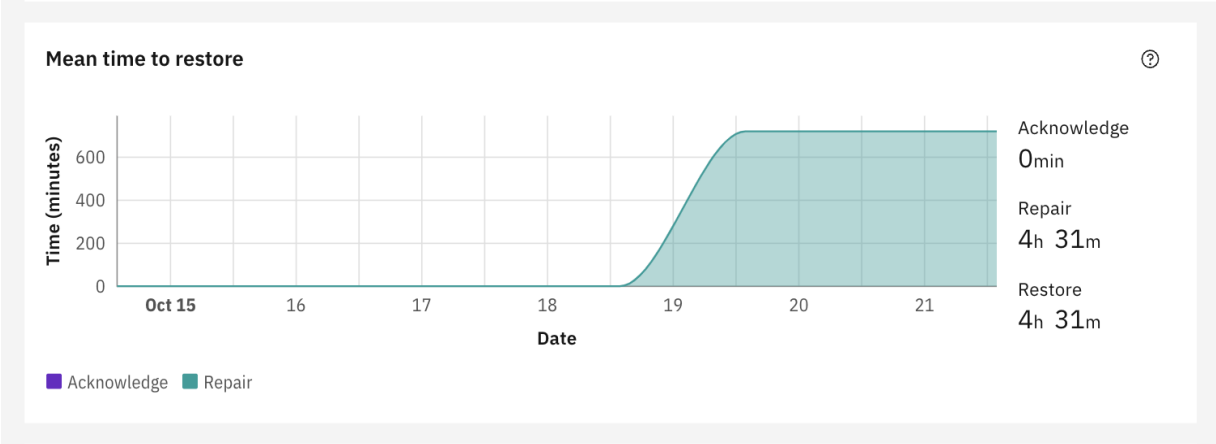


Figure. Mean time to restore chart

Chart definitions:

Acknowledge: the mean time that it takes to begin work on an incident. Toggle the purple *Acknowledge* box in the legend to display only this data on the chart.

Repair: the mean time taken to reestablish a service. Toggle the green *Repair* box in the legend to display only this data on the chart.

Restore: the total mean time (*Acknowledge* time and *Repair* time combined) taken to restore service.

Note: All mean values that are generated are based on dividing the item total by the number of data points on the chart. Because of the time intervals, this is always the selected time period +1 unit. So 24-hour totals are divided by 25, 7-day totals are divided by 8, 30-day totals are divided by 31.

Incident activity

Within IBM Cloud Pak for AIOps, an incident is a collection of alerts, insights, and potential solutions to help drive incident remediation.

By clicking the chart icon in the *Incident activity* chart, you can select to view by *Time series* (default view) or *Summary*.

- The *Time series* chart displays the number of incidents that were opened and closed during the selected time period.

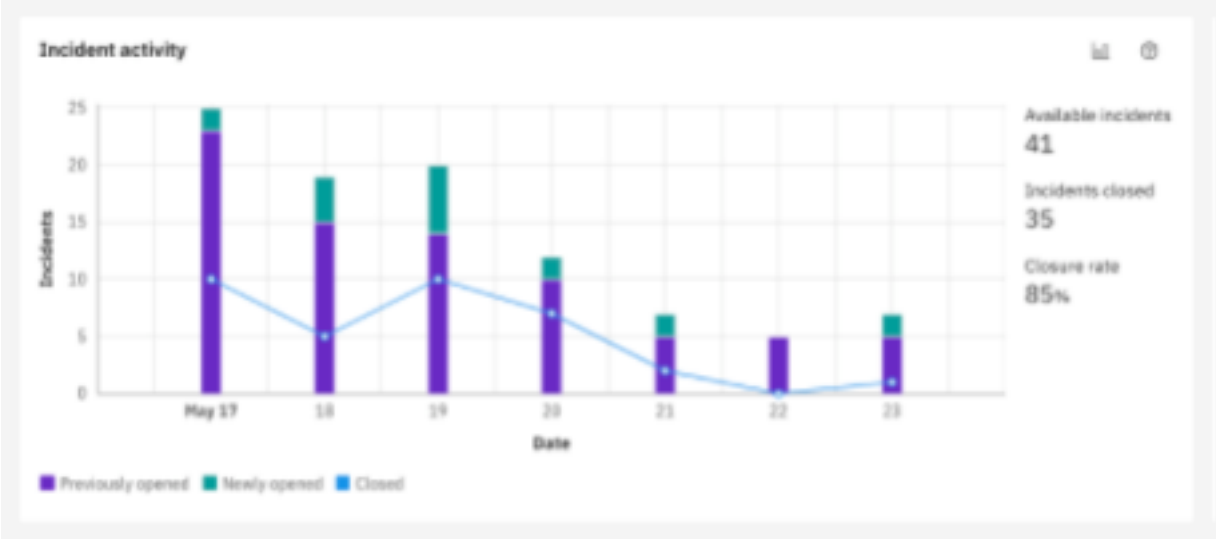


Figure. Incidents closure rate

- The *Summary* chart displays open and closed incidents according to their severity level, over the selected time period.



Figure. Summary setting

Chart definitions:

Previously opened: incidents carried over from the preceding time unit (for example, in a 7-day period, it would be incidents that are carried over from the day before), and previously opened incidents that are not yet resolved. Hover the cursor over the purple bar to see specific numbers for each day. Toggle the purple *Previously opened* box in the legend to display only this data on the chart.

Newly opened: incidents opened in the selected time period. Hover the cursor over the green bar to see specific numbers for each day. Toggle the green *Opened* box in the legend to display only this data on the chart.

Closed: the number of incidents that closed in the selected time period. Expressed as a line graph. Toggle the blue *Closed* box in the legend to display only this data on the chart.

Available incidents: the total number of incidents that were in an open state over the selected period. Combines the number of previously opened incidents – carried over when the selected time period began - with the aggregate number of new incidents (combined value of all the green bars) opened.

Incidents closed: the aggregated number of incidents closed in the selected period.

Closure rate: *Incidents closed* expressed as a percentage of the *Total incidents*.

Noise reduction

The *Noise reduction* chart displays how IBM Cloud Pak for AIOps reduces the number of IT events and alerts that your operations staff must evaluate, speeding up recovery time and reducing employee fatigue.



Figure. Noise reduction dashboard

Note: Updated events first created more than 30 days earlier are not incremented into the *Noise reduction* chart. For more information, see this Known issue.

Chart definitions:

Noise reduction occurs in three phases, as depicted in the chart:

1. *Events*: Duplicate or irrelevant events are removed.
2. *Alerts*: Events are correlated and grouped into alerts based on when they occurred and the relationships between the associated IT resources.
3. *Incidents*: Alerts are grouped into prioritized incidents that also include insights and potential solutions to help drive incident remediation.

The *Noise reduction* percentage is the reduction that occurs in the transition from the original number of events that are generated to the final number of incidents.

IBM Cloud Pak for AIOps includes three noise reduction algorithms, two of which require no pre-training. You can also create policies to suppress events and precisely control how alerts are grouped into incidents. For more information, see About event grouping, Deduplication, and About policies.

Incidents and noise reduction:

Leveraging **Incidents** is fundamental to deriving value from IBM Cloud Pak for AIOps. If you limit the number of policies that create incidents, by default you also limit the level of noise reduction that can be tracked and presented in the *Noise Reduction* chart.

Policies connect events and alerts to incidents. Without that connection, the value of the *Noise Reduction* chart is unnecessarily restricted. It can still display the number of *Events* and *Alerts* but not the level of *Noise Reduction*, which will instead display as empty (double dashes).



Figure. Noise reduction dashboard with incidents restricted

Note: Alerts that surface in *Resource Management* may not flow into the Noise Reduction chart. This is because they may not be part of a policy that directly ties them to an incident.

Runbook usage

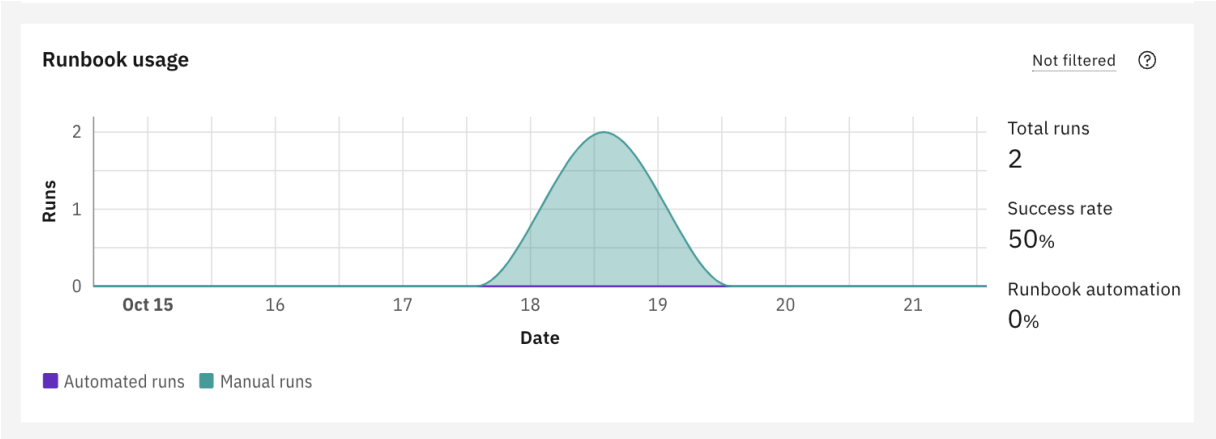


Figure. Runbook usage

Use the runbook usage chart to view the number of automated versus manual runbook runs and the success rates of them.

Chart definitions:

Automated runs: Runbooks that were run without any operator interaction, in the selected period. Toggle the purple *Automated runs* box in the legend to display only this data on the chart.

Manual runs: Runbooks that were implemented manually by an operator who followed an exact procedure, in the selected period. Toggle the green *Manual runs* box in the legend to display only this data on the chart.

Total runs: The combined number of automated and manual runs, in the selected period.

Success rate: The percentage of automated and manual runbooks successfully implemented.

Runbook automation: Percentage of total runbooks that were automated - that needed no operator interaction.

Chart tooltips

White dots highlight particular points on a chart. Clicking on a chart and then clicking one of these dots generates a tooltip data box for that particular point in time.
Note: This is not available for Noise reduction chart as it does not use a line graph.

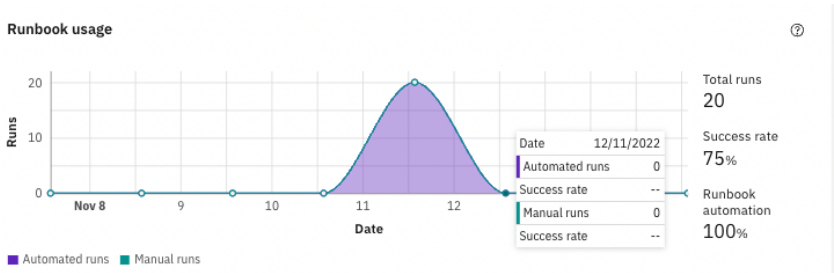


Figure. Tooltip

Tooltip information is also available for metrics in the sidebar of a chart.

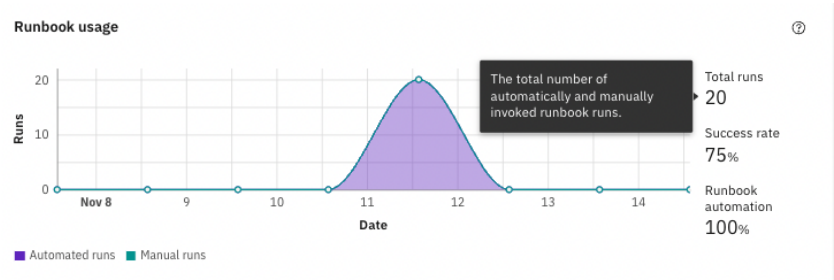


Figure. Tooltip sidebar

AIOps Insights Scopes and Filter

Use customized scopes and the filter bar in **AIOps Insights** to refine your charts, to achieve more focused insights.

Create a scope

A scope is a customized view of data. Use a scope to segment data and base your metrics on a meaningful set of IT resources.

Initially, **AIOps Insights** includes a single scope, *All resources*. When selected, metrics are scoped to all of the IT resources monitored by IBM Cloud Pak® for AIOps. Create custom scopes to display metrics for smaller, more meaningful sets of IT resources.

Prerequisites

Creating and managing scopes requires administrative privileges. For more information, see Roles and permissions.

Procedure

Do the following to create a custom scope.

1. On the toolbar, when you first open **AIOps Insights** go to the dropdown menu with the words *All resources*. This indicates that all data that is related to Cloud Pak for AIOps is included in the charts. Click the dropdown menu.

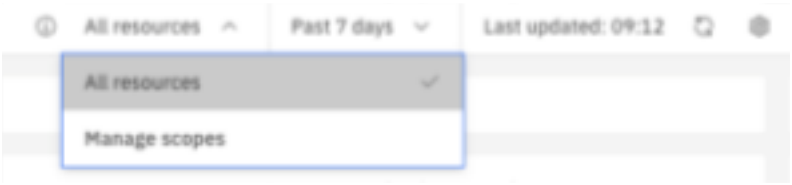


Figure. Scope dropdown

2. From the dropdown menu, select **Manage scopes**. This opens the **Scopes** panel in *AIOps insights settings*, and lists all scopes available to you. **Note**: another route to the Scopes panel is to click *AIOps Insights settings* and then click **Scopes**.

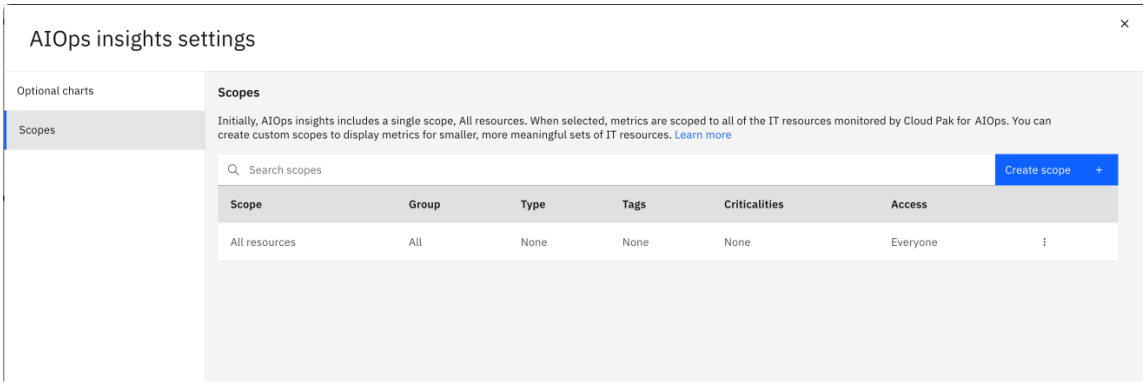


Figure. Scope panel

3. In the **Scopes** panel, click **Create scope**.
4. The *Name and description* panel opens. Enter a short title for your scope – *Banking Sales*, for example – and a description (optional), explaining what this scope shows. When completed, click **Next**.
5. In the **Resources** panel, select either *Applications* or *Resource groups*.
   - Under *Tags*, use the dropdown menu to select how to filter by selected tags:
     - For **Applications**, select either **Include applications with all of the tags selected below** or **Include applications with any of the tags selected below**.
     - For **Resource groups**, select either **Include resource groups with all of the tags selected below** or **Include resource groups with any of the tags selected below**.
   - If a tag is attached to an application or resource group, it appears under *Tag*. To find a specific tag, enter the name in the *Search tags* field.
   - Under *Tag*, select an available tag for the scope. A blue border around the tag name indicates it is selected. Click again, to deselect.
   - To scope by *Business criticalities*, click *Criticalities* and select from the criticalities you assigned to define levels of criticality, for example Tier1, Tier2, and so on. For more information on criticalities, see Defining business criticality.
6. Click **Next**.
7. Use the **Access control** panel to control who can access the scope that you are creating. You can also use this panel to enable and disable a scope, which allows you to create and store scopes for future use.

   Once the scope is enabled, use the dropdown menu to select who can access the scope. There are three choices: **Only me**, **Specified user groups** and **Everyone**.

   Use the search box to find any specified user groups who you want to allow access to this scope. Any group that is selected is displayed under *User groups*. A blue border around the user group indicates it is selected. Click again, to deselect.

   When you complete your selections, click **Create scope**.

The scope that you created is now added to the list of scopes in the main **Scopes** panel. Click the three vertical dots to edit or delete a scope. **NOTE**: You can access a description of the scope by hovering your mouse over its title. Where a number appears, hover the cursor over it, to see what this references (a tag, for example).

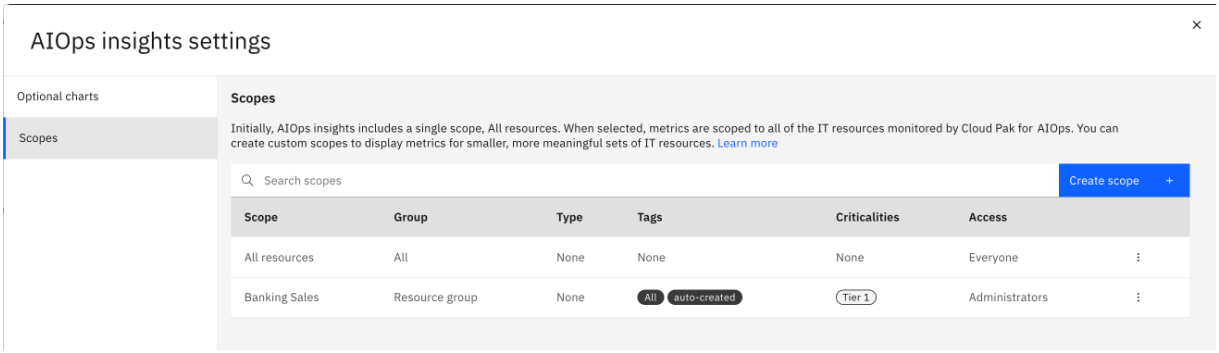


Figure. Scopes panel

On the AIOps Insights dashboard toolbar, the new scope is added to the dropdown menu. Only users who have access to this scope see it in the scopes list.

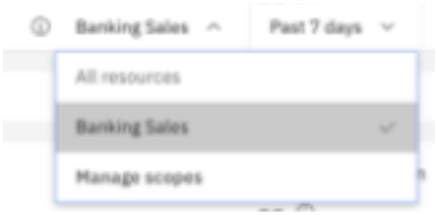


Figure. Scopes dropdown customized

Click the information button beside the scope to view a summary of the key information about the scope.

Adding a filter

Use the filter to refine by incident priority. Changes take effect immediately on the dashboard. **Note**: The filter is not applied to runbook usage metrics.

1. Click the filter icon to access the **Filter** panel.
2. Click *Incident priority* and then select one or more priority (**P**) level. A blue border around the level name indicates it is selected. To clear selections, click the level name again and the border disappears.
3. Click **Apply**.

The incident priority filters that you chose are applied to the dashboards (excluding *Runbook usage*).

Managing AI modelling

Leverage your data to build models that IBM Cloud Pak® for AIOps can use to learn patterns in your data. IBM Cloud Pak for AIOps can then apply these patterns to provide insights to your Operations and SRE teams.

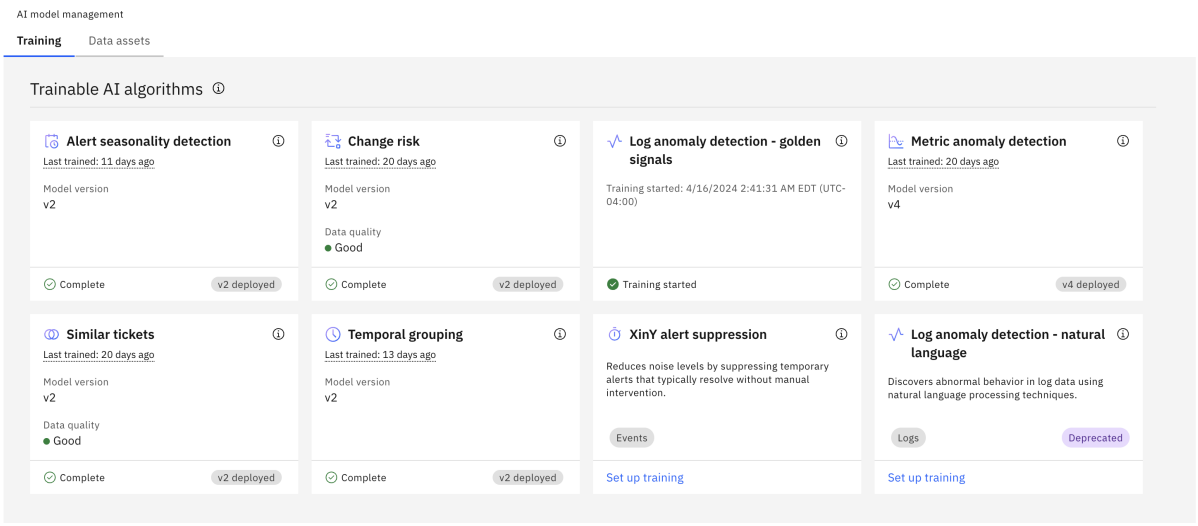


Figure. AI model management trainable AI algorithms

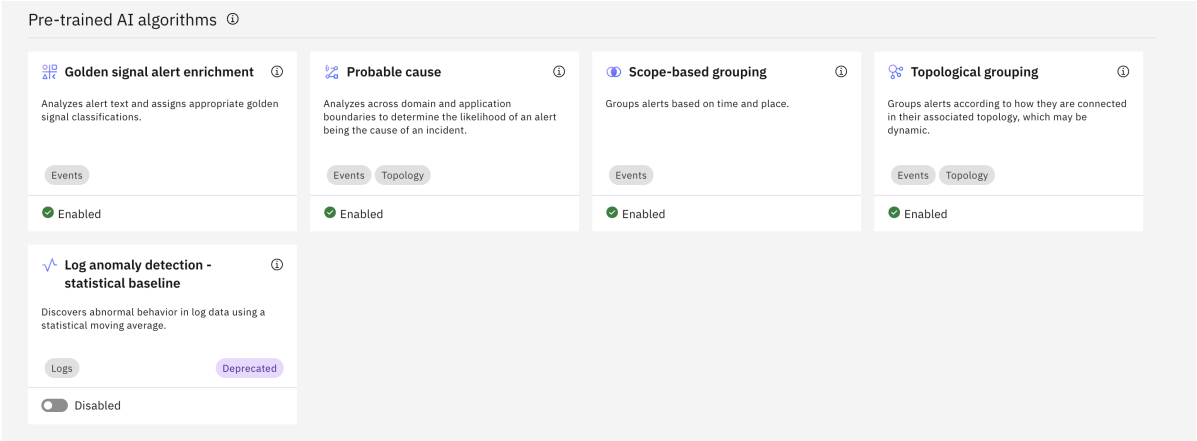


Figure. AI model management pre-trained AI algorithms

For more information about configuring AI training, see the following topics:

- Preparing the AI environment
- About AI algorithms
- Preparing AI algorithms
- Managing AI algorithms
- Displaying data assets

About AI models

IBM Cloud Pak® for AIOps provides a wide range of AI algorithms to gather data and learn from your environment, and provide insights to support resolution of health issues across your application, service, and network infrastructure.

For each of the following AI algorithms, this topic describes the algorithm, explains how it helps you, and, where configuration is necessary, points to the relevant configuration tasks.

- Alert seasonality detection
- Change risk
- Event grouping
- Log anomaly detection
- Metric anomaly detection
- Probable cause
- Similar tickets

Alert seasonality detection

This AI algorithm groups alerts that tend to occur within a seasonal time window. Periodicity of alert occurences is an indication of a common underlying cause, which might help you to diagnose the underlying cause of the alert. The model also enables you to take different actions on alerts depending on if they are occuring within the seasonal window, or outside. If a seasonal alert is considered benign when occuring within the time window, you can also suppress seasonal alerts.

To train this algorithm, you must initially configure it using the console, as described in Configuring alert seasonality detection.

When this algorithm is successfully trained and the resulting AI model deployed, alerts that occur within a seasonal time window are grouped together. Site reliability engineers (SREs) and other users responsible for application and service availability can then view the details in the Alert Viewer, as described in Displaying alert seasonality.

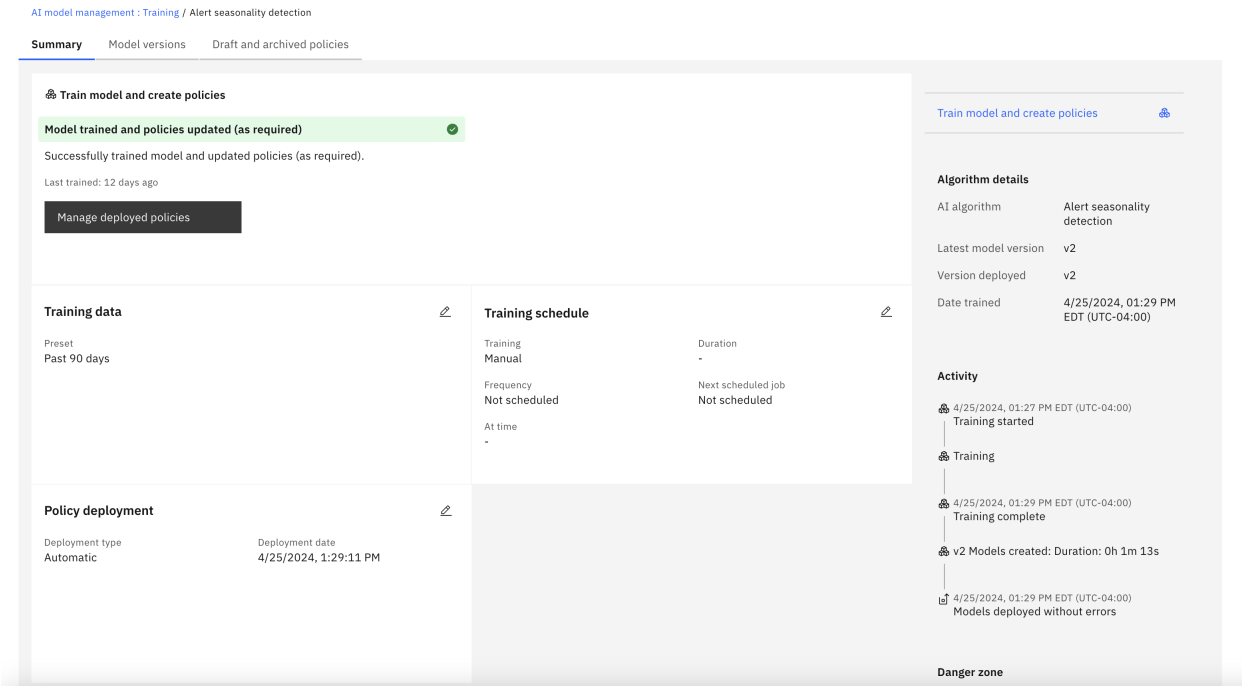


Figure. Alert seasonality detection summary

Change risk

This AI algorithm provides an assessment of the risk of implementing a proposed change; for example, a code change or a new software version.

Hundreds of changes can affect an application during its lifecycle. Change risk takes historical data about all of those changes and helps you determine how likely it is that a specific change would cause a problem, based on how successfully that similar change was deployed in the past. Using the assessment score provided by this AI model, you can determine how safe it is to proceed with the change.

Training this AI algorithm helps you ensure that risky changes for an application are assessed before deployment. To train this algorithm, you must initially configure it using the console, as described in Configuring change risk.

Once this algorithm is successfully trained and the resulting AI model deployed, then when someone opens a change request ticket, a *Change risk assessment* appears in the proactive ChatOps channel. For more details, see Change risk.

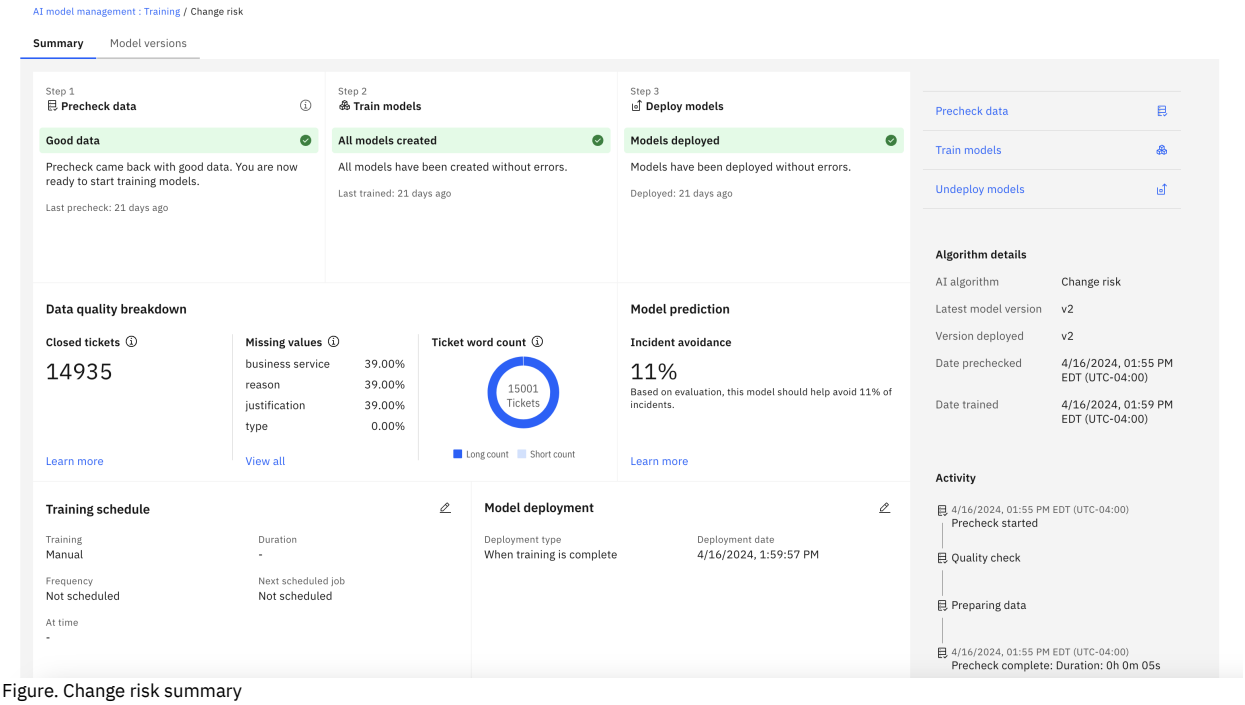


Figure. Change risk summary

Event grouping

Three AI algorithms are provided that group events together and present the groups within a single incident.

The three algorithms are the following:

- Temporal grouping
- Topological grouping
- Scope-based Grouping

Temporal grouping

This AI algorithm groups events that are discovered to co-occur over time. When a problem arises, there are typically multiple parts of a system or environment that are impacted. When events in different areas co-occur, it makes sense to look at them together and treat them as one problem to try to determine what might have happened.

Grouping co-occurring events together reduces the number of tickets and incidents opened and the number of people looking at the same problem, thereby significantly reducing noise in your monitoring systems. It helps you to understand the context of an issue so you can prioritize, triage, and resolve it more quickly.

To train this algorithm, you must initially configure it using the console, as described in Configuring temporal grouping.

When this algorithm is enabled related alerts are grouped based on when they occur, Site reliability engineers (SREs) and other users responsible for application and service availability are able to view the details in the Alert Viewer, as described in About alerts.

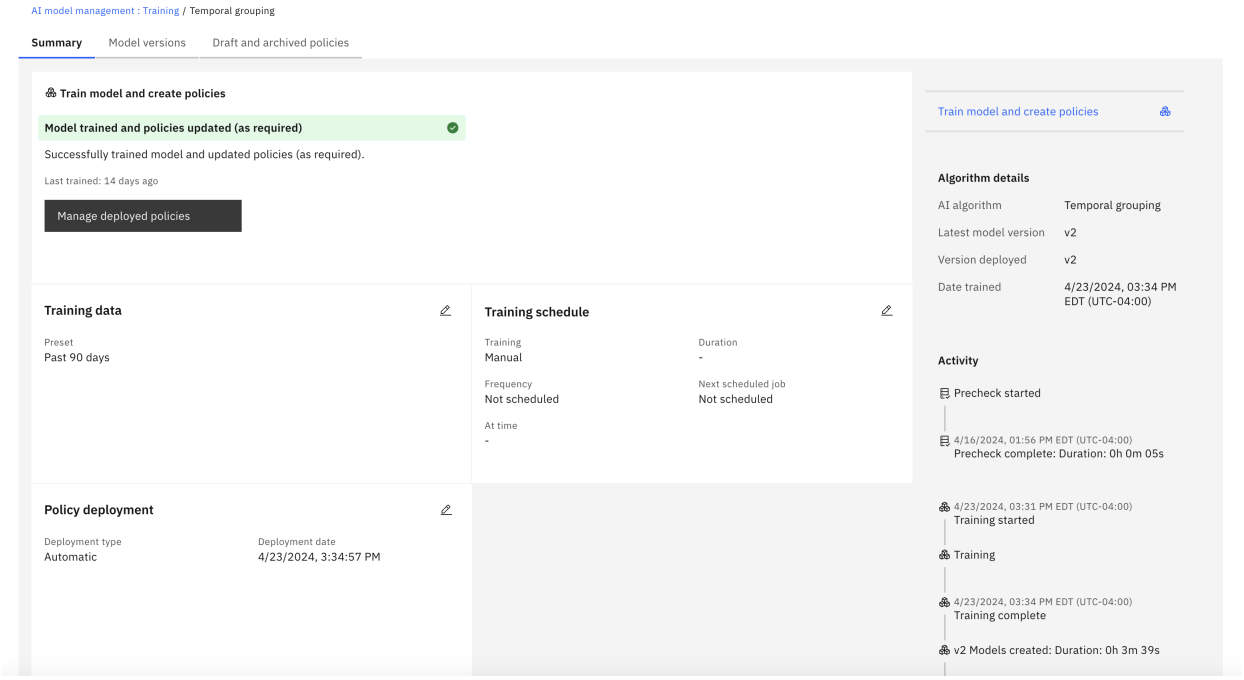


Figure. Temporal grouping summary

Topological grouping

This AI algorithm groups your events based on the resource groups in which events events occur. For example if you have a resource group made up of all the resources within a given Kubernetes namespace, then any events on pods, microservices, or other resources in that namespace will be grouped together in a single topological group.

Topological grouping helps you understand when events are connected based on their topology, providing valuable context for why related events might occur together.

This algorithm is enabled so when related events are grouped based on their topology, Site reliability engineers (SREs) and other users responsible for application and service availability will be able to view the details in the Alert Viewer, as described in About alerts.

Scope-based grouping

This AI algorithm automatically groups events relating to an incident if they have the same defined scope and occur during the same period of time. A scope can be used to identify where events originate based on a common attribute, for example, the location of a server room.

By understanding when events are related based on both time and location, you can more quickly diagnose incidents.

This algorithm is enabled so when related events are grouped based on their scope, Site reliability engineers (SREs) and other users responsible for application and service availability will be able to view the details in the Alert Viewer, as described in About alerts.

Log anomaly detection

This pair of AI algorithms gather log data from one or multiple components in the application architecture, identifies a baseline of expected log message types, and uses this baseline to discover abnormal behavior in your live log data. For more information on log anomaly detection algorithms, see About log anomaly detection algorithms.

Metric anomaly detection

Metric anomaly detection first learns normal patterns of metric behavior by analyzing metric values at regular intervals. If that behavior significantly changes, it raises anomalies or alerts.

For more information, see About metric detection algorithms.

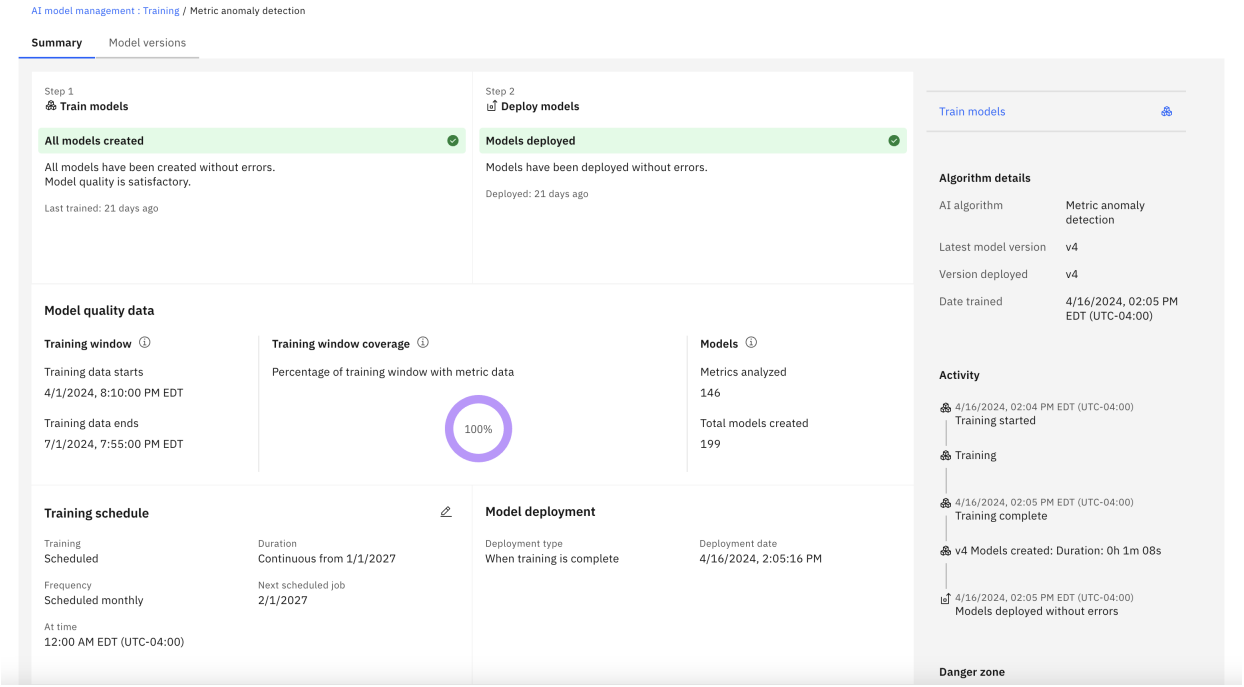


Figure. Metric anomaly detection summary

Probable cause

This AI algorithm identifies the event with the greatest probability of being the cause of an incident.

Probable cause analyzes event and topology data to understand how events are related to each other. It then uses that understanding to try to determine the root cause of a problem.

All events in an incident are given a probability score that indicates the likelihood of that event being the cause of the incident. events that occur on resource that is lower down in the topology are giving a higher probability score. events are also categorized and are scored based on the category that they fall into. This ensures that if two events occur on the same resource in the topology, the event with the more significant category will have a higher probability score.

Once the SRE has identified the probable cause event, they can take action on the resource that is generating that event and quickly resolve the incident.

The probable cause AI algorithm is enabled by default, so that information about the probable cause of a problem is automatically included in your incident details, which are surfaced in your ChatOps interface, as described in Managing incidents.

Similar tickets

This AI algorithm discovers details about similar messages, anomalies, and events within your tickets that occurred in the past and are impacting the current application.

When an incident occurs, it can be helpful to review details for similar tickets to help determine a resolution. This model aggregates information about similar messages, anomalies, and events for a component or application. This model can also extract the steps used to fix previous incidents, if documented.

Training this AI model helps you discover historical incidents to aid in the remediation of current problems. To train this algorithm, you must initially configure it using the console, as described in Setting up training for similar tickets.

Site reliability engineers (SREs) and other users responsible for application and service availability are able to access incidents in their ChatOps interface and within each incident view historical similar tickets, as described in Managing incidents.

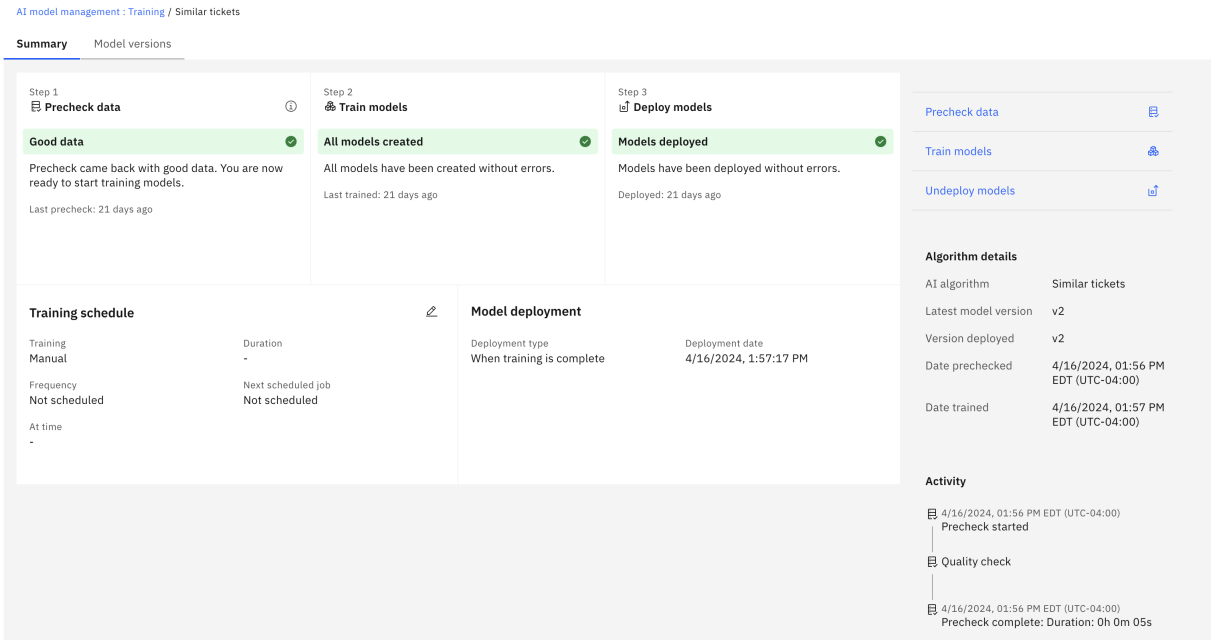


Figure. Similar tickets summary

Prerequisites for AI modelling

For each different AI algorithm, you must integrate with the right tools and gather specific types of data so the AI can begin learning.

This topic describes the prerequisites for each of the following AI algorithms.

- Change risk and Similar tickets
- Log anomaly detection
- Alert grouping, Alert seasonality detection, and Probable cause

Change risk and Similar tickets

Both the Change risk algorithm and the Similar ticket algorithm require ticket data from ServiceNow. Before attempting to configure these algorithms, ensure that you have created a functioning data integration to ServiceNow. For more information, see Creating a ServiceNow integration.

Log anomaly detection

The Log anomaly detection algorithms require log data from one or more of your log systems. Before attempting to configure these algorithms, ensure that you have created a functioning data integration to at least one of the log systems listed in the following table.

Log system	For more details, see:
Falcon LogScale	Creating Falcon LogScale integrations
Mezmo	Creating Mezmo integrations
Splunk	Creating Splunk integrations

You can also load log data into the system using one of these generic data loading methods.

Data loading method	For more details, see:
Custom	Creating custom integrations
ELK	Creating ELK integrations
Kafka	Creating Kafka integrations

Event grouping, Alert seasonality detection, and Probable cause

The three alert grouping algorithms, the Alert seasonality detection algorithm, and the Probable cause algorithm all require event data. In the case of the Topological grouping algorithm and the Probable cause algorithm, topological data is also required. This topological data must be correlated with the event data.

Event data

In some cases, your event data is loaded into IBM Cloud Pak® for AIOps from a IBM® Netcool® Operations Insight®-based data source by using the following Netcool Operations Insight integration.

Data loading method	For more details, see:
Connecting an external ObjectServer	Creating IBM® Tivoli® Netcool®/OMNIbus integrations

You can also load event data into the system by using other integration types. For more information, see Incoming integrations.

Topological data

Load topology data into IBM Cloud Pak® for AIOps by configuring observers. For more information, see Defining observer jobs.

Preparing the AI environment

Prepare the AI environment by configuring performance and other setttings for the various AI algorithms.

For more information about preparing the AI environment, see the following topics:

- Adjusting model training resources
- Configuring alert expiry time
- About Elasticsearch indices and shards
- Managing embedded JSON
- Scaling log anomaly detection pods

How to modify training resources

You can modify training resources, such as to alter requirements.

Procedure

1. Find the custom resource of the algorithm for which you wish to alter the default resource requirements by running the following command:

   ```
   oc get algorithms
   ```

2. This should return something similar to the following:

   ```
   oc get algorithms
   ```

   Output:

   ```
   NAME                AGE
   change-risk         46h
   log-anomaly         46h
   similar-incidents   46h
   ```

3. These CRs contain all the algorithm specifications. You can go ahead and edit the specific algorithm by running:

   ```
   oc edit algorithm <name>
   ```

4. Then you decode the *spec.manifestBase64* field. Once decoded you see a resources field defined as follows:

   ```
   resources:
      requests:
         cpu: 2
         memory: 4Gi
      limits:
         cpu: 2
         memory: 4Gi
   ```

   Note: You can edit requests and limits (For training algorithms, make sure that the **limits=requests**), re-encode to base64, and then replace the *manifestBase64* field in the CR with this newly encoded manifest.

5. Save the changes, and proceed with your training.

Configuring expiry time for log anomaly alerts

By default, alerts that are generated by a log anomaly expire in six hours (21,600 seconds). You can modify this value.

1. Access the IBM Cloud Pak Administration panel by following the instructions at UI console.
2. In the main navigation menu, click **Workloads** > **Deployments**.
3. Search by name, and type **log-anomaly** in the field. The **aimanager-aio-log-anomaly-detector** deployment is listed in the table.
4. Click **aimanager-aio-log-anomaly-detector** to visualize deployment details.
5. Click the **Environments** tab. A list of environment variables is displayed.
6. From the list, click **Add more** to add another environment variable.
7. Name the environment variable **EXPIRY_SECONDS**.
8. Set the value of the **EXPIRY_SECONDS** environment variable in seconds. For example, if you want to increase the expiry time to 24 hours, which is 86,400 seconds, set the value to `86400`.
9. Click **Save** to save the changes. This action triggers a restart of the **aimanager-aio-log-anomaly-detector** pod, which takes a few minutes.

After the pod restarts, the new expiry time applies to any alerts that are generated by a log anomaly.

About indices and shards

When setting up AI training, a critical factor is the number of shards available to support the creation of Elasticsearch indices. By default, a production installation provides 1000 shards for use by the various AI training types and by the AI platform itself.

Each Elasticsearch index requires a certain number of shards. If the number of Elasticsearch indices used by the various AI training components requires more than the default of 1000 available shards, then some of the AI training might fail. A training failure returns a **Needs improvement** data qualtiy level, as described in Generating an AI model. In order to avoid training failure, you can do one or more of the following:

To avoid the possibility of running out of shards, a periodic cron job utility runs every hour to delete all the logtrain elastic indices that have an index creation date older than 14 days from the current timestamp.

Determining your current shard consumption

Monitor the number of indexes and shards currently in use so that you can plan your AI training accordingly.

About this task

When log data is collected for IBM Cloud Pak® for AIOps, the data is persisted within Elasticsearch for training purposes. While the collection of historical or live data for training is occurring or after the data collection completes, you can connect to Elasticsearch to check the progress and look at the indexes.

Connect to Elasticsearch

1. Log in to your Red Hat® OpenShift® cluster, or SSH into a control plane node or the loadbalancer VM on your Linux® deployment. Then swtich to the namespace where IBM Cloud Pak for AIOps is installed.

   ```
   oc project <project>
   ```

   Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in.

2. Run the following command to list the indices stored in Elasticsearch. Look for index files that have a **1000-1000-<date>-logtrain** name format, such as **1000-1000-20210510-logtrain**.

   ```
   export ELASTIC_SET="$(oc get statefulset -l app.kubernetes.io/managed-by=ibm-elasticsearch -o
   jsonpath='{.items[0].metadata.name}')"
   oc exec "statefulset/${ELASTIC_SET}" -- curl -XGET -k --fail "http://127.0.0.1:19200/_cat/indices?v" | sort
   ```

   Example output, where the index name is **1000-1000-20220302-logtrain**:

   ```
   green open 1000-1000-20220302-logtrain        eEvaBufQ06yG074hmeHKG  8 1 244 6 72.7kb 72.7kb
   ```

   The following table lists the elements in this line of output:

   Table. elements

Element	Description	Example
Index health	Health of the index.	green
Index Status	Status of the index.	open
Index name	Name of the index.	1000-1000-20220302-logtrain
UUID	Universal unique identifier for this index.	eEvaBufQ06yG074hmeHKG
Number of shards	Number of shards required by default for this index. This value varies based on the type of index. For example, an index that is associated with a day of ingested log data for natural language log anomaly training requires 8 shards for both a large and a small installation.	8
Number of replicas	Number of replicas required by default for this index. This value varies based on the type of index and the size of the installation. For example, an index that is associated with a day of ingested log data for natural language log anomaly training requires 1 replica for a large installation and 0 replicas for a small installation. A replica value of 1 replicates all of the shards for that index, meaning that a shard replica setting of 8/1 assigns 16 shards to the index.	1
docs.count	Detailed statistic associated with this index.	244
docs.deleted	Detailed statistic associated with this index.	6
Store size	Detailed statistic associated with this index.	72.7 kb
Primary store size	Detailed statistic associated with this index.	72.7 kb

3. Check the count of the records for an index by using the following command:

   ```
   export INDEX_NAME=<index name>
   export ELASTIC_SET="$(oc get statefulset -l app.kubernetes.io/managed-by=ibm-elasticsearch -o
   jsonpath='{.items[0].metadata.name}')"
   oc exec "statefulset/${ELASTIC_SET}" -- curl -XGET -k --fail "http://127.0.0.1:19200/${INDEX_NAME}/_count"
   ```

   Where **<index_name>** is the name of an index returned by the previous step

   Example output:

```
{"count":318713,"_shards":{"total":1,"successful":1,"skipped":0,"failed":0}}
```

4. You can also look at sample data by using the following command:

```
export INDEX_NAME=<index name>
export ELASTIC_SET="$(oc get statefulset -l app.kubernetes.io/managed-by=ibm-elasticsearch -o
jsonpath='{.items[0].metadata.name}')"
oc exec "statefulset/${ELASTIC_SET}" -- curl -XGET -k --fail "http://127.0.0.1:19200/${INDEX_NAME}/_search"
```

Where **<index_name>** is the name of an index returned by the previous step.

Shard consumption in a large installation

Use the following table to help interpret the results returned. All values that are provided are for a production (large) installation.

Table. shard consumption

N.	AI component	Index name	Shard usage	Notes
	Natural language log anomaly detection (total)		Varies	
1	Each day of ingested log data	`1000-1000-`*date*`-logtrain`	16	The index for each extra day of data uses 16 shards.
2	Each model version total	`1000-1000-v_`x-contenttype	50	The index for each extra model version uses 50 shards.
3	Dedicated index for saving training data	`1000-1000-windowed_logs`	5	
4	Dedicated index for deploying models	`1000-1000-log_modles_latest`	5	
	Statistical baseline log anomaly detection (total)		20	
5	Statistical baseline log anomaly detection	`1000-1000-log_models_latest`	5	
6	Statistical baseline log anomaly detection	`1000-1000-reference_embedding`	5	
7	Statistical baseline log anomaly detection	`1000-1000-reference_oob`	5	
8	Statistical baseline log anomaly detection	`1000-1000-si_models_latest`	5	
9	**Change risk**	`1000-1000-cr_models_latest`		
10	**Similar tickets**	`1000-1000-si_models_latest`	10	
	Lifecycle services		Varies	
11	Each service	`aiops-searchservice-v10` *datetime*	5	The index for each extra lifecycle service uses 5 shards.
	AI platform		12	
12	AI platform	`postchecktrainingdetails0`	2	
13	AI platform	`prechecktrainingdetails0`	2	
14	AI platform	`trainingdefinition`	2	
15	AI platform	`trainingrun`	2	
16	AI platform	`trainingrunning`	2	
17	AI platform	`trainingstatus`	2	

Best practices

An analysis of the table in Shard consumption in a large installation shows that the most effective way to keep shard usage down is to follow these practices:

- Train natural language log anomaly detection on a smaller number of days. As you can see from the table row that is labeled 1, the index for each extra day of data uses 16 shards. The recommended number of days for training is 14.

  Note To avoid the possibility of running out of shards, a periodic cron job utility runs every hour to delete all the logtrain elastic indexes that have an index creation date older than 14 days from the current timestamp.

- Regularly delete new versions of natural language log anomaly detection models. As you can see from the table row that is labeled 2, the index for each extra model version uses 50 shards.

- When you define a log integration, such as for Elasticsearch, Logstash, and Kibana (ELK), Humio, or Mezmo, the Log Ops UI in IBM Cloud Pak for AIOps doesn't show the start and end date time zone when you use the **Historical data for initial AI training** data flow mode. You can't use the Log Ops UI to enter the start and end times. Instead, you can specify only the start and end dates. The start and end date time zone is Greenwich mean time (GMT).

Managing embedded JSON

Log lines containing embedded JSON might not be parsed correctly by the natual language log anomaly detection algorithm. If your log lines contain embedded JSON, then apply the methods described here to convert your log lines to unstructured data.

The following snippet shows an example of a log line containing embedded JSON.

```
021-09-23T21:43:40.067+0000  I NETWORK  [conn5083706]  received client metadata from 127.0.0.1:49736  conn5083706: {
name: \"PyMongo\",  version: \"3.8.0\", type: \"Linux\", architecture: \"x86_64\",  version: \"3.10.0-
1160.25.1.el7.x86_64\", platform: \"CPython3.7.3.final.0\" }
```

The braces (curly brackets) and the section between those braces is embedded JSON.

You can handle these log lines by applying one of the following options:

- **None**: The default option. The JSON is not processed or modified. This option results in the example JSON resembling the following code:

  ```
  021-09-23T21:43:40.067+0000  I NETWORK  [conn5083706]  received client metadata from 127.0.0.1:49736  conn5083706:
  { name: \"PyMongo\",  version: \"3.8.0\", type: \"Linux\", architecture: \"x86_64\",  version: \"3.10.0-
  1160.25.1.el7.x86_64\", platform: \"CPython3.7.3.final.0\" }
  ```

- **Flatten**: This option flattens the JSON object by removing the opening and closing braces. This option results in the example JSON resembling the following code:

  ```
  021-09-23T21:43:40.067+0000  I NETWORK  [conn5083706]  received client metadata from 127.0.0.1:49736  conn5083706:
  name: \"PyMongo\",  version: \"3.8.0\", type: \"Linux\", architecture: \"x86_64\",  version: \"3.10.0-
  1160.25.1.el7.x86_64\", platform: \"CPython3.7.3.final.0\"
  ```

- **Filter**: This option extracts the JSON object and replaces it with an empty string. This option results in the example JSON resembling the following code:

  ```
  021-09-23T21:43:40.067+0000  I NETWORK  [conn5083706]  received client metadata from 127.0.0.1:49736  conn5083706:
  ```

  Note: Examine your data before applying this option, especially if your log lines are complete JSON objects.

Which option to select

Perform the following steps to get an idea of which option to select:

1. Retrieve a sample of your log data of at least 30,000 lines.
2. In your sample, determine the percentage of log lines that contain either JSON objects, or lines made up of both text and JSON objects.
3. Use the following table to help you select an option.

JSON percentage in data	Suggested option
Less than 30% of log lines contain JSON objects	**none**
More than 30% of log lines contain JSON objects	**filter**
More than 80% of log lines contain JSON objects	**none** or **flatten**

Limitations

Be aware of the following limitations:

1. Neither the **flatten** nor **filter** option checks the validity of the JSON object.
2. In some cases log data can include partial JSON. In this case, the **filter** options will extract the message preceding the first opening brace of the partial JSON object.

Selecting an option

To select a JSON processing option, see Adding an integration.

Scaling log anomaly detection pods

You can scale up the number of log anomaly detection pods for your instace.

Each `log-anomaly-detection` replica supports roughly 10 components. By default, a starter deployment has 2 `log-anomaly-detection` pods, while a production deployment has 4.

To scale the replicas to handle processing for additional components, modify the `AIManagerMainProd` custom resource.

Before you begin

Clusters can be for starter or production deployments, as described in Hardware requirements. A starter deployment cluster has limited resources, so you might want to check your cluster resources before you complete the following task to ensure that you are able to allocate adequate resources.

Procedure

1. Log in to your cluster by running the `oc login` command.

   ```
   oc login -u kubeadmin -p <password>
   ```

   For more information, see Logging in to the OpenShift CLI.

2. Edit the IBM Cloud Pak for AIOps customer resource. The following command opens the file in the `vi` editor for editing.

   ```
   oc edit AIManagerMainProd aimanager
   ```

3. Search for the following configuration:

   ```
   spec:
     helmValues:
       global:
         logAnomaly:
           replicas: 2
   ```

4. Modify the replica count, such as to change the number of replicas to `4`.

   ```
   spec:
     helmValues:
       global:
         logAnomaly:
           replicas: 4
   ```

   Note: Each instance of the `log-anomaly-detector` pod requires an additional 0.5 CPU cores and 0.5Gi of memory.

5. Save the file.

6. Restart the `ai-platform-api-server` pod to put the changes into effect. Restart the pod by deleting the associated pod. A new pod automatically starts up and uses the new parameters that are specified in the configmap.

   1. First, determine the name of the currently running pod.

      ```
      oc get po | grep ai-platform-api-server
      ```

      Your output can resemble the following sample output:

      ```
      aimanager-aio-ai-platform-api-server-7d75768cbc-zx5dh              1/1     Running     0          6d1h
      ```

      Where,

      - The string `aimanager-aio-ai-platform-api-server-7d75768cbc-zx5dh` is the pod ID. The character string `7d75768cbc-zx5dh` uniquely identifies the pod. This string is the value for `<unique pod reference>` in the following steps.
      - The other information indicates that the pod is running and has been running for 6 days and 1 hour.

   2. Delete the pod by running the following command:

      ```
      oc delete po aimanager-aio-ai-platform-api-server-7d75768cbc-zx5dh
      ```

   3. Wait a short time, then run the following command to check that the new pod started up, and to show the new pod ID.

      ```
      oc get po | grep ai-platform-api-server
      ```

Configuring log formats for WebSphere log anomaly detection

To access log messages for WebSphere log anomaly detection, configure the domains with the correct type of log formats.

- Configure IBM WebSphere Application Server to use High Performance Extensible Logging (HPEL).
  - HPEL provides a mechanism for storing and accessing log, trace, `System.err`, and `System.out` information that is generated by the server or generated by applications that use `java.util.logging`.
  - To render data stored by HPEL in JSON format, use the LogViewer command.
- Configure IBM WebSphere Liberty servers to output JSON to the `messages.log` file.
- Only log messages from IBM WebSphere Liberty server code have IBM message IDs, so IBM Cloud Pak for AIOps can use only those messages. IBM Cloud Pak for AIOps can't run log anomaly detection on logs from applications that run on IBM WebSphere Application Server.
  - Thus, the upstream log data source for IBM Cloud Pak for AIOps is either HPEL for IBM WebSphere Application Server or a `messages.log` file for an IBM WebSphere Liberty server.

About AI algorithms

AI algorithms analyze different kinds of data and produce AI models, which automate research, reducing the need for manual work in solving problems. These unsupervised algorithms enable the use of unlabeled data for training AI models without any human intervention. IBM Cloud Pak® for AIOps training generates models for a number of different AI algorithms. You can train the models for each of these AI algorithms independently of each other.

You use the AI Model Management to set up training for various AI algorithms.

The AI Model Management has these tabs: **Training** and **Data assets**.

1. The **Training** tab allows you to view two types of algorithms in tile format:

   - **Trainable AI algorithms** These algorithms must be trained on certain data types before they can create deployable models. They include change risk, log anomaly detection-natural language, metric anomaly detection, similar tickets, and temporal grouping. These algorithms are listed as tiles. You can set up the training for each algorithm separately.

- **Pre-trained AI algorithms** These algorithms don't require models to derive insights. They are enabled by default. These algorithms include probable cause, scope-based grouping, topological grouping, and log anomaly detection-statistical baseline. Again each of these algorithms is represented by a tile.

2. The **Data assets** tab shows the data integrations that are feeding your AI training. It lists them in a table and gives details about each integration, such as Integration name, Data flow status (Running or Finished), Integration type, Data type, and Data collected (Live or Historical).

Note: It is worth noting that naming the training is an optional step in setting up training for trainable AI algorithms.

For more information about AI algorithms, see the following topics:

- About change risk
- About event grouping
- About log anomaly detection algorithms
- About metric anomaly detection
- About probable cause
- About similar tickets

About change risk

This AI algorithm provides an assessment of the risk of implementing a proposed change; for example, a code change or a new software version.

Hundreds of changes can affect an application during its lifecycle. *Change risk* is an unsupervised learning algorithm that takes historical data about all of those changes and helps you determine how likely it is that a specific change would cause a problem, based on how successfully that similar change was deployed in the past.

Using the assessment score provided by this AI model, you can determine how safe it is to proceed with a change.

To train this algorithm, you must complete some setup tasks, as described in Setting up training for change risk.

When a *Change risk* algorithm is successfully trained and the resulting AI model deploys, if someone opens a change request ticket, a *Change risk* assessment appears in the proactive ChatOps channel (for more information, see Change risk in ChatOps). When the assessment is deemed to be high risk, then an alert is generated and is displayed in the Alert Viewer.

Note: Some training features are specific to the change risk algorithm. For more information, see Managing algorithms.

Prerequisites

To train a change risk algorithm, you need change request and incident ticket data from ServiceNow. Starting with IBM Cloud Pak® for AIOps version 4.1.0, similar incidents, such as from a ServiceNow integration, are called similar tickets.

Before you set up training for this algorithm, create a functioning data integration to ServiceNow. For more information, see Creating a ServiceNow integration.

Note: If you enable collection of historical data when you integrate with ServiceNow, then data for the selected timeframe is available to train on.

To provide the best data for training, you need a mix of successful and failed change tickets. Around 10,000 change tickets, with a minimum 2% failure rate, is considered ideal. However, models can also be created with much less data.

Language support

For information about supported languages for this algorithm, see Language support.

About event grouping

Three AI algorithms are provided with IBM Cloud Pak for AIOps that group events together and present the groups within a single incident.

Event grouping AI algorithms:

- Temporal grouping
- Topological grouping
- Scope-based Grouping

Temporal grouping

This AI algorithm is an unsupervised learning algorithm that groups alerts that are discovered to co-occur over time. When a problem arises, there are typically multiple parts of a system or environment that are impacted. When alerts in different areas co-occur, it makes sense to look at them together and treat them as one problem to try to determine what might have happened.

Grouping co-occurring alerts together reduces the number of tickets and opened incidents, and the number of people looking at the same problem, thereby significantly reducing noise in your monitoring systems. It helps you to understand the context of an issue so you can prioritize, triage, and resolve it more quickly.

To train this algorithm, complete some initial setup tasks. This AI algorithm also needs at least three months of historical event data to learn alert patterns and discover groups. For more information, see Setting up training for temporal grouping.

Just like alert seasonality detection algorithm, no configuration is required to activate temporal grouping. The model training needs to be configured and run.

When the Temporal grouping algorithm is enabled related alerts are grouped based on when they occur, Site reliability engineers (SREs) and other users responsible for application and service availability are able to view the details in the event Viewer, as described in About alerts.

Topological grouping

This AI algorithm is an unsupervised learning algorithm that groups alerts based on the resource groups in which those alerts occur. For example if you have a resource group made up of all the resources within a given Kubernetes namespace, then any alerts on pods, microservices, or other resources in that namespace will be grouped together in a single topological group.

Topological grouping helps you understand when alerts are connected based on their topology, providing valuable context for why related alerts might occur together.

When this algorithm is enabled related alerts are grouped based on their topology, site reliability engineers (SREs) and other users responsible for application and service availability are able to view the details in the Alert Viewer, as described in About alerts.

Notes:

1. Topology resources in the same group do not need to be connected for alert grouping to happen.
2. Alerts that target topology resources in the same application (service) are also grouped.
3. Alerts must occur within a 15-minute time window to get correlated.

Topological grouping requires that there be a topological group creator.

Scope-based grouping

This AI algorithm is an unsupervised learning algorithm that automatically groups alerts relating to an incident if they have the same defined scope and occur during the same period of time. A scope can be used to identify where events originate based on a common attribute, for example, the location of a server room.

By understanding when alerts are related based on both time and location, you can more quickly diagnose incidents.

This algorithm is enabled so when related alerts are grouped based on their scope, Site reliability engineers (SREs) and other users responsible for application and service availability will be able to view the details in the Alert Viewer, as described in About alerts.

Prerequisites

The three event grouping algorithms require event data. In the case of the Topological grouping algorithm, topological data is also required. This topological data must be correlated with the event data.

For correlations, they must exist within the same topological group or template.

Event data

In most cases your event data will be loaded into IBM Cloud Pak® for AIOps from a IBM® Netcool® Operations Insight®-based data source, by using one of the methods listed in the following table.

Data loading method	For more details, see:
Connecting on-premises probes	Integrating on-premises probes to the in-cluster ObjectServer
Connecting an external ObjectServer	Connecting an external ObjectServer into the system by using an XML Gateway

You can also load event data into the system by using the generic custom and Kafka methods.

- For more information on the custom integration, see Creating custom integrations.
- For more information on the Kafka integration, see Creating Kafka integrations.

Topological data

Load topology data into IBM Cloud Pak® for AIOps by configuring observers. For more information, see Defining observer jobs.

Language support

For information about supported languages for this algorithm, see Language support.

About log anomaly detection algorithms

IBM Cloud Pak® for AIOps processes incoming logs as part of the log anomaly detection process.

There are two log anomaly detection AI algorithms, each of which can run independently of the other. If both algorithms are enabled, then any log anomalies discovered by both will be reconciled, so that only one alert is generated. In this case, the severity of the combined alert will be equal to the highest severity of the two alerts.

Site reliability engineers (SREs) and other users responsible for application and service availability are able to display log anomalies as alerts within the context of an incident, as described in Viewing relevant events. They will also be able to view these log anomaly alerts in the Alert Viewer, as described in About alerts.

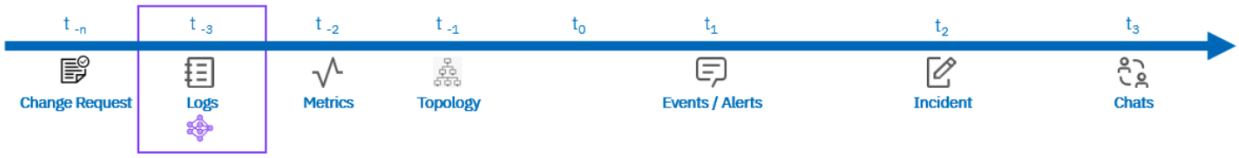


Figure. Log anomaly detection algorithm

The preceding diagram highlights the machine learning models that are used within Cloud Pak for AIOps. The models utilize advanced algorithms to analyze data from various sources, including logs, metrics, topology, and alerts. The analyzed data is used to detect anomalies, predict changes, and provide real-time insights.

The primary goal of these Cloud Pak for AIOps machine learning models is to enable IT teams to proactively identify and resolve issues before these issues impact the business. The advanced language models, deep NLP, and semantic parsing enable IT teams to extract and summarize incident information quickly and accurately. The machine learning models identification of patterns and trends enables IT teams to quickly identify the root cause of the issues, take corrective actions to prevent downtime, help ensure optimal system performance, and security, ultimately leading to improved efficiency, reliability, and security of IT operations.

With Cloud Pak for AIOps policies, IT operators can automatically assign runbooks to events based on pre-configured conditions. They can also identify the root cause and scope of affected components using blast radius and fault localization. Additionally, operators can execute assigned runbooks automatically or manually using operator-configurable policies.

Cloud Pak for AIOps policies automate routine tasks and provide real-time system performance visibility, enable teams to streamline workflows, enhance incident response, reduce downtime, and boost overall efficiency.

Prerequisites

Log data

Logs are sent from an application environment to a logging system, such as Mezmo or Splunk, through data shippers, such as Logstash. Each logging system has a mapping definition in Cloud Pak for AIOps.

The Log anomaly detection algorithms require log data from one or more of your log systems. Before attempting to set up training for these algorithms, ensure that you have created a functioning data integration to at least one of the log systems listed in the following table.

Table. Log system details

Log system	For more details, see:
Falcon LogScale	Creating Falcon LogScale integrations
Mezmo	Creating Mezmo integrations
Splunk	Creating Splunk integrations

You can also load log data into the system using one of these generic data loading methods.

Table. Data loading method details

Data loading method	For more details, see:
Custom	Creating custom integrations
ELK	Creating ELK integrations
Kafka	Creating Kafka integrations

Cloud Pak for AIOps has various integrations with log aggregators such as ELK (Elasticsearch, Logstash, Kibana), Falcon LogScale, Mezmo, Splunk, and comes with a REST service for custom integrations. These integrations connect to the source and pull the log data. If you do not have a log aggregator that is compatible with Cloud Pak for AIOps, there is an alternative option that utilizes Kafka-based integration. The Kafka-based integration also supports log processing where historic logs are fed in offline mode. Additional time needs to be allocated for planning and development in this scenario, since custom logic might be needed to parse and transform the logs in to a format that is expected by Cloud Pak for AIOps.

Performance tuning

Before training can occur, logs are pulled from a logging system, such as Mezmo or Splunk, and are stored in Elasticsearch.

Before attempting to set up training for these algorithms, review the performance tuning tasks at the following location: Setting up training for log anomaly detection.

Language support

For information about supported languages for this algorithm, see Language support.

Algorithms

The algorithms are as follows:

- Natural language
- Statistical baseline
- Golden signals

Best practices

Log anomaly detection algorithms are useful when you have logs that represent a healthy system. When Cloud Pak for AIOps is trained on a healthy system, anomalous problems can then be detected. For instance, if the infrastructure or application logs are accurate real-time representations of system health, then log anomaly alerts can help to minimize the mean time to error detection, which includes the early warning signs of potential problems that might occur later.

Log anomaly detection is resource-intensive, and each log anomaly algorithm has different data requirements. You are recommended to limit log analysis to your infrastructure and applications that are considered business critical to ensure that you do not see hardware requirements increase significantly.

About log anomaly detection - natural language

Natural language log anomaly detection is an unsupervised learning algorithm that uses Natural Language Processing (NLP) on a subset of your log data to discover abnormal behavior.

Deprecated: The natural language log anomaly detection algorithm is to be deprecated in a future release. The recommended action is to use log anomaly detection - golden signals as a replacement.

Log anomaly detection takes large amounts of log data and trains on it to learn what is normal behavior for a particular component. This model goes beyond just looking at error states or frequency of metadata around log messages. Instead, it determines when something becomes an anomaly compared to what patterns it typically exhibits.

To train this AI algorithm, you must complete some initial setup tasks, as described in Setting up training for log anomaly detection - natural language

For more information about natural language log anomaly detection, see the following topics:

- Workflow
- Examples

Workflow of natural language log anomaly detection

Find out how natural language log anomaly detection processes your log data to generate log alerts.

The workflow of natural language log anomaly detection is made up of two parts:

- Model generation
- Application of the model to the live log data stream

Model generation

The following figure shows a simplified workflow illustrating how the natural language log anomaly detection algorithm generates an AI model from log data. This part of the workflow is known as **training**.

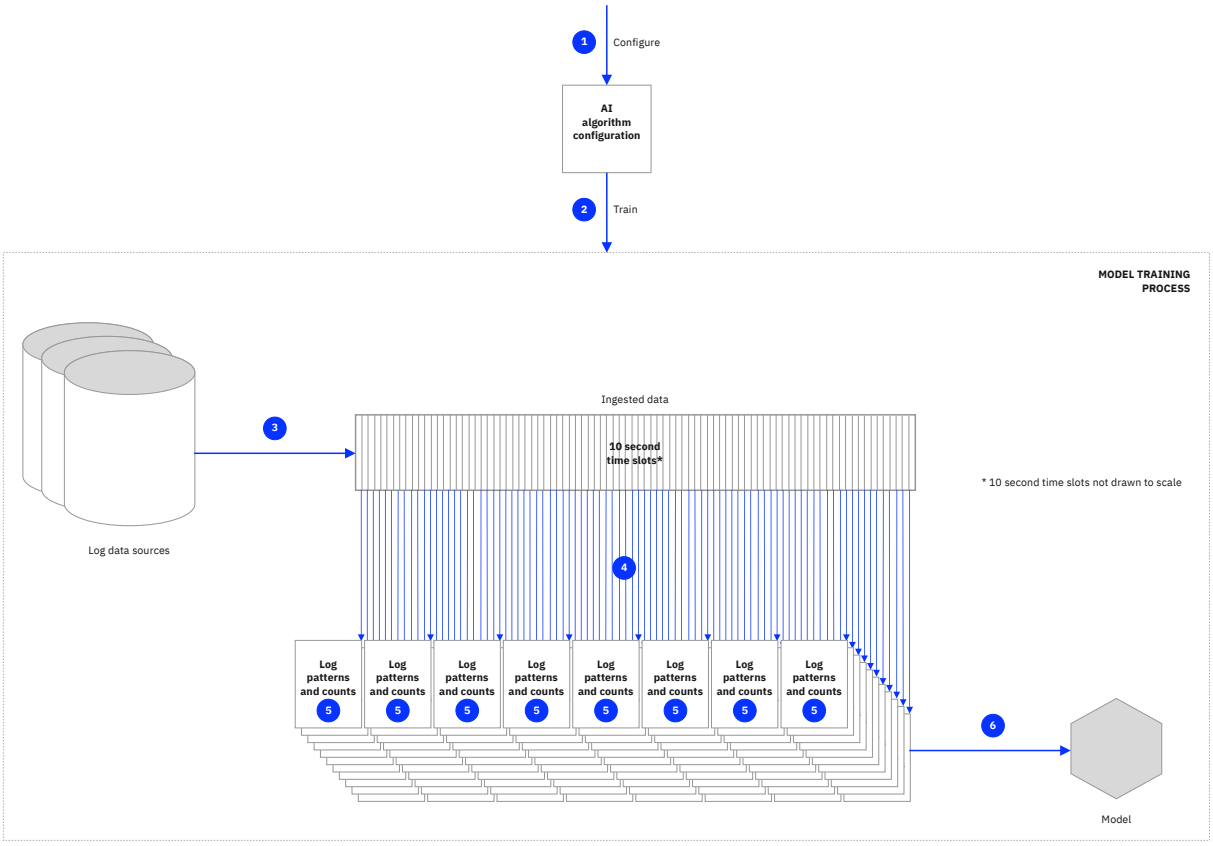


Figure. Workflow

1. **Algorithm is configured and trained**

   The algorithm configuration includes specification of log data sources, the period of data to train on, whether to run training on a schedule or on demand, and whether to deploy the training automatically or manually. Training is started and the model training process begins, as described in steps 2 to 6.

2. **Training jobs are identified**

   Assume that you have a microservice that is made up of multiple components such as orders, catalog, front-end, and more, each of these components with its own set of logs. As it is likely that each of the components has different logging behavior, the log anomaly detection algorithm creates a separate model for each component. Each of these models is trained by a separate training job.

   The logging data for the different components is differentiated by using the **`instance-id`** parameter that is specified during log integration mapping. For an example of log integration mapping for a Mezmo log system, see Specifying field mapping.

   The following steps 3 to 6 explain the process for training a unique model for each component. The steps are repeated for each component identified.

   Note: For more information about limits and throughput rates, see Supported resource number and throughput rates.

3. **Data is imported**

   Based on the specification of log data sources in the algorithm configuration, data is imported from log data sources for a given component.

4. **Data is divided into 10-second slots**

   The imported data is divided into 10-second time slots. For example, if two weeks' worth of data was impoirted, then that data is divided into x time slots based on the following calculation.

   2 weeks = 14 days = 336 hours = 20,160 minutes = 120,960 time slots

5. **Log patterns are identified and counted**

   Within each time slot, the system identifies and counts log patterns. Log patterns are identified in the following way:

   1. Each log message is separated into variant and invariant parts. For example, consider the following log messages:

```
2021-03-12T03:05:31.608355778+00:00 stdout F
{
  "name":"@instana collector",
  "__in":5,
  "hostname":"cart-846b9595c9qcqfk",
  "pid":35261,
  "module": "announceCycle/agentHostLookup",
  "level":2,
  "msg":"Agent cannot be contacted through system ABC nor through default gateway 123. Scheduling reattempt 0
}
```

```
2021-03-12T03:05:33.608355778 +00:00 stdout F
{
  "name":"@instana collector",
   "__in":5,
   "hostname":"cart-846b9595c9qcqfk",
   "pid":89561,
   "module": "announceCycle/agentHostLookup",
   "level":2,
   "msg":"Agent cannot be contacted through system DEF nor through default gateway 456. Scheduling reattempt 1
 }

2021-03-12T03:05:35.321654987+00:00 stdout F
{
 "name":"@instana collector",
 "__in":5,
 "hostname":"cart-846b9595c9qcqfk",
 "pid":75421,
 "module": "announceCycle/agentHostLookup",
 "level":3,
 "msg":"Agent cannot be contacted through system GHK nor through default gateway 789. Scheduling reattempt 2
}
```

These log messages can be expressed by using the following log pattern, where the variant parts of the log messages are expressed by using wildcards, such as **<*>** and **<NUM>**.

```
<*> stdout F
{
  "name":"@instana collector",
  "__in":<NUM>,
  "hostname":"cart-846b9595c9qcqfk",
  "pid":<NUM>,
  "module": "announceCycle/agentHostLookup",
  "level":<NUM>,
  "msg":"Agent cannot be contacted through <*> nor through default gateway <*>. Scheduling reattempt <NUM>
}
```

2. Log patterns are counted in each 10-second slot and means and standard deviations are stored for each log pattern.

6. **Model is created**

   On completion of training, a model is created that includes log pattern statistics.

   Note: Each time that the algorithm is run, steps 2 to 5 are repeated from scratch and a new version of the model is generated.

Application of the model to the live log data stream

The following figure shows a simplified workflow illustrating how the natural language log anomaly detection algorithm applies the AI model to the live log data stream. This part of the workflow is known as **inference**, and it involves applying the findings from the model.

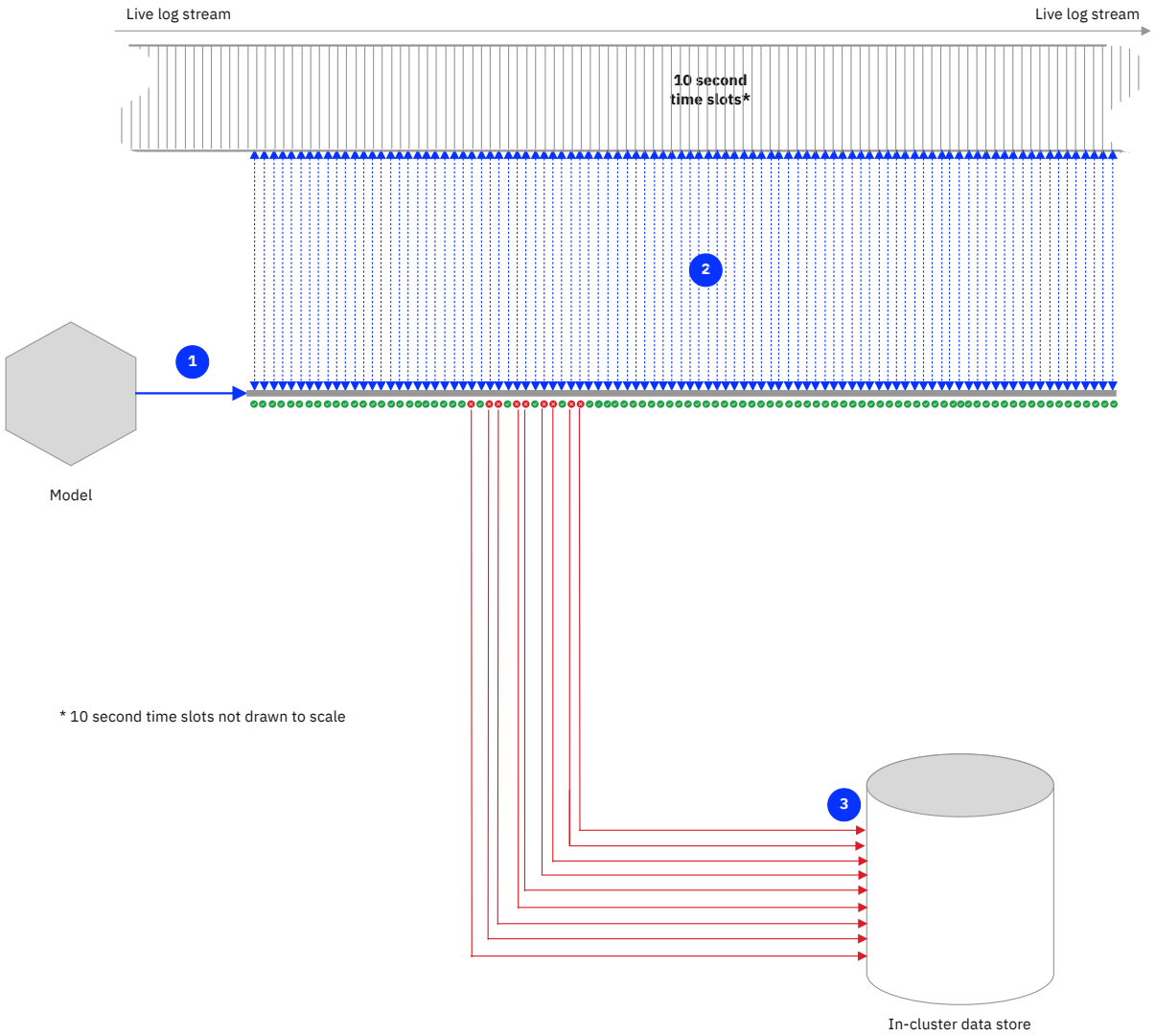


Figure. Workflow model generation

1. **Model is deployed**

   Depending on how the algorithm was configured a new version of the model is deployed automatically on completion of training, or manually.

2. **Review live stream for anomalies**

   Now that you have a deployed model, the system starts monitoring the live log stream for anomalies, which are statistically significant deviations from the baseline. It evaluates each component's logs every 10 seconds for anomalies.

3. **Log alerts are generated**

   Log anomalies are generated at the end of each 10-second interval for each log line in which there is a statistical difference between the log pattern counts and the reference values and log patterns – or both – are identified in the live log stream that does not exist in the model.

   Based on this analysis, the log anomaly detection algorithms identify one or more of the following log anomaly types:

   1. Pattern is expected but is not detected
   2. Pattern is detected but is not expected
   3. Pattern is detected but is not expected, and includes an error
   4. Pattern is detected with lower frequency than expected
   5. Pattern is detected with higher frequency than expected

   The algorithm creates a log alert for each log anomaly. The severity of the log alert is calculated based on the following considerations:

   - All alerts that are generated by this algorithm are given an intial severity of 4 (Minor)
   - If the pattern is detected but not expected (log anomaly type 2) or it is an error pattern, then the severity is increased to 5 (Major).
   - If the pattern is detected with a lower frequency than expected (log anomaly type 4), then the severity is increased to 6 (Critical).

Example of natural language log anomaly detection

This topic provides an examples of log anomalies generated by the natural language log anomaly detection algorithm and when they are likely to be found.

Example: unexpected log pattern

You recently installed an application, which is made up of multiple microservices. Most of the microservices that make up this application generate their own log files.

Note For the natural language log anomaly detection algorithm to work, the following minimum data requirements must be met. At least 75% of the component microservices must have at least 2000 data points, where a data point is usually a log line.

The example proceeds as follows:

1. Data from the microservices is trained, as described in Workflow of natural language log anomaly detection.
2. During training, log patterns statistics are gathered and stored in the initial version of the model, as described in Workflow of natural language log anomaly detection. One of the patterns identified in the model is the following pattern within the **message** keyword.

   ```
   incident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined subtype=undefined level=notice
   ```

   This pattern occurs on average roughly 2.5 times in every 10 second time slot. See Clean log file lines for an example of what a log like this might look like.
3. When the model is applied to the live data stream as described in Workflow of natural language log anomaly detection, at a certain point the log pattern described in the previous step starts occurring in a number of the 10 second time slots. See Error log file lines for an example of what a log like this might look like.
4. Log anomaly alerts are generated for each log anomaly and sent to the in-cluster ObjectServer. These alerts report that a pattern has been detected with higher frequency than expected. See Log anomaly alert format for an example of what a log anomaly alert might look like.

Reference information

The following information is referenced in this task.

- Clean log file lines
- Error log file lines
- Log anomaly alert example

Clean log file lines

These example JSON log snippets display log lines for one of the microservices within the application.

Clean log file lines

The pattern of interest is:

```
incident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined subtype=undefined level=notice
```

You can find this pattern in the **message** parameter within the log lines below by searching for the invariant code **type=undefined subtype=undefined level=notice**. This will show you that the pattern of interest only occurs once, in the first line.

In a real life example, the number of lines and the number of occurrences of the pattern in a 10 second time slot would be consistent with the two or three times that is expected based on the values in the model.


```
[
    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"zpprensBxt31inA6cZbF","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:00.283Z","stream":"stdout","input":{"type":"container"},"container":
{"runtime":"docker","id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53","image":
{"name":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-354180024"}},"ecs":{"version":"1.8.0"},"log":
{"offset":640306,"file":{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-rating-
bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"}},"message":"incident euid=2570 epid=7324
dsteuid=2411 dstepid=7148 vd=Jarret10 type=undefined subtype=undefined level=notice","kubernetes":{"container":
{"name":"qotd-rating","image":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-354180024"},"deployment":
{"name":"qotd-ratings"},"node":{"uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":{"node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux","kubernetes_io/hostna
me":"padme-okd.coc-ibm.com","node-role_kubernetes_io/compute":"true","node-
role_kubernetes_io/infra":"true"},"hostname":"padme-okd.coc-ibm.com","name":"padme-okd.coc-
ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287","pod":{"name":"qotd-ratings-5444ddd79c-
gp7kb","uid":"d9b9549f-0529-11ec-9802-0617a6095287"},"namespace":"qotd","replicaset":{"name":"qotd-ratings-
5444ddd79c"},"labels":{"pod-template-hash":"1000888357","app":"qotd-rating"}},"agent":{"hostname":"padme-okd.coc-
ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-
okd.coc-ibm.com","type":"filebeat","version":"7.12.1"},"host":{"name":"padme-okd.coc-ibm.com"}},"sort":
[1629846480283,640306]}

    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"z5prensBxt31inA6d5Z9","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:02.065Z","agent":{"version":"7.12.1","hostname":"padme-okd.coc-
ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-
okd.coc-ibm.com","type":"filebeat"},"log":{"offset":11804878,"file":{"path":"/var/log/containers/qotd-web-5cbb5d9f64-
srmc4_qotd_qotd-web-
f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c.log"}},"stream":"stdout","message":"Starting new request
token: 783499 for IP: 10.128.0.41","input":{"type":"container"},"kubernetes":{"container":{"name":"qotd-
web","image":"registry.gitlab.com/quote-of-the-day/qotd-web:latest"},"deployment":{"name":"qotd-web"},"node":{"labels":
{"kubernetes_io/hostname":"padme-okd.coc-ibm.com","node-role_kubernetes_io/compute":"true","node-
role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux"},"hostname":"padme-
okd.coc-ibm.com","name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287"},"namespace_uid":"dcc7d8e8-
01bf-11ec-9802-0617a6095287","pod":{"uid":"efbf8ddd-04fc-11ec-9802-0617a6095287","name":"qotd-web-5cbb5d9f64-
srmc4"},"namespace":"qotd","replicaset":{"name":"qotd-web-5cbb5d9f64"},"labels":{"app":"qotd-web","pod-template-
```

```
hash":"1766185920"}},"container":
{"runtime":"docker","id":"f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c","image":
{"name":"registry.gitlab.com/quote-of-the-day/qotd-web:latest"}},"ecs":{"version":"1.8.0"},"host":{"name":"padme-okd.coc-
ibm.com"}},"sort":[1629846482065,11804878]}

    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"0JprensBxt31inA6d5Z9","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:02.066Z","host":{"name":"padme-okd.coc-ibm.com"},"ecs":
{"version":"1.8.0"},"stream":"stdout","message":"[783499] Web request: /.","input":{"type":"container"},"kubernetes":
{"pod":{"uid":"efbf8ddd-04fc-11ec-9802-0617a6095287","name":"qotd-web-5cbb5d9f64-srmc4"},"namespace":"qotd","replicaset":
{"name":"qotd-web-5cbb5d9f64"},"labels":{"pod-template-hash":"1766185920","app":"qotd-web"},"container":{"name":"qotd-
web","image":"registry.gitlab.com/quote-of-the-day/qotd-web:latest"},"deployment":{"name":"qotd-web"},"node":
{"hostname":"padme-okd.coc-ibm.com","name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":
{"kubernetes_io/hostname":"padme-okd.coc-ibm.com","node-role_kubernetes_io/compute":"true","node-
role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux"}},"namespace_uid":"dc
c7d8e8-01bf-11ec-9802-0617a6095287"},"container":
{"runtime":"docker","id":"f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c","image":
{"name":"registry.gitlab.com/quote-of-the-day/qotd-web:latest"}},"log":{"offset":11805003,"file":
{"path":"/var/log/containers/qotd-web-5cbb5d9f64-srmc4_qotd_qotd-web-
f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c.log"}},"agent":{"name":"padme-okd.coc-
ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-
f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057"}},"sort":[1629846482066,11805003]}

    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"0ZprensBxt31inA6d5Z9","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:02.182Z","input":{"type":"container"},"kubernetes":{"deployment":{"name":"qotd-
web"},"node":{"name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":{"node-
role_kubernetes_io/compute":"true","node-role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux","kubernetes_io/hostna
me":"padme-okd.coc-ibm.com"},"hostname":"padme-okd.coc-ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-
0617a6095287","pod":{"uid":"efbf8ddd-04fc-11ec-9802-0617a6095287","name":"qotd-web-5cbb5d9f64-
srmc4"},"namespace":"qotd","replicaset":{"name":"qotd-web-5cbb5d9f64"},"labels":{"app":"qotd-web","pod-template-
hash":"1766185920"},"container":{"name":"qotd-web","image":"registry.gitlab.com/quote-of-the-day/qotd-
web:latest"}},"host":{"name":"padme-okd.coc-ibm.com"},"agent":{"id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-
okd.coc-ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-ibm.com","ephemeral_id":"39a50623-e967-
451f-b30d-f06fd421d0cf"},"stream":"stdout","message":"[783499] Getting daily quote.","log":{"file":
{"path":"/var/log/containers/qotd-web-5cbb5d9f64-srmc4_qotd_qotd-web-
f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c.log"},"offset":11805098},"container":{"image":
{"name":"registry.gitlab.com/quote-of-the-day/qotd-
web:latest"},"runtime":"docker","id":"f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c"},"ecs":
{"version":"1.8.0"}},"sort":[1629846482182,11805098]}

    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"3pprensBxt31inA6h5Yj","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:02.195Z","kubernetes":{"namespace":"qotd","replicaset":{"name":"qotd-quote-
585979769"},"labels":{"app":"qotd-quote","pod-template-hash":"141535325"},"container":{"name":"qotd-
quote","image":"registry.gitlab.com/quote-of-the-day/quote-service:latest"},"deployment":{"name":"qotd-quote"},"node":
{"uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":{"node-role_kubernetes_io/compute":"true","node-
role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux","kubernetes_io/hostna
me":"padme-okd.coc-ibm.com"},"hostname":"padme-okd.coc-ibm.com","name":"padme-okd.coc-
ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287","pod":{"name":"qotd-quote-585979769-
wpg25","uid":"e810e7e8-04fc-11ec-9802-0617a6095287"}},"container":{"image":{"name":"registry.gitlab.com/quote-of-the-
day/quote-
service:latest"},"runtime":"docker","id":"ef6bf31678a9d80b5ac95a27dbb72f68a2f855aa09874e6eda222e4ff1c3184e"},"log":
{"file":{"path":"/var/log/containers/qotd-quote-585979769-wpg25_qotd_qotd-quote-
ef6bf31678a9d80b5ac95a27dbb72f68a2f855aa09874e6eda222e4ff1c3184e.log"},"offset":6052045},"input":
{"type":"container"},"agent":{"id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-okd.coc-
ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-
f06fd421d0cf"},"ecs":{"version":"1.8.0"},"host":{"name":"padme-okd.coc-ibm.com"},"stream":"stdout","message":"[783499]
Quote request: /daily."},"sort":[1629846482195,6052045]}
]
```


Error log file lines

These example JSON log snippets display log lines for one of the microservices within the application. You can see that the pattern of interest occurs in every line in this log snippet, in all five times.

The pattern of interest is:

```
incident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined subtype=undefined level=notice
```

You can find this pattern in the **message** parameter within the log lines below by searching for the invariant code **type=undefined subtype=undefined level=notice**. This will show you which lines of code instantiate this pattern.

In a real life example, the number of lines and the number of occurrences of the pattern in a 10 second time slot would be significantly larger. You might have 100 or so log lines in 10 seconds and the pattern might occur on the order of 20 times, which is a significant deviation from the two or three times that is expected based on the values in the model. Based on this, log anomaly alerts would be generated.

Error log file lines


```
[
    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"pppqensBxt31inA6h5Qa","_score":null,"_source":
{"@timestamp":"2021-08-24T23:07:00.053Z","input":     {"type":"container"},"container":
{"id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53","image":{"name":"registry.gitlab.com/quote-of-
the-day/qotd-ratings-service:build-354180024"},"runtime":"docker"},"host":{"name":"padme-okd.coc-ibm.com"},"log":{"file":
{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-rating-
bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"},"offset":610747},"stream":"stdout","message":"inci
dent euid=5840 epid=9800 dsteuid=3612 dstepid=7788 vd=Corbin_Gerlach59 type=undefined subtype=undefined
level=notice","kubernetes":{"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287","pod":{"name":"qotd-ratings-
5444ddd79c-gp7kb","uid":"d9b9549f-0529-11ec-9802-0617a6095287"}, "namespace":"qotd","replicaset":{"name":"qotd-ratings-
5444ddd79c"},"labels":{"app":"qotd-rating","pod-template-hash":"1000888357"},"container":{"name":"qotd-
```

```
rating","image":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-354180024"},"deployment":{"name":"qotd-
ratings"},"node":{"hostname":"padme-okd.coc-ibm.com","name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-
0617a6095287","labels":
{"beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux","kubernetes_io/hostname":"padme-okd.coc-
ibm.com","node-role_kubernetes_io/compute":"true","node-role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true"}}},"ecs":{"version":"1.8.0"},"agent":{"name":"padme-okd.coc-ibm.com","type":"filebeat",
"version":"7.12.1","hostname":"padme-okd.coc-ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-
f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057"}}
    ,"sort":[1629846420053,610747]},

    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"qppqensBxt31inA6h5Qa","_score":null,"_source":
{"@timestamp":"2021-08-24T23:07:00.578Z","stream":"stdout","message":"incident euid=5896 epid=8870 dsteuid=4250
dstepid=5530 vd=Sunny39 type=undefined subtype=undefined level=notice","kubernetes":{"pod":{"name":"qotd-ratings-
5444ddd79c-gp7kb","uid":"d9b9549f-0529-11ec-9802-0617a6095287"},"namespace":"qotd","replicaset":{"name":"qotd-ratings-
5444ddd79c"},"labels":{"app":"qotd-rating","pod-template-hash":"1000888357"},"container":
{"image":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-354180024","name":"qotd-rating"},"deployment":
{"name":"qotd-ratings"},"node":{"name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":
{"beta_kubernetes_io/os":"linux","kubernetes_io/hostname":"padme-okd.coc-ibm.com","node-
role_kubernetes_io/compute":"true","node-role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64"},"hostname":"padme-okd.coc-
ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287"},"ecs":{"version":"1.8.0"},"host":{"name":"padme-
okd.coc-ibm.com"},"log":{"offset":611253,"file":{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-
rating-bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"}},"input":{"type":"container"},"container":
{"image":{"name":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"},"runtime":"docker","id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53"},"agent":
{"id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-okd.coc-
ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-
f06fd421d0cf"}},"sort":[1629846420578,611253]},

    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"vJpqensBxt31inA6i5QD","_score":null,"_source":
{"@timestamp":"2021-08-24T23:07:01.057Z","input":{"type":"container"},"kubernetes":{"labels":{"app":"qotd-rating","pod-
template-hash":"1000888357"},"container":{"name":"qotd-rating","image":"registry.gitlab.com/quote-of-the-day/qotd-
ratings-service:build-354180024"},"deployment":{"name":"qotd-ratings"},"node":{"uid":"193183f9-01bd-11ec-926b-
0617a6095287","labels":{"node-role_kubernetes_io/compute":"true","node-role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux","kubernetes_io/hostna
me":"padme-okd.coc-ibm.com"},"hostname":"padme-okd.coc-ibm.com","name":"padme-okd.coc-
ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287","pod":{"name":"qotd-ratings-5444ddd79c-
gp7kb","uid":"d9b9549f-0529-11ec-9802-0617a6095287"},"namespace":"qotd","replicaset":{"name":"qotd-ratings-
5444ddd79c"}},"container":{"id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53","image":
{"name":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-354180024"},"runtime":"docker"},"host":
{"name":"padme-okd.coc-ibm.com"},"log":{"file":{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-
rating-
bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"},"offset":611542},"stream":"stdout","message":"inci
dent euid=2210 epid=3789 dsteuid=3713 dstepid=4212 vd=Garrick40 type=undefined subtype=undefined level=notice","agent":
{"name":"padme-okd.coc-ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-
ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057"},"ecs":
{"version":"1.8.0"}},"sort":[1629846421057,611542]},

    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"vZpqensBxt31inA6i5QD","_score":null,"_source":
{"@timestamp":"2021-08-24T23:07:01.584Z","log":{"offset":611726,"file":{"path":"/var/log/containers/qotd-ratings-
5444ddd79c-gp7kb_qotd_qotd-rating-bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"}},"input":
{"type":"container"},"container":{"image":{"name":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"},"runtime":"docker","id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53"},"message":"incide
nt euid=8003 epid=5254 dsteuid=2568 dstepid=6719 vd=Sylvan.Romaguera type=undefined subtype=undefined
level=notice","stream":"stdout","kubernetes":{"pod":{"uid":"d9b9549f-0529-11ec-9802-0617a6095287","name":"qotd-ratings-
5444ddd79c-gp7kb"},"namespace":"qotd","replicaset":{"name":"qotd-ratings-5444ddd79c"},"labels":{"app":"qotd-rating","pod-
template-hash":"1000888357"},"container":{"name":"qotd-rating","image":"registry.gitlab.com/quote-of-the-day/qotd-
ratings-service:build-354180024"},"deployment":{"name":"qotd-ratings"},"node":{"name":"padme-okd.coc-
ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":{"node-role_kubernetes_io/compute":"true","node-
role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux","kubernetes_io/hostna
me":"padme-okd.coc-ibm.com"},"hostname":"padme-okd.coc-ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-
0617a6095287"},"host":{"name":"padme-okd.coc-ibm.com"},"agent":{"ephemeral_id":"39a50623-e967-451f-b30d-
f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-okd.coc-
ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-ibm.com"},"ecs":{"version":"1.8.0"}},"sort":
[1629846421584,611726]},
    {"_index":"filebeat-7.12.1-2021.08.20-000001","_type":"_doc","_id":"0ppqensBxt31inA6jpTq","_score":null,"_source":
{"@timestamp":"2021-08-24T23:07:02.092Z","stream":"stdout","input":{"type":"container"},"kubernetes":{"node":
{"name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":{"node-
role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux","kubernetes_io/hostna
me":"padme-okd.coc-ibm.com","node-role_kubernetes_io/compute":"true"},"hostname":"padme-okd.coc-
ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287","pod":{"uid":"d9b9549f-0529-11ec-9802-
0617a6095287","name":"qotd-ratings-5444ddd79c-gp7kb"},"namespace":"qotd","replicaset":{"name":"qotd-ratings-
5444ddd79c"},"labels":{"pod-template-hash":"1000888357","app":"qotd-rating"},"container":{"name":"qotd-
rating","image":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-354180024"},"deployment":{"name":"qotd-
ratings"}},"ecs":{"version":"1.8.0"},"agent":{"version":"7.12.1","hostname":"padme-okd.coc-
ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-
okd.coc-ibm.com","type":"filebeat"},"log":{"file":{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-
rating-bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"},"offset":612021},"message":"incident
euid=2850 epid=7575 dsteuid=3306 dstepid=2473 vd=Keshaun_Berge96 type=undefined subtype=undefined
level=notice","container":{"id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53","image":
{"name":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-354180024"},"runtime":"docker"},"host":
{"name":"padme-okd.coc-ibm.com"}},"sort":[1629846422092,612021]}
]
```


Log anomaly alert example

This example alert incorporates data from the identified log anomalies into an ObjectServer alert structure. Note that based on the values in the **details.log_anomaly_detector.causality.template_list** field, three different templates have been identified as occurring with significant deviations from the model and have been grouped together within this anomaly.

```
[
    {"id": "a21bd7c9-b3f6-3684-b900-12057634cb66",
    "occurrenceTime": "2021-08-24T23:07:00.000Z",
    "sender": {"name": "Log Anomaly"},
    "event_link": "https://api.us-south.logging.cloud.ibm.com/v1/export?to=1629846430000&from=1629846420000",
    "summary": "Abnormal behavior found in component: qotd-rating",
    "description": "[<NUM>] Ratings request, id: <NUM>: 3\n[<NUM>] The monkey's dart hit the <NUM>: 4\nHeartbeat status:
CPU=\"OK\" Memory=\"OK\": 2\nincident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined
subtype=undefined level=notice: 1",
    "severity": 4,
    "type":
        {"eventType": "problem",
        "classification": "log anomaly",
        "condition": "0d33c050-101f-11ec-881b-acde48001122"},
    "resource":
        {"name": "qotd-rating",
        "type": "application",
        "application": "qotd-rating",
        "entity": "qotd-rating"},
    "expirySeconds": 900,
    "details":
        {"log_anomaly_detector":
          {"start_timestamp": 1629846420000,
          "end_timestamp": 1629846430000,
          "original_group_id": "0d33c050-101f-11ec-881b-acde48001122",
          "causality":
            {"service": ["qotd-rating"]},
            "detected_at": 1632252299670.0,
            "source_application_id": "qotd-rating",
            "log_anomaly_confidence": 0.9152,
            "log_anomaly_model": ["PCA"],
            "prediction_error":
              {"pca_prediction_error": 26.884115203263626,
              "embedding_prediction_error": 0.02826894412475822},
            "error_templates": [],
            "template_list": ["Heartbeat status: <*><*><*>", "Heartbeat status: <*><*><*><*><*>", "incident euid=<NUM>
epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined subtype=undefined level=notice", "Heartbeat status: CPU=\"OK\"
Memory=\"OK\"", "[<NUM>] Ratings request, id: <NUM>", "[<NUM>] That doesn't seem fair.", "[<NUM>] The monkey's dart hit
the <NUM>", "Heartbeat status: <*><*><*><*><*><*>", "Heartbeat status: <*><*><*><*>", "Unknown_normal", "Unknown_error"],
            "count_vector": [0, 0, 21, 0, 5, 0, 5, 2, 0, 0, 0],
            "text_dict":
              {"template_ids": ["7eddc431", "b59d9df4", "7cd5f6b4"],
              "template_list": ["Heartbeat status: CPU=\"OK\" Memory=\"OK\"", "Heartbeat status: <*><*><*><*><*><*>",
"incident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined subtype=undefined level=notice"],
              "count_vector": [0, 2, 21],
              "expected_count_vector": [14.24, 14.91, 15.13],
              "expected_count_lower_bound": [5.34, 6.45, 10.46],
              "expected_count_upper_bound": [23.14, 23.36, 20.02]},
              "application_group_id": "1000",
              "application_id": "1000",
              "model_version": "v4",
              "severity_from_model": 0,
              "description": "Heartbeat status: <*><*><*><*><*><*>: 2\nincident euid=<NUM> epid=<NUM> dsteuid=<NUM>
dstepid=<NUM> <*>type=undefined subtype=undefined level=notice: 21\n[<NUM>] Ratings request, id: <NUM>: 5\n[<NUM>] The
monkey's dart hit the <NUM>: 5"}},
              "timestamp": 1632252299670.0}
  ]
```


About log anomaly detection - statistical baseline

Statistical baseline log anomaly detection is an unsupervised learning algorithm that uses a statistical moving average on all of your log data to discover abnormal behavior. This algorithm automatically detects unusual patterns in logs and notifies you when they occur.

Deprecated: The statistical baseline log anomaly detection algorithm is to be deprecated in a future release. The recommended action is to use log anomaly detection - golden signals as a replacement.

Data that is used for analysis is updated every 30 minutes, so this algorithm provides value quickly. After it takes the first 30 minutes to establish a baseline, statistical baseline log anomaly detection uses the computed baseline to identify potential errors.

For domain-specific log anomaly detection, such as for IBM MQ or WebSphere, the statistical baseline model already has a baseline for domain-specific logs. Thus, the model can detect potential errors without waiting for 30 minutes to establish an initial baseline. Other than the means of establishing a baseline, domain-specific log anomaly detection has identical behavior to standard statistical baseline log anomaly detection. For more information about domain-specific log anomaly detection, see this video.

The algorithm for both statistical baseline log anomaly detection and domain-specific log anomaly detection supplements the natural language log anomaly detection algorithm by reporting extra log anomalies to your ChatOps interface and to the Alert Viewer.

Domain-specific log anomaly detection is available for the following products:

- WebSphere
- IBM MQ

Enabling this algorithm helps IBM Cloud Pak for AIOps understand normal behavior so when an anomalous situation arises, you are notified. Anomalies are raised to your ChatOps interface. For information on an anomaly, you can view related alerts to see the patterns and associated log messages.

By default the training of this AI algorithm is enabled and it is suggested that you leave it on. If you disable it, you won't see details about statistical baseline anomalies in either your ChatOps feed or in the Alert Viewer.

For more information about natural language log anomaly detection, see the following topics:

- Workflow
- Examples

Workflow of statistical baseline log anomaly detection

Find out how statistical baseline log anomaly detection processes your log data to generate log alerts.

This algorithm extracts specific entities from the logs, such as short text strings that indicate error codes and exceptions. Combined with other statistics, such as the number of logs for each component, the algorithm uses these statistics as a baseline to discover abnormal behavior in your live log data.

These extracted entities are not just a simple dictionary, but rather use *clue words* to interpret other data in the logs, such as numbers. For example, the number **`500`** appearing in a log message might be a count, identifier, or some other value, but if it is accompanied by clue words, such as **`http status`**, then the system knows that this number refers to an error code. When enough of these deviations are detected to be statistically significant, then the system knows it needs to report an anomaly issue.

Domain-specific log anomaly detection, such as that for IBM MQ or WebSphere, uses the same algorithm as statistical baseline log anomaly detection. However, the entities that are used in domain-specific log anomaly detection and that are included in the input logs are already provided from prior knowledge about the specific domain. After entities like the message ID and log level are detected, the workflow of a domain-specific log anomaly is similar to the workflow of a general statistical baseline log anomaly.

The following figure shows a simplified workflow of the statistical baseline log anomaly detection algorithm.

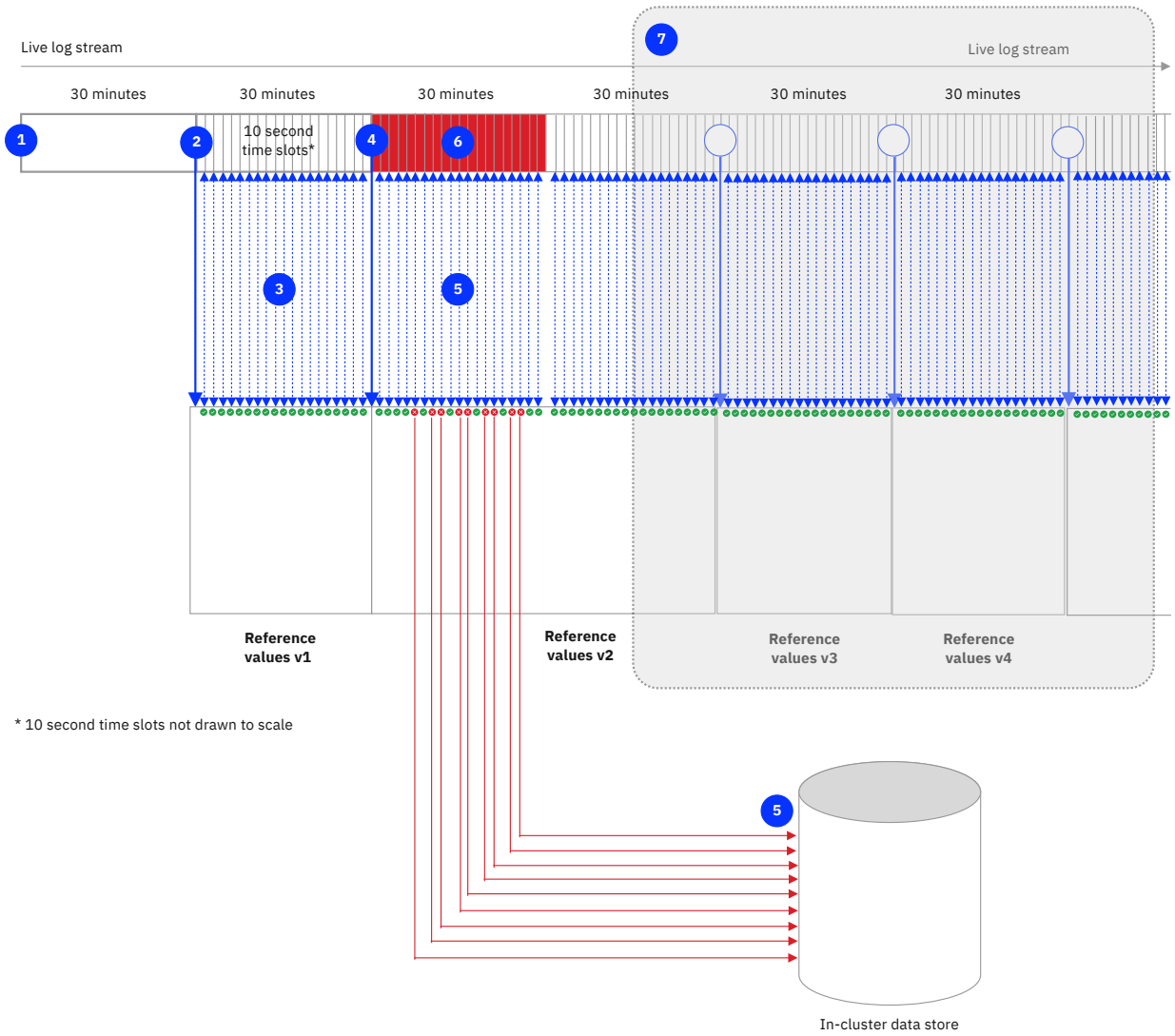


Figure. Workflow

1. **Algorithm switched on**

The algorithm is first switched on. **Note:** See Supported resource number and throughput rates for limits on throughput rates.

2. **Extract initial reference values**

   At the end of the first 30-minute time interval, the algorithm reviews the log messages from the previous 30 minutes and extracts text entities from the logs. It saves the number of occurrences of each entity as reference values.

3. **Review live stream for anomalies**

   Now that it has the reference values, the system starts monitoring the live log stream for anomalies, which are statistically significant deviations from the baseline. It evaluates each component's logs every 10 seconds for anomalies.

4. **Extract values from next 30-minute interval and update reference values**

   At the end of the next 30-minute time interval, the algorithm again reviews the log messages from the previous 30 minutes. As before, the algorithm extracts text entities and statistics from the logs. It then combines this data with the previous baseline to compute a new baseline, which now covers 60 minutes' worth of normal behavior data.

   Note: If the live data that is being processed is not from the last 30 minutes, then the algorithm still looks for anomalies, but the reference values are not updated. One reason why the last 30-minute time interval might contain older data is when a delay in data processing occurs due to a temporary malfunction of a data processing component, such as a Flink job.

5. **Log alerts are generated**

   The system is continuously monitoring the live log stream for anomalies and evaluates each component's logs every 10 seconds for anomalies. Log anomalies are generated at the end of each 10-second interval for each log line in which a statistical difference exists between the number of entities that are identified and the reference values and numbers, or both, that are identified and accompanied by clue words.

   Based on this analysis, the log anomaly detection algorithm identifies one or more of the following log anomaly types:

   - An entity is detected but not expected and includes an error.
   - An entity is detected with lower frequency than expected.
   - An entity is detected with higher frequency than expected.

   The algorithm creates a log alert for each log anomaly and sends these alerts to the in-cluster data store. Any alerts that are generated by this algorithm are assigned severity 5 (Major).

6. **Interval is potentially tagged as biased**

   Based on the number of log anomalies detected in the previous step, a threshold calculation is performed. The number of log anomalies that are detected is divided by the number of windows. If the resulting value exceeds the predefined threshold of 0.15, then the entire interval is tagged as biased, and none of the reference values are updated based on the entity extraction values for this interval.

   Note: The predefined threshold ensures that frequently occurring anomalies do not bias the statistical baseline reference. If the anomalies are not frequently occurring, such as when the resulting value does not exceed the predefined threshold of 0.15, the interval is not tagged as biased, and the reference values are updated. It is possible that such infrequent anomalies are learned as normal behavior, and they might not be detected again. To reset the learning, you need to manually reset the reference of the statistical baseline. For more information, see Resetting the reference of the statistical baseline.

7. **Processing of the live log stream continues**

   The algorithm continues processing 30-minute time intervals in the live log stream.

Examples of statistical baseline log anomaly detection

This topic provides examples of log anomalies generated by the statistical baseline log anomaly detection algorithm and when they are likely to be found.

Examples presented in this topic

The following examples are described in this topic.

- Unexpected 500 error
- WebSphere log anomaly
- IBM MQ log anomaly

Example: unexpected 500 error

You recently installed the sock-shop application, which is made up of multiple microservices, such as **`front-end`** and **`orders`**. Most of the microservices that make up this application generate their own log files.

Note For the statistical baseline log anomaly detection algorithm to work, there is no formal minimum log line count. The main requirement is that all microservices generate a reasonable amount of log data.

The example proceeds as follows:

1. Initially the microservices are all running OK and generate clean log files, without any text entities that might indicate an error. See Clean log file lines for an example of what a log like this might look like.

2. Shortly after this, some of the pods associated with the microservices start shutting down unexpectedly; as a result of this, **`HTTP 500`** errors appear for the first time in the live log stream.

3. Live log data is analyzed in 10 second slots. Now, when the latest logs are compared to the baseline reference values, several error entities are identified: **500**, **error**, and **illegIllegalStateException**. The algorithm determines that the number **500** is an error, as it is accompanied by the entity **status**, which points to an HTTP 500 error, a general HTTP status code indicating that something has gone wrong on a server but the server could not be more specific about what the exact problem is. These entities have not been identified before during normal behavior and since they are statistically significant (trivially so in this example) thus point to an anomaly. See Error log file lines for an example of what a log like this might look like.

   Note If this was an error message that occurs on a regular basis, say an optional library that is not used in a given piece of code, then the statistical baseline log anomaly detection algorithm will automatically learn this, and consequently treat this error message as normal and therefore no longer alert on it.

4. Log anomaly alerts are generated for each log anomaly and sent to the in-cluster ObjectServer. See Log anomaly alert format for an example of what a log anomaly alert might look like.

Reference information

The following information is referenced in this task.

- Clean log file lines
- Error log file lines
- Log anomaly alert example

Clean log file lines

These example JSON log snippets display clean log lines for the front-end and orders microservices within the sock-shop application. The log algorithm extracts entities from the **_line** keyword within these log lines and then performs the analysis described in Workflow of statistical baseline log anomaly detection on these extracted entities.

Clean log file lines from the front-end microservice

The following log files lines do contain numbers but they are not accompanied by error entities. The numbers in these log message are actually part of user address data (for example **322 Main Street**) and credit card data, and can be seen in the **_line** parameter in the code snippet below. You might need to scroll to the right to see the full content of the **_line** parameter.


```
[
  {
    "_app":"front-end",
    "_file":"/var/log/containers/front-end-85c54f4fb-jvzqn_sock-shop_front-end-
e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42.log",
    "_host":"front-end-85c54f4fb-jvzqn",
    "_ip":"172.30.148.208",
    "_ipremote":"52.117.50.10",
    "_lid":"1264263893737455616",
    "_line":"Posting Address: {\"number\":\"\",\"street\":\"322 Main Street\",\"city\":\"Gotham
City\",\"postcode\":\"12345\",\"country\":\"United States\",\"userID\":\"1q2w3e4r5t6y7u8i9o0p\"}",
    "_logtype":"json",
    "_mac":"00:00:00:00:94:60",
    "_tag":["k8s"],"_ts":1601424000273,
    "city":"Gotham City",
    "container":"front-end",
    "containerid":"e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42",
    "country":"United States",
    "namespace":"sock-shop",
    "node":"kube-bqvhtmdd031d903ud4g0-simulationd-default-00000105",
    "number":"",
    "pod":"front-end-85c54f4fb-jvzqn",
    "postcode":"12345",
    "street":"322 Main Street",
    "userID":"0p9o8i7u6y5t4r3e2w1q"
  },
  {
    "_app":"front-end",
    "_file":"/var/log/containers/front-end-85c54f4fb-jvzqn_sock-shop_front-end-
e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42.log",
    "_host":"front-end-85c54f4fb-jvzqn",
    "_ip":"172.30.148.208",
    "_ipremote":"52.117.50.10",
    "_lid":"1264263893737455617",
    "_line":"\u001B[0mPOST /addresses \u001B[32m200 \u001B[0m9.663 ms - -\u001B[0m",
    "_logtype":"customapp",
    "_mac":"00:00:00:00:94:60",
    "_tag":["k8s"],
    "_ts":1601424000282,
    "container":"front-end",
    "containerid":"e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42",
    "namespace":"sock-shop",
    "node":"kube-bqvhtmdd031d903ud4g0-simulationd-default-00000105",
    "pod":"front-end-85c54f4fb-jvzqn"
  }
]
```


Clean log file lines from the orders microservice

The following log files lines do contain numbers but they are not accompanied by error entities. The numbers in these log message are actually part of user address data (for example **322 Main Street**) and credit card data, and can be seen in the **_line** parameter in the code snippet below. You might need to scroll to the right to see the full content of the **_line** parameter.


```
[
    {"_app":"orders",
```

```
    "_file":"/var/log/containers/orders-b8cb69f5c-gn87g_sock-shop_orders-
9d699fcc72f3283bf66ecb0bb850251e73dd5a7f5a0026b4fcb6facc6c533baf.log",
    "_host":"orders-b8cb69f5c-gn87g",
    "_ip":"172.30.148.208",
    "_ipremote":"52.117.50.10",
    "_lid":"1264263925219901440",
    "_line":"2020-09-30 00:00:07.959  INFO [orders,4ecc59a2bbf00a18,276a9b5feb58dcd3,true] 7 --- [p-nio-80-exec-8]
w.w.s.o.controllers.OrdersController     : Sending payment request: PaymentRequest{address=Address{id=null, number='',
street='322 Main Street', city='Gotahm City', country='United States', postcode='12345'}, card=Card{id=null,
longNum='1234567812345678', expires='12/56', ccv='321'}, customer=Customer{id=null, firstName='John', lastName='Doe',
username='John_Doe', addresses=[], cards=[]}}",
    "_logtype":"json",
    "_mac":"00:00:00:00:94:60",
    "_tag":["k8s"],
    "_ts":1601424007959,
    "addresses":"[],",
    "cards":"[]}}",
    "container":"orders",
    "containerid":"9d699fcc72f3283bf66ecb0bb850251e73dd5a7f5a0026b4fcb6facc6c533baf",
    "firstName":"'John',",
    "lastName":"'Doe',",
    "level":"INFO",
    "message":"[orders,4ecc59a2bbf00a18,276a9b5feb58dcd3,true] 7 --- [p-nio-80-exec-8]
w.w.s.o.controllers.OrdersController     : Sending payment request: PaymentRequest{address=Address{id=null, number='',
street='322 Main Street', city='Gotham City', country='United States', postcode='12345'}, card=Card{id=null,
longNum='1234567812345678', expires='12/56', ccv='321'}, customer=Customer{id=null, firstName='John', lastName='Doe',
username='John_Doe', addresses=[], cards=[]}}",
    "namespace":"sock-shop",
    "node":"kube-bqvhtmdd031d903ud4g0-simulationd-default-00000105",
    "pod":"orders-b8cb69f5c-gn87g",
    "username":"'John_Doe',"}
]
```

Error log file lines

The following example JSON log snippet displays error log lines for the front-end microservice within the sock-shop application. The log algorithm extracts entities from the **_line** keyword within these log lines and then performs the analysis described in Workflow of statistical baseline log anomaly detection on these extracted entities.

Error log file line from the front-end microservice

The following log line contains numbers accompanied by error entities; in particular, the **_line** parameter contains the number **500**, as well as the error entities **status**, **exception**, and **IllegalStateException**. You might need to scroll to the right to see the full content of the **_line** parameter.

```
[
    {"_account": "6a8e297e13",
    "_cluster": "ld70",
    "_host": "front-end-85c54f4fb-jvzqn",
    "_ingester": "mezmo-agent/1.6.2 (Ubuntu/16.04) k8s",
    "_logtype": "json",
    "_mac": "00:00:00:00:94:60",
    "_tag": ["k8s"],
    "_file": "/var/log/containers/front-end-85c54f4fb-jvzqn_sock-shop_front-end-
e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42.log",
    "_line": "Order response: {\"timestamp\":1599236098757,\"status\":500,\"error\":\"Internal Server
Error\",\"exception\":\"java.lang.IllegalStateException\",\"message\":\"Unable to create order due to unspecified IO
error.\",\"path\":\"/orders\"}",
    "_ts": 1599236098759,
    "_app": "front-end",
    "pod": "front-end-85c54f4fb-jvzqn",
    "namespace": "sock-shop",
    "container": "front-end",
    "containerid": "e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42",
    "node": "kube-bqvhtmdd031d903ud4g0-simulationd-default-00000105",
    "_ip": "172.30.148.208",
    "_ipremote": "52.117.50.10",
    "__key": "logline:6a8e297e13:front-end:front-end-85c54f4fb-jvzqn:ld70",
    "_bid": "e4975ea4-6634-4f88-accc-8eba410d2617:61301:ld70",
    "level": null,
    "timestamp": 1599236098757,
    "error": "Internal Server Error",
    "exception": "java.lang.IllegalStateException",
    "message": "Unable to create order due to unspecified IO error.",
    "_status": 500,
    "_path": "/orders",
    "_id": "1255087168836907020"}
]
```

Log anomaly alert example

This example alert incorporates data from the identified log anomalies into an ObjectServer alert structure.

- This alert has been assigned a severity of 5.
- The **summary** parameter explains that 24 errors have been found in the front-end component.
- The **details.log_anomaly_detector.text_dict.template_list** parameter lists the error entities identified.

```
[
    {"id": "d708346f-046a-35ae-b0ab-1a9ad8f1bafa",
    "occurrenceTime": "2020-09-04T16:15:40.000Z",
```

```
    "sender": {"name": "Log Anomaly"},
    "event_link": "https://api.us-south.logging.cloud.ibm.com/v1/export?to=1599236150000&from=1599236140000",
    "summary": "24 errors found in component: front-end",
    "description": "500: 8\njava.lang.IllegalStateException: 8\nerror_log: 8",
    "severity": 5,
    "type":
        {"eventType": "problem",
        "classification": "log anomaly",
        "condition": "c9680808-14bb-11ec-b72f-acde48001122"},
    "resource":
        {"name": "front-end",
        "type": "application",
        "application": "front-end",
        "entity": "front-end"},
    "expirySeconds": 900,
    "details":
        {"log_anomaly_detector":
            {"start_timestamp": 1599236140000,
            "end_timestamp": 1599236150000,
            "original_group_id": "c9680808-14bb-11ec-b72f-acde48001122",
            "causality":
                {"service": ["front-end"]},
            "detected_at": 1632338288103.0,
            "source_application_id": "front-end",
            "log_anomaly_confidence": 1.0,
            "log_anomaly_model": ["oob_entities"],
            "prediction_error":
                {"oob_entities_error": 24},
            "error_templates": [0, 1, 2],
            "template_list": ["500", "java.lang.IllegalStateException", "error_log"],
            "count_vector": [8, 8, 8],
            "text_dict":
                {"template_ids": ["oob_entities_500", "oob_entities_java.lang.IllegalStateException",
"oob_entities_error_log"],
                "template_list": ["500", "java.lang.IllegalStateException", "error_log"],
                "count_vector": [8, 8, 8],
                "expected_count_vector": [0.29, 0.29, 0.48],
                "expected_count_lower_bound": [0, 0, 0],
                "expected_count_upper_bound": [3.0, 3.0, 3.21]
                },
                "application_group_id": "ss",
                "application_id": "1",
                "model_version": "5",
                "severity_from_model": 0,
                "description": "500: 8\njava.lang.IllegalStateException: 8\nerror_log: 8"
            }
        },
    "timestamp": 1632338288103.0}
  ]
```

Example: WebSphere log anomaly

The following example shows a WebSphere log. In log anomalies for a specific domain like WebSphere, entities are included in the logs. WebSphere log anomaly detection identifies which entities are error indicators. Message ID, **ibm_messageId**, and log levels, **loglevel**, are examples of WebSphere entities.

```
{
  "type": "was_message",
  "host": "abc-123.myserver.com",
  "ibm_cellName": "gsV900Cell",
  "ibm_nodeName": "node4",
  "ibm_serverName": "nodeagent",
  "ibm_sequence": "1616166373781_0000000000002",
  "message": "TRAS0017I: The startup trace state is *=info.",
  "ibm_datetime": "2021-03-19T11:06:13.781-0400",
  "ibm_messageId": "TRAS0017I",
  "ibm_threadId": "00000001",
  "module": "com.ibm.ejs.ras.ManagerAdmin",
  "loglevel": "INFO",
  "unix_ts": 1616166373781,
  "start": 1616166370000,
  "end": 1616166380000
}
```

On the **Incidents and alerts** page, the following example from the **Summary** parameter explains that two logs were found in the **qotd-qrcode** container. This parameter also says that a primary message code of **SSLC0008E** is contributing to the error. This alert is further explained by the **Alert details** panel. For more information, see Domain-specific log anomaly alert details.

```
Log Anomaly found 2 logs containing errors for container(s): qotd-qrcode with a primary message code of SSLC0008E
```

Example: IBM MQ log anomaly

The following example shows an IBM MQ log. In log anomalies for a specific domain like IBM MQ, entities are included in the logs. IBM MQ log anomaly detection identifies which entities are error indicators. Message ID, **ibm_messageId**, and log levels, **loglevel**, are examples of IBM MQ entities.

```
{
  "ibm_messageId": "AMQ6287I",
  "ibm_arithInsert1": 0,
  "ibm_arithInsert2": 0,
  "ibm_commentInsert1": "Linux 4.18.0-193.70.1.el8_2.x86_64 (Cloud Pak Linux (x86-64 platform) 64-bit)",
  "ibm_commentInsert2": "/opt/mqm (Installation1)",
  "ibm_commentInsert3": "9.3.0.0 (p930-L220526)",
```

```
  "ibm_datetime": "2022-06-01T01:13:49.673Z",
  "ibm_serverName": "QMA",
  "type": "mq_log",
  "host": "slot11-qma-ibm-mq-0",
  "loglevel": "INFO",
  "module": "amqxeida.c:6755",
  "ibm_sequence": "1654046029_673594295",
  "ibm_processId": "37",
  "ibm_threadId": "3",
  "ibm_version": "9.3.0.0",
  "ibm_processName": "amqzmuc0",
  "ibm_userName": "mqm",
  "ibm_installationName": "Installation1",
  "ibm_installationDir": "/opt/mqm",
  "message": "AMQ6287I: IBM MQ V9.3.0.0 (p930-L220526)."
}
```

On the **Incidents and alerts** page, the following example from the **Summary** parameter explains that two logs were found in the `qotd-qrcode` container. This parameter also says that a primary message code of `SSLC0008E` is contributing to the error. This alert is further explained by the **Alert details** panel. For more information, see Domain-specific log anomaly alert details.

```
Log Anomaly found 2 logs containing errors for container(s): qotd-qrcode with a primary message code of SSLC0008E
```

About log anomaly detection - golden signals

Detect log anomalies with golden signals.

Log anomaly detection - golden signals, introduced in Version 4.7.0, is an algorithm that improves signal-to-noise management of anomaly detection over previous log anomaly detection algorithms. It uses autoclassification of log templates to pre-filter logs of interest and provides administrative control to further tune the anomalies that are raised as alerts. It also improves anomaly alert explainability by providing anomaly periods, baselines, and forecasts in a chart and access to raw logs.

How do golden signals help with log anomaly detection?

Working with metric anomaly detection, **Log anomaly detection - golden signals** discovers important log message patterns and detects when they change significantly. The algorithm takes in raw log messages and converts the textual information to log patterns with both known and unknown variables and static content. These log patterns are called templates. Domain knowledge is used to assign a golden signal type to each template.

After this templatization, log data is filtered based on the golden signal type and converted to metric data. This converted metric data is passed through the metric anomaly detection algorithm to train models and infer anomalies.

IT operations teams can use the alerts that are generated to identify the log patterns that deviated from normal behavior and the time period in which the abnormalities were observed. Teams can use this information for root cause analysis to help resolve incidents.

Note: If the **Log anomaly detection - golden signals** algorithm is enabled, the **Log anomaly detection - natural language** or **Log anomaly detection - statistical baseline** algorithms cannot be active. Likewise, if **Log anomaly detection - natural language** or **Log anomaly detection - statistical baseline** is enabled, **Log anomaly detection - golden signals** cannot be active.

What is a golden signal?

Golden signals help IT operations teams to classify and organize events when they diagnose incidents.

- Seven types of golden signals exist within IBM Cloud Pak® for AIOps.
- Golden signal types can fall into the following classes:
  - Causes include latency, error, and availablity.
  - Effects include exception, traffic, saturation, and information.
  - The information type is not assigned to any class and shows up with none.

Golden signal types

- **Latency:** These events include a duration and are not related to resources.
  - Examples include transaction load, transfer, or any other metrics where the unit of measurement is time.
  - Example events include disk read/write times and I/O wait times.
- **Error:** These events are associated with a timeout, drop, failure, reset, retry, retransmission of request, packets, or a notification.
  - An error might also be understood as a limiting condition of latency. Examples include HTTP code 5xx, requests failed, crash affected user rate or count, count of errors, or any user actions with errors, error count, or rate associated with reported or requests, and event failure.
- **Availability:** These events are indicative of the availability or readiness of the resource.
  - Examples include HTTP code 4xx, host unhealthy, resource availability, readiness, and health status.
- **Exception:** These events are composed of stack trace or any indication of other errors in the log data.
  - Examples include `DSRA0010E : SQL State = 08S01, Error Code = 18,456: (FailedLoginException) Login error: com.ibm.security.krb5.KrbException`, and `status code: 6 message: Client not found in Kerberos database`.
- **Traffic:** These events represent the volume or rate of transactions that flow through the system.
  - Examples include HTTP code 2xx, traffic in/out, bytes received/sent/outstanding, disk/bytes read/write operations, throughput, service idle time, and HTTP 2xx/4xx/5xx success counts.

    - Events that represent idle time might also be classified as traffic because they might indicate that a service is experiencing idle time, indicating low traffic.
- **Saturation:** These events can be measured in terms of count or percentage and can be exhausted and throttled. These events are also resource-oriented, such as CPU, GPU, memory, swap, kernel thread, cache, containers, link, connection, process, queue, disk, and session.
    - At the infrastructure level, such resources might be as granular as CPU, GPU, memory, and disk space or as broad as a Kubernetes cluster, pod, and VM.
    - At the middleware level, such resources might include connection pool and thread pool.
    - At the application level, such resources might be user sessions.
- **Information:** Events that do not fall into any of the previous golden signal types are assigned as information logs, for example, `<<UNKNOWN TYPE>> on Success type = 0 data=true`.

Enabling or disabling the log anomaly golden signals pipeline

To enable the pipeline, complete the following steps:

1. Before you enable the log anomaly golden signals pipeline, disable the **Log anomaly detection - natural language** and the **Log anomaly detection - statistical baseline** algorithms from the AI Model Management UI.

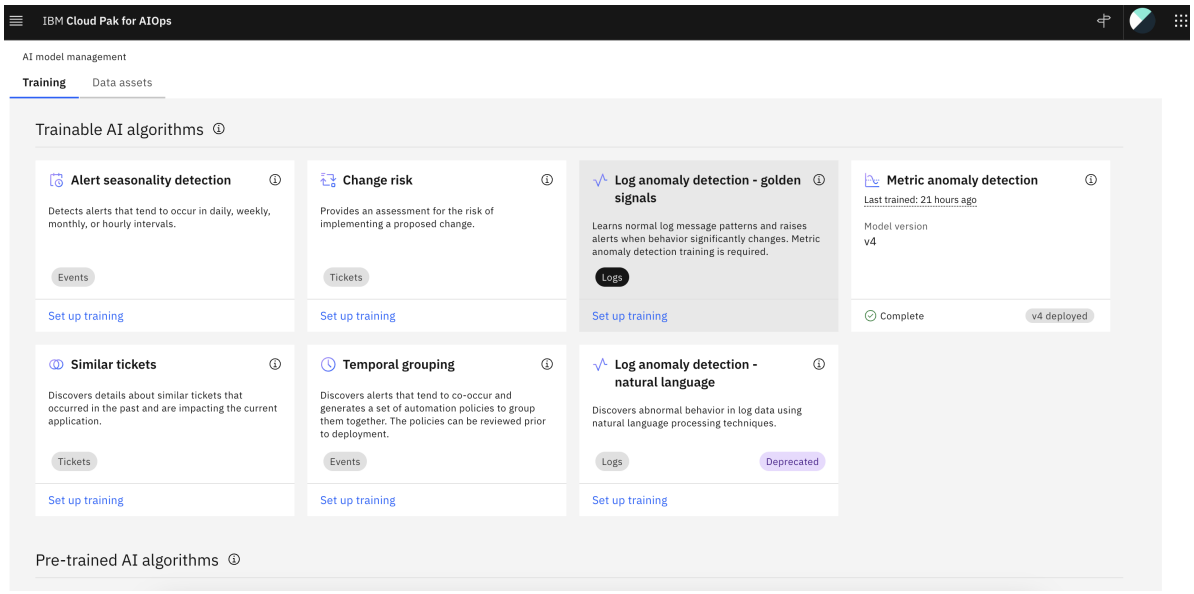


Figure. AI Model Management UI page

2. Enable the metric anomaly detection training schedule from the AI Model Management UI.
3. Click **Set up training** and complete the steps in Starting the training setup to enable the pipeline.
4. Verify that the **Log anomaly detection - golden signals** policy is enabled so the log anomaly golden signals pipeline can run as expected. The policy is enabled by default. After the pipeline runs, it produces an alert.

To disable the pipeline, complete the following steps:

1. Click the **Log anomaly detection - golden signals** tile within the AI Model Management page and delete the training definition.

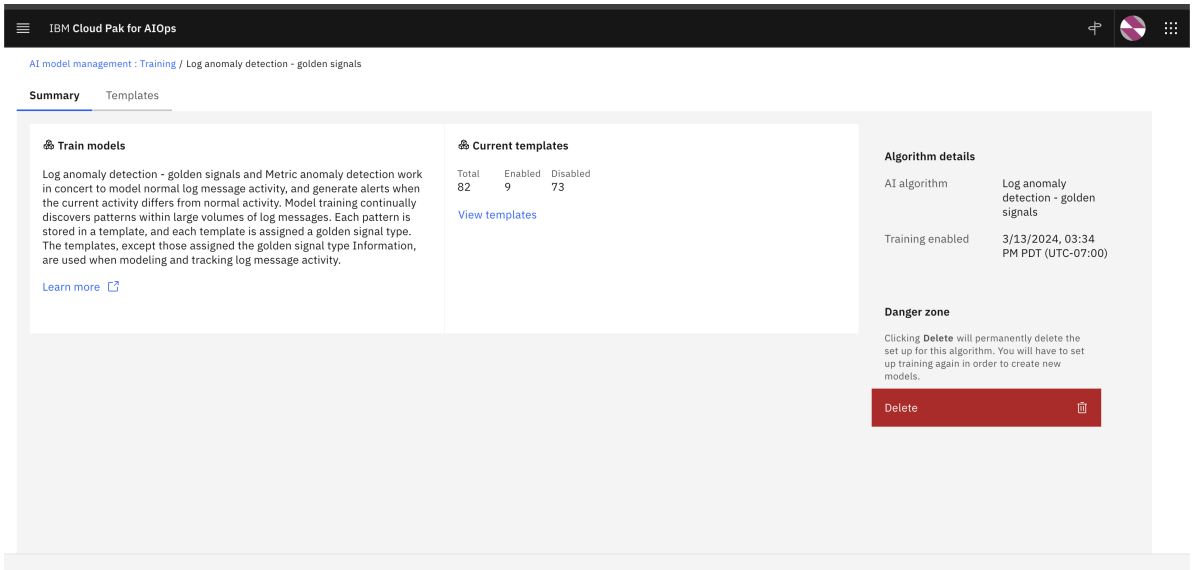


Figure. Log anomaly detection - golden signals UI page

2. Enable the **Log anomaly detection - natural language** or the **Log anomaly detection - statistical baseline** algorithms from the AI Model Management UI. Now the pipeline is configured not to use golden signals, and all incoming log lines are instead sent to the log anomaly detection pipeline.

How to set up a training definition for log anomaly detection - golden signals

For more information, see Setting up training for log anomaly detection - golden signals.

Handling large log message loads for template training

Starting with IBM Cloud Pak® for AIOps version 4.7.1, log message sampling is enabled by default. The training initially runs every 10 minutes until at least 100,000 log messages are processed. Then, the training runs once every hour. Regardless of the log message load, 450,000 logs are sampled by default for every training run.

If the log message load contains more than 10 million logs each hour, tune some parameters to override the defaults with one of the following options.

Option 1

Increase the pod resources and the **MINIMUM_SAMPLE_SIZE_PER_TRAINING_RUN** and **MAXIMUM_SAMPLE_SIZE_PER_TRAINING_RUN** values.

1. After you log in to the terminal with the **oc login** command, edit the deployment file by entering the following command:

```
oc edit deployment aimanager-aio-log-anomaly-golden-signals
```

2. Modify the values in the **container.resources** file to the following values:

```
containers:
  - resources:
      limits:
        cpu: '3'
        memory: 6Gi
      requests:
        cpu: '2'
        memory: 4Gi
```

3. In the **env** section, add or modify the following values:

```
- name: MINIMUM_SAMPLE_SIZE_PER_TRAINING_RUN
  value: '800000'
- name: MAXIMUM_SAMPLE_SIZE_PER_TRAINING_RUN
  value: '900000'
```

Option 2

Alternatively, run the training more frequently even after the initial 100,000 log messages are processed by modifying the **JOB_FREQUENCY_TIME** value.

1. After you log in to the terminal with the **oc login** command, edit the deployment file by entering the following command:

```
oc edit deployment aimanager-aio-log-anomaly-golden-signals
```

2. In the **env** section, add or modify the following value to change the number of minutes:

```
- name: JOB_FREQUENCY_TIME
  value: '10'
```

Viewing templates

Now the training setup is complete. The template training continuously runs in the background.

After the data integration collects the threshold number of log messages, the algorithm triggers template training on the data that is collected. The algorithm starts to process the log messages to categorize them into the required templates.

When template training gets completed, you can see a summary count of all the templates that got generated, as shown in the following example:



Figure. Summary count of all templates

Any template with a golden signal type of **Information** gets categorized as a **`disabled`** template, and all other templates with a golden signal type other than **Information** get categorized as **`enabled`**.

You can view more details about these templates by clicking the **Templates** tab.

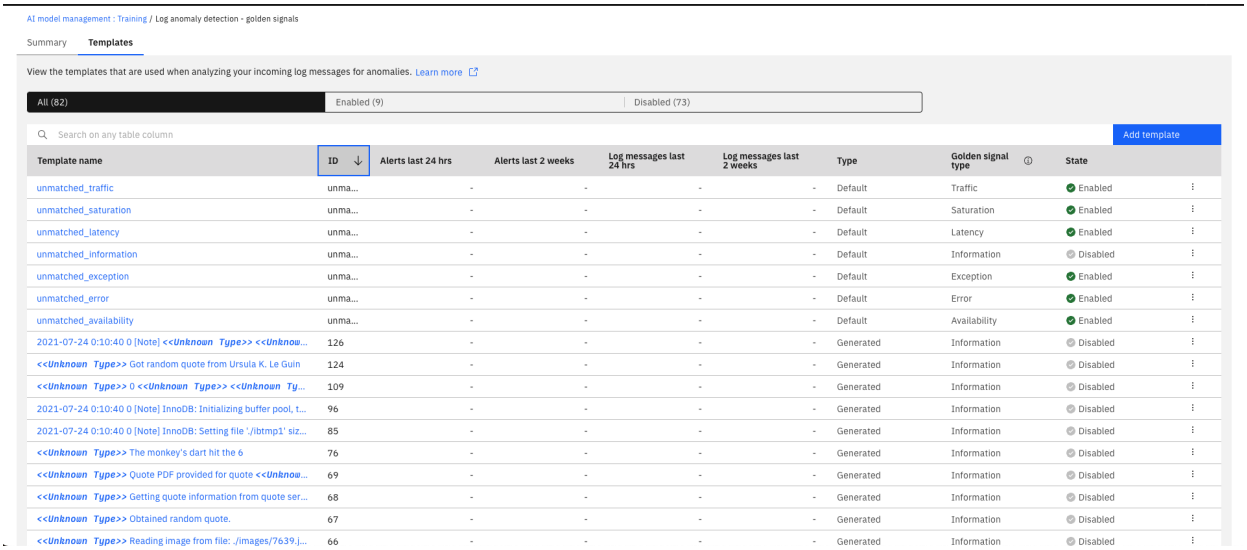


Figure. View templates

The templates are displayed in a tabular format with the following details:

- The **Template name** section shows the log pattern that matches the input log messages.
  - Logs that were not matched to any existing template are assigned to a **Template name** that includes *unmatched* in its name. A specific *unmatched* template is assigned based on the golden signal that is classified on the log content.
  - Click a **Template name** to view template **Details**.
- The **ID** numbers indicate the unique template ID value that is assigned by the training algorithm.
- The **Alerts last 24 hours** and **Alerts last 2 weeks** sections display the alert count if an alert was generated for that particular template in the last 24 hours or 2 weeks. If any log message patterns or frequencies change unexpectedly, the template triggers an alert to be generated.
- The **Log messages last 24 hours** and **Log messages last 2 weeks** sections show the count of any log messages that were collected in the last 24 hours or 2 weeks that match a specific template pattern.
- The **Type** section indicates whether the template is **Custom**, **Generated**, or **Default**.
  - You can create a **Custom** template with **Add template**.
  - A **Generated** template is created from the training algorithm.
  - All *unmatched* templates are categorized as **Default**.
- The **Golden signal type** section shows the golden signal type that is attached to a template. Open the menu for the template that you want to modify. Click **Set golden signal** to select a golden signal type, or click **Set state** to set the state.
- The **State** section shows **`enabled`** or **`disabled`** based on the golden signal type for the specific template. **`Enabled`** templates are used to track log message patterns over extended periods of time. This data is then sent to metric anomaly detection. Alerts are generated when log message patterns or their frequencies change unexpectedly.

If the counts against template patterns from the log anomaly detection - golden signals algorithm are not updated in the training UI table, enable historic alert storing in Elasticsearch to access alert counts. For more information, see Counts against template patterns are not updated in the training UI.

Viewing unmatched templates

Log messages that don't have any common pattern to them are grouped based on their golden signal type and matched to a template pattern with an `unmatched_<golden_signal_type>` format.

After enough unmatched log messages reach the threshold count, an incremental training run is triggered. The incremental training might generate new log template patterns to help match the log lines that are similar to current unmatched logs and add the logs to a template.

Click a **Template name** of an unmatched template to view template **Details**.

Viewing log messages

View up to 10,000 log messages that match a template pattern.

1. Click a **Template name**. Then, click the **Log messages** tab.
   - **Received** shows timestamps for when the messages were received.
   - **Log message** shows the log message content that matches the template pattern.
2. Click **Export to CSV file** to generate a CSV file if you want to further analyze the data.

Viewing metric anomalies in the Alert Viewer

For more information about viewing **Log anomaly detection - golden signals** metric anomalies in the Alert Viewer, see Viewing metric anomaly details.

Adding templates

If the automatically generated templates didn't capture a pattern that is important to you, click **Add template** to add your own custom template patterns.

1. From the **Templates** tab, click **Add template**. The **New custom template** page opens.
2. Enter an optional **Template name**.
3. Enter a **Sample log message** for the pattern that needs to be generated.
4. Click **Generate pattern**. Based on prior knowledge and predefined variable rules, the model generates a **Log message pattern** for the sample log message. Along with the pattern, a default **Golden signal type** is assigned based on the log message that you entered. The **State** is **Enabled** by default. However, you can modify these values.
5. To edit the generated pattern, click **Edit**. Use the menus to customize the log message pattern. For example, to change a variable type, click an **Unknown Type** menu and select the option that you want.
6. Click **Done**.
7. If the assigned category isn't correct, select a different **Golden signal type**.
8. Select the **State** as needed. If the template is **Disabled**, models aren't trained with it.
9. When your template is ready for use, click **Create template**.

Editing templates

Edit templates from the **Templates** tab or the **Details** tab.

Edit a template from the **Templates** tab:

1. Find the template that you want to edit in the table.
2. Click **Golden signal type**. A menu opens.
3. Select **Set golden signal type** to set the golden signal type of the template or select **Set state** to enable or disable the template.
4. If the template **Type** is **Custom**, you can also edit the **Template pattern**.

Edit a template from the **Details** tab:

1. In the table, find the template that you want to edit and click the **Template name**. The template **Details** tab opens.
2. Update the **Template name**, select a **Golden signal type**, or change the **State** to **Enabled** or **Disabled**.
3. If the template **Type** is **Custom**, you can also edit the **Template pattern**.
4. After your changes are complete, click **Save**.

Deleting templates

You can delete any templates except Default templates.

1. Find the template that you want to delete in the table.
2. Click **Template edit options** (the three dots at the end of the row) and select **Delete**.

About shape correlation

BETA Shape correlation is available only for technology preview, not for production usage.

Provide a metric and pattern to the shape correlation algorithm, and it can search whether another metric shows a similar metric pattern.

Enabling shape correlation

Enable the shape correlation algorithm by creating a policy.

1. In IBM Cloud Pak® for AIOps, select **Automations** and then **Create policy**.
2. In *Policy templates*, select the **Group alerts based on scope** template and click **Create**.

3. From the *Policy details* section, enter a **Policy name**.
4. From the *Condition sets* section, click **Add condition** and select **Alert property**. An alert property condition is added.
5. Select **AND** so that the alerts are grouped when all the conditions are met.
6. Enter the following values for the alert property condition:
    - Set the *Property* field to **`Value of: alert.sender.name`**.
    - Set the *Operator* field to **contains**.
    - Set the *Matches* field to **only**.
    - Set the *Value* field to **`String: anomaly-detection`**.
7. In the *Create a scope-based grouping* section, set the *ScopeID* to **`Value of: alert.sender.sourceId`**.
8. Click **Create policy**. The shape correlation algorithm is enabled.

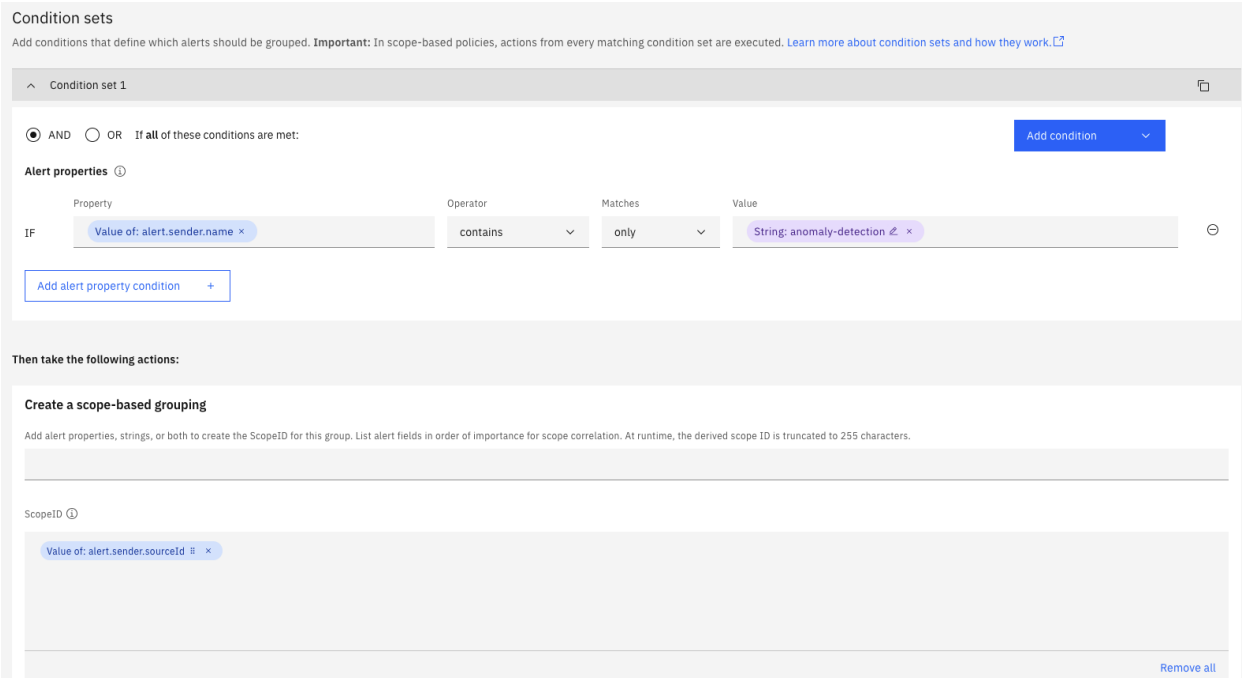


Figure. Create a policy with the appropriate values to enable shape correlation.

About metric anomaly detection

This AI algorithm generates alerts when it detects anomalous behavior in your metrics.

Description

Metric anomaly detection is composed of a set of unsupervised learning algorithms. These algorithms learn normal patterns of metric behavior by analyzing metric values at regular intervals. Then, metric anomaly detection raises anomalies or alerts when that behavior significantly changes.

Enabling metric anomaly detection helps IBM Cloud Pak® for AIOps discover metric anomalies. The first model, simple baseline, is produced after 4 data points. Simple baseline is also known as the naive baseline.

Unless otherwise stated, other models are produced when at least 7 days of data are present in the system, and training is completed. The algorithm can use up to 14 days of data to learn. After models are trained, you can be alerted to problems before services or applications are impacted.

Metric anomalies are generated based on the following analyses:

- Dynamic calculation of baseline values for each metric. For example, a metric baseline might be within the 1 to 25 range at a particular time of day. If a metric value is returned outside of this range, then a metric anomaly alert is immediately generated.
- Simple baseline, where less than 50% of data is available for a time series when training occurs, or the analytics determines that a dynamic baseline isn't a good fit for a particular time series. This baseline starts to be produced after 4 data points. After the analytics training occurs, this algorithm might raise alerts when values exist outside the simple baseline. Before the analytics training occurs, values can fall outside the simple baseline and not raise an alert.
- Flatlining, whereby the system identifies that the metric started unexpectedly returning a constant value. Models are produced when at least 3.5 days of data are present in the system, and training is completed. The algorithm can use up to 7 days of data to learn.
- Finite Domain, whereby the system detects an anomaly when a metric value elevates to a level not reached previously.
- Predominant Range, whereby the system detects an anomaly when the variation in a metric value exceeds the range within which the metric normally varies.
- A metric with learned variance and that is later found to vary significantly is also flagged as a metric anomaly alert.

Site reliability engineers (SREs) and other users responsible for application and service availability are able to display metric anomalies as alerts within the context of an incident, as described in the following topics:

- Viewing alerts in ChatOps
- Viewing probable cause alerts in an incident

They can also view metric anomaly alerts in the Alert Viewer and associated graphs that display metrics over time, baseline values of metrics, highlighting of anomalous metrics, and forecasting of metric values, as described in the following topics:

- Viewing metric anomaly details
- Metric anomaly forecasts

Note: Individual anomolous or non-anomalous metrics can be viewed on a timeseries chart using the Metric search function located on the Cloud Pak for AIOps console.

Prerequisites

Metric anomaly detection must be trained before anomalies are generated. For more information about training metric anomaly detection based on time series from integrated data sources, see Setting up training for metric anomaly detection.

To train metric anomaly detection, you need time series data that takes measurements at regular intervals.

- Up to 14 days of the most recent timeseries data are used to build the models.
- Models require 50% of their training window to have data before they are built and used to detect anomalies. For more information, see Description.
- Models that do not satisfy the 50% data availability requirement might raise anomalies with simple baseline. Anomalies from simple baseline are always produced. Simple baseline is trained with a minimum of 4 data points.

Data ingestion

Data comes from one of the following sources

- Ingest metric data by posting data using the REST API. For more information, see Metrics API and the POST entry in the associated Swagger document.
- Check that ingested data is being ingested by retrieving metric data using the REST API. For more information, see Metrics API and the GET entry in the associated Swagger document.
- If you have a metric data integration set up (Instana, Dynatrace, New Relic, Zabbix, for example), you can ingest data using the relevant integration, as described in Integration.

Automating metric anomalies to create incidents

You can automatically create incidents from a subset of anomalies, for example, if you want to allow for better prioritization. You need to create a policy to promote alerts to an incident like you would for any other alert. For more information, see Promote alerts to an incident.

The following example shows a policy to promote alerts to an incident:

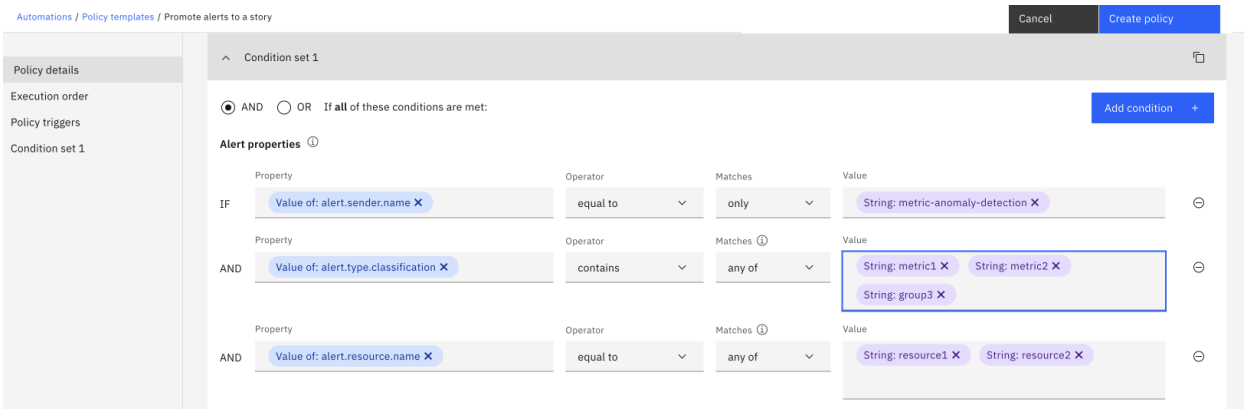


Figure. Policy template alerts

This policy means that the next time there is a new alert on **`resource1`** or **`resource2`** for **`metric1`** or **`metric2`**–or any metric in group3–then an incident is automatically created for it.

For more information, see Promote alerts to an incident.

Language support

For information about supported languages for this algorithm, see Language support.

About golden signal alert enrichment

This AI algorithm analyzes the text of an alert and assigns the golden signal that applies to it.

Enabling and disabling the golden signal alert enrichment policy

Golden signal alert enrichment is a default policy that comes with IBM Cloud Pak® for AIOps. You can enable or disable this policy.

1. From IBM Cloud Pak® for AIOps, click **Automations**. The *Policies* tab opens.
2. In the search bar, search for *story creation*. In the *Policies* table, *Golden signal and persistence story creation* appears.
3. In the *Policies* table, on the *State* column, select **Enabled** or **Disabled** to enable or disable the policy. Or click the **Golden signal and persistence story creation** table row. The *Golden signal and persistence story creation* panel opens. The panel displays details, and you can enable or disable the policy from the panel.

Viewing and editing the golden signal alert enrichment policy

The policy creates stories if specified conditions are met. View the default conditions and edit them if wanted.

1. From IBM Cloud Pak® for AIOps, click **Automations**. The *Policies* tab opens.
2. In the search bar, search for *story creation*. In the *Policies* table, *Golden signal and persistence story creation* appears.
3. Click the **Golden signal and persistence story creation** table row. The *Golden signal and persistence story creation* panel opens.
4. Click the **Specification** tab to view the conditions and the action that is taken if they are met. For more information about editing policies, see Policy editor.

Accessing the golden signal alert enrichment tile

View information about this AI algorithm from the tile.

1. From IBM Cloud Pak® for AIOps, click **AI model management**. The *Training* tab opens.
2. In the *Pre-trained AI algorithms* section, find *Golden signal alert enrichment* and confirm that it is enabled.
3. For more information about this AI algorithm, click **Golden signal alert enrichment**.

Viewing the golden signal classifications

See the golden signals that the AI algorithm assigned to the alerts.

1. From IBM Cloud Pak® for AIOps, click **Alerts**. The *Alerts* tab opens.
2. On the table row that you want more information about, click the arrow to view more details.
3. Scroll horizontally in the table until the *Golden signal* column comes into view. This column displays the golden signal that is associated with the alerts. The golden signals can fall into the following classifications:

- *Effect* golden signals indicate that users are negatively impacted because of the service.
  - *Effect- Error:* Users call the service and receive errors in the service responses, such as an HTTP status code 500. This classification might be the result of latency.
  - *Effect - Latency:* Users receive correct responses from the service, but the response is delayed beyond an acceptable time. This classification might be the result of saturation.
- *Cause* golden signals indicate situations that might eventually impact users.
  - *Cause - Saturation:* This classification indicates potential problems with system resources and might be the result of high traffic.
  - *Cause - Traffic:* This classification indicates workload. Higher traffic can lead to saturation.
- If the alert isn't significant, or the algorithm can't specify a classification, it is classified with *None*.
  - *None - Availability:* This classification indicates that the service is available.
  - *None - Information:* If the algorithm can't specify a classification for the alert, the alert is classified as *Information*.

About probable cause ranking

This AI algorithm identifies the alert with the greatest probability of being the cause of an incident.

It is enabled by default, so that information about the probable cause of a problem is automatically included in your incident details.

Description

Probable cause is a set of approaches and customizations that analyses alerts and topologies together. It then uses that understanding to try to determine the root cause of a problem. After the probable cause alert is identified, action can be taken on the resource that is generating that alert to quickly resolve the incident.

A number of techniques are used to make a probable cause determination. These are derived from topological relationships and natural language calculations. The natural language model has been built to classify the alerts in the golden signal classes, based on their summary. This information is then combined with the topological information (path calculation between the affected resources on which the alert exists). Then, the path and class are combined together to determine the rank of the alert, within the context of the other alerts in the incident.

- Keyword in the **Summary**: For example, if the word "down" appears, it might be more indicative of the root cause than the word "up".
- Severity: The highest severity normally means it's more likely to be indicative of what to look for in the root cause.
- First to occur: Set off by default. Often the first alert to arrive in a group of alerts can be seen as the root cause of the problem, but it is not always the case.
- The ability to inherit previous root causes from IBM Tivoli Network Manager (ITMN) root causes and CauseWeights that exist from scope-based alert grouping in Netcool. (For more information, see the **Event Grouping** section here.)
- Scoring config: The following options contribute to probable cause determination:
  - `PathCalculationEnabled`: This option uses topological data to calculate the path between sources. You must create the topological resource group. For the resource group, set the alert correlation to `true`. This setting allows alerts to be correlated by toplogical correlation. When a problem is

encountered in the system, a path is calculated between all the alerts in the group. This path is the shortest number of hops. After the path is calculated, a path score is calculated based of the edge types in the system.

- **`wordEnabled`**: You can add or remove words from the system with an API. If a specified word is in the summary field of an alert, the word score is added.
  - Another option exists for case sensitivity.
  - The higher the score, the more likely a problem is to be the root cause.
  - The word score is the sum of the score. For example, if 2 words in the summary are found, and one word has a value of 100, and the other has a value of 10, the total word score is 110.
- **`SeverityEnabled`**: The problem severity is accounted for. By default, the score is the severity multiplied by 10.
- **`causeScoreEnabled`**: If the field **`CauseWeight`** is present in the details section of an alert then the **`causeScore`** is added to the total score in **`CauseWeight`**. If this field is not present in the details section, there is no effect. This field cannot be changed.
- **`itnmproccessing`**: If this option is turned on and ITMN marked an alert as the root cause(**`NmosCauseType=1`**), then **`itnmcauseweight`** is added to the score.
- **`firstBoost`**: If **`firstBoost`** is set to true, the amount that is specified in the **`firstBoostWeight`** is added to the alert that has the lowest **`firstoccurrence`** time.

The techniques work together to determine the root cause. One technique alone does not work in an environment as an alert might, or might not, have an associated topology.

Prerequisites

For topology and natural language probable cause, the alert data must have topological data and must also be topologically co-related. For more information, see Topological grouping.

Viewing rankings

Probable cause rankings from 1 (highest) to 3 (lowest) in order of probability can be viewed in the Incident overview UI.



Figure. Probable cause Incidents Overview

They also display (showing rankings greater than 3) in the Alerts tab of the Incidents viewer.



Probable cause Alerts tab

Rankings can also appear in ChatOps notifications. (For more information, see Probable cause in ChatOps.)

Language support

For information about supported languages for this algorithm, see Language support.

Related topic

- Configuring probable cause
- Probable cause customization APIs

About similar tickets

This AI algorithm discovers details about similar tickets that occurred in the past and are impacting the current application.

Important: Starting with IBM Cloud Pak® for AIOps version 4.1.0, similar incidents, such as from a ServiceNow integration, are called similar tickets.

When an incident occurs, it can be helpful to review details for similar tickets from the past to help determine a resolution. Similar tickets is an unsupervised learning algorithm that aggregates information from past similar tickets. It can also extract the steps used to fix previous tickets, if documented.

Training this AI algorithm will help you discover historical tickets to aid in the remediation of current problems. To train this algorithm, you must complete some initial setup tasks, as described in Setting up training for similar tickets.

Site reliability engineers (SREs) and other users responsible for application and service availability are able to access tickets in their ChatOps interface and within each ticket view historical similar tickets, as described in Managing incidents.

Prerequisites

The Similar tickets algorithm requires ticket data from ServiceNow. Before attempting to set up training for this algorithm, ensure that you have created a functioning data integration to ServiceNow. For more information, see Creating a ServiceNow integration.

After training is complete, enable the similar ticket policy in the **Policies** page. For more information, see Setting up training for similar tickets.

Language support

For information about supported languages for this algorithm, see Language support.

Language support

The AI algorithms in the AI Model Management are able to analyze and train on a limited set of non-English language data. In an operations environment, languages often occur in combination. For example, English and Spanish, or English and German. Analysis and training on the following non-English languages data are supported: German, and Spanish.

The use of mixing languages in a feed of records is supported. However, mixing languages in an individual record is not supported. For example, a feed of support tickets can contain a mixture of English and Spanish tickets, but a mixture of languages in a single support ticket is not supported.

If data in an unsupported language is detected during pre-training of any of the algorithms, **Summary** *Step 1 - Precheck data* will contain the warning message **Data needs improvement**.

Click **View recommendations** and, if unsupported langage is a problem, the following instructions appear :

```
We have detected that a portion of this data is in an unsupported language and could impact the quality of this model.

We recommend that you remove any data containing unsupported languages. Then, run training again.
```

Supported language combinations

The following table lists supported language combinations for each of the AI algorithms.

Table. Supported language combinations

Algorithm	Data used	Supported language combinations	Not supported
Alert Grouping	Events	Not language sensitive	N/A
Change risk	Tickets	English and Spanish, English and German, English and Spanish and German	N/A
Log anomaly detection (deprecated)	Logs	English and Spanish, English and German, English and French, English and Italian, English and Korean	N/A
Log anomaly detection - golden signals	Logs	English only	N/A
Metric anomaly detection	Metrics	Not language sensitive	N/A
Probable cause	Events	English only	If event data in Spanish or German is processed by this algorithm, the ranking of probable cause in an incident might be unreliable or missing.
Similar tickets	Tickets	English and Spanish, English and German, English and Spanish and German, English and French, English and Italian, English and Japanese	N/A

Preparing AI algorithms

Use this information for how to prepare AI algorithms. Some algorithms require setting up training, training, and deploying, such as trainable AI algorithms. Others simply need to be enabled, and others still are enabled by default.

For more information about preparing AI algorithms, see the following topics:

- About algorithm types
- About pre-trained AI algorithms
- Setting up training for trainable AI algorithms

About algorithm types

The AI algorithms can be categorized into two types: ones that analyze your data in real time, and ones that must first be trained to generate an AI model.

The following are the two types of AI algorithms:

- Trainable AI algorithms
- Pre-trained AI algorithms

Trainable AI algorithms

These algorithms require user interaction as they must first be trained to generate an AI model. Models are well-defined computations that are formed as a result of an algorithm that takes some value, or set of values, as input and produces some value, or set of values as output. When deployed, it is your model that acts on the data and derives insights.

You must first set up training for these AI algorithms, and then train that algorithm to generate a model. The following table lists the trainable AI algorithms in alphabetical order, along with a link to the relevant training setup task.

Algorithm	Action
Change risk	Set up training
Metric anomaly detection	Set up training
Natural language log anomaly detection	Set up training
Similar tickets	Set up training
Temporal grouping	Set up training

Pre-trained AI algorithms

These algorithms analyze your data in real time and derive insights without the need to generate a model.

You do not need to set up training for these AI algorithms. You can enable them. In some cases, these algorithms are automatically enabled for you. The following table lists the online algorithms in alphabetical order, along with their default state, and whether you can take action on them to change that state.

Algorithm	Default state	Action
Probable cause	Enabled	None
Scope-based grouping	Enabled	None
Statistical baseline log anomaly detection	Enabled	Disable or Enable
Topological grouping	Enabled	None

About pre-trained AI algorithms

These algorithms don't require models to derive insights. They are enabled by default. You do not need to set up training for these AI algorithms; they are automatically enabled for you, except for the statistical baseline log anomaly detection algorithm, which is enabled by default, but can be disabled.

Click here for more information about the pre-trained AI algorithms:

- About probable cause
- About scope-based grouping
- About statistical baseline log anomaly detection
- About topological grouping

About probable cause

The probable cause AI algorithm is enabled by default and cannot be disabled. To indicate this, the word **Enabled** appears in grayed out text at the bottom of the "Probable cause" tile in the AI Model Management.

For more information on the probable cause AI algorithm, see About probable cause ranking.

About scope-based grouping

The scope-based grouping algorithm is enabled by default and cannot be disabled. To indicate this, the word **Enabled** appears in grayed out text at the bottom of each of the event grouping tiles in the AI Model Management.

For more information on the event grouping AI algorithms, see About scope-based grouping.

About statistical baseline log anomaly detection

Leave this AI algorithm turned on to see details about statistical baseline anomalies in your ChatOps feed and in your Alert Viewer.

Statistical baseline log anomaly detection is an unsupervised learning algorithm that automatically detects unusual patterns in logs and notifies you when they occur. Data that is used for analysis is updated every 30 minutes so this algorithm provides value quickly. This algorithm supplements the natural language log anomaly detection algorithm by reporting extra log anomalies to your ChatOps interface and to the Alert Viewer.

By default, this algorithm is turned on and it is recommended that you leave it turned on. If you turn it off, you won't see details about statistical baseline anomalies in either your ChatOps feed or in your Alert Viewer.

Note: If both statistical baseline and natural language log anomaly detection algorithms are active, then any log anomalies discovered by both will be reconciled, so that only one alert is generated.

About topological grouping

The topological grouping AI algorithm is enabled by default and cannot be disabled. To indicate this, the word **Enabled** appears in grayed out text at the bottom of each of the event grouping tiles in the AI Model Management.

For more information on the event grouping AI algorithms, see About topological grouping.

Setting up training for trainable AI algorithms

AI algorithms analyze different kinds of data and produce AI models, which automate research, reducing the need for manual work in solving problems. IBM Cloud Pak® for AIOps training generates models for a number of different AI algorithms. You can train the models for each of these AI algorithms independently of each other.

For more information about setting up training for trainable AI algorithms, see the following topics:

- Setting up training for alert seasonality detection
- Setting up training for change risk
- Setting up training for log anomaly detection - golden signals
- Setting up training for natural language log anomaly detection
- Setting up training for metric anomaly detection
- Setting up training for similar tickets
- Setting up training for temporal grouping

Setting up training for alert seasonality detection

Set up alert seasonality detection to train policies. Then, choose how to deploy the policies.

For more information about alert seasonality, see Alert seasonality detection algorithm.

Before you begin

To train a definition for alert seasonality detection, you need event data. By default, this AI algorithm can load up to 90 days of historical event data. A set of data creates a set of policies. For more information, see Prerequisites.

This AI algorithm can start to discover seasonal trends after a pattern is learned, which is typically ranging between days or months. For example, if an alert instance occurs on each weekday at two PM, the seasonality algorithm will detect a daily seasonal pattern after one week. However, if this only occurs on Tuesdays at a given time, then it will take multiple weeks to detect this pattern in the alert data.

About this task

There are multiple parts to this task:

- Starting the training setup
- Scheduling the training
- Select data
- Deciding how to deploy the training
- Running the training again
- What to do next

Starting the training setup

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. In the **Training** tab, click **Set up training** within the **Alert seasonality detection** tile under the Trainable AI algorithms section.

Scheduling the training

Decide whether to run the training set up on demand or to schedule it to run on an ongoing basis.

Note: When you schedule each of the trainable AI algorithms, it is strongly recommended to schedule training on quieter periods of the day, and to stagger them out. For example, schedule training runs at least one hour apart from each other. This helps to spread the analytical processing load out.

1. Proceed as follows:
   - To run the training set up on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training set up on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).
3. Click **Next** to move to the next panel.

Select data

Select a date range (either Preset or Custom) for the event data that will be included in the AI training. Including additional data typically improves the effectiveness of the AI models and policies that are generated by the training.

Deciding how to deploy the training

The option to review training results before deployment is disabled for this algorithm. However, you can choose when to deploy the policies that are generated by training the algorithm.

- To review and manually deploy the model, set **Deployment type** to **Manual**.
- To automatically deploy the model, set **Deployment type** to **On completion**.

Click **Done** to save the training setup for the algorithm.

Running the training again

- If you run the training again with the same data, expect to receive the same policies.
- If you run the training again with different data, you might receive new policies.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Launching the training.

Setting up training for change risk

Set up training for this AI algorithm to determine how likely it is that a proposed change would cause a problem.

About this task

Hundreds of changes can affect an application during its lifecycle. Change risk is an unsupervised learning algorithm that correlates historical change data with historical incident data and determines how likely it is that a specific code change would cause a problem. Using the assessment score provided, you can determine how safe it is to proceed with the change. Training this AI algorithm helps you ensure that risky changes for an application are assessed before deployment.

Training setup of the algorithm is done by completing some setup tasks. When the training setup is complete, you need to go to a different part of the AI Model Management to train the model.

Before you begin

To train this algorithm, you need change request and incident ticket data from ServiceNow. If you enable the collection of historical data when you connect to ServiceNow, then data for the selected timeframe is available to train on.

To provide the best data for change risk, you need a mix of successful and failed change tickets. Around 10,000 change tickets, with a minimum 2% failure rate, is ideal. However, models can also be created with much less.

Closed change ticket count requirements

For a closed change ticket count less or equal to 150, you need at least three problematic tickets. For a closed change ticket count greater than 150, you need at least 2% of change tickets to be problematic. A *problematic change ticket* is defined as a change ticket that can "either be associated with incident tickets or have an 'unsuccessful' value set in the change ticket close code field".

Failure rate

If you provide data with a change ticket failure rate of less than 2%, the training fails with a **Needs improvement** data quality error. For more information, see Data quality messages.

To check whether you meet these requirements, review your ServiceNow integration. For more information, see Creating a ServiceNow integration.

To resolve the issue, you can optionally configure the failure percentage. Edit the deployment yaml of the ai-platform server and reduce the value for the failure percentage to do so. The following example shows the failure percentage set to 1.9.

```
oc set env deploy/aimanager-aio-ai-platform-api-server MIN_LABEL_PERCENT="1.9"
```

Where **MIN_LABEL_PERCENT** is the failure percentage.

Note Model quality deteriorates with a MIN_LABEL_PERCENT less than 2.0

Running the task

There are multiple parts to this task:

- Starting the training setup
- Checking data integrations
- Deciding how to run the training
- Deciding how to deploy the training
- What to do next

Starting the training setup

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. Under the Training tab, in the Trainable AI algorithms section, click the **Change risk** tile.

Note: If you want to edit configurations for algorithms after the first time set-up, you can simply click the algorithm tile.

Checking data integrations

Review the information on the page. It explains what this AI algorithm is, how it helps in your production environment, and provides a list of integrations that are needed to generate a model.

1. On the Integrations, check that at least one integration is listed.

   Note: If no integrations are listed, or you expected to see more integrations, click **Go to Integrations** to modify your integrations. For more details, see Creating a ServiceNow integration.

2. Click **Next** to move to the next section.

Deciding how to run the training

Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.

1. Proceed as follows:
   - To run the training setup on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training setup on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).
3. Click **Next** to move to the next section.

Deciding how to deploy the training

Decide whether to review the results of training before deploying the model, or to have the system automatically deploy the model.

1. Proceed as follows:
   - To review the results of training before deploying the model, ensure that **Deployment type** is set to **Review first**.
   - For the system to automatically deploy the model, set **Deployment type** to **On completion**.
2. Click **Save** to save the training setup for the algorithm.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Launching the training.

Enabling full data profiling report in Change Risk

Note: To enable this feature, a user must have administrator privilege.

The following procedure generates a JSON data profiling report. This JSON report provides information about word count and parts of speech in the tickets that are used for training, which can be helpful in understanding model behavior.

1. Obtain the change-risk algorithm of the corresponding namespace as a yaml file.

   ```
   oc -n get algorithm change-risk -o yaml >temp.yaml; cp temp.yaml backup.yaml
   ```

2. Use a text editor to extract the value of the **manifestBase64** field from **temp.yaml**, then use **base64** to decode it.

   ```
   echo <value of manifestBase64> | base64 -d > manifest.tmp
   ```

3. Edit **manifest.tmp** to add **GENERATE_FULL_PROFILING_REPORT** to the environment. Set it to **True**.

   ```
   env:
     - name: GENERATE_FULL_PROFILING_REPORT
   ```

```
    value: "True"
```

4. Encode the updated file. Update the value of **manifestBase64** in **temp.yaml** with this result.

```
cat manifest.tmp | base64
```

5. Apply the modified **temp.yaml** on the cluster.

```
oc apply -f temp.yaml
```

6. When enabled, the full data profiling report will be generated from the next run of the change-risk training.

Accessing full data profiling report

The report of the corresponding training is stored in Elasticsearch under the index name **change-risk-data-profiling-<Version-Number>** Access the *version-number* from the AI model management tool when you run training.

Example: if Version number is **v1** then the report is in **change-risk-data-profiling-v1**

Run the following command:

```
export ELASTIC_SET="$(oc get statefulset -l app.kubernetes.io/managed-by=ibm-elasticsearch -o
jsonpath='{.items[0].metadata.name}')"
oc exec "statefulset/${ELASTIC_SET}" -- curl -XGET -k --fail "http://127.0.0.1:19200/change-risk-data-profiling-
v1/_search?pretty"
```

Training a Change Risk model

Note: Make sure ServiceNow Integration is first set to *Historical* mode.

1. Go to IBM Cloud Pak for AIOps console.
2. Click **AI Model Management**.
3. Find the Change risk algorithm under the *Training* tab.
4. Click **Set up training**.
5. Verify your ServiceNow integration is listed.

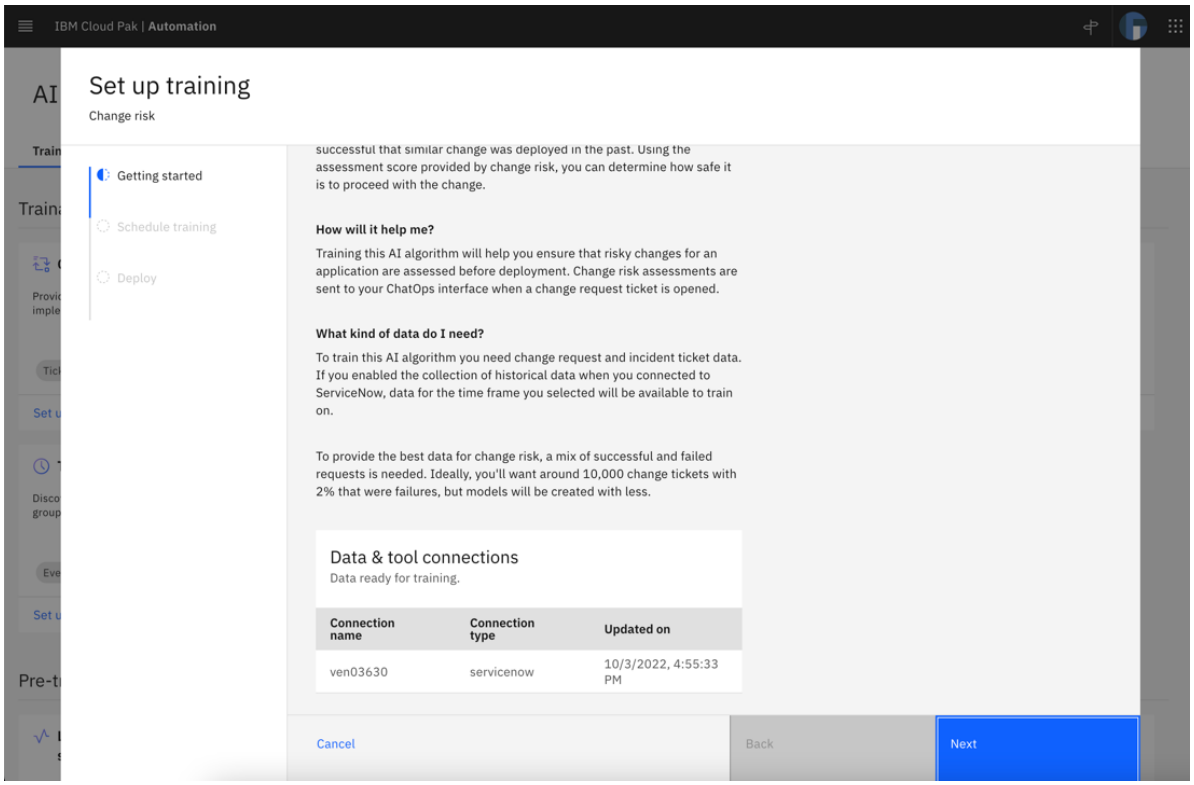


Figure. Set up training page

6. Click **Next** for **Schedule training**. No need to schedule the training. It starts training in a few moments.
7. Click **Next** for **Deploy**. Set to deploy model *On completion*.
8. Click **Done**.
9. View details on the Change risk algorithm.

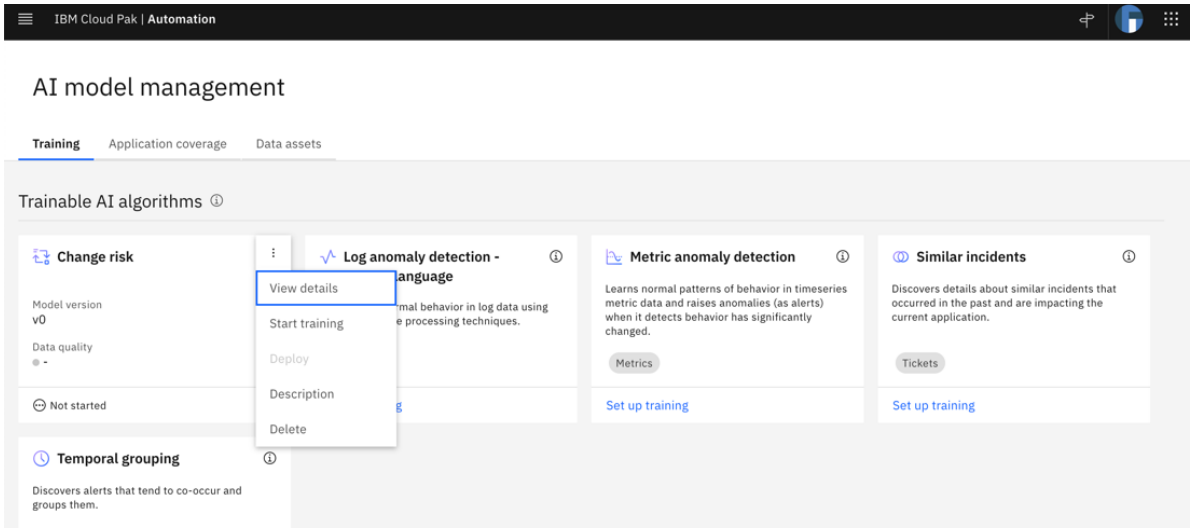


Figure. View details menu option

10. Click **Start training**. IBM Cloud Pak® for AIOps will first check data quality before training.

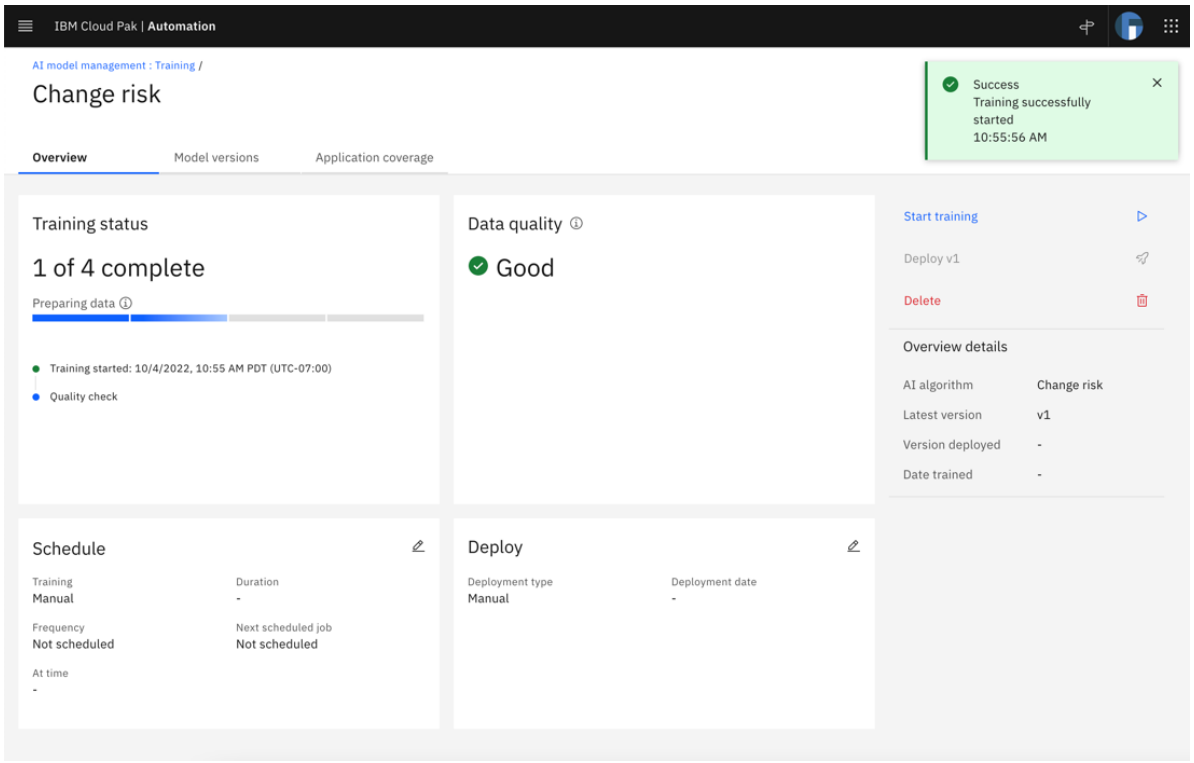


Figure. Data quality check

11. Deploy model to enable change risk assessment functionality. Remember to change ServiceNow Integration to Live data mode

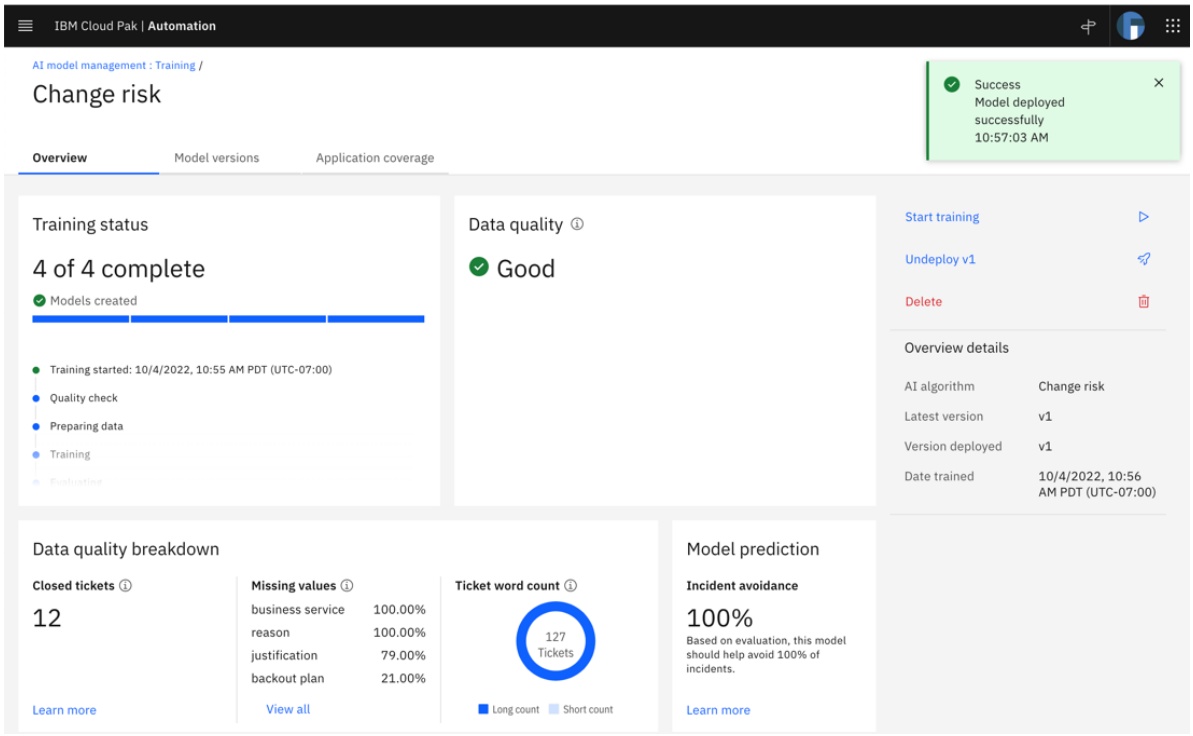


Figure. Model deployed successfully

Configuring change risk

The more data that you provide, the better the quality of your change risk analysis. At a minimum, your AI modeling requires about 2 percent of change tickets to be problematic. If you train with 20,000–50,000 tickets with a 2–5 percent failure rate, you would yield more robust results. Training does not start automatically when you reach that threshold. If you selected *Live data for initial AI training* in your ServiceNow integration, you would need to manually disable the integration when you reach the minimum threshold of the training model.

The following change ticket fields are used for training. To help produce a good model, provide various values for these change ticket fields:

- *sys_domain*
- *business_service*
- *type*
- *short_description*
- *impact*
- *reason*
- *justification*
- *description*
- *backout_plan*

For more information about training change risk AI models, see Configuring change risk.

Change risk assessment

If all of these steps are complete, when someone opens a change request ticket, a *Change risk assessment* appears in your proactive ChatOps channel. (For more information about channels, see ChatOps insight management). Depending on the risk level, you might also receive a second notification: *Most relevant change requests*.

The *Change risk assessment* includes the ticket number and a link to the change request ticket in ServiceNow and message links to the approver and creator of the ticket.

Confidence percentage

Change risk is scored on a 0–100 percent scale. The higher the percentage, the higher the confidence in the prediction. The amount of information that is provided depends on the severity of the risk.

Most relevant change requests

If the outcome is deemed as high risk change, IBM Cloud Pak® for AIOps provides links to historical relevant problematic tickets. You can view these tickets, with their accompanying explanations, to gain a better understanding as to why the perceived risk is high.

Three types of relevant change tickets can be generated. Each can be identified by a particular term:

1. *successfully completed* – provides details of successfully completed change and incident that is caused by that change. Includes timeline and closing remarks.

2. *closed as being unsuccessful* – occurs when the ticket was not resolved and includes closing remarks, which can indicate that further investigation was undertaken. Also, gives information why this ticket was deemed relevant to new change request.
3. *closed as being successful but change mentioned in incident(s)* – provides details of successfully completed change that was later mentioned in one or more incident tickets.

Setting up training for log anomaly detection - golden signals

Log anomaly detection - golden signals uses the Metric anomaly detection pipeline for log data sets. It takes in raw logs and converts the textual information to log patterns with variables (both known and unknown) and static content.

The patterns are also referred to as templates. Log data after templatization is filtered, based on golden signals, and converted to metric data.

Prerequisites

Take the following actions.

1. To set up training for this algorithm, you need log data that shows normal operational behavior. If your logs include anomalous situations, such as a critical incident or a technical glitch, filter out this anomalous data before you train the algorithm.
2. In the Cloud Pak for AIOps home page, click the navigation icon (four horizontal bars) to go to the main navigation menu.
3. In the main navigation menu, click **Operate** > **AI Model Management** to open the AI Model Management.
4. Disable the **Log anomaly detection - natural language** and **Log anomaly detection - statistical baseline** algorithms.
5. Create a training definition for *Metric anomaly detection* with a training schedule.

Starting the training setup

1. Within the **Log anomaly detection - golden signals** tile, click **Set up training**.

   Note: If this AI algorithm is already set up for training, **Set up training** is not available. For more information about changing the algorithm training setup, see Editing the training setup for an algorithm.

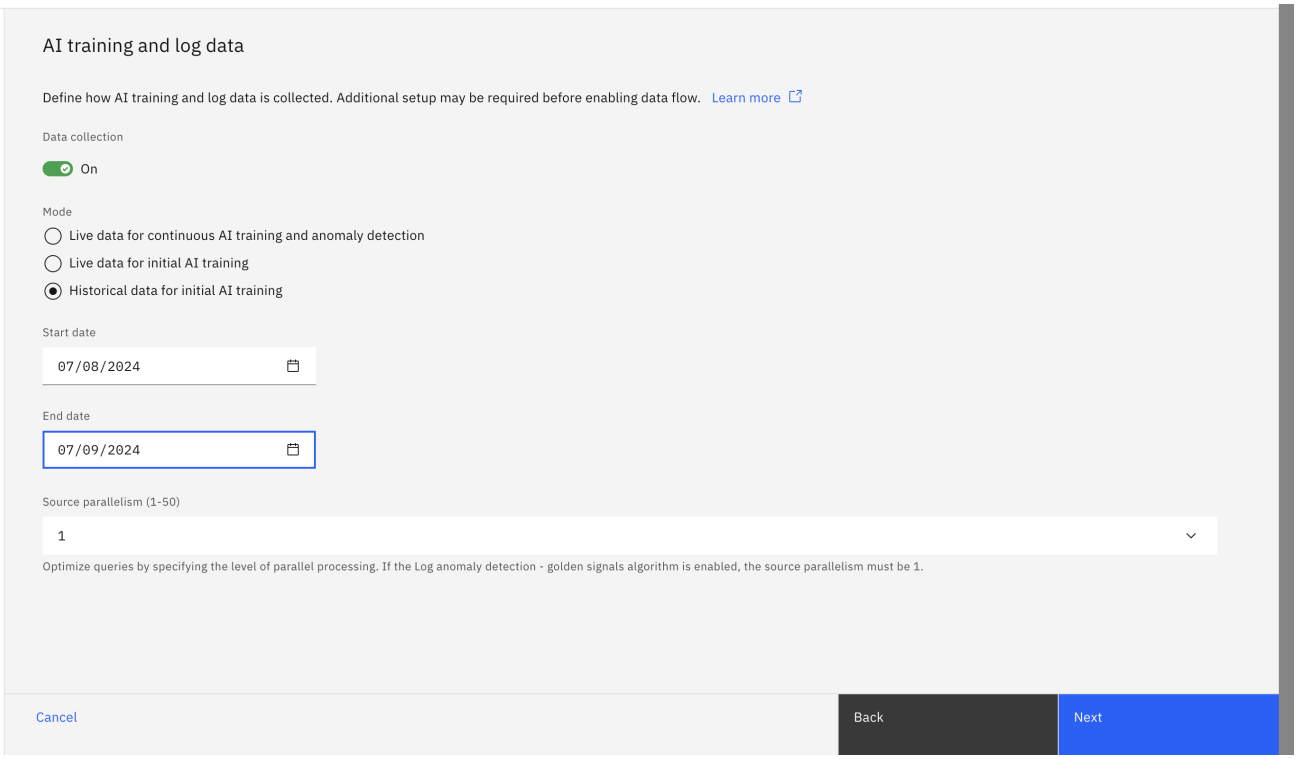
2. Click **Next** to open the **Getting started** panel. The panel explains this AI algorithm and how it helps in your production environment. The panel also provides a list of integrations that are available to generate a model. At least one integration must exist in this list for the model to collect data and start training.
3. If no integrations are listed or if you expected to see more integrations, click **Integrations** to modify your integrations.
4. Add at least one integration by using the Integrations area of IBM Cloud Pak for AIOps. Alerts are generated after the following conditions are true:

   - At least 100,000 log messages are processed for template quality.
   - The logs span at least 3.5 days, and metric anomaly detection training is run on that data.

   On the Integrations, check that at least one integration is listed.

   Note: If no integrations are listed, or you expected to see more integrations, click **Go to Integrations** to modify your integrations.

   Note: Make sure that all the log data integrations that are created in the **Integrations** section are in **Live data for initial training** or **Historical data for initial AI training** mode. If you choose **Historical data for initial AI training**, the source parallelism must be set to **1**. You can set up log data integration before or after you enable log anomaly golden signals.



5. Click **Next**.
6. In the **Training setup** panel, click **Done**. The training begins.

What happens next

Now the training setup is complete.

The template training continuously runs in the background and in parallel data is sent to Metric anomaly detection.

If you chose the *Historical data for initial AI training* mode, and the integration status displays as *Done*, edit your integration and select *Live data for initial AI training* mode for continuous AI training and anomaly detection.

Upon receiving sufficient data, Metric anomaly detection trains models and generates alerts if any anomalous logs are detected.

Disabling the training

You can turn off the algorithm by toggling the switch to **Disabled** in the *Log anomaly detection - golden signals* tile in the AI model management UI.

To restart, toggle the switch to **Enabled**.

The training pipeline sets `la_golden_signals_enabled_flag` to `True` in **Elasticsearch**. **Flink jobs** is now enabled to start with log anomaly golden signal components.

To initiate the change, the training pipeline restarts all the **Flink jobs**.

Setting up training for log anomaly detection - natural language

Set up training for this AI algorithm to detect unusual patterns in your logs and notify you by generating an alert when they occur.

About this task

For more information on the how the different types of log anomaly detection work, see About log anomaly detection algorithms.

To set up training for this algorithm, you need log data that shows normal operational behavior. If your logs include anomalous situations, such as a critical incident, technical glitch, or some other change that deviates from the norm, you can filter that data out before training the algorithm.

Training setup of the algorithm is done by completing some setup tasks. When the training setup is complete, you need to go to a different part of the AI Model Management to train the model.

Note If both natural language and statistical baseline log anomaly detection algorithms are active, then any log anomalies that are discovered by both will be reconciled, so that only one alert is generated.

Before you begin

When providing log data, select a date range that includes at least 10,000 lines of messages for each resource that you want insights from. An ideal amount would be around 100,000 lines. When selecting longer date ranges, consider that it might take more time to process the data and might have system impacts.

To check that you have met these requirements, review the prerequisites for this AI algorithm. For more details, see the rows that are associated with Log anomaly detection in the prerequisites table on this page: Prerequisites.

In addition, you must perform the following tasks:

- Configure resources such as CPU and memory to ensure good training performance. For more details, see Configuring resources for log anomaly detection

Running the task

There are multiple parts to this task:

- Starting the training setup
- Checking data integrations
- Specifying which data to train on
- Excluding irregular data
- Deciding how to run the training
- Deciding how to deploy the training
- What to do next
- Viewing the running jobs

Starting the training setup

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. In the AI algorithms tab, click **Set up Training** within the **Log anomaly detection - natural language** tile.

Note: If this AI algorithm has already been set up for training, then the word Set up appears in grayed out text at the bottom of the tile. In this case, you can edit the algorithm training setup, as described in Editing the training setup for an algorithm.

Checking data integrations

Review the information on this page. It explains what this AI algorithm is, how it helps in your production environment, and provides a list of integrations that are needed to generate a model.

1. On the Integrations, check that at least one integration is listed.

   Note: If no integrations are listed, or you expected to see more integrations, click **Go to Integrations** to modify your integrations. For more details, see this page: Prerequisites.

2. Click **Next** to move to the next panel.

Specifying which data to train on

Review and modify your data integrations, if necessary.

1. Review the Data assets table.

   This table lists the data integrations from the **Getting started** panel, and provides the following information for each data integration:

   - How is the data being collected? Options are Live or Historical.
   - Is the data flow on?

   To make any changes, click **Go to Integrations** to modify your integrations. For more details, see this page: Prerequisites.

   Optionally remove a data integration by clicking the checkbox to the left of the data integration name, and clicking the Delete icon that appears.

2. Specify a date range to train on. Select **Preset** for a preset option or **Custom** for a range of dates.

   Tip If you are planning to schedule the execution of the training for this algorithm then you must select one of the preset options that retrieves a fixed window of data relative to the current time.

   - **Preset**: Click this option and select one of the following time ranges. Note that 14 days of log data is the maximum amount that is supported with the default values.

     - Past day
     - Past 3 days
     - Past 7 days
     - Past 10 days
     - Past 14 days

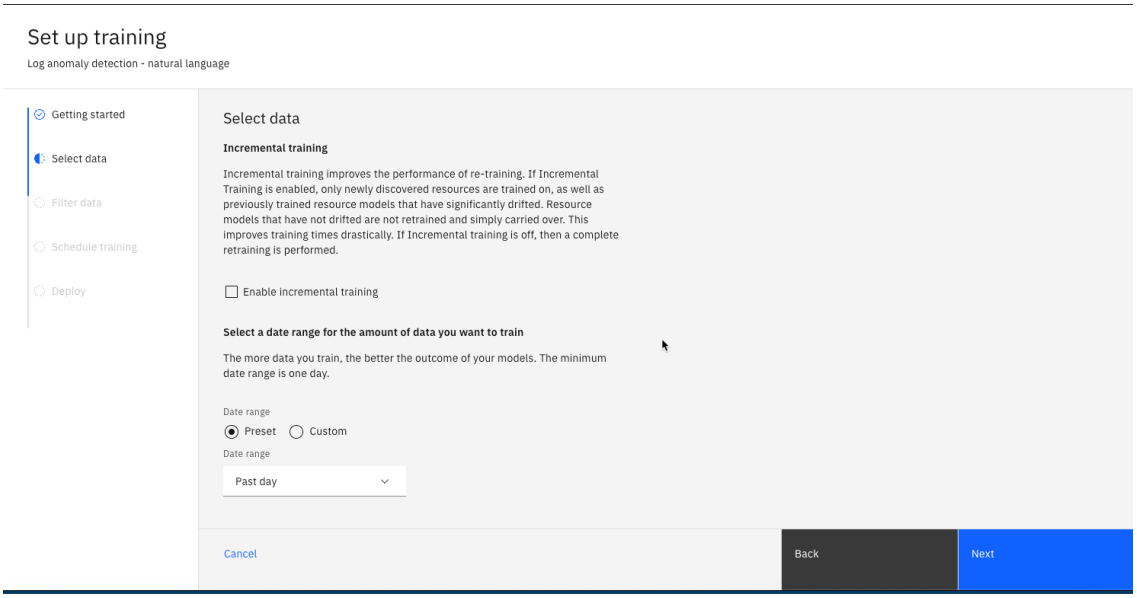


Figure. Schedule training

- **Custom**: This is an easy and quick option to use if you are not concerned with training models on data from a very specific window of time. Click this option and specify a fixed date range.

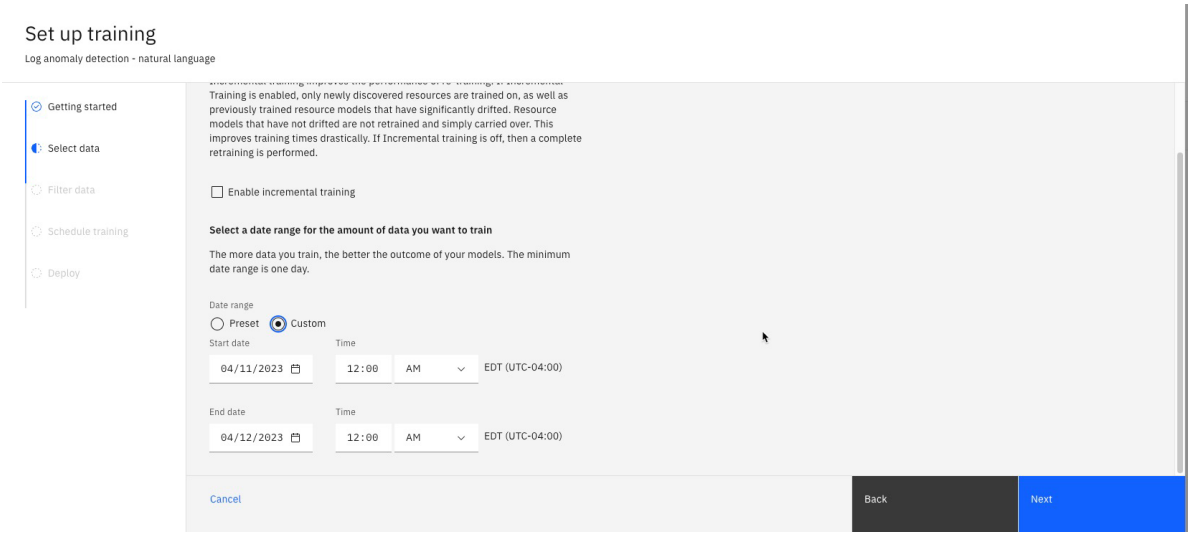


Figure. Custom

Note When you set custom dates and times, you must be aware of the following:

- You have the option to specify start and end times in local time or UTC. By default, the time is in local time.
- Ingested log data is stored in Elasticsearch in UTC.
- To ensure that you select the correct data intervals, specify both the start and end times in this panel in UTC.

For example, if you select November 15 12:00 AM through December 15 11:59 PM UTC time, the training window includes data from the start day of November 15 through to the end of the day on December 15. If you instead select December 15 12:00 AM, then the training window includes data from November 15 through December 14 11:59 PM.

- If you choose to use local time, you need to adjust the start and end date and time to match the period for which log data is available in ES.

3. Click **Next** to move to the next panel.

Filter data

Filter data from the training. This includes data associated with event storms, maintenance windows, major events, and other unusual outages. This will enable the Log anomaly detection algorithm to baseline normal behavior and record expected log messages.

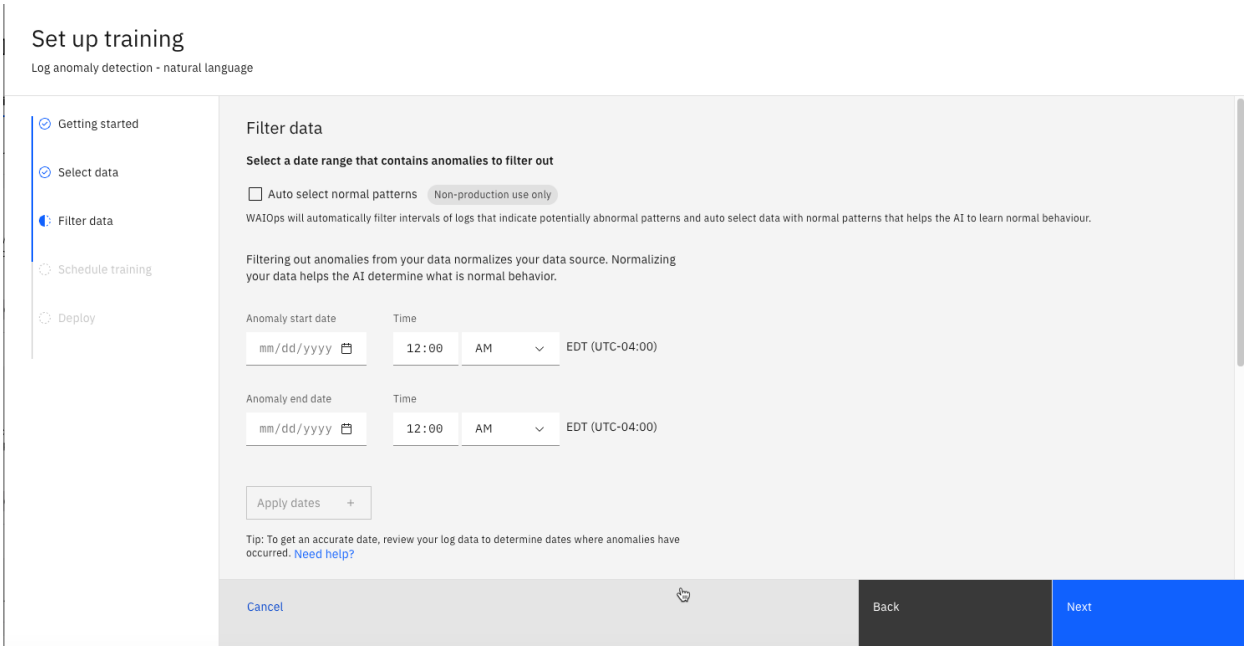


Figure. Log anomaly

1. (Optional) Select date range of irregular data to filter out. Then click **Apply dates**. Your selection appears in the **Applied dates** table below.

   Tip Go to your data source to view your log data. This will help you determine when there were incidences of irregular data.

   Note When you set custom dates and times, you must be aware of the following:

   - You have the option to specify start and end times in local time or UTC. By default, the time is in local time.
   - Ingested log data is stored in Elasticsearch in UTC.
   - To ensure that you select the correct data intervals, specify both the start and end times in this panel in UTC.

     For example, if you select November 15 12:00 AM through December 15 11:59 PM UTC time, the training window includes data from the start day of November 15 through to the end of the day on December 15. If you instead select December 15 12:00 AM, then the training window includes data from November 15 through December 14 11:59 PM.
   - If you choose to use local time, you need to adjust the start and end date and time to match the period for which log data is available in ES.

2. (Optional) Repeat step 1 for each date range of irregular data you want to apply.

   The data set is listed in the **Data sets** table at the bottom of the screen.

3. Click **Next** to move to the next panel.

Schedule the training

Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.

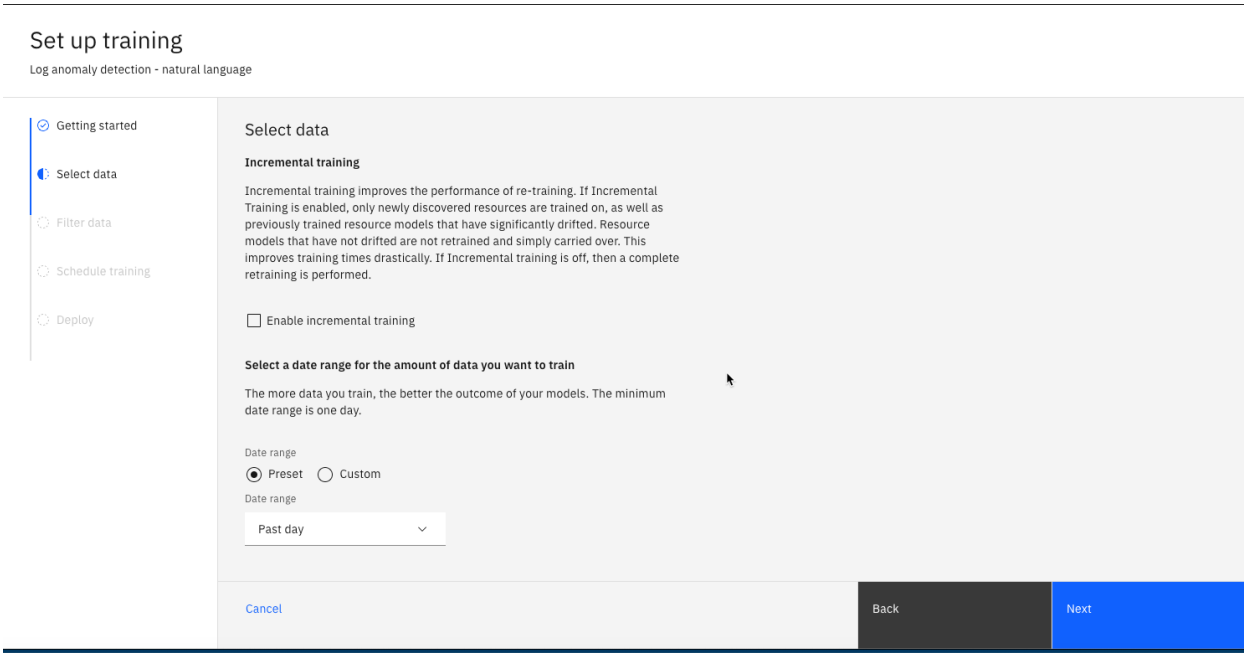


Figure. Schedule training

1. Proceed as follows:
   - To run the training setup on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training setup on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).

   Note:

   - 'Start' is when you want the scheduler to begin launching the training run. You can select 'now' for the scheduler to immediately start running, or you can specify a custom time by using the 'on date' option.
   - 'End' lets you specify how long you want the scheduler to manage the training runs. By selecting 'Never', the scheduler will keep running trainings (on the defined frequency) forever. You can also specify a custom end date by selecting 'on date'.
   - 'Frequency' lets you select how often they want the model to be retrained. The possible options are daily, weekly, bi-weekly, or monthly.
3. Click **Next** to move to the next panel.

Deciding how to deploy the training

Decide whether to review the results of training before deploying the model, or to have the system automatically deploy the model.

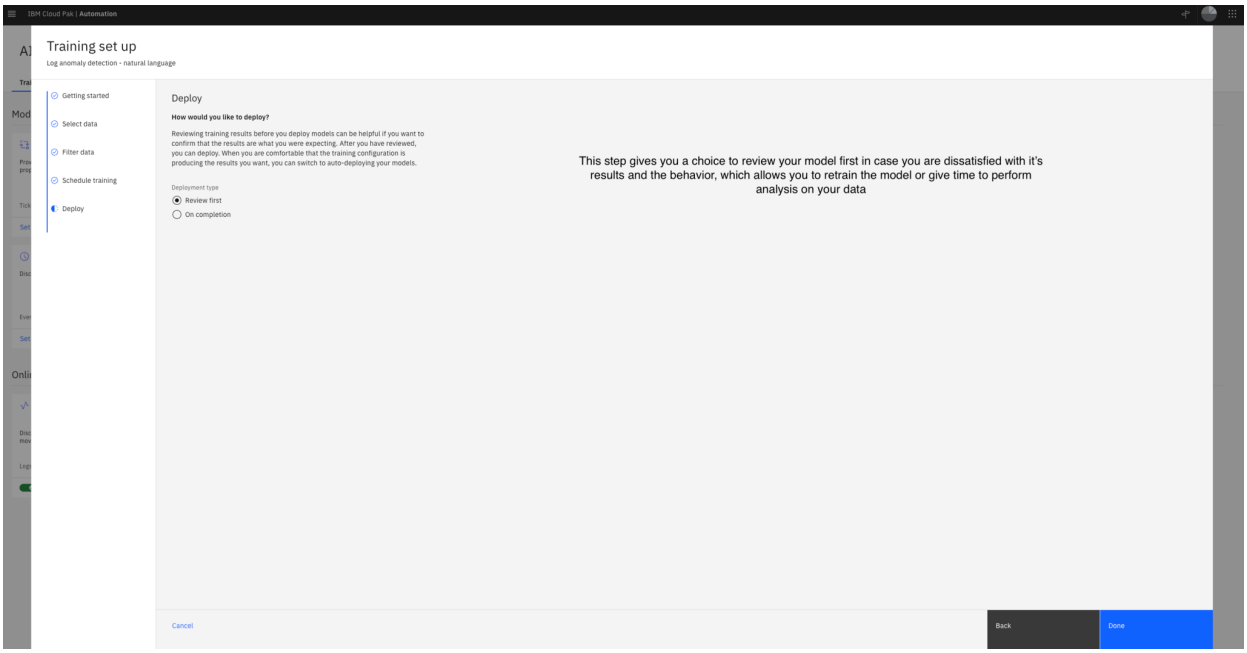


Figure. Deploy training

1. Proceed as follows:
   - To review the results of training before deploying the model, ensure that **Deployment type** is set to **Review first**.

- To have the system automatically deploy the model, set **Deployment type** to **On completion**.

2. Click **Done** to save the training setup for the algorithm.

Viewing the running jobs

1. From the main navigation of the console, click **AI model management**.
2. Click on the tile of the AI algorithm you want to view.
3. From the **Summary** tab, click **Start training** to begin training your AI algorithm. While the training runs, the jobs and their statuses are displayed in the **Activity** section.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Launching the training.

Setting up training for metric anomaly detection

Set up training for this AI algorithm to detect anomalous behaviour in your metrics and notify you by generating an alert when they occur.

For more information on the metric anomaly detection AI algorithm, see About metric anomaly detection.

About this task

There are multiple parts to this task:

- Starting the training setup
- Getting started
- Deciding how to run the training
- Deciding how to deploy the training
- What to do next

Starting the training setup

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. In the AI algorithms tab, click **Set up training** within the **Metric anomaly detection** tile.

Getting started

Review the information on this page. It explains what this AI algorithm is, how it will help in your production environment, and provides information on the data needed for the algorithm.

1. Review the information on this panel to understand more about metric anomaly detection.
2. Click **Next** to move to the next panel.

Schedule the training

Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.

1. Proceed as follows:
   - To run the training setup on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training setup on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).
3. Click **Next** to move to the next panel.

Deciding how to deploy the training

The option to review training results before deployment is disabled for this algorithm. The metric anomalies generated by training the algorithm will automatically be displayed in the Alert Viewer when training is complete.

Click **Done** to save the training setup for the algorithm.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Generating AI models.

Setting up training for similar tickets

Set up training for this AI algorithm to enable the discovery of historical tickets to aid in the remediation of current tickets.

About this task

When an incident occurs, it can be helpful to review details for similar tickets from the past to help determine a resolution. This model aggregates information from past similar tickets. It can also extract the steps that are used to fix previous tickets, if documented.

Training setup of the algorithm is done by completing some setup tasks. When the training setup is complete, you need to go to a different part of the AI Model Management to train the model.

Before you begin

To train similar tickets, you must provide ticket information. There is no set minimum of how much data to provide; the AI trains on whatever amount is available.

However, the more ticket data you can provide, the better. Any data that is collected for historical training is available to use as data. You can also collect data from the live integration and as future tickets are created, they are processed automatically.

Qualifying tickets

Important: Starting with IBM Cloud Pak® for AIOps version 4.1.0, similar incidents, such as from a ServiceNow integration, are called similar tickets.

To pass data precheck, a minimum of five qualifying tickets are needed. Qualified tickets are closed:

- The `state` and `closed_at` fields from the raw ticket schema indicate closure.
- The `close_notes` field from the raw ticket schema includes meaningful resolution notes with actions and entities that the similar tickets algorithm can extract. If your tickets do not include resolutions, or if they include default entries like *Closed by Caller*, these values do not identify paths to remediation, and the similar tickets algorithm is unable to use the tickets to provide recommended actions through similar tickets.

The following required fields are needed for a ServiceNow ticket to be qualified for the similar tickets AI algorithm.

Required attributes for the similar tickets algorithm

Attribute	Description
`short_description`	Short description or title of the ticket.
`number`	Identification number of the ticket.
`opened_at`	Coordinated Universal Time at which the ticket started formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`state`	Current state of the ticket.
`closed_at`	Coordinated Universal Time at which the ticket was closed formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`close_notes`	Comments or closed notes that talk about the list of actions that are taken, or comments that are added by the user who is attending to the ticket.

Running the task

There are multiple parts to this task:

- Starting the training setup
- Checking data integrations
- Deciding how to run the training
- Deciding how to deploy the training
- What to do next

Starting the training setup

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. On the **Training** tab, click **Set up Training** within the **Similar tickets** tile under the Trainable AI algorithms section.

Checking data integrations

Review the information on this page. It explains what this AI algorithm is, how it will help in your production environment, and provides a list of integrations needed to generate a model.

1. On the Integrations, check that at least one integration is listed.

   Note: If no integrations are listed, or you expected to see more integrations, click **Go to Integrations** to modify your integrations. For more information, see ServiceNow integration.

2. Click **Next** to move to the next panel.

Schedule the training

Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.

1. Proceed as follows:
   - To run the training setup on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training setup on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).
   - Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.
   - Training loads existing qualified ServiceNow tickets into the similar tickets AI algorithm.

- If you set the ServiceNow integration to continuous data collection, scheduling frequent trainings helps ensure that the similar tickets AI algorithm includes the new tickets in its recommendations.

3. Click **Next** to move to the next panel.

Deciding how to deploy the training

The option to review training results before deployment is unavailable for this algorithm. After training is complete, the model is automatically deployed. To see similar tickets appear in your subsequent incidents, go to the **Automations** page and enable **Default story query similar incidents service policy** in the policy list.

Click **Done** to save the training setup for the algorithm.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Launching the training.

Setting up training for temporal grouping

Set up temporal grouping to train policies. Then, choose how to deploy the policies.

For more information about the temporal grouping AI algorithm, see About temporal grouping.

Before you begin

To train a definition for temporal grouping, this AI algorithm needs at least three months of historical event data to learn alert patterns and discover groups. A set of data creates a set of policies. For more information, see Prerequisites.

This AI algorithm can start to discover groups after a pattern is learned, which might take hours or days. For example, if a problem occurs in the first and the fourth week of January and again in March, the algorithm learns to group the alerts that tend to co-occur when the problem happens.

About this task

There are multiple parts to this task:

- Starting the training setup
- Scheduling the training
- Select data
- Deciding how to deploy the training
- Running the training again
- What to do next

Starting the training setup

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. In the **Training** tab, click **Set up training** within the **Temporal grouping** tile under the Trainable AI algorithms section.

Scheduling the training

Decide whether to run the training set up on demand or to schedule it to run on an ongoing basis.

1. Proceed as follows:
   - To run the training set up on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training set up on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).
3. Click **Next** to move to the next panel.

Select data

Select a date range (either Preset or Custom) for the event data that will be included in the AI training. Including additional data typically improves the effectiveness of the AI models and policies that are generated by the training.

Deciding how to deploy the training

The option to review training results before deployment is disabled for this algorithm. However, you can choose when to deploy the policies that are generated by training the algorithm.

- To review and manually deploy the model, set **Deployment type** to **Review and manually deploy policies**.
- To automatically deploy the model, set **Deployment type** to **Automatically deploy all policies**.

Click **Done** to save the training setup for the algorithm.

Running the training again

- If you run the training again with the same data, expect to receive the same policies.
- If you run the training again with different data, you might receive new policies.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Launching the training.

Managing AI algorithms

Use this information to understand how to generate AI models, and how to perform various administration tasks on the models.

For more information about managing AI algorithms, see the following topics:

- Generating AI models
- Deploying AI models
- Editing algorithm configurations
- Deleting algorithm configurations
- Managing AI model versions
- Draft and archived policies

Generating an AI model

Generate an AI model by training one of the AI algorithms. The AI model that is generated by the training acts on live data and generate insights to your SREs.

About this task

You only need to generate an AI model for trainable AI algorithms. Pre-trained AI algorithms, in contrast, analyze your data in real time and derive insights without the need to generate a model. For more details, see Algorithm types.

Before you begin

You must save at least one algorithm training setup, as described in Setting up training for trainable AI algorithms.

Procedure

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the Training page.

   A list of algorithms is shown in tile format under sections depending on whether they are trainable or pre-trained AI algorithms.
3. Click 'Set up training' on the trainable algorithm tile for which you want to generate an AI model. When the setup is complete for that algorithm, the details from the 'Training Set up' page appear. To understand a particular model - what it does and what kind of data you need to run it – click *Description* (information icon) next to the tile's title.

   Note: You can run precheck before training your model. Precheck checks the quality of the data first. Regardless, if you try to train models without first running the precheck, AI Models still run the precheck first anyhow. Even if the training data changes, the precheck will be rerun on the new data.
4. If the setup is complete for an algorithm, click the algorithm's tile for more details on that AI model and to begin the training.

   When the model detail page opens, you can click **Precheck data** from the panel on the right side of the screen, before training. This checks the quality of the data. Regardless, if you try to train models without running the precheck, AI models still run the precheck first, anyhow. Even if the training data changes, the precheck will be rerun on the new data. **Note**: There is no data precheck for Metric anomaly detection.
5. In the right side panel, click **Train Models**.

   After a few moments the IBM Cloud Pak for AIOps console responds as follows:

   - A message appears with the text: **Training successfully started**.
   - The **AI Training** tile displays the message: **In Progress**. Training is now in progress.
6. As training progresses it passes through phases, which you can view under *Activity*, in the side panel.

There are six possible phases that are shown, though they vary depending on which algorithm you chose to train:

```
1. **Queued**
2. **Data retrieving**
3. **Quality check**
4. **Preparing data**
5. **Training**
6. **Saving models**
```

Only change risk contains all six phases. Once training is complete, the **AI Training** tile displays the messages:

- **Complete**
- **Deployed** (if indeed it has also been deployed).

At this point AI models have been created, with a specified data quality, which can take one of the following values:

- **Good**
- **Needs improvements**

  Note: No quality check is carried out on the Metric anomaly detection or Temporal Grouping models, as the check is not supported by the API that provides data to these algorithms.

For more details, click the information symbol in the **Data quality** tile. For a full set of data quality measures and messages, see Data quality messages.

For log anomaly detection training, an extra **Models** tile is displayed containing information on the models that were generated during training. For more details, see Displaying log anomaly detection resources and models.

Change risk tiles

For **change risk**, there is a new section entitled **Ticket data quality**, consisting of three tiles:

- *Closed tickets*: A number. For example, 10000. Historical ticket data must be in a terminal state (closed tickets) in order for the algorithm to train.
- *Missing tickets*: Data in high-importance fields within a ticket is needed to create good models. The more complete the data that you have in all fields (such as type, business service, sys domain and short description), the better your models perform. **Note:** if less than 0.5% of the total closed tickets are missing the relevant field (type, business service, and so on), it is displayed as 0% in the UI.
- *Ticket Word Count*: Tickets with a short (<5 words) overall word count generally don't provide enough information for the algorithm to create good models. This field combines long and short count words for the number of tickets, say 10000.

There is also a **Model prediction** tile, which shows the number of incidents causing tickets as a percentage.

What to do next

If you selected manual deployment when setting up training for the model, then you can now proceed to deploy the AI models that you created. For more details, see Deploying models.

If you encountered errors in the training, see the following sections for more information:

- Troubleshooting documentation at Troubleshooting AI model management.
- Release notes at Known issues and limitations.

Reference information

The following information is referenced in this task.

List of algorithm training setups

Column	Description
AI algorithm	The AI algorithm that is related to this record. Possible values are: **Change risk**, **Log anomaly detection**,**Metric anomaly detection**, **Similar incidents**, and **Temporal grouping**.
Latest model version	Latest version of the model. A new version of the model is generated every that time the algorithm training setup is run.
Version deployed	This is the currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
Date trained	Date and time this training setup was last run. If it has never been run then this field is blank.
Activity	Displays one of the following status values: (a) **Not started**: this training setup has not yet been run; (b) **Training**: the training setup is currently running; (c) **Training queued**: training is currently waiting on another training job to complete; (d) **Training complete**: the most recent training setup run was successful; (e) **Failed**: Training failed.
Danger zone Contains the **Delete** option, which will delete all models for the algorithm.	

Displaying log anomaly detection resources and models

During natural language log anomaly detection, models are produced for each of the resources identified in the log data provided as input to the algorithm. On completion of log anomaly detection training, the system provides information on the models that were generated during training.

About this task

In order to produce a model for a resource, a minimum of 2,000 log lines is required for that resource and the log data must meet formatting requirements. On completion of training, a **Models** tile is displayed containing the details described below.

Models

The main part of the tile displays a donut graph showing resources with models in green and resources without models in red.

- Total number of models identified is shown in the middle of the donut graph.
- Hover over the donut graph to display the number of resources with models and the number of resources without models.

View resources

Click this link at the top right of the **Models** tile to display resource and model details.

Column	Description
Resource name	Name of the resource for which models were produced.
Log lines found	Number of log lines found for this resource in the data supplied when configuring the algorithm. A minimum of 2,000 log lines is required per resource to meet data sufficiency requirements. **Note** The number of log lines displayed here might be smaller than the number of log lines provided by the log data integration and presented in a log aggegator dashboard. This is because log lines are deduplicated during the training process. A log line is said to be duplicated when the combination of **timestamp**, **instance_id** and **message** is exactly the same. The duplicate log lines are not saved on Elasticsearch and are also not used for inference in log anomaly detection. Deduplication has no impact on the quality of the models generated but does have the advantage of lower data storage requirements and also helps to manage Flink restart issues due to collection of the same data multiple times.
Sufficient data	Displays one of the following values: **Passed**: 2,000 log lines or more were found for this resource. **Failed**: Less than 2,000 log lines were found for this resource.
Data format	An indication of whether the data found for this resource meets data formatting needs. This column takes one of the following values: **Passed**: Formatting needs are met. **Failed**: Formatting needs not met. You might need to modify the field mapping of the data. For more information see Mapping data.
Model status	Indicates whether a model was created for this resource. A model is only created if both sufficiency and formatting needs are met.

Data quality messages and alerts

The data quality precheck stage always runs on your input data before it can be approved for training. Once the precheck stage is complete, you are presented with an assessment of the data quality.

The data quality precheck examines whether enough data is available to return meaningful results (*data sufficiency*), and whether the data meets mapping requirements (*data formatting*). A result is displayed on the console, and this result affects the execution of the model training, as if the precheck fails, the training is disabled, and you cannot run training.

If you change the selected training data, the precheck stage is always run again.

Mapping data

Poor *data formatting* is often associated with incorrect field mapping.

Data is collected when you select a source on the Integrations and add a integration. Within the window for configuring the integration, use the **Field mapping** section to map source fields to standard fields. For example, the `_timestamp` in Mezmo data should be mapped to `timestamp` in the standard format. One reason that schema validation might fail, is when incorrect data mappings are entered in the **Mapping** field. For more information on the overall data loading and AI training process, see Planning data loading and training.

Good data quality

Good data quality means that requirements for *data sufficiency* and *data formatting* are met. Good data quality is represented on the Cloud Pak for AIOps console using the following label: **Good data**.

Data quality	Recommendation	Description
Good data	`Precheck came back with good data. You are now ready to start training models.`	Requirements for *data sufficiency* and *data formatting* are met.

Poor data quality

Poor data quality messages vary with the AI algorithm that you are training. Poor data quality is represented on the Cloud Pak for AIOps console using the following label: **Data needs improvement**.

For each of the following AI algorithms, this topic describes the possible data quality messages, provides a brief description of the message, and points to any relevant documentation task, which can support remediation action.

- Change risk
- Log anomaly detection
- Similar tickets

Change risk

This algorithm takes as input change and incident data from ServiceNow. To meet overall data quality requirements, both change and incident data must meet *data sufficiency* and *data formatting* requirements.

Data quality alert	Recommendation	Description	Link

Data quality alert	Recommendation	Description	Link
Data needs improvement	`Part of your data did not pass precheck. View recommendations to see how you might improve data quality.`	**Data precheck issue**: Insufficient number of problematic change tickets for a good model. For a closed change ticket count less or equal to 150, you need at least three problematic tickets. For a closed change ticket greater than 150, you need at least two percent of change tickets to be problematic. A problematic change ticket is defined as a change ticket that can either be associated with incident tickets, or have an 'unsuccessful' value set in the change ticket close code field.	See ServiceNow integration
Data needs improvement	`Change data: The data is not valid according to schema. Check input data mapping or data source.`	**Data formatting issue**: Change data does not meet *data formatting* needs. You might need to modify the data's field mapping.	See ServiceNow integration
Data needs improvement	`More data is needed. You need at least _X_ change tickets for a good model.`	**Data sufficiency issue**: Change data does not meet *data sufficiency* needs. More change ticket data is needed	See ServiceNow integration
Data needs improvement	`Incident data: The data is not valid according to schema. Check input data mapping or data source.`	**Data formatting issue**: Incident data does not meet *data formatting* needs. You might need to modify the data's field mapping.	See ServiceNow integration
Data needs improvement	`More data is needed. You need at least 1 incident data point for a good model.`	**Data sufficiency issue**: Incident data does not meet *data sufficiency* needs. More change ticket data is needed	See ServiceNow integration
Data needs improvement	`Data is insufficient for a good model. You need at least 2 percent of change tickets to either be associated with incident tickets or have an 'unsuccessful' value set in the change ticket close code field.`	**Data sufficiency issue**: Incident data does not meet *data sufficiency* needs. You must provide data with a change ticket failure rate of 2% or higher.	See ServiceNow integration

Log anomaly detection

This algorithm takes as input log data from one or more log systems. The log anomaly detection algorithm itself has multiple underlying microservices. *Data sufficiency* and *data formatting* requirements for this algorithm work as follows:

- Each of these microservices requires a minimum of 2000 data points. To meet the minimum *data sufficiency* requirement for the algorithm as a whole, at least 75% of the algorithms must meet this 2000 data point requirement. For example, if only two microservices out of 10 have less than 2000 data points, and the rest have more than 2000 data points, then *data sufficiency* requirement for the algorithm as a whole is good.
- To meet the *data formatting* requirement for the algorithm as a whole, at least 75% of all data (from all of the microservices) must be valid with respect to the schema.

Data quality alert	Recommendation	Description	Link
Data needs improvement	`Part of your data did not pass precheck. There were not enough logs for X out of Y resources. Models were only created for those where sufficient data existed. You can expand the date range to gather more data and train again to expand coverage.`	**Data precheck issue**: Data does not meet data precheck needs. You might need to modify the data's quality before running the training.	See Train model precheck
Data needs improvement	`There were not enough logs for _X_ out of log_anomaly_detection components. Models were only created for those where sufficient data existed. You can expand the date range to gather more data and train again to expand coverage.`	**Data sufficiency issue**: Data does not meet *data sufficiency* needs. More than 25% of the component microservices have less than the required 2000 data points. More log data is needed. Try expanding the data range of the data provided.	See Natural language log anomaly detection: data to train on
Data needs improvement	`We have detected that a portion of this data set is in an unsupported language, and could affect the quality of this model. We recommend that you remove this data from the data set and retrain the model.`	**Language support issue**: Log anomaly detection only supports the following languages and language combinations: English, Spanish, German, English and Spanish, English and German. Your data contains either an unsupported language or an unsupported language combination.	For more information, see Language support. To remediate this issue, see Managing unsupported languages.

Similar tickets

This algorithm takes as input incident data from ServiceNow. To meet overall data quality requirements, both change and incident data must meet *data sufficiency* and *data formatting* requirements.

- To meet *data sufficiency* requirements, a minimum of five closed incidents with associated resolution is required.

Data quality alert	Recommendation	Description	Link
Data needs improvement	`Part of your data did not pass precheck. View recommendations to see how you might improve data quality.`	**Data precheck issue**: More data is needed. You need at least five closed incidents with resolution for a good model.	See ServiceNow integration
Data needs improvement	`More data is needed. You need at least 5 closed incidents with resolution for a good model.`	**Data sufficiency issue**: Data does not meet *data sufficiency* needs. More closed incident ticket data is needed.	See ServiceNow integration

Note There is a new data quality alert that appears in the console if the precheck is launched (for all of the three algorithm failures) without any data loaded. The alert tells you that the 'Precheck failed', and the recommendation is 'Cannot find any data'.

Managing unsupported languages

If natural language log anomaly training reports that the model needs improvement because of unsupported language issues, you can remove the offending data from the data set by deleting component-specific logs, and then retrain.

About this task

Log anomaly detection only supports the following languages and language combinations:

- English
- English and French
- English and German
- English and Italian
- English and Spanish
- French
- German
- Italian
- Spanish

If your data contains either an unsupported language or an unsupported language combination, then you will get the following data quality alert on completion of training:

Needs improvement We have detected that a portion of this data set is in an unsupported language, and could affect the quality of this model. We recommend that you remove this data from the data set and retrain the model.

To remediate the situation you must delete all the logs from components whose logs are all or partially in an unsupported language.

Procedure

Perform the following steps to remediate the situation.

1. Log in to your cluster.

   ```
   oc login -u kubeadmin -p <password>
   ```

   For more information, see Logging in to the OpenShift CLI.

2. Run the `oc project` command to set the context to the project where Cloud Pak for AIOps is deployed.

3. Find the name of the apiserver pod.

   ```
   oc get pod | grep api-server
   ```

4. Open a terminal on that pod.

   ```
   oc exec -it <api-server_podname> bash
   ```

5. Run the following `curl` command exactly as specified below to retrieve data quality alert details.

   ```
   curl -k -u $ES_USERNAME:$ES_PASSWORD -X GET "https://$ES_ENDPOINT/prechecktrainingdetails/_search?pretty" -H
   'Content-Type: application/json' -d'{"_source":"dataQualityDetails.languageInfo.components"}'
   ```

   In this command:

   - `ES_USERNAME` is the Elasticsearch username.
   - `ES_PASSWORD` is the Elasticsearch password.
   - `$ES_ENDPOINT` is the Elasticsearch environment variable in the pod.
   - `https://$ES_ENDPOINT/prechecktrainingdetails` is the Elasticsearch endpoint for the `prechecktrainingdetails` index, which stores language information associated with the data set sampled for a specific component.

6. In the output that is generated, navigate to the `dataQualityDetails` field, and find the nested `languageInfo` field. Then retrieve the names of the component where the language list has `unknown` as one of the values. For example, in the example below you would select `abc`.

   ```
   "components" : [
                {
                  "name" : "abc",
                  "language" : [
                      "English",
                      "unknown"
                              ],
   ```

7. For each component that you identified in the previous step, delete the logs from all log data indices by running the following command.

   ```
   curl -k -u $ES_USERNAME:$ES_PASSWORD -X POST "https://$ES_ENDPOINT/*-logtrain/_delete_by_query" -H 'Content-Type:
   application/json' -d'{"query": {"bool": {"must": [{"term": {"instance_id.keyword": "abc"}}]}}}'
   ```

   Where `abc` must be replaced with the name of the component.

   Note If all components are detected as having logs in an unsupported language, then this step will delete all data from the Elasticsearch indices for log anomaly training. You will need to train again when fresh data arrives.

8. Once you have deleted the relevant logs, you must relaunch the training of your natural language log anomaly detection algorithm, as described in Generating AI models.

Determining which algorithm generated a log anomaly

There are two log anomaly detection AI algorithms, each of which can run independently of the other: natural language and statistical baseline. If both algorithms are enabled, then any log anomalies discovered by both will be reconciled, so that only one alert is generated. Use the information in this topic to work out which log anomaly detection algorithms contributed to the detection of the anomaly.

About this task

There are two ways to check which log anomaly detection AI algorithms generated a given log anomaly:

- By viewing the details of the alert in the Alert Viewer.
- By monitoring the Kafka topic **cp4waiops-cartridge.lifecycle.input.events kafkatopics**, and parsing out the values in the **log_anomaly_model** field.

Viewing details in the Alert Viewer

1. In the Alert Viewer, double click the log anomaly alert of interest. The **Alert Details** panel opens on the right, displaying the properties of the alert.
2. In the **Properties** table, locate the **Description** field.
3. Within the **Description** field, locate the **log_anomaly_model** key. The nested values within this key identify which log anomaly detection algorithm contributed to the detection of this anomaly. Possible nested values are listed in Possible values within the log_anomaly_model JSON key below.

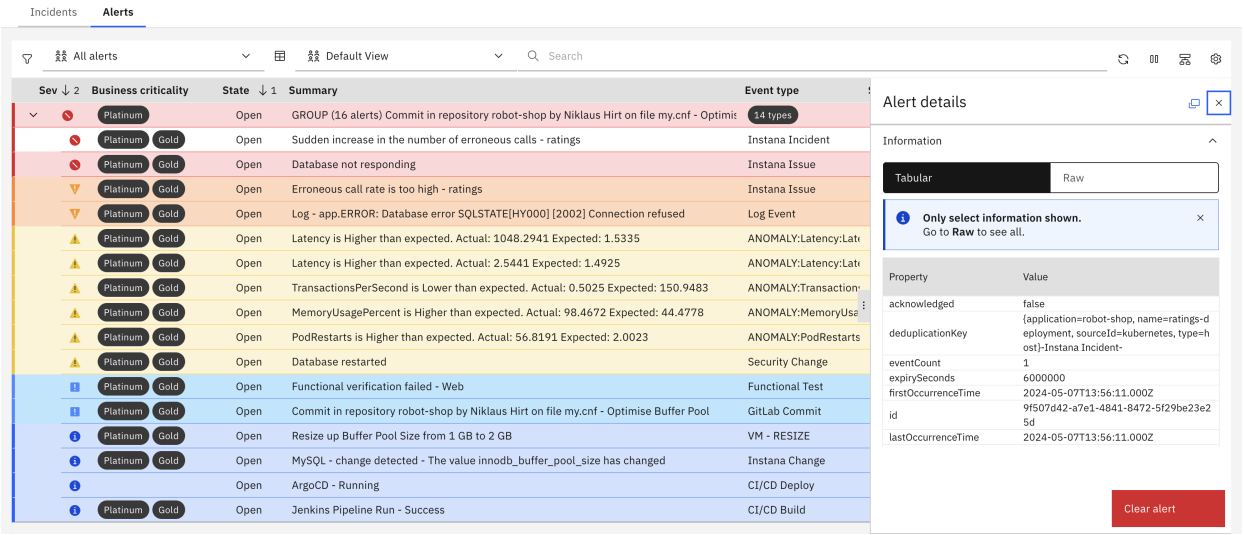


Figure. The Alert details panel

Monitoring the relevant Kafka topic

1. Monitor the Kafka topic **cp4waiops-cartridge.lifecycle.input.events kafkatopics**. This topic provides a Kafka stream of alert instances.
2. Look for a **sender** key where name attribute is "Log Anomaly" to identify those alert instances that were generated by the log anomaly detection AI algorithms.
3. Within that section of the JSON, locate the **log_anomaly_model** key. The nested values within this key identify which log anomaly detection algorithm contributed to the detection of this anomaly. Possible nested values are listed in Possible values within the log_anomaly_model JSON key below.

Reference information

The following information is referenced in this task.

Possible values within the log_anomaly_model JSON key

Value	Algorithm	Meaning
PCA_Stats	Natural language	The time window in which this alert occurred is statstically different to the training data.
PCA_error_or_unknown_template	Natural language	There are error templates or unknown templates in this time window which the algorithm has identified as anomalous.
PCA_low_frequency_error_template	Natural language	There are error templates which occur with low frequency in this time window and the algorithm has identified this as anomalous.
RSM_entities	Statistical baseline	The algorithm has determined that there are significantly different error entities in this time window.
RSM_embeddings	Statistical baseline	The algorithm has found word embeddings for this time window that are distant from the embedding vectors for normal data.

Value	Algorithm	Meaning
RSM_websphere	Statistical baseline	This value is specific to WebSphere logs. The algorithm has found significantly different error message identifiers or log levels for this time window.

Deploying models

If the model is not autodeployed, then if you are happy with the results, you can now deploy the model.

Before you begin

You only need to perform this task if you generated a model that requires manual deployment.

Procedure

1. From the AI Model Management, click the options menu for the algorithm configuration that includes the model that you want to deploy.
2. Click **Deploy**.

Reference information

The following information is referenced in this task.

List of algorithm configurations

Column	Description
Version	Latest version of the model. A new version of the model is generated every time the algorithm configuration is run.
Deployed version	Currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
AI algorithm	The AI algorithm related to this record. Possible values are: **Change risk**, **Log anomaly detection**, **Similar tickets**, and **Temporal grouping**.
Schedule	Displays the schedule frequency (**Daily**, **Weekly**, **Biweekly**, or **Monthly**). If no schedule was set, then this field displays the text **Run manually**.
Last trained	Date and time this configuration was last run. If it has never been run then this field is blank.
Status	Displays one of the following status values: (a) **Not started**: this configuration has not yet been run; (b) **Training**: the configuration is currently running; (c) **Training queued**: training is currently waiting on another training job to complete; (d) **Training complete**: the most recent configuration ran was successful; (e) **Failed**: Training failed.

Editing the training setup for an algorithm

You can edit the training setup for any of the existing algorithms.

Before you begin

You must save at least one algorithm training setup, as described in Setting up training for trainable AI algorithms.

About this task

Only algorithms that generate models can be edited. For more details, see Algorithm types.

Procedure

1. From the AI Model Management, under the Training tab, click the tile of the algorithm you want to edit.

   - **Change risk**: Deployment, Training schedule
   - **Log anomaly detection - natural language**: Training data, Deployment, Training schedule
   - **Similar tickets**: Training schedule
   - **Temporal grouping**: Training schedule
   - **Metric Anomaly Detection**: Training schedule

2. Modify the algorithm settings by clicking the **Edit data** icon  next to the settings that you want to modify. After you make your changes click **Save** to return to the Overview page.

   Your changes are now saved.

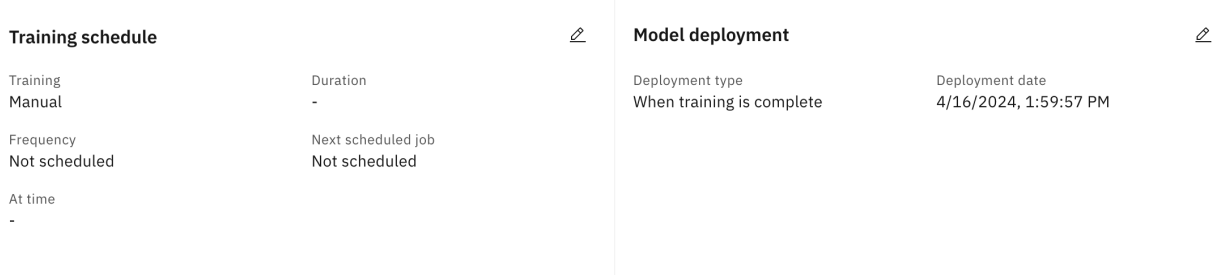


Figure. Editing configurations for the change risk algorithm

Reference information

The following information is referenced in this task.

List of algorithm details

Column	Description
Latest model version	Latest version of the model. A new version of the model is generated everytime that the training succeeds.
Version deployed	Currently the deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
AI algorithm	The AI algorithm that is related to this record. Possible values are: **Change risk**, **Log anomaly detection - natural language vs statistical baseline**, **Similar tickets**, **Metric anomaly** and **Temporal grouping**.
Date prechecked	Date and time this training setup was prechecked.
Date trained	Date and time that this training setup was last run. If it has never run then this field is blank.

Deleting the training setup for an algorithm

You can delete the training setup for any of the existing algorithms.

About this tasks

If an algorithm training setup has associated AI models, then the action of deleting the training setup for the algorithm automatically removes any currently deployed model generated based on that training setup. It then deletes all of the models associated with this training setup.

Before you begin

You must have saved at least one algorithm training setup, as described in Setting up training for trainable AI algorithms.

Procedure

1. From the AI Model Management, click the options menu for the algorithm training setup that you want to delete.
2. Click **Delete**.

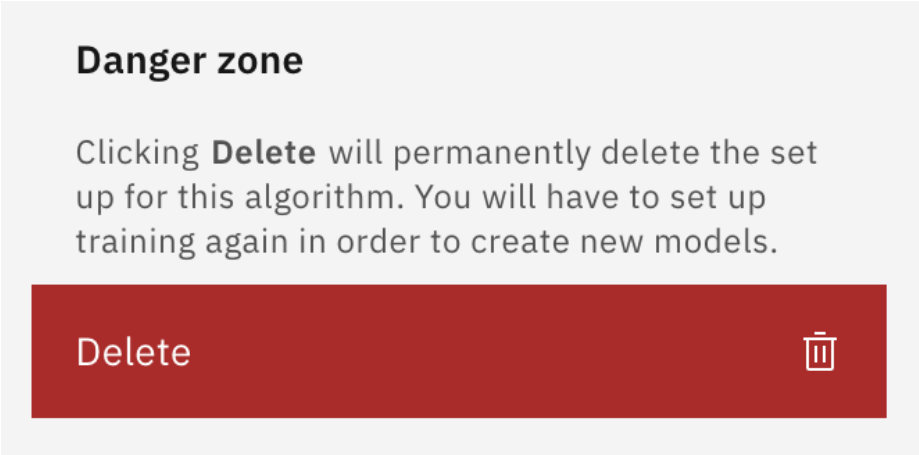


Figure. Deleting an algorithm

Reference information

The following information is referenced in this task.

List of training setups for an algorithm

Column	Description
Version	Latest version of the model. A new version of the model is generated every time the training setup for the algorithm is run.
Deployed version	Currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
AI algorithm	The AI algorithm related to this record. Possible values are: **Change risk**, **Log anomaly detection**, **Similar tickets**, and **Temporal grouping**.
Schedule	Displays the schedule frequency (**Daily**, **Weekly**, **Biweekly**, or **Monthly**). If no schedule was set, then this field displays the text **Run manually**.
Last trained	Date and time this training setup was last run. If it has never been run then this field is blank.
Status	Displays one of the following status values: (a) **Not started**: this training setup has not yet been run; (b) **Training**: the training setup is currently running; (c) **Training queued**: training is currently waiting on another training job to complete; (d) **Training complete**: the most recent training setup ran was successful; (e) **Failed**: Training failed.

Managing AI model versions

You can change the version of a model that is deployed. You can also delete a model version.

For more information about managing AI model versions, see the following topics:

- Changing the deployed version
- Deleting model versions

Changing the deployed version of a model

Only one version of a model can be deployed at any one time. Where you have multiple versions of a model, you can change the deployed version.

Before you begin

You must have generated at least two versions of an AI model, and deployed one of them as described in Deploying models.

Procedure

1. In the AI Model Management, under *Trainable AI algorithms*, click the tile of the configuration you want to change.
2. When the overview page opens, click the *Model versions* tab. A list of model versions is displayed. For more information, see the List of model versions.
3. To change the deployed version, click the options menu at the far end of the table in the row corresponding to the version that you want to deploy. Then, click **Deploy**. The table updates to show that the version that you selected is now the deployed version.

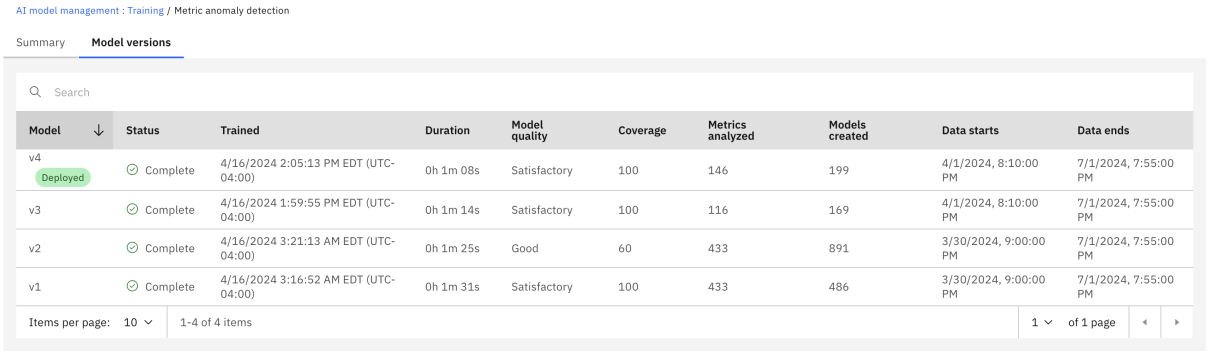


Figure. Model versions in Metric anomaly detection

Reference information

The following information is referenced in this task.

List of algorithms configurations

Column	Description
Name	Name of the algorithm configuration.
Latest model version	Latest version of the model. A new version of the model is generated every that time the algorithm configuration is run.
Version deployed	This is the currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.

Column	Description
AI algorithm	The AI algorithm that is related to this record. Possible values are: **Change risk**, **Log anomaly detection**, **Similar tickets**, and **Temporal grouping**.
Data prechecked	Date and time this configuration was last prechecked.
Date trained	Date and time this configuration was last run. If it has never been run, then this field is blank.

List of model versions

Note: The below columns are common to most algorithms and then they can vary in terms of additional columns. For example, Log anomaly also has 'Totals resources found', and 'Models created', whereas Change risk also includes 'Closed tickets' and 'Incidence avoidance'.

Column	Description
Model	Version of the model. A new version of the model is generated every that time the algorithm configuration is run.
Status	The deployed version is marked as **Deployed** with a comforting green tick to the left of the word **Deployed**. The other versions have the status **Complete**.
Last trained	Date and time when this version of the model was trained.
Duration	Length of time taken to train this version of the model.
Data Quality	Possible values are **Good** and **Needs improvement**. Only deploy model versions that have the status **Good**.

Deleting model versions

You can delete versions of a model.

About this task

You cannot delete a deployed version of the model directly. If you want to delete a deployed version, you must first change the deployed version of the model, and then delete the original model.

Procedure

1. From the AI Model Management **Training** tab, click **Log anomaly detection - natural language**.
2. From the **Log anomaly detection - natural language** page, click the **Model versions** tab.
3. To delete a model version, go to the row of the version that you want to delete and click the options menu to expand it. Then, click **Delete**.

   Note: You cannot delete a deployed version of the model directly. If you want to delete a deployed version, you must first change the deployed version of the model and then delete the original model.

Resetting the log anomaly training environment

If you need to reset your log anomaly training environment, complete the following steps. Proceed with caution and do not attempt these steps on a production or critical Cloud Pak for AIOps environment.

1. Run the `oc project` command to set the context to the project where Cloud Pak for AIOps is deployed.
2. **Important:** Do not delete any of the following index files. Deleting these files might delete data that corresponds to other AI activities, such as metric anomaly detection training. However, you can use `_delete_by_query` to delete documents that are referenced by these index files.
   - `trainingdefinition`
   - `trainingsrunning`
   - `trainingrun`
   - `trainingstatus`
3. Specify the variables and delete the following index files. Deleting an index uses a cascading delete and deletes the documents that are referenced by the index.
   - Any `1000-1000-*-logtrain` index files
   - The `1000-1000-vN-*` or `1000-1000-v*-models` index files of the particular version that you want to reset. Specify values for the asterisk (*) and `N` variables. The `N` variable is an integer number that indicates the version, such as in `1000-1000-v1-applications`. If you want to delete all log anomaly model versions, use `1000-1000-vN-*` or `1000-1000-v*-models`.
   - The `1000-1000-log_models_latest` index file
   - See the following command as an example:

     ```
     export ELASTIC_SET="$(oc get statefulset -l app.kubernetes.io/managed-by=ibm-elasticsearch -o
     jsonpath='{.items[0].metadata.name}')"
     oc exec "statefulset/${ELASTIC_SET}" -- curl -XDELETE -k --fail "http://127.0.0.1:19200/1000-1000-v1-templates"
     ```

Reference information

The following information is referenced in this task.

List of algorithm configurations

Column	Description
Name	Name of the algorithm configuration.
Version	Latest version of the model. A new version of the model is generated every time the algorithm configuration is run.
Deployed version	Currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
AI algorithm	The AI algorithm related to this record. Possible values are: **Change risk**, **Log anomaly detection**, **Similar tickets**, and **Temporal grouping**.
Schedule	Displays the schedule frequency (**Daily**, **Weekly**, **Biweekly**, or **Monthly**). If no schedule was set, then this field displays the text **Run manually**.
Last trained	Date and time this configuration was last run. If it has never been run then this field is blank.
Status	Displays one of the following status values: (a) **Not started**: this configuration has not yet been run; (b) **Training**: the configuration is currently running; (c) **Training queued**: training is currently waiting on another training job to complete; (d) **Training complete**: the most recent configuration ran was successful; (e) **Failed**: Training failed.

List of model versions

Column	Description
Version	Version of the model. A new version of the model is generated every time the algorithm configuration is run. The deployed version has a green **Deployed** tag to the right of the version number.
Status	The deployed version is marked as **Deployed** with a comforting green tick to the left of the word **Deployed**. The other versions have the status **Training complete**.
Last trained	Date and time when this version of the model was trained.
Duration	Length of time taken to train this version of the model.
Data Quality	Possible values are **Good** and **Needs improvement**. Only deploy model versions that have the status **Good**.

Draft and archived policies

You can manage the alert seasonality and temporal grouping policies that are created by AI analytics.

Procedure

1. In the AI Model Management, under Trainable AI algorithms, click the **Alert seasonality detection** or **Temporal grouping** tile.
2. On the overview page, click the **Draft and archived policies** tab.

   A policies table is displayed with two views: **Draft** and **Archived**. To search the list of policies in either view, click anywhere in the **Search** field and enter a search term. The system shows in the table all policies that contain the search term in the policy names and the policy conditions. Click Refresh to refresh the table at any time.
3. If present, draft policies once reviewed can be deployed if preferred or moved to archived if unwanted.

   When applying actions to draft or archived policies, you can multi-select policies in the table and apply actions to all of your selections at one time. Use the checkboxes at the beginning of the table rows to select multiple policies. You can also select all policies in the table by clicking the checkbox in the table header beside **Policy name**. A banner above the table displays the number of policies that are selected, together with a menu of options. The actions for draft and archived policies are explained below.

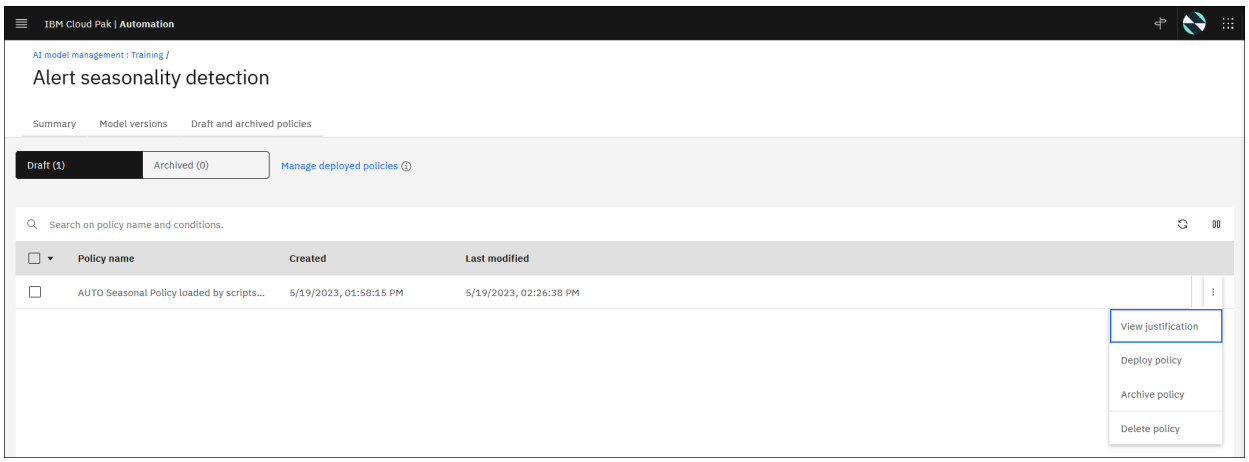


Figure. Draft and archived policies

- **Draft policies**

  The **Draft** tab displays policies that are suggested by AI analytics. Click the options menu (three vertical dots) at the end of a table row to display all of the options available for draft policies:

- View justification

  Further investigate the seaonality or temporal details and the justification for the policy creation. For more information, see Seasonality policy details and Temporal policy details.

- Deploy

  Deploy policies that you want to act on incoming alerts. Policies that are deployed from here appear in the Cloud Pak for AIOps Policies table. Click **Manage deployed policies** to display the Policies table that is filtered to include only temporal or seasonal policies. Any existing filters are removed.

- Archive policy

  Archive policies that are not in use.

- Delete policy

  Delete the policy. Note, deleted policies are removed from both the Draft and Archived views. This action cannot be undone.

- **Archived policies**

  Policies that have been archived by administrators or senior operators are shown in the **Archived** view. Archived policies do not act on incoming alerts. Click the options menu on any archived policy to display the available options:

  - View justification

    Further investigate the seaonality or temporal details and the justification for the policy creation. For more information, see Seasonality policy details and Temporal policy details.

  - Deploy

    Deploy policies that you want to act on incoming alerts. Policies that are deployed from here appear in the Cloud Pak for AIOps Policies table. Click **Manage deployed policies** to display the Policies table that is filtered to include only temporal or seasonal policies. Any existing filters are removed. Draft policies can also be either **Archived** or **Deleted**.

  - Unarchive policy

    When a policy is unarchived, it is visible again in the **Draft** view.

  - Delete policy

    Delete the policy. Note, deleted policies are removed from both the Draft and Archived views. This action cannot be undone.

Related links

- Policies table
- Archiving policies
- Seasonality policy details
- Temporal policy details

Displaying data assets

List the data assets that are available for analysis by your algorithm configurations.

Before you begin

Before you perform this task, make sure that you load the relevant data for your AI types. For more details, see Prerequisites for AI modelling.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. Click **Data assets**. A list of data assets is displayed. For details of the information is displayed in each row, see the List of data assets below.

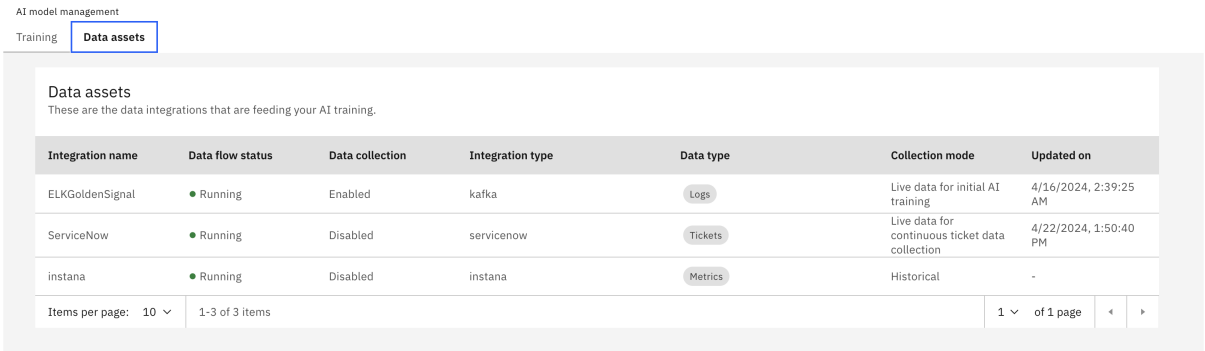


Figure. Data assets in AI model management

Reference information

The following information is referenced in this task.

List of data assets

To modify any of these data integration, you must modify the relevant data integration. For more information, see Configuring Integrations.

Column	Description
Integration name	Name of the data integration.
Data flow	Indicates whether the integration is enabled or disabled.
Integration type	Provides a tag specifying the system from which the data originates.
Data type	Specifies the type of data provided.
Data collected	Indicates whether the data feed is live or historical.
Updated on	Time and date when this integration was last updated.

Managing applications (services) and viewing topologies

Using IBM Cloud Pak for AIOps console you can create applications (services) from bespoke groups of resources, which you can then view in topology or table format. You can analyse incidents and historical data at a granular level using various topology tools, and customize your topology management experience to suit your operational needs.

To access applications (services) and their constituent elements, you launch the *Resource management* page from the main navigation menu. Here data is presented on separate tabs as resources, resource groups, and applications (services).

Note: The following procedure uses 'application' for the data object type that is being managed. Depending on your UI configuration, the term 'service' can be used instead.

Tip for IBM Cloud Pak for AIOps version 4.1.0 and later

You can now configure the UI to label your groups-of-groups collections of resources as either 'applications' or 'services', depending on which term more accurately describes your situation. The system default is 'application', which is also used in the documentation.

- **During installation**, you can set the use of either 'services' or 'applications' by assigning the appropriate value to the TOPOLOGY_TERMINOLOGY environment variable.
- **For an existing installation**, you can run a patch command to switch to the use of 'service'.

  First, confirm the CR name used during installation by running the `oc get installation` command. Use the value returned as **NAME** in the following command.

  ```
  oc patch installations.orchestrator.aiops.ibm.com/<NAME> --type merge -p '{"spec":{"topologyModel":"service"}}'
  ```

Resources

Resources are the lowest level of element in the system. They are represented in topology visualizations, search experience, and APIs, and form the basis of any topology-based event analytics. Resources are created in the system from observations of different technologies, or from data entered directly, and can represent a multitude of different elements, which are interconnected by their specific relationships and related properties. Relationship types can represent physical connections or logical connections, and all resources together provide an end-to-end view of an environment across data sources and management silos.

Resource groups

Resource groups are an abstraction layer on top of resources and are used to both manage their constituent resources and form the building blocks of applications. You can have resource groups from different sources within an application, or separate from an application.

Resource groups have been collected into a group in order to capture a business or technical context within the environment for both search and correlation purposes. They are either created directly from the topological data discovered via observer jobs, or they are created by defining templates that represent a specific business logic, which dynamically produces resource groups based on defined criteria, such as a model topology structure or a set of tags. Importantly, resource groups are dynamic, which means they change as and when the environment is updated, thereby ensuring the topology data is up-to-date.

Groups allow us to impart meaning to potentially opaque resources, for example, a network interface problem can be prioritized far more effectively if it is known that the resource is part of an important customer application. If required you can create many thousands of groups as determined by their environment; for example, you could provide tens-of-thousands of applications to your customers, or have a hundred-thousand geographical locations.

Resource groups can only be created and managed through a connected inventory provider, such as Kubernetes, Instana or VMware, or through a resource group template, and cannot be created or deleted from the Resource management page; while the resources within a resource group must be created and managed through the integration with the inventory provider (that is, an observer job). If you need to change resources within a resource group, you do not directly manipulate the resource group. Instead, an administrator user must use the resource group template, or manage the group within the connected inventory provider.

To open a resource group topology view of all of the resources that make up or have relationships to a group, see Viewing topologies.

Applications

Applications can be conceptualised as 'groups-of-groups' and are the highest level of abstraction, considered the starting point for correlating the current environment and managing it in a more coarse-grained and abstract way. An application consists of a collection of resource groups, which can each be created and managed from different sources, designed to best represent a specific business application or service, considering that no one source is likely to provide end-to-end visibility. When combined with resource-level event correlation and grouping mechanisms, applications provide rapid visibility of overall states, allowing operators to prioritize work orders and focus on specific sets of resources that relate to a specific application, thereby excluding resources and resource groups that are not of interest.

For example, an application might be formed by a number of virtual machines and their hypervisors, network equipment, cloud-native applications, and a build pipeline. All of this can form a single holistic representation of the application, which can be used as a shared context between many different teams.

Remember: IBM Cloud Pak for AIOps can model many applications or other 'groups-of-groups' configurations (like services) that you might require for your business application management.

For more information, see the following topics:

- Resource management: Searching and filtering
- The *Resource management* page
- Defining and editing applications
- Managing topologies

Resource management: Searching and filtering

You can search for applications, resource groups, and resources using the Search function. You can further filter the search results by defining filter conditions, which can be saved for future use. For instance, you can define filter conditions to display resources that have a specific tag applied to them, or for RBAC

After you log into IBM Cloud Pak for AIOps console, you open the *Resource management* page from the main navigation menu.

Lists of applications, resource groups and resources are displayed on separate tabs. If many resources are listed, you can use the 'search' or 'filter' functions to identify specific ones, which you can then manage using the tools provided, or display as topologies for further analysis.

Remember: Beginning with IBM Cloud Pak for AIOps version 4.8.0, you can configure the UI to label your groups-of-groups collection of resources as either 'applications' or 'services'. For more information, see the following tip on changing terminology.

Tip: You can now also access the search and filter functionality from the new Resource map (for resources only).

To find aplications, resource groups and resources

1. Activate the Search field.

   As you insert your cursor into the **Search** field, the five most recent searches are listed, and can be selected.

   - You can delete a recent search by clicking the delete button (**X**) next to the suggested search, or by clicking **View all** to display all recent searches.
   - When viewing all recent searches, you can filter the results further using the Filter table Search field, or you can select them and then delete them from the recent searches list.
2. Enter data into Search.

   As you type in a search term, suggested search results are listed, both from the topology database and from recent searches, with your entered text highlighted.

   Tip: You can perform more complex searches, such as searching with multiple search terms. For more information about advanced query syntax, see the topology query reference topic.

3. Select a Search result.

   When you click a search result, the results are listed on the *Results* page under the *Favorites* and *All results* sections, separated into *Applications*, *Resource groups*, and *Resources* tabs, and search is closed with the search term displayed in the Search field.

   Note: (For resources only), you can select the **Include deleted resources** toggle in order to include deleted resources on the *Results* page (unless an administrator has disabled that feature). Deleted resources are marked as such, and the date and time that they were deleted is displayed. If you click the resource name, the deleted resource is displayed as a topology with the history timeline open, and set to 1ms before deletion.

   - For each search result, the name, type and other properties are displayed.
   - If a status other than `clear` exists for a search result, the maximum severity is displayed, and a color-coded information bar above each result displays all non-clear statuses (in proportion).
   - To render a topology view from a search result, click the name.

   Note: You can click the delete button (**X**) next to the search term to return to the previous *Resource management* screen.

4. You can further reduce the number of search results by defining (or using predefined) filters.

To filter applications, resource groups and resources

If filtering from the Resource management page, the Services and Resource group tabs are unavailable (grayed out). Once you have identified a resource you wish to view on a map (defined earlier), click the map icon for that resource in the table (to the right of the 'More details' link).

To see a resticted access filter do the following:

1. Click **Resouce Managment**.

2. Click the **Resources** tab, and then the filter icon in the top left-hand corner.
3. Set the filter as you need, for example you can set the **severity** section to *major*, and **business criticality** to *Auto-medium-criticality*.
4. Then click the three dots icon in the top right-hand corner of the Filter conditions section. 5. Click **Save as new filter**.

   Note: Alternatively, if you are an administrator who has *manage profiles* priviledges enabled, you can set Role Based Access Control (RBAC) functionality that allows you see and click the **Save as new restriction filter** option, thus applying certain restriction filters to applications, resource groups and resources.

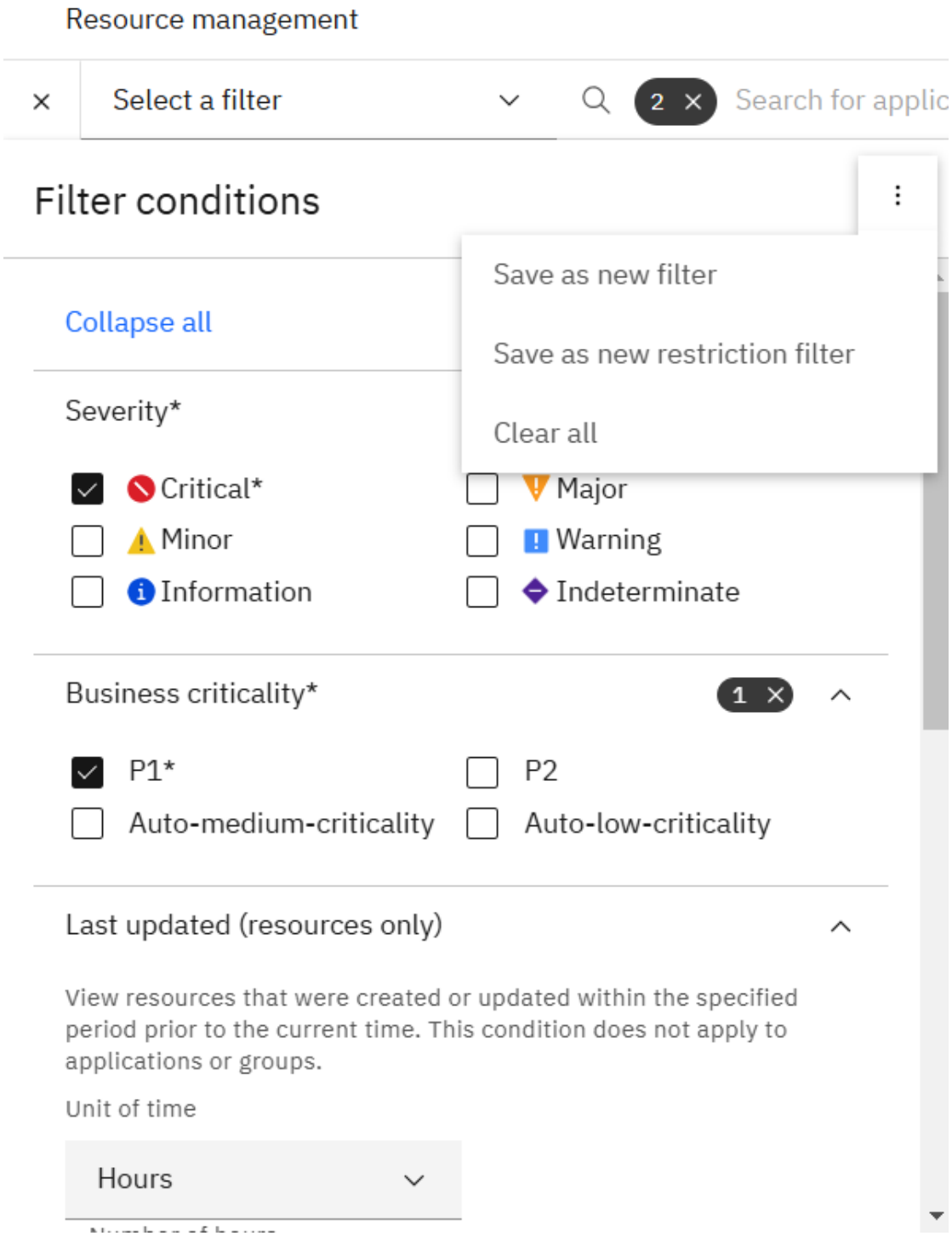


Figure 1. Resource management new restriction filter

If filtering from a Resource map, filters are applied immediately and the map is adjusted.

1. Filter applications, resource groups and resources.

   If many applications, resource groups or resources are displayed, you can filter them using predefined filters, or by defining new filters. You can save filters for your future use, or for use by specified users types.

   - To use an existing filter, click **Select a filter** to the left of the Search field to select any previously defined filters to which you have access. If you select a filter, the list of displayed applications, resource groups or resources is updated.

- To define a new filter, click **Open filter** to the left of the **Select a filter** drop-down to open the *Filter conditions* panel. From here, you can define the following filters:
  - **Severity:** Select one or more severity as a filter condition.
  - **Business criticality:** Select a business criticality as a filter condition. Business criticality can be customized by an administrator.
  - **Other properties:** You can set more complex filter condition by clicking **Add condition**. See To use properties to define filters for more details.
  - **For resources only**:
    - Using **Last updated**, you can choose to display only the resources that were created or changed during a specific time period (prior to 'now'). Note that changes in severity or status in this time period are not flagged. Choose either minutes, hours, or days, and select a number between 1 (one) and 500.
    - Using **Geospatial**, you can filter resources that have been enriched with a geographical location. See To define a geospatial filter for more details.

2. To save a filter for future use, complete the following steps:

   a. After you have defined the conditions for a new filter, click **Save as new filter** on the overflow menu.

   b. Define a **Filter name** and **Description** for the filter. The name must be unique.

   c. Define the filter user and their level of access.

   - **`Only me`**: Select for private use.
   - **`Everyone`**: Select either **`Use`** or **`Edit`**. If you select Edit, everyone will be able to change the filter.
   - **`Specified users, user groups or both`**: Select from the available **`Users`** or **`User groups`**, and then specify the level of control, that is, either **`Use`**, **`Use and edit`**, or **`Use, edit and manage`**

   d. Click **Save as new filter**. The specified users will be able to select the filter from the **Select a filter** drop-down list.

   Note: You can delete only the filters that you have created. To do so, select **Delete filter** from the overflow menu.

To use properties to define filters

1. To define a filter condition, complete the following steps:

   a. Select a property from the **Choose a property** drop-down.

   Note: The default Filter builder properties are **`entityTypes`**, **`name`** and **`tags`**. An administrator can add additional properties for filtering.

   b. Select an operator appropriate to that property from the **Choose an operator** drop-down.

   c. Depending on the property selected, either type in a value or select one using the **Value** field (or drop-down).

   d. Click **Apply** to filter the Results table.

   Note: You can delete, edit or disable the filter, or save it for future use. Any filter conditions you set are maintained in the URL, and can therefore be shared via direct-launch URL string.

   Example filters:

   - Example 1: Select the **`tags`** property, then the **`excludes all`** operator, then choose one of the available tags from the Value drop-down. Once applied, all results with that tag are excluded from the *Results* table.
   - Example 2: Select the **`name`** property, then the **`equal to`** operator, then type a resource name into the **Value** field. Once applied, only results of that name are included in the *Results* table.

2. Save your filter.

To define a geospatial filter

1. To set a geospatial filter, define the following scope:

   - **All resources with geospatial data**:
     - All resources that have geospatial properties are displayed.
     - The Area filter parameter is unavailable.
   - **All resources without geospatial data**:
     - All resources without geospatial properties are displayed.
     - The Area filter parameter is unavailable.
   - **Within area**:
     - Click Select area to open a world map. Use the cursor and zoom function to move to a desired location, then use the selection tool to specify the area. Click Apply selection to return to the previous screen, where all resources that have geospatial properties and are located within the selected area are listed.
     - From here, you can clear or edit the area.
   - **Outside area**:

- Click Select area to open a world map. Use the cursor and zoom function to move to a desired location, then use the selection tool to specify the area. Click Apply selection to return to the previous screen, where all resources that have geospatial properties and are located outside the selected area are listed.
- From here, you can clear or edit the area.

- **Include resources in overlapping area**:
  - Click Select area to open a world map. Use the cursor and zoom function to move to a desired location, then use the selection tool to specify the area. Click Apply selection to return to the previous screen, where all resources that have geospatial properties and are located within the selected area are listed, as are resources adjacent to or overlapping with the selected area.
- **Exclude resources in overlapping area**:
  - Click Select area to open a world map. Use the cursor and zoom function to move to a desired location, then use the selection tool to specify the area. Click Apply selection to return to the previous screen, where all resources that have geospatial properties and are located within the selected area are listed, though resources adjacent to or overlapping with the selected area are excluded.

2. Save your filter.

The *Resource management* page

You can view your applications (services) and details about them, such as their constituent resource groups and resources, on the *Resource management* page. If you have many applications, resource groups or resources in your topology database, you can use sophisticated searches or define search filters to find them.

Lists of applications, resource groups and resources are displayed on separate tabs.

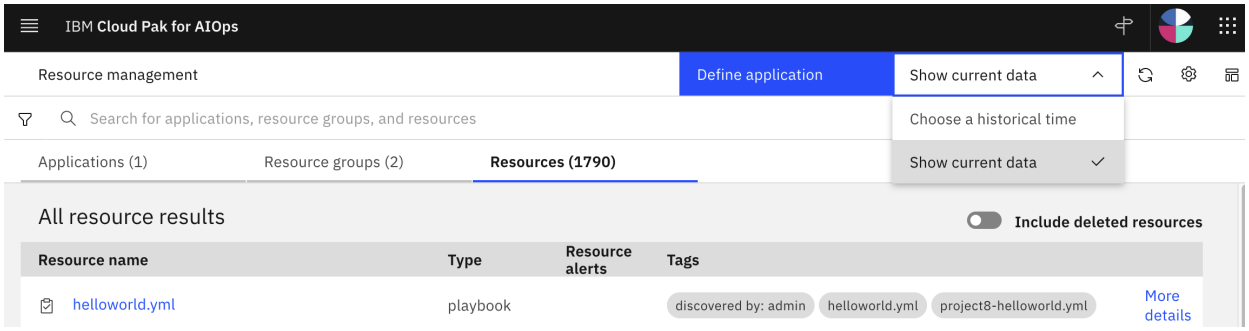


Figure. Resource management page (part)

- About resource management
  - The Applications tab
  - The Resource groups tab
  - The Resources tab

Remember: You can configure the UI to label your groups-of-groups collection of resources as either 'applications' or 'services'. For more information, see the following tip on changing terminology.

Before you begin

In order to view applications and their constituent elements, you must have obtained resource data, created resource groups, and defined applications, as documented in the following sections:

- Configure observers and run jobs.
- Use templates to generate resource groups.
- Define or edit applications.

About resource management

After you log into IBM Cloud Pak for AIOps console, you open the *Resource management* page from the main navigation menu.

Lists of applications, resource groups and resources are displayed on separate tabs. applications and resource groups are divided into 'All results' and 'Favorites' sections. You can display your favorite applications and resource groups either as cards or in a table; however, resources are always displayed in a table, and cannot be designated as favorites.

If many applications, resource groups or resources are listed, you can use the 'search' or 'filter' functions to identify specific ones, which you can then edit or view as topologies. For instance, you can define filter conditions to display the resources that have a specific tag applied to them. For more information on searching, see Resource management: Searching and filtering.

From here, you can define new applications, view your data at a historical point in time, and access the topology configuration, user preferences and template builder functionalities.

If the status for an application, resource group or resource is available, an icon that depicts the status severity level and the number of events that caused the severity are displayed. The severity of a status ranges from `clear` (white tick on a green square) to `critical` (white cross on a red circle).

Severity status icons

Icon	Severity

Icon	Severity
	clear
	indeterminate
	information
	warning
	minor
	major
	critical

The Applications tab

The applications tab lists the name, icon, status, tags, business criticality, and number of resource groups for each of the applications in the 'All applications' table. You can define new applications or edit or delete existing ones, assign them a business criticality, add estimated costs for application disruptions, assign favorites, or display them either as a topology or in a tabular format.

For information about defining new applications, see Defining and editing applications.

Application details displayed:

- The overall status of the application.
- The application creation date
- The last updated date
- The included resource groups or resources
- Any associated tags
- Any business criticality set
- Any associated incidents

  An incident is a collection of insights that comes from different data sources such as logs, events, and alerts. Incidents represent the determination of an incident and are categorized by priority from 1 (high) to 5 (low). They build understanding and help drive remediation.

  For more information, see Incidents.

- A link to the *More details* tearsheet (page).

  You can also access more details from the overflow menu to display a sidebar that shows all the current stored properties and values for the specified application in tabular and JSON format. In addition to the Properties tab, the *More details* page also contains Related resources and Related resource groups tabs.

  ```
  **Tip:** If a property has been excluded from view by an administrator, it is removed from the tabular view, but remains visible in JSON format.
  ```

- A heart icon (*Add favorite*) to add the item to the favorites section. When selected as a favorite, the tooltip for the icon changes to *Remove favorite*.
- Click the application name to open a topology view of the application.

  The topology shows you all the resources that make up your business application and how the resources are connected. You can use this interactive tool to drill into individual resource detail and status, enable timeline to view changes over time, and more. You can also use this view to triage issues when an issue occurs.

  For more information, see Managing topologies.

  Note When viewing an application as a topology, you can click **View Legend** to open a Help panel that describes the visual elements, such as symbols, icons and line styles, that are used in the rendering of the topology. You can also click the **Tabular view** button to switch from a topology view to a table view.

- Click **Choose a historical time** on the toolbar to view applications at a historical point in the past (the default selection is **Show current data**.)

  Choose a specific date and time in the past, and click **Apply**.

The Resource groups tab

The Resource groups tab lists the name, icon, type, status, tags, business criticality and number of resources for each of the resource groups in the *All resource groups* table. You can add resource groups to applications, assign them a business criticality, assign them as favorites, or view them either as a topology or in tabular format; however, you cannot create resource groups from here.

For information about using templates to create and manage resource groups, see the template topics.

Resource group details displayed:

- The overall status of the resource group
- The creation date
- The last updated date
- The included resources
- Any associated tags

- Any business criticality set
- Any associated incidents

  An incident is a collection of insights that comes from different data sources such as logs, events, and alerts. Incidents represent the determination of an incident and are categorized by priority from 1 (high) to 5 (low). They build understanding and help drive remediation.

  For more information, see Incidents.
- A heart icon (*Add favorite*) to add the item to the favorites section. When selected as a favorite, the tooltip for the icon changes to *Remove favorite*.
- A link to the *More details* tearsheet (page).

  You can also access *More details* from the overflow menu to display a page that shows all the current stored properties and values for the specified resource group in tabular and JSON format. In addition to the Properties tab, the *More details* page also contains Related resources and Related applications tabs.

  `**Tip:** If a property has been excluded from view by an administrator, it is removed from the tabular view, but remains visible in JSON format.`
- Click the resource group name to open a topology view of the resource group.

  The topology shows you all the resources that make up your resource group and how the resources are connected. You can use this interactive tool to drill into individual resource detail and status, enable timeline to view changes over time, and more.

  For more information, see Managing topologies.

  Note When viewing a resource groups as a topology, you can click **View Legend** to open a Help panel that describes the visual elements, such as symbols, icons and line styles, that are used in the rendering of the topology. You can also click the **Tabular view** button to switch from a topology view to a table view.
- Click **Choose a historical time** on the toolbar to view resource groups at a historical point in the past (the default selection is **Show current data**.)

  Choose a specific date and time in the past, and click **Apply**.

The Resources tab

The Resources tab of the *Resource management* page lists all available resources by name, icon, status, type, tags and business criticality in the 'All resources' table. You can view them as topologies, assign them a predefined business criticality, compare resource details, or open the *Resource details* window to display all resource information.

Optionally, you can list deleted resources.

Resource details displayed:

- The overall status of the resource
- The creation date
- The last updated date
- Any associated tags
- Any business criticality set
- A link to a resource map (if defined)
- A link to **More details**. You can also access more details from the overflow menu to display a sidebar that shows all the current stored properties and values for the specified resource in tabular and JSON format. The details also indicate the data origin, such as observer and observer job name, status, related and neighbor resources and relationships, and location (if defined).

  Tip: If a property has been excluded from view by an administrator, it is removed from the tabular view, but remains visible in JSON format.
- Click **Compare resources** on the overflow menu to open the *Compare resources* dialog. From here, you can search for and select a second resource to compare properties in table format.
- Click a resource name to display it as a topology view.

  From this view, you can drill into individual resource detail and status, enable the historical timeline to view changes over time, and more.

  For more information, see Managing topologies.

  Note When viewing a resource as a topology, you can click **View Legend** to open a Help panel that describes the visual elements, such as symbols, icons and line styles, that are used in the rendering of the topology. You can also click the **Tabular view** button to switch from a topology view to a table view.
- Click **Choose a historical time** on the toolbar to view resources at a historical point in the past (the default selection is **Show current data**.)

  Choose a specific date and time in the past, and click **Apply**.

Defining and editing applications (services)

You can define and edit applications by adding (or removing) resource groups, icons and tags, assigning business criticality levels, and setting application disruption costs.

Remember: Beginning with IBM Cloud Pak for AIOps version 4.8.0, you can configure the UI to label your groups-of-groups collection of resources as either 'applications' or 'services'. For more information, see the following tip on changing terminology.

The following tasks are documented here:

- Defining an application
- Editing an application
- Assigning business criticality
- Adding estimated costs for service disruption
- Deleting an application

Before you begin

- Ensure that you have your system set up with your required inventory integrations, such as for Kubernetes, VMware, or Instana, completed and working. These integrations provide the topological resource data for your applications.
- (Optionally) ensure that business criticality definitions have been created by an administrator user so that you can assign them to an application. This is described in the Defining business criticality topic.
- Ensure that groups of resources are created within your system and are available on the *Resource management* page.
    - Resource groups are used to manage specific resources. You can have resource groups within an application and also separate from any application. These groups can be from different sources.
    - Resource groups can be created and managed through a connected inventory provider, such as Kubernetes, VMware, or Instana, or through a group template. The resources within a group must be created and managed through the integration with the inventory provider.
    - If you need to change the resources within a resource group, do not directly manipulate your group. Use the group template or manage the group within the connected inventory provider.
    - When created and available, resource groups display on the *Resource groups* tab of the *Resource management* page and included within applications.
- **Remember** that you can use the searching and filtering functionality to identify and gather the components to include in your applications. For instance, you can define filter conditions to display the resources that have a specific tag applied to them. For more information, see Resource management: Searching and filtering.

About Instana applications

Instana applications are identified by their logo and tag, as well as by their second entity type of `aiopsManagedApplication`; however, they cannot be deleted. Instana resource groups cannot be removed from an Instana application, but any user-added resource groups or tags can be removed, and the icon can be changed.

Defining an application

Defining an application involves selecting the groups of resources to compose the application, previewing the application, and then adding any additional details as required.

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**. You can select either the **Applications**, **Resource groups** or **Resource** tabs.
3. Click **Define application**. The page for defining an application is displayed.
4. Select the groups of resources that you want to include in the application.

   When you select a group, the group name is added to the sidebar. This sidebar shows all of the groups that you have selected for the application.

   You can browse through the table of resource groups to select groups or use search to find specific groups. You can search for groups by name, by the group type, or by tags to view similar groups. Use search to narrow the results to help you find specific groups. By default the results display the first results in the system. As you enter a search term, the results are narrowed down based on your entered term. The auto-suggest menu also displays suggested terms based on existing groups.
5. Click **Next** to preview the application.

   When you preview an application, you can use the left checkbox to deselect resource groups on the sidebar in order to visualize your application without those groups. To fully remove any superfluous resource groups from the application, use the **X** to the right of the resource group.
6. Click **Next** to add additional details.
    - Enter a name for the application.
    - Set an icon for the application by using the drop-down list. This icon is used to identify the resource type in the topology view of your applications.
    - (Optional:) Mark the application as a favorite.
    - (Optional:) Select any tags for classifying the application. Tags can be used to identify similar applications and to distinguish applications from other applications. To add a tag, you need to search for the tag by name and then select the tag. To create a new tag, type the tag and then select the **Create tag** option in the dropdown.
    - (Optional:) Select a business criticality from the **Select business criticality** drop-down list.
    - (Optional:) Enter an estimated disruption cost in the form of a numerical value in the **Estimated cost per minute for service disruption** box.

- (Optional:) Remove any groups that you do not want in the application.

7. Click **Define application**. The application is added to the list of all applications on the page, and if selected, to the list of favorite applications.

After the application is defined, you can view the application in the list of applications on the *Resource management* page. You can also view the application details page to see the status and topology, and any associated incidents for the application. For more information, see Viewing applications.

Editing an application

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**, then select the **Applications** tab.
3. In *Favorites* or *All application results*, find the application that you want to edit. You can use pagination or the search field to find the specific application.
4. Click the options (3-dot icon) for the application entry in the table. From the menu, click **Edit application**. The *Edit application* page is displayed.
5. Edit the groups of resources, then preview the application. Edit any additional details as required.
6. After you complete your changes, click **Update application**.

Assigning business criticality

Note: If you do not want to assign the default business criticalities, bespoke definitions must first be created by an administrator user as described in the Defining business criticality topic.

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**, then select the **Applications** tab.
3. In *Favorites* or *All application results*, find the application that you want to edit. You can use pagination or the search field to find the specific application.
4. Click the options (3-dot icon) for the application entry in the table. From the menu, click **Assign criticality**. The *Assign business criticality* dialog is displayed.
5. Select a pre-defined business criticality from the drop-down list. **Remember:** Custom definitions can only be created by an administrator, as described in Defining business criticality.
6. Click **Save** to set the priority level for your application.

Adding estimated costs for service disruption

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**, then select the **Applications** tab.
3. In *Favorites* or *All application results*, find the application that you want to edit. You can use pagination or the search field to find the specific application.
4. Click the options (3-dot icon) for the application entry in the table. From the menu, click **Edit application**.
5. In the *Edit Application* page, click the *Add Details* tab.
6. Enter a numerical value in the *Estimated cost per minute for service disruption (optional)* box. This value is used to chart disruption costs in the **AIOps Insights** dashboard. For more information, see AIOps Insights dashboard.

Deleting an application

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**, then select the **Applications** tab.
3. In *Favorites* or *All application results*, find the application that you want to delete. You can use pagination or the search field to find the specific application.
4. Click the options (3-dot icon) for the application entry in the table. From the menu, click **Delete service**. (Reminder: Instana applications **cannot** be deleted. The delete function is grayed out.)
5. On the *Confirm delete* dialog, click **Delete application** to confirm the deletion. The application is deleted and no longer displays in the list of all applications. The groups and resources that composed the application are not deleted.

Managing topologies

You can render, view, and manipulate topologies for applications (services), resource groups and resources to help you better understand how they are constructed. You can also configure the topology views to customize the display of topology elements, and add custom elements and tools.

Remember: Beginning with IBM Cloud Pak for AIOps version 4.8.0, you can configure the UI to label your groups-of-groups collection of resources as either 'applications' or 'services'. For more information, see the following tip on changing terminology.

- Rendering a topology

- Viewing a topology
- Templates types
- Defining business criticality
- Running data administration routines
- Configuring topology user preferences
- Configuring topologies

Rendering (visualizing) a topology

You define the scope of the topology that you want to render by specifying a seed resource, the number of relationship hops surrounding that resource, as well as the types of hops. The topology service then supplies the data that is required to visualize the topology.

Before you begin

To visualize a topology, ensure that inventory integrations are configured.

Note: The IBM Cloud Pak for AIOps console has a default timeout set at 30 seconds. If service requests are not received in that time, a timeout message is shown, as in the following example:

```
A time-out has occurred. No response was received from the Proxy Service within 30 seconds.
```

For more information, see Topology render timeout.

Tip: For information about using the File and REST SDK observers to build custom topologies, see Building a custom topology.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. Open the *Resource management* page from the main navigation menu.

   Resources are displayed arranged in *Applications*, *Resource groups*, and *Resources* tabs. Both the *Search* and *Filtering* functionality is accessible at the top left.
3. Find a resource to use as the seed resource of the topology visualization.

   You can find the required seed resource within existing applications (services), resource groups or resources, or use the filtering or search functionality to identify it.

   Search: As you type in a search term that is related to the resource that you want to find, such as name or server, a drop-down list is displayed. This contains the suggested search terms that exist in the topology service. If the resource that you want to find is unique and you are confident that it is the first result in the list of search results, you do not need to select a result from the suggested search terms. Instead, choose to click the **View topology for closest matching resource** icon in the **Suggest** drop-down. This renders and displays the topology for the closest matching resource.

   For more information on searching and filtering, see the Resource management: Searching and filtering topic.

   For more information about advanced query syntax, see the topology query reference topic.

   If you select one of the suggested results, the Search Results page is displayed listing possible resource results. The Results are listed under separate **Resources** and **Groups** tabs.

   - Deleted resources

     You can select the **Include deleted resources** toggle in order to include deleted resources in the Results page (unless an administrator has disabled that feature). Deleted resources are marked as such, and the date and time that they were deleted is displayed.

     If you click **View topology**, the resource is displayed in the Topology Viewer with the history timeline open, and set to 1 ms before deletion. When closed, the topology is cleared. If you display applications (services) or resource groups as topologies, the Incidents panel is opened by default on the right hand side. You can close the panel to maximise the topology display area, or switch from a graphical topology view to a tabular view.

     When you search for resources with a previous point in time set in the history timeline, deleted resources are **not** returned and the toggle to include them is not available.

     Deleted templates or defined topologies are **not** included in any search results.

   - Groups

     Groups **must** be defined before they are listed. If you are defining Group templates in the template builder, search results are listed under separate **Resources** and **Groups** tabs.

     To designate as a favorite a group search result and add it to the collection of topologies accessible from the *Resource management* page, tag it as a favorite by selecting the **Add to favorites** icon next to it.

   - Tabular view

     Resource groups and applications (services) can be displayed in table format, which lists the included resources alphabetically.

   For each result, the name, type and other properties are displayed.

If a status other than clear exists for a search result, the maximum severity is displayed in the information returned. Also, a color-coded information bar before each result displays all non-clear statuses (in proportion).

You can expand a result in order to query the resource or defined topology further and display more detailed, time-stamped information. Examples of this include its state and any associated severity levels, or when the resource was previously updated or replaced (or deleted).

You can click the **View Topology** button next to a result to render the topology.

Restrictions:

- When you load a predefined topology, it is displayed in a 'group' version of the Topology viewer, which has restricted functionality. You are unable to follow its neighbors, or change its hops, or use its advanced filters.
- You can recenter the defined topology from the context menu, which loads it in the Topology viewer with all its standard functionality.

4. Select a number to define the number of relationship hops to be visualized from the seed resource. The default value is set at `1`.

   For more information about customizing the maximum hop count, see Configuring advanced settings.

5. Choose one of the following hop types:

   - The **Host to Host** hop type uses an aggregate traversal across elements with the entity type 'host' to generate a view that shows host to host connections.
   - The **Element to Element** hop type (default) performs the traversal by using all element types in the graph.
   - The **Element to Host** hop type provides an aggregated hop view like the 'Host to Host' type, but also includes the elements that are used to connect the hosts.

   For more information about customizing the hop type, see Configuring advanced settings.

6. Filter the topology before rendering it.

   Open the Filter toolbar by using the **Filter** toggle, and apply the resource, relationship, or severity filters required. You can filter resources that are displayed in the topology, or set filters before rendering a topology to prevent a large, resource-intensive topology from being loaded. If a filter is applied to a displayed topology, the text 'Filtering applied' is displayed in the status bar under the topology.

   For more information about using filters, see Viewing a topology.

7. Click **Render** to render the topology.

The *Topology viewer* connects to the topology service and renders the topology. By default, the view is refreshed every thirty seconds, unless specified otherwise.

What to do next

Next, you can refine and manipulate the view for further analysis.

If needed, you can rebuild a topology. For instance, open a topology and use the visualization tools to modify the topology settings and then clicking **Render**. This action runs the topology visualization to render the topology based on your new settings.

Tip - Incidents: The topologies of applications (services) and resource groups are displayed with the Incidents panel open on the right hand side, which you can collapse to increase the topology display area. If you switch to a tabular view and use the Incidents panel functionality to sort or filter the incidents, the table is dynamically updated to display only the resources affected by the selected incidents.

Troubleshooting: Topology render timeout

If you receive a timeout message, this can be due to a number of reasons:

- Large amounts of data being retrieved for complex topologies
- Too many hops counts specified
- Issues with the back-end services

Workarounds

- Check that all services are running.

  You can verify that the containers are running by using the following command:

  ```
  oc get pods -n <namespace> | grep <pod_name>
  ```

  The system should return text indicating that all containers have a state of Up.

- Increase the timeout when rendering topology views. The Red Hat OpenShift route used when rendering topologies in the UI has a default timeout of 30 seconds, which might not be long enough to render a large or complex topology view. To increase the timeout value, for example to 300 seconds, add the following code to the spec section of the custom resource of the operator (which is usually the `asm` instance; you can find the instance name using the `oc get asm` command):

  ```
  spec:
     helmValuesASM:
        global.asm.uiApi.topologyServiceTimeout: 300
        global.asm.uiApi.layoutServiceTimeout: 300
  ```

  The up-api pod should restart with the values of the environment variables TOPOLOGY_SERVICE_TIMEOUT and LAYOUT_SERVICE_TIMEOUT now set to 300 seconds.

- You can list the routes by using the `oc get routes` command. To increase the timeout for the UI change, use the following example:

```
oc annotate route noi-ui-api --overwrite haproxy.router.openshift.io/timeout=300s
oc annotate route noi-ibm-hdm-common-ui-uiserver --overwrite haproxy.router.openshift.io/timeout=300s
```

- Lower the hop count to reduce the service load. For more information about customizing the maximum hop count, see the Defining advanced topology settings topic.

Troubleshooting: No results found

When you enter text into the **Resource management** field, surround the text with quotation marks, such as in `"Search value"`. If the quotation marks are omitted, topology viewer doesn't display results even if the search value was included as data.

Building custom topology

Custom topology can be added to the system by using the File and REST SDK observers. These resources can be merged with data from the other observers.

Topology model

The basic topology model consists of the following items:

- Resources can be used to represent anything, such as network elements, software, and people.
- Edges are the relationships between resources.
- Groups collect resources to allow them to be easily viewed and to provide topological correlation.
- Applications or services are a collection of groups.
- Status indicates that alarms are mapped to a resource.

Topology model `partOf` edges

The **`partOf`** composition edge from child resources to parents is a special case. This edge is not visible in the resource management view and associated configuration. The edge is used for Host to host and Element to host resource management views. The edge is considered when you match the alerts to resources. An alert that is matched against a part of the child is also seen against the parent.

The **`partOf`** edges have the following limitations:

- A **`partOf`** relationship is a relationship from many children to a single parent.
  - A child cannot have multiple parents.
- A parent cannot be **`partOf`** another parent.

Resource lifecycle

Each resource or group within the scope of a provider has a lifecycle that is managed by the job that modeled it. Load jobs are expected to take snapshots of the entire system. If a load job finds the resources X, Y, Z on one run, and the next run finds resources A, B, X, then Y and Z are deleted. The listener jobs need to explicitly delete resources.

Resource lifecycle APIs:

- The **`insertReplace`** PUT method removes any properties that are not listed.
  - The exception is a placeholder with only **uniqueId** and **_references** properties.
- The **`insertUpdate`** POST method adds to the existing properties of the Topology model.

Edge lifecycle

To maintain the edge lifecycle, keep the following items in mind:

- Model edges by using **_references** nested within a resource.
- Always use **`insertReplace`** PUT method semantics.
- Model edges in a consistent direction.
- Edge type and direction per resource provides the replacement scope.
- With the **`insertReplace`** PUT method, all the edges of the same edgeType and direction are replaced.
- Direction is implied by **_toUniqueId** or **_fromUniqueId**, or use **using _direction** to remove all edges.
- The edge lifecycle can be maintained independently of properties by using a payload with only the **uniqueId** and **_references** properties.

Edge deletion

Edges can be deleted in multiple ways.

For example, if a resource X is defined with an outgoing *uses* edge to another:

```
"_references":[{"_edgeType":"uses","_toUniqueId":"Y"}]}
```

This edge is deleted if:

- Either the **`X`** or **`Y`** end is deleted.
- **`X`** is replaced with:

```
"_references":[{"_edgeType":"uses","_toUniqueId":"not Y"}]
```

- `Y` is replaced with:

```
"_references":[{"_edgeType":"uses","_fromUniqueId":"not X"}]
```

Note: The following examples show the syntax to remove all the edges of that type and direction.

- `X` is replaced with:

```
"_references":[{"_edgeType":"uses","_direction":"out"}]
```

- `Y` is replaced with

```
"_references":[{"_edgeType":"uses","_direction":"in"}]
```

Viewing topologies

A topology is a graphical representation of an application (service), or a resource group, or an individual resource. When you render a topology, it displays all of the constituent elements and relationships that make up your topology, and then lets you refine and manipulate the view.

You can use the interactive topology tools to drill into the individual resource details and status, enable a timeline to view changes over time, and more. You can also use this view to triage issues when an incident occurs.

When you view a topology, you can zoom in and out of specific areas of the topology or pan across it in various ways. You can auto-fit the topology into the available display window, draw a mini map, or redraw the entire topology. You can also filter your types of resources that are displayed, or your types of relationships that are rendered.

Tip: While viewing a topology, you can click **View legend** to open a Help panel that describes the visual elements used in the display of the topology. Alternatively, you can click **Tabular view** to display the topology of an application or resource group in table format instead.

Severity status

Severity status icons

Icon	Severity
	clear
	indeterminate
	information
	warning
	minor
	major
	critical

You can manipulate the topology by using a number of tools:

- To view a topology
  - Modifying the presentation
  - Tabular view
- Visualization tools
- Interacting with a topology

Before you begin

To view a topology, it must be defined first. The following procedure details how to view *existing* topologies for applications, resource groups and resources.

- For more information about creating a topology, see Rendering (visualizing) a topology.
- For more information about creating applications, see Defining and editing applications.

About this task

You view a topology from the **Resource management** page.

To view a topology

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**.

   On the *Resource management* dashboard, applications, resource groups and resources are presented on separate tabs. In addition, the dashboard is divided into 'All applications results' and 'Favorites' sections. You can display favorite applications and resource groups either as cards or in a table. Resources are always displayed in a table, and cannot be designated as favorites.

3. To view an application, resource group or resource as a topology, click the name. Alternatively, use Search.

   - **Search:** As you type in a search term, such as a name, tag, entity type or unique Id, a drop-down list is displayed with suggested search terms that exist in the topology service. If you select one of the suggested results, the Search Results page is displayed listing possible resource results that

match the selected search term. The Results are listed under separate tabs.

- If the resource that you are searching for is unique, the resource can be included in the drop-down list of suggestions. If you are confident that the listed suggestion is the resource that you are searching for, you can click the **View topology for the closest matching resource** icon in the drop-down list to render the topology for the closest matching resource to that suggestion.
- **Optional:** If the resource that you are looking for might be deleted, you can select **Include deleted resources** (unless an administrator disables this feature). When this toggle is enabled, the search results include any deleted resources. Deleted templates or resource groups are **not** included in any search results. Deleted resources are marked as deleted and include the date and time that they were deleted. If you click **View topology**, the resource topology is displayed with the history timeline open. The timeline is set to 1 ms before the resource was deleted. When you close the timeline, the topology is cleared. When you search for resources with a previous point in time set in the history timeline, deleted resources are **not** returned and the toggle to include them is not available.
- For each serch result, the name, type and other properties are displayed. You can expand a search result entry for more details, such as its status and any associated severity levels, or when the resource was previously updated. If a status other than clear exists for a search result, the maximum severity is displayed.
- **Tip:** You can perform more complex searches, such as searching with multiple search terms. For more information about advanced query syntax, see the topology query reference topic.

Viewing and modifying the topology presentation

When you view a topology, you can zoom in and out of specific areas of the topology, and pan across it in various ways. You can auto-fit the topology into the available display window, draw a mini map, redraw the entire topology, or filter the types of resources displayed or the types of relationships rendered.

Incidents: The topologies of applications (services) and resource groups are displayed with the Incidents panel open on the right hand side, which you can collapse to increase the topology display area. If you switch to a tabular view and use the Incidents panel functionality to sort or filter the incidents, the table is dynamically updated to display only the resources affected by the selected incidents.

Tip: Click **View Legend** to open a Help panel that describes the visual elements that are used in the rendering of the topology (such as lines styles, icons, and symbols).

Tip: By default, a topology is displayed in graphical view format. Click the **Tabular view** button to switch to a table view, or the **Map view** button to view the topology resources displayed on a map.

Information bar: A bar at the bottom of the topology visualization displays the status of the rendered topology, including a timestamp on the left to indicate the time of the most recent refresh. If two time periods are being compared, both are indicated. Additional information on the right side of the information bar indicates the number of resources that are rendered, and the number of associated relationships. The bar also indicates whether any of the resources or relationships were added or removed since the last refresh, and whether a filter is applied.

When changes are made to a topology, the following conventions are used to identify the changes:

- Deleted resources

  A minus icon indicates that a resource was deleted since the topology was last rendered. This icon displays when a topology is updated, and within the history views.

- Added

  A purple plus icon indicates that a resource was added since the topology was last rendered. This icon displays when a topology is updated, and within the history views.

- Added (neighbors)

  A blue asterisk icon shows that a resource was added by using the 'get neighbors' function.

You can manipulate the topology by using a number of tools. Depending on whether you are viewing the topology for a resource, resource group or an application, the tools that are available can be different.

Tabular view

If you click **Tabular view** to display the topology in table format, it lists the **`Name`**, **`Type`**, **`Resource alerts`**, **`Tags`** and **`Business criticality`** (if defined) for each resource. The (collapsable) Incidents panel on the right of the topology is open by default.

- **Tip:** If you use the Incidents panel functionality to sort or filter the incidents, the table is dynamically updated to display only the resources affected by the selected incidents.
- You can also dynamically filter the table by creating a filter, using a predefined filter or by typing names, types or tags into the **`Search`** field.

  Note: Elements that are not displayed when viewing resources in table format, such as relationship types, are not available for filtering.

- You can display resources at a historical point in the past by clicking **Choose a historical time** on the toolbar. From the *Choose a historical time* dialog, select a new moment in time up to the data retention limit set for your system (the default is thirty days in the past). If you return to the current time, the data is refreshed.

  Tip: If you switch back from a tabular to a topology view, the time pin has moved according to the historical date you have selected.

- You can click **`Resource details`** on the overflow menu to open the *Resource Details* window.
- You can perform a number of additional context-dependent activities from the overflow menu of each displayed item in the table, such as viewing comments, showing the item in a map, assigning business criticality, opening it in the topology viewer, or building a template.

Tip: You can share a link to the application (or resource group) in tabular format with other users.

Troubleshooting: When moving between a tabular and a graphical view several times while also using the timeline toggle, the display may freeze. To avoid this issue, do not disable the timeline while using historic mode in the tabular view. If the issue does occur, refresh the page to reset.

Tabular view overflow menu

The overflow menu (the three dots next to each table entry) gives you access to the following functionality:

- Click **Resource details** on the overflow menu.

  The *Resource details* page has a number of tabs.

  Properties: Displays a window that shows all the current stored properties for the specified resource in tabular and JSON format. If a property has been excluded from view by an administrator, it is only visible in JSON format. The resource name is an active link that opens a topology view of the resource in context. Toggle the Show JSON switch On or Off to display the properties in JSON format.

  Alerts: Displays a dialog that shows the time-stamped statuses related to the specified resource in table format.

  Data origin: Displays the point of origin of the resource data, such as a specific observer job. If you have the appropriate user permission, such as the administrator role, the observer job name is an active link, which opens the observer job page from where you can re-run the job, view its history, or edit the job.

  Related applications: Displays all related applications as active links, which open the service topologies in new windows. Also lists the group type, number of groups, status, and tags.

  Related resource groups: Displays all related resource groups as active links, which open the resource group topologies in new windows. Also lists the resource type, number of resources, status, and tags.

  Neighbor resources: Displays all neighboring resources in a table. Click a resource name to open the details page for the resource.

  Location: Opens a map in which the resource is geo-located.

  Structure view: Displays all resources that are part of a host or parent resource.

- Click **Resource alerts** on the overflow menu to access the *Resource details* page for the selected resource with the Alerts tab selected.
- Click **View related incidents** on the overflow menu to display related incidents for the selected resource on the Incidents panel.
- Click **Comments** on the overflow menu to access the commenting functionality.

  When selected, this displays any comments recorded against the resource. By default, resource comments are displayed by date in ascending order. You can sort them in the following way:

  - Oldest first
  - Newest first
  - User Id (A to Z)
  - User Id (Z to A)

  Users with the appropriate permissions can view or add new comments.
- Click **Assign business criticality** on the overflow menu to define business criticalities for the selected resource.
- Click **Build a template** on the overflow menu to opens the *Create a new template* dialog, from where you can create a new resource group template.
- Click **Compare resources** on the overflow menu to select a second resource for side-by-side comparison in a table.
- Click **Open in Topology viewer** on the overflow menu to open a resource as a topology view.

  From here, you can drill into individual resource details and status, enable a timeline to view changes over time, and more. You can also use this view to triage issues.
- Click **Show on map** on the overflow menu to display the resource on a map.

Map view

The map view displays the topology elements on a map. For more information, see Viewing resources on a map.

Vizualisation tools

A number of tools and functions let you manipulate the topology presentation.

Refresh rate

You can expand the **Refresh rate** drop-down field to select your preferred refresh rate or to pause the option. This rate determines when the topology page auto-updates to search for and include new changes.

You can pause the topology data refresh, or specify the following values: 10 seconds, 30 seconds (default), one minute, or five minutes.

Refresh

You can click **Refresh** to manually refresh the topology data, which also clears all user-specific session data.

About this application

Opening this twisty displays a summary for the selected application topology, such as the date and time it was created, the associated groups and whether any have been added, any tags and business criticalities associated with it, and the date and time it was last updated.

About this resource group

Opening this twisty displays a summary for the selected resource group topology, such as the date and time it was created, the associated applications and whether any have been added, any tags associated with it, and the date and time it was last updated.

Filters

You can either use a predefined filter by selecting it from the **Select a filter** drop-down, or create a new one by clicking the **Open filter** icon and defining new filter conditions.

For more information, see the searching and filtering topic.

Search

While viewing a topology, the Search functionality remains available. As you type in a search term, a drop-down list is displayed that includes suggested search results for the currently rendered topology and other topology entries from the topology database. You can search by name, tag, uniqueId, or entity type.

Tip: If you hover over a search result, the resource is highlighted in the topology window. When you click a search result, the topology highlights the selection and direct relationships, and search is closed.

For more information, see the searching and filtering topic.

Settings

From this menu, you can select to open pages for managing the following types of settings:

- **Topology configuration**

  You can configure the following topology elements to change how users can visualize, manipulate, and use topologies:

  - **Advanced settings**

    Control how the topology user interface works for all users.

  - **Topology tools**

    Create tools that can be accessed from within the topology's context menu.

  - **Resource types**

    Customize the style of your resources.

  - **Relationship types**

    Customize the style and label for relationships.

  - **Custom icons**

    Add custom icons for resources that are displayed in the topology.

  - **Rules**

    Define automated processing rules for your topology data.

  For more information, see Configuring topology elements.

- **Create exact template**

  Opens the *Using resource group templates* page already populated with the current view elements including the preview pane to the right of the Template builder. From here, you can create a new exact resource group template.

  Restriction: While viewing topologies, you can only create exact templates based on element-to-element topologies. The Create exact template option will not appear in the Settings drop-down for other views.

  Tip: As a further refinement, you can adjust your layout by manually positioning the resources in the topology, and this layout is then replicated when you generate a resource group from any exact template that you build from this topology.

- **User preferences**

  You can set your personal preferences for how topologies are rendered. The settings that can personalize are grouped into the following categories:

  - Updates

    These settings determine how and how often topologies are automatically updated.

  - Layout

    These settings determine the default presentation of topologies.

  - Miscellaneous

    Other settings, such as displaying messages, screen reader support, and troubleshooting.

  For more information, see User preferences.

Sharing options

You can share a topology either by obtaining a direct URL linking to the topology view, or by exporting a view of the topology as an image. To select a sharing option, click the **Sharing options** icon and then click your preferred option:

- **Obtain direct URL**

  This option displays the *Direct topology URL* dialog. This dialog includes a displayed URL, which captures the current topology configuration, including the layout type. The layout orientation is not tracked.

  Click the **Copy direct URL to clipboard** icon to the right of the URL to copy the direct-launch URL string for accessing the topology view. You can then share the URL with a colleague or use the URL to directly access the topology later.

  Then, click **Close**.

- **Export as PNG**

  Click this option to export a snapshot of the current topology in PNG format. The snapshot is automatically downloaded based on your browser download settings. By default, the file is saved as `topology.png`. You can now share the image as required.

- **Export as SVG**

  Click this option to export a snapshot of the current topology in SVG format. The snapshot is automatically downloaded based on your browser download settings. By default, the file is saved as `topology.svg`. You can now share the image as required.

Number of hops (resource topologies only)

The number of relationship hops to visualize from the seed resource. You can define the number of relationship hops to be performed, which can be from zero to four, unless this setting has been customized. The default is set at 'one'.

For more information about configuring the maximum hop count, see the Configuring advanced topology settings.

Type of hop (resource topologies only)

The type of graph traversal used. The options are:

- Element to element (default)

  This type performs the traversal by using all element types in the graph. If no hop type has been defined, a topology is rendered using the element to element hop type as default.

- Host to host

  This type generates a view showing host to host connections.

- Element to host

  This type provides an aggregated hop view like the Host to Host type, but also includes the elements that are used to connect the hosts.

  Tip: The URL allows you to set `h2h` for host to host, `e2e` for element to element, or `e2h` for element to host.

Render (resource topologies only)

This button causes the topology visualization to run to update a topology based on any changes to the topology or the navigation toolbar settings. This process integrates any new resources into the topology. Choosing this option recalculates the topology layout based on your current display settings, which can adjust the displayed topology significantly. Once the rendering is complete, the topology refreshes on a 30-second interval by default. You can pause the auto-update refresh, or select a custom interval.

Tip: The IBM Cloud Pak for AIOps console can time out when a large amount of data is being retrieved for rendering. For more information about how to address this issue, see Rendering (visualizing) a topology.

Delta

You can toggle the **Delta** button to show how your selected topology changes between two points in time.

Timeline

Use the **Timeline** toggle to open and close the History toolbar below the topology. This view shows how the selected resource changed over time. For more information about the details and how you can use this view, see Viewing topology history.

View legend

Click **View Legend** to open a Help panel that describes the visual elements that are used in the rendering of the topology, such as symbols, icons and line styles.

Topology view tools

A list of tool icons is displayed near the top right hand side. You can hover over some icons to display a submenu.

- **Select tool**

  Use this icon to select individual resources by using a mouse click, or to select groups of resources by creating a selection area (using click-and-drag).

- **Pan tool**

  Use this icon to pan across the topology by using click-and-drag on a blank area of the visualization panel.

- **Zoom Select tool**

Use this icon to zoom in on an area of the topology by using click-and-drag.

- **Zoom In**

  Use this icon to zoom in on the displayed topology.

- **Zoom Out**

  Use this icon to zoom out of the displayed topology.

- **Zoom Fit**

  Use this icon to fit the entire topology in the current view panel.

- **Overview**

  Use this icon to create the overview mini map in the lower right corner. The mini map provides an overview of the entire topology while you zoom in or out of the main topology. The mini map displays a red rectangle to represent the current topology view. **Note:** In some browsers, the use of the mini map might cause performance issues when displaying large topologies.

- **Layout**

  Use this icon to recalculate, and then render the topology layout again. You can choose from a number of layout types and orientations.

  - Layout 1

    A layout that displays all resources in a topology without applying a specific layout structure.

  - Layout 2

    A circular layout that is useful when you want to arrange a number of entities by type in a circular pattern.

  - Layout 3

    A grouped layout is useful when you have many linked entities, as it helps you visualize the entities to which a number of other entities are linked. This layout helps to identify resource groups of interconnected entities and the relationships between them.

  - Layout 4

    A hierarchical layout that is useful for topologies that contain hierarchical structures, as it shows how key vertices relate to others with peers in the topology being aligned.

  - Layout 5

    A peacock layout is useful when you have many interlinked vertices, which resource group the other linked vertices.

  - Layout 6

    A system board rank layout is useful when you want to view how the topology relates to a vertex in terms of its rank, and also how vertices are layered relative to one another.

  - Layout 7

    A rank layout is useful when you want to see how a selected vertex and the vertices that are immediately related to that rank relative to the remainder of the topology (up to the specified number of hops). The root selection is automatic.

    For example, vertices with high degrees of connectivity outrank lower degrees of connectivity. This layout ranks the topology automatically around the specified seed vertex.

  - Layout 8

    A root rank layout similar to layout 7, except that it treats the selected vertex as the root. This layout is useful when you want to treat a selected vertex as the root of the tree, with others being ranked below it.

Interacting with a topology

A displayed topology consists of resource nodes and the relationship links connecting the resources. You can interact with these nodes and links. You can use the topology to also view critical issues and interact with the related resource nodes to troubleshoot the issue.

You can select, move, and view more options for managing nodes and the associated resources.

- Select a node

  Selecting a node highlights that node, and emphasizes its first-order connections by fading all other resources.

- Move a node

  Click and drag a node to move it.

- View the context menu

  Right-click a node to view the context menu. From this menu you can select more options for viewing details and managing the topology. For the included options, see Context menu

Context menu

You open the context menu by using the right-click function. The context menu provides access to the resource-specific actions that you can perform.

For resource entities, you can access the following context menu items:

- Resource details
  - Alerts
  - Structure view
- Comments
- Assign criticality
- Compare resources
- Build a template
- Get neighbors
- Follow relationship
- Show last change in timeline
- Show first change in timeline
- Find path
- Recenter View

Resource details

Note: The details dashboard is accessed either from the topology context menu or the Resource management dashboard. Some items may be unavailable.

If opened from the Resource management tool in the UI, the resource name is an active link that opens a topology view of the resource in context, provided you have the correct user permissions (the **`waiops_manage_integrations`** or **`waiops_view_integrations`** permissions).

If a map has been defined, the Map link next to the resource name opens the Resource map view. When selected while viewing a topology history with delta mode On, the Resource details window displays the properties of the resource at both the reference time and at the delta time. 'Related services' and 'Related resource groups' are only displayed when not in historical mode.

Properties tab: Displays all the current stored properties for the specified resource in tabular and JSON format. If a property has been excluded from view by an administrator, it is removed from the tabular view, but remains visible in JSON format.

Tip: Toggle the **Show JSON** switch On or Off to display the properties in JSON format.

Alerts tab: Displays a dialog that shows the time-stamped statuses related to the specified resource in table format. The Severity and Time columns can be sorted, and the moment (that is, the historical time point) that Resource status was selected is also time-stamped.

You can use the **Show active only** toggle to remove inactive statuses. In addition, if any status tools have been defined, the status tool selector (three dots) is displayed next to the resource's statuses.

Click the status tool selector to display a list of any status tools that have been defined, and then click the specific tool to run it. Status tools are only displayed for the states that were specified when the tools were defined. The severity of a status ranges from 'clear' (white tick on a green square) to 'critical' (white cross on a red circle). For the list of status, see the Severity status table.

Tip: If statuses related to a specific resource are available, the resource is marked with an icon depicting the status severity level, and the Resource status tab will appear on the Resource details window, accessible through the resource context menu.

Data origin tab: Displays the point of origin of the resource data, such as a specific observer job. If you have the appropriate user permission, such as the administrator role, the observer job name is an active link, which opens the observer job page from where you can rerun the job, view its history, or edit the job.

Related applications tab: Displays all related applications as active links, which open the application topologies in new windows. Also lists the group type, number of groups, status, and tags.

Related resource groups tab: Displays all related resource groups as active links, which open the resource group topologies in new windows. Also lists the resource type, number of resources, status, and tags.

Neighbor resources tab: Displays all neigboring resources and their names, relationships and types, alerts (max severity only) and tags.

Location tab: Opens a map in which the resource is geo-located. Zoom functions are enabled, though the map is limited to the More details dialog. To open the full map, use the Map link next to the resource name.

Structure view tab: Displays all resources that are part of a host or parent resource (such as a device), which are linked together in a relationship via the 'partOf' edge regardless of their position; that is, it displays linked resources even if they are several hops removed from one another or from the host (parent) resource.

The Structure view tab displays all the current stored properties for a selected resource on the right in tabular and JSON format (as for the properties tab), with an additional breadcrumb above the details showing its distance from its parent resource.

On the left, a resource tree is shown, expanded to the resource from which the structure view was opened, with the parent resource at the highest level. The highest severity resource status is shown next to the name, although if a status is shown in parenthesis, the status applies to a resource nested at a lower level.

You can click on a resource name to open a topology view of the resource and its parent resource. When you display a parent resource, all resource linked to it via the partOf edge are also depicted. If you activate the Problem view toggle, all resources with no reported issues are filtered out.

Comments

When selected, this displays any comments that are recorded against the resource.

By default, resource comments are displayed by date in ascending order. You can sort them in the following way:

- Oldest first
- Newest first
- User ID (A to Z)

- User ID (Z to A)

To add a comment, enter text into the New Comment field, and then click **Add Comment** to save.

Assign criticality

Business criticality sets the priority level for a resource (or application, or resource group).

To assign a priority, select a definition from the **Business criticality** drop-down, then click **Save**.

Remember: Business criticality definitions must be created by an administrator user before you can assign them to a resource. This is described in the Defining business criticality topic.

Compare resources

You can select two resources and their properties for side-by-side comparison, with different values highlighted.

In Resource management, select the Resources tab, and next to a resource to be compared (Resource A), click **Compare resources** from the overflow menu.

The *Compare resources* page opens with the standard details for Resource A displayed in a table.

Note: Alternatively, you can launch the *Compare resources* page from a specific resource's context (right-click) menu in the topology viewer.

Use the **Search* field to identify and select a second resource (Resource B) for comparison.

The table displays the properties for Resource A and Resource B in adjacent table columns.

- Different property values are highlighted.
- If a historical time point was selected, the properties from that time point are displayed.

Build a template

Select the type of template to use to group resources

- Dynamic template

  This template automatically finds and updates multiple resource groups based on your criteria and how your services and infrastructure change.

- Tag based template

  This template creates and updates a single resource group containing the resources that have a specified set of tags.

- Exact template

For more information, see Resource group templates.

Get neighbors

When selected, a menu opens to display the resource types of all neighboring resources. Each resource type lists the number of resources of that type, and the maximum severity associated with each type.

You can click the menu option **Choose specific** to open a dialog to select specific resources that you want to add to the current topology visualization. You can also choose to include all neighboring resources. When you select a resource, a confirmation option displays for you to either **Add neighbors** or **Cancel**. After you select **Add neighbors**, **Undo** is available for you to undo the last add neighbor request.

Selecting **Get Neighbors** overrides any existing filters.

Follow relationship

When selected, opens a menu that displays all adjacent relationship types. Each relationship type lists the number of relationships of that type, and the maximum severity associated with each type. You can choose to follow all relationships, or only the neighbors of a specific type.

Show last change in timeline

When selected, this displays the history timeline, depicting the most recent change made to the resource.

Show first change in timeline

When selected, this displays the history timeline, depicting the first change made to the resource.

Find path

When you click the **Path tool**, the *Find path* dialog opens. From this dialog, you can define criteria for finding the shortest path between the currently selected resource and a second specified resource within the topology.

Within the **Path definition** section, you need to specify the start and end points (resources) for the path:

- **Path start resource**

  Identifies the selected resource from the topology.

- **Path end resource**

Search for the resource that you want to use with the starting resource for calculating the shortest path.

Within the **Filters** section, specify the resource types and relationship types to include or exclude from the calculated path topology.

- **Filter mode**

  Choose to **Exclude** or **Include** the resource or relationship types from the path.

- **Resource types**

  Search for or browse the drop-down list of resource types and select any types to be included or excluded. The list includes all defined types.

- **Relationship types**

  Search for or browse the drop-down list of relationship types and select any types to be included or excluded. The list includes all defined types.

- **Maximum number of hops**

  The maximum number of hops you can permit when calculating the path. You can set the value at 1-30 hops. The default is 30 hops.

When you have completed your selections, click **Find path**. A browser tab for the Topology viewer tab opens and is labeled *Path topology*. The shortest path between the start and end resources is calculated and is shown on this *Path topology* tab. This view shows up to a maximum of five paths. The time taken to calculate the shortest path depends on the complexity of the links between the selected resources. You can continue to work with the current topology while the path is being calculated and the associated topology rendered.

Within the path topology, the start and end resources are highlighted initially. When you hover over a link between any resources, all the paths that include those resources are highlighted.

If needed, you can further filter the displayed topology by using the Filter toolbar.

Recenter View

When selected, the displayed topology is updated from the specified seed resource to include any changes.

Viewing resources on a map

You can view a resource on a map, which gives you access to geospatial analysis functionality.

Before you begin

To geo-locate resources and view them on a map, your administrator must have previously defined a location tile server, and either created a file enrichment rule, or loaded resources with geolocation data another way, for example by using the File or REST observers.

About this task

You access a map from the *Resource* tab on the **Resource management** page, or the **More details** page accessed from the context meny in a topology

Restriction: A resource map is available only for individual resources, not for resource groups or services (applications).

Restriction: For composite resources, geolocation data should only be added to a single resource in the composite.

To view a resource on a map

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**. Open the *Resource* tab.
3. Find the resource you wish to view on a map, and click the **Map** link in the table (on the far right).

   Tip: You can also access a resource map from the *More details* page by clicking the map link next to the resource name. You can access a reduced map with limited functionality from the *More details* 'Location' tab, which is also accessed when you click the location pin in a map.

4. The Resource map gives you access to the following functionality:

   - Hover on the location pin to view a tooltip with the resource name, its icon and type.
   - Click the pin to open the More details page for the resource.
   - Use the mouse wheel (or **Zoom in** or **Zoom out** buttons) to zoom in or out. Alternatively, double-click on the map to zoom in.
   - Zoom out to view the selected resource as part of a cluster of nearby resources. The cluster ring matches the highest severity (status) of the grouped resources.
   - A cluster displays the number of resources it contains. When you hover your cursor over a cluster, the dispersal of resources is indicated as a shape.

   Tip: The granularity of a cluster, that it, the number of resources included in a cluster and at what pixel size, are customizable settings.

   - Click a cluster to zoom into the area of the clustered resources. Continue to drill down to the location pins of individual resources.

- Use the Zoom fit button to adjust the size of the view to show all the visible locations based on the current viewing criteria.

Viewing topology history

You can view the history of a topology to compare and contrast the current topology with historical versions to see how the topology has changed.

Before you begin

You must have an existing topology. For more information about rendering a topology, see Rendering (visualizing) a topology.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**.

   On the *Resource management* dashboard, applications, resource groups and resources are presented on separate tabs. In addition, the dashboard is divided into 'All applications results' and 'Favorites' sections. You can display favorite applications and resource groups either as cards or in a table. Resources are always displayed in a table, and cannot be designated as favorites.
3. Search for the resource or resource group to find the topology that you want to view.

   As you type in a search term, such as a name, tag, entity type or unique Id, a drop-down list is displayed with suggested search terms that exist in the topology service. If you select one of the suggested results, the Search Results page is displayed listing possible resource results. The Results are listed under separate **Resources** and **Resource groups** tabs.

   - If the resource that you are searching for is unique, the resource can be included in the drop-down list of suggestions. If you are confident that the listed suggestion is the resource that you are searching for, you can click the **View topology for the closest matching resource** icon in the drop-down list to render the topology for the closest matching resource to that suggestion.
   - **Optional:** If the resource that you are looking for might be deleted, you can select **Include deleted resources** (unless an administrator disables this feature). When this toggle is enabled, the search results include any deleted resources. Deleted templates or resource groups are **not** included in any search results. Deleted resources are marked as deleted and include the date and time that they were deleted. If you click **View topology**, the resource topology is displayed with the history timeline open. The timeline is set to 1 ms before the resource was deleted. When you close the timeline, the topology is cleared. When you search for resources with a previous point in time set in the history timeline, deleted resources are **not** returned and the toggle to include them is not available.
   - For each result, the name, type and other properties are displayed. You can expand a search result entry for more details, such as its status and any associated severity levels, or when the resource was previously updated. If a status other than clear exists for a search result, the maximum severity is displayed.
   - **Tip:** You can perform more complex searches, such as searching with multiple search terms. For more information about advanced query syntax, see the topology query reference topic.
4. Click **View topology**.

Note When viewing a topology, you can click **View Legend** to open a Help panel that describes the topology's visual elements.

You can display and refine the topology history in a number of ways:

- Delta mode

  You can enable and disable delta by using the **Delta** toggle in the topology navigation toolbar.

  When Delta mode is **On**, differences in topology are displayed as purple plus or minus symbols next to the affected resource.

  When Delta mode is **On** with History mode **On**, you can compare two time points to view differences in topology. Historical change indicators (blue dots) are displayed next to each affected resource.

  Note: For efficiency reasons, historical change indicators are only displayed for topologies with fifty or fewer resources. You can decrease, but not increase, this default by changing the Historical Change Threshold. For more information, see Defining advanced topology settings.

  - Lock time pin

    On the timeline graph, click the **Lock** icon on a time pin's head to lock a time point in place as a reference point for the comparison. You can then use the second time slider to view topology changes in comparison to the locked point in time.
  - Compare resource properties

    Click **Resource details** on a resource's context menu to compare the resource's data at the two selected time points. You can view and compare the property names and values in table format, or raw JSON format.
- History timeline

  You open the Topology History toolbar using the **Timeline** toggle in the Topology Visualization toolbar (on the left).

  Note: Resources with a large amount of history may take longer to display.

  You use the time pins to control the topology shown. When you move the pins, the topology updates to show the current topology representation.

While in delta mode you can move both pins to show a comparison between the earliest pin and the latest. The timeline shows the historic changes for a single selected resource, which is indicated in the timeline title. You can lock one of the time pins in place to be a reference point.

When you first display the history timeline, coach marks (or tooltips) are displayed, which contain helpful information about the timeline functionality. You can scroll through these, or switch them off (or on again) as required.

To view the timeline for a different resource, you click it, and the heading above the timeline changes to display the name of the selected resource. If you click the heading, the topology centers (and zooms into) the selected resource.

The history timeline is displayed above a secondary time bar, which displays a larger time segment and indicates how much of it is depicted in the main timeline. You can use the jump buttons to move back and forth along the timeline, or jump to the current time.

You can use the time picker, which opens a calendar and clock, to move to a specific second in time.

To view changes made during a specific time period, use the two time sliders to set the time period. You can zoom in and out to increase or decrease the granularity using the **+** and **–** buttons on the right, or by double-clicking within a timeframe. The most granular level you can display is an interval of 1 second. The granularity is depicted with time indicators and parallel bars, which form **buckets** that contain the recorded resource change event details.

The timeline displays changes to a resource's state, properties, and its relationships with other resources. These resource changes are displayed through color-coded bars and dash lines, and are elaborated on in a tooltip displayed when you hover over the change.

You can exclude one or more of these elements from displaying:

- **Relationships**

  This toggle displays or hides the number of relationships with neighboring resources, and whether the relationships were changed. Examples of this include the removal or addition of a relationship, or the modification of an existing relationship.

- **State**

  This toggle displays or hides the number of resource state changes that a resource has undergone.

- **Property**

  This toggle displays or hides the number of times that resource properties were changed. Each time that at least one property was changed is displayed as one property change event regardless of whether one or more properties were changed at the time.

Viewing topologies with direct-launch URLs

You can view specific topologies using a direct-launch URL by setting the topology navigation settings in the URL parameters to directly load a specific topology configuration. This configuration gives you quick access to specific views.

You can obtain a direct-launch URL string from a topology in the IBM Cloud Pak for AIOps console, or create the URL manually.

Before you begin

To obtain a direct-launch URL from the Cloud Pak for AIOps console, complete the process of visualizing a topology. To share a direct-launch URL with other users, those users must have the appropriate role permissions.

About this task

A typical URL starts with the host URL (**cluster_address**) and the tenant ID:

```
https://<cluster_address>/aiops/<TENANT ID>/topology-viewer
```

The **cluster_address** is the Cloud Pak for AIOps console route. For more information, see Accessing your cluster by using the console.

Tips

- For filtering a topology, you can specify 'type', and include 'Advanced' resource or relationship filters in your URL (not available for 'Simple' filters).
- For setting conditions for filters, any filter conditions you set on the Resource management page are maintained in the URL, and can therefore be shared.

Example topology URL

```
https://cp-console-cp4aiops.apps.mycluster.mydomain.com/aiops/cfd95b7e-3bc7-4006-a4a8-a73a79c71255/topology-viewer?resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=1&layoutType=5&refreshTime=30000
```

This example URL includes the following parts:

- **resourceId** - Identifies the seed resource.
- **hops** - The number of hops from the seed to be included in the topology. In this example, the **hops** number is set at '1'.
- **layoutType** - Specifies the type of display. In this example, the layout type is **5**.
- **refreshTime** - Specifies how often the view is refreshed. In this example, the refresh time is set at **30000** ms (30 seconds).
- The hop type is 'element-to-element', which does not have to be specified as it is the default.

For more information about the layout type, see layout information in the topic Viewing a group or resource topology.

Tips: By specifying 'type', you can include **Advanced** resource or relationship filters in your URL, which are not available for **Simple** filters. For more information, see the filtering information in the topic Viewing a group or resource topology.

Procedure

Obtain a topology URL from the Cloud Pak for AIOps console

Once you render a topology in the Topology viewer, use the **Sharing options** drop-down to obtain a direct-launch URL string.

Define a URL manually

You can define a URL by editing the parameters, as shown in the following examples:

- Empty topology URL

  For a URL to open the Topology viewer with no configurations, use this type of URL:

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer
  ```

- Specify seed resource by ID in the URL

  To open the page and create a visualization from a specific seed resource by ID, use this format:

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceId=<resource_id>
  ```

- Specify seed resource by name in the URL

  To open the page and create a visualization from a specific seed resource by name, use this format.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>
  ```

- Specify seed resource and hops URL

  To open the page and create a visualization from a specific seed resource with a specific number of hops and hop type, use this format.

  The hopType parameter is only required if you use the host-to-host (that is, `host`) hop type (element-to-element is the default).

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>&hops=<hops>&hopType=<host>
  ```

- Specify focus areas for large topologies in the URL

  When you load it, this topology view zooms in on the area specified, preventing very large topologies from being displayed with potentially too much information on view.

  The focusId parameter is required for the topology view to zoom into the area of a particular resource.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceId=<resource_id>&hops=<hop number>&focusId=
  <resource_id>
  ```

- Specify more resources to be highlighted in the URL

  Specify an array of resource IDs to highlight more resources in the topology view. These selections override any already selected seed resource.

  The selectedIds parameter is required for the topology view to highlight non-seed resources when the topology is loaded.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceId=<resource_id>&hops=<number>&selectedIds=["
  <resource_id>" "" …]
  ```

- Specify seed resource, hops and get neighbor expansions URL

  To open the page and create a visualization from a specific seed resource with a specific number of hops, as well as resource neighbors, use this format.

  This example sets the hop type as element-to-element, and it uses the parameter neighbourRequests, which expects an array of space separated ID strings.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>&hops=<hops>&neighbourRequests=["
  <resource_id>" "<resource id>" …]
  ```

- Specify search parameter (partial resource name) in the URL

  To open the page and create a visualization from a specific resource search value, use this format.

  - If a single resource is found that matches the search parameter, the resource is used as a seed and the topology is drawn.
  - If more than one resource is found that matches the parameter, the Search Results page is displayed listing possible results, including their names, types and other resource properties.
  - If no matches are found, an error message is displayed.

  This example searches for a resource by name, and will return results that match the name either fully or partially.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?search=<name>
  ```

- Load topology using unique ID to set seed

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceUniqueId=<unique id>
  ```

- Load topology using advanced resource filters

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceFilter=["Type1" "Type2"]
  ```

- Load topology using advanced relationship filters

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?relationFilter=["Type1" "Type2"]
  ```

- Show topology at a specific point in time.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>..<any other configuration>..&time=<unixTimeMilliseconds>
  ```

- Show topology in delta history mode, with 'time' as a reference point and deltaTime as the point to delta against

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>..<any other configuration>..&time=<unixTimeMilliseconds>&deltaTime=<unixTimeMilliseconds>
  ```

- Load topology with side toolbar hidden

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?hideToolbar=<true>
  ```

- Load topology with top search bar hidden

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?hideSearch=<true>
  ```

- Load topology using simple filters in the URL

  You can use simple filters to include specific resource or relationship types, or severities.

  simpleResourceFilter

  ```
  https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..<any other configuration>..&simpleResourceFilter=type1,type2
  ```

  simpleRelationshipFilter

  ```
  https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..<any other configuration>..&simpleRelationshipFilter=type1,type2
  ```

  simpleSeverityFilter

  ```
  https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..<any other configuration>..&simpleSeverityFilter=clear,critical
  ```

  You can extend the use of these simple filters by activating 'Include mode' for a group using the following parameters:

  - includeResourceType
  - includeRelationshipType
  - includeSeverityType

  Examples:

  ```
  https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..<any other configuration>..&simpleResourceFilter=type1,type2&includeResourceType=true
  ```

  ```
  https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..<any other configuration>..&simpleRelationshipFilter=type1,type2&includeRelationshipType=true
  ```

  ```
  https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..<any other configuration>..&simpleSeverityFilter=clear,critical&includeSeverityType=true
  ```

What to do next

With direct-launch URLs for topology visualizations created, you can save the URLs for quick access to specific views, or share them with other users to provide them with specific, instant information.

Viewing static topologies with API

You can use the API to render a static topology view through a cURL command, which you then download as an image. You use the resourceId parameter to seed the topology, and then define a number of other parameters to generate a static server-side topology image.

Deprecation notice: The ability to use the UI-API to render a static topology view through cURL command is no longer being used and can be removed in a future release.

Before you begin

To define a cURL string that renders a topology through the API, you must know the URL for your core host. You must also know the resource ID for your seed resource, as well as the required authentication credentials.

About this task

The API is available on the following endpoints:

- `https://<host>/1.0/ui-api/render/svg`
- `https://<host>/1.0/ui-api/render/png`

After you construct your cURL command in the following steps, you can use the command to download a static topology view as either a PNG or SVG image.

In addition to the **resourceId** parameter, you can use a **Basic** authentication header when calling the API. With this header, you can call the API without needing to log in to the IBM Cloud Pak for AIOps console first.

Tips:

- You can use the width and height URL parameters (exclusively for use with the UI-API) to set the size of the image produced. The default size is **1000 x 700**.
- You can include advanced resource or relationship filters in your URL by specifying 'type'. For more information about topology filtering, see Viewing a topology.
- You can change the label length for your resource types. For more information, see Editing resource type styles.

Example URLs

- PNG example

```
curl -X GET "https://example.host.com/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

- SVG example

```
curl -X GET "https://example.host.com/1.0/ui-api/render/svg?resourceId=sl1RQ5NCR_ykgciQCTkFGA" -H "accept: image/svg+xml" -H "authorization: Basic YXNtOmFzbQ=="
```

Where:

- The IBM Cloud Pak® for AIOps core host is **asm.example.host.com**
- The resourceId that identifies the seed resource is **sl1RQ5NCR_ykgciQCTkFGA**
- The output format is specified by either **png** or **svg**
- The authorization method is specified as **Basic**
- The encrypted user credentials (username and password) are **YXNtOmFzbQ==**

Procedure

1. Obtain your resource ID, authentication credentials, and host URL.
2. Define a cURL call by editing the parameters, as shown in the following examples.

Example commands

Specify a seed resource by ID

The following cURL call example renders a simple IBM Cloud Pak for AIOps topology view with a host URL of **example.host.com** and a port number of **3080**. The view is based on the **resourceId** parameter with the value of **sl1RQ5NCR_ykgciQCTkFGA**. It returns a **png** image, and performs **Basic** access authentication using the encrypted username and password credentials (**YXNtOmFzbQ==**).

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify the hop numbers and types

The image that is returned by this cURL call has the same seed resource as before, but now with the amount of **hops** set at **3**, and a **hopType** of **host**. The **hopType** parameter is only required if you want to use the host-to-host hop type (element-to-element is the default).

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&hopType=host" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify focus areas for large topologies

The image that is returned by this cURL call has the same seed resource as before (now with a hop count of three), but now zooms in on the area that is specified by the **focusId** parameter, preventing large topologies from being displayed with potentially too much information on view.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&focusId=<resource_id>" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify more resources to be highlighted

The image returned that is by this cURL call has the same seed resource as before with a hop count of three, but now uses the **selectedIds** parameter to specify other non-seed resources through their IDs to highlight when the image is rendered.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&selectedIds=<resource_id> <resource_id>" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify 'get neighbor' expansion

The image that is returned by this cURL now includes a **neighbourRequests** parameter that gathers a number of other neighboring resources for display, each identified by its resource ID. The **neighbourRequests** parameter expects an array of space-separated ID strings.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&neighbourRequests=<resource_id> <resource_id>" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify advanced resource filters

The image returned now includes a **resourceFilter** parameter that filters out all resources of a specified type. The **resourceFilter** parameter expects an array of space-separated, previously defined resource types.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&resourceFilter=
<Type1> <Type2>" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify advanced relationship filters

The image that is returned includes the relation filter parameter (**relationFilter**) to filter out all relationships of a specified type. The **relationFilter** parameter expects an array of space-separated, previously defined resource types.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&relationFilter=
<Type1> <Type2>" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify time point

The image that is returned reflects the state of the topology at a specific point in history (in Unix time). The **time** parameter expects the time entered in Unix time.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&time=
<unixTimeMilliseconds>" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify delta history mode

The image returned shows the topology in delta history mode, with the specified **time** as a reference point and the **deltaTime** parameter as the point to delta against.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&time=
<unixTimeMilliseconds>&**deltaTime**=<unixTimeMilliseconds>" -H "accept: image/png" -H "authorization: Basic
YXNtOmFzbQ=="
```

What to do next

With the topology images created, you can download and save the images for quick access to specific views, or share the images with other users to provide them with specific, instant information.

Topology query syntax

The Inventory service is used by the UI for the querying features that are provided through the search bar and advanced filtering functionality.

While previous searching technology indexed only a subset of the information of a resource or group, the Inventory service can query all of the information.

Query syntax

The main format of a supported query string consists of the following types:

- Fuzzy or full-text search: A text to search its occurrence (or close similarity) over a set of well-known properties.
- Parameterized query: Tries to find items that contain properties with the expected value. For example, matching a certain property=value content. All of the properties of a resource or group can be queried, including custom properties.
- Compound query: A combination of the other two to provide a more complex query where we try to find items matching a full-text search that we further restrict to those also matching properties with some expected value.

Fuzzy or full-text search

A fuzzy query conducts a full-text search to find fragments of information that contain a certain string within the following subset of an item's information:

- uuid
- uniqueId
- name
- keyIndexName
- description
- businessCriticality
- types (that is, entityTypes)
- tokens (all types)

Supported syntax

Provide a string to be found in any part of the queried properties, such as **mystring**, where typically this is a substring of an expected value, although it can also be the full string.

Note: The string can contain spaces.

You can use the wildcard (*) character to specify how a substring should be matched:

- ***string** - expects a suffix matching
- **string*** - expects a prefix matching
- ***string*** - expects a wrapped matching

Note: The wildcard character will only be interpreted when used at the beginning or end of the string.

You can use quotation marks to specify an exact match, as in `"my expected match"`. This syntax functions like the `mystring` matching, but in combination with the wildcard it allows you to define the prefix/suffix matching criteria for a string with spaces, for example:

- `"expected match"*` - looks for the specified string (including the spaces) as the matching prefix

Parameterized query

A parameterized query lets you find items that contain a property matching an expected value. You can query or match any property of a resource or group including any custom property, which means that this query is not limited to a set of well-known properties, unlike the fuzzy query type (or the former Search type).

Supported syntax

The syntax is based on the one defined for the fuzzy or full-text search.

You can specify the exact value, such as `property=value`

You can specify the expected value by using wildcards. For example:

- `property=*value`
- `property=value*`
- `property=*value*`

You can use quotation marks to wrap the expected value (which can have spaces), such as. For example:

- `property="complex value"` (which can also use wildcards)

Operators

This type of query also supports different operators that modulate how the user expects the matching of the expected value, or substring of a matching value.

- **Comparison operators**: `=`, `>`, `>=`, `<`, `<=`

  The comparison operators other than `=` are targeted at numeric properties; so if the property specified has a string as the value, then the comparison is likely to be done on the numeric counterpart of the characters, thereby potentially rendering some unexpected results.

  Note: For the date values either an epoch value or timestamp value is valid.

- **Collection operators**: `:`, `!:`

  Comparison and collection operators accept a comma-separated list of values.

Complex queries

You can specify several property-matching terms in the same query linking them with the `AND / OR` boolean operators to define a complex query criteria. For example:

- `property1=value1 AND property2>value2 OR property3=value3`

Also, each term can in turn be logically negated by using the `NOT` operator, as in the following example:

- `property1=value1 AND NOT property2>value2` where the expected result contains items for which `property2` value is `<=value2` (per the negated term).

Compound query

A compound query combines a fuzzy or full-text query and parameterized query to let you specify a full-text search term and also properties to match as part of the same criteria, such as `string AND property=value`.

The syntax that is described for these two types applies to a compound query, with the following restrictions. If not followed, the query is rejected as being of an invalid format.

- The fuzzy query segment must be specified first.
- The fuzzy query must be connected to the parameterized part by using `AND`.

Examples

Fuzzy query example

Fuzzy matching of items with a couple of words:

- `my server`

Fuzzy query example 2

Fuzzy matching of items with a certain prefix, for example, all FQDN servers for which the user knows the first part of the string:

- `host-*`

Parameterized query example: Discovery date

Finding servers that are discovered after a certain date:

- `entityType=server AND createTime>="2023-02-16T13:30:00.000Z"`

Parameterized query example: Tags

Finding servers tagged with certain values:

- `entityType=server AND tags:production,backend`

Parameterized query example: Busines criticality

Finding servers who have been assigned a business criticality of 'gold':

- `entityTypes=server AND businessCriticality=gold`

  Tip: Defining business criticality is described here.

Parameterized query example: Status

Finding servers with a status of 'critical':

- `entityTypes=server AND alert_severity=critical`

Parameterized query example: Group membership

Finding every member of a specified resource group:

- `memberOfGroup=ID`

You can find the resource group ID listed on the 'Properties' tab of the resource group *More details* page.

Compound query example

Finding images with the string test that also match a custom property for a certain release:

- `test AND entityType=image AND release=1`

Template types

Choose from three different template types to group your resources.

- **Dynamic template**

  This template automatically finds and updates multiple resource groups based on your criteria and how your applications and infrastructure change.

  A dynamic template builds multiple resource groups that are similar to the specified seed resource, and relationship or resource type criteria. Resource group creation and membership is dynamically maintained based on the available topology data and how it changes.

  For example, you can use a dynamic template to create dynamic resource groups, one for each of your virtual machines and the Kubernetes services they are running. Resource groups are then added, removed, and updated as you add or remove virtual machines or applications running on them.

- **Tag-based template**

  This template creates and updates a single resource group containing the resources that have a specified set of tags.

  Resource group membership is dynamically maintained to reflect changes in the topology for resources that gain or lose the specified tags.

  For example, you can use a tag-based template to create a dynamic resource group tagged with both 'red' and 'green'. Then, when you add new resources with both of those tags, the resource group will be updated to include them.

- **Token template**

  A token template applies rules to resources as they are received via observer jobs.

  Token templates are based on rules which use resource parameters as tokens, whose values then become group names for groups which will contain the resources to which each rule is applied. A distinct group is created for each token value.

  To form group names from multiple resource property values, you can construct rule tokens using variable substitution (as is possible with all rules).

- **Exact template**

  An exact template builds a single resource group that are centered on a specified seed resource, your criteria and how your applications and infrastructure change. The membership of the resulting resource group includes the seed resource and those matching the specified relationship or resource type criteria. Resource group membership is dynamically maintained for the resulting resource group provided the specified seed resource type remains unchanged.

  For example, you can use an exact template to create a resource group only for a specific Jenkins build pipeline and any new builds or physical server and its connected network switch.

  As a further refinement, you can adjust your layout by manually positioning the resources in the topology, and this layout is then replicated when you generate a resource group from any exact template that you build from this topology.

Viewing resource group templates

You can access **Resource group templates** from the *Resource management* page or while viewing a topology. From the **Resource group templates** page you can edit, delete, or create new templates.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**.
3. Click the **Resource group templates** icon on the upper right corner of the page.
4. Any templates that exist in your system are listed in the Resource group templates table.

   From the **Resource group templates** page, you can delete or edit existing templates, or click **Create a new template**.

   For new templates, select from the following available template types:

   - **Dynamic template**

     This template automatically finds and updates multiple resource groups based on your criteria and how your services and infrastructure change.

   - **Tag-based template**

     This template creates and updates a single resource group containing the resources that have a specified set of tags.

   - **Token template**

     This template applies rules to resources as they are received via observer jobs.

   - **Exact template**

     This template creates and updates a single resource group based on the specified seed resource, your criteria and how your services and infrastructure change.

5. Click **Start** to begin building your selected template type.

   For more information, see Building resource group templates.

Templates table columns

Column name	Description
Name	The template name and the icon associated with the template when it was defined.
Type	The template type; Dynamic, Tag based, or Exact.
Resource groups	The number of resource groups that have been created from the template (Dynamic templates can be more than one).
Description	A description of the template.
User	The user ID of the user who last modified the template.
Date modified	The date that the template was last modified.

Building resource group templates

You can use the template builder to create topology templates to generate resource groups, which search your topology database for instances that match its conditions. These resource groups are then listed on the Resource management page.

Before you begin

You can access the Template builder from the **Resource group templates** page. You can also right-click a resource in a Topology viewer diagram and select **Build a template** from the context menu.

For more information about the type of template to use to group resources, see Template types.

About this task

You can create a new template, or edit or delete an existing template.

To create a new template, you construct a topology view centered around a seed resource. This lets you search your database for topologies with resources and relationships that match the conditions defined in the template.

These defined topologies are dynamically generated, maintained and updated, and can be searched for and accessed via the Resource management page. Resource groups are dynamically refreshed when accessed.

Dynamic changes to resource group: If edges or resources are removed that render an existing resource group that has already been indexed incomplete, it will remain available, and can therefore still be found and displayed via the search functionality, or via a previously created direct URL. If edges or resources are added or changed that render a resource group that was not indexed previously complete, the resource group will be indexed automatically, thereby becoming accessible via the search service.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**.
3. Click the **Resource group templates** icon on the upper right corner of the page.

   The Resource group templates page is displayed listing existing templates in a table format by name, type, number of resource groups, description, user, and date modified.

   Tip: To reduce the number of templates displayed in the table, use the Filter table text field. The items displayed are filtered as you enter the search text.

   From here, you can perform the following actions:

   - To reload the templates, click **Refresh** (top right). This can be useful if other users are customizing the templates.
   - To delete an existing template, select it in the table, then click **Delete topology template**. If you intend to recreate the template, be aware of the restrictions.
   - To edit an existing template, select it in the table, then click **Edit topology template**.

     Note: When you edit an existing template, the template definition and topology areas are already populated. Also, the Name and Type parameters cannot be edited. You edit the other parameters as for a new template, and when you do, the template definition and topology areas will change. If you require a template with a different name, you must create a new template.
4. To create a new template, click **Create a new template** (top right). You are presented with a choice of template types.
5. Select either a Dynamic, Tag based, Token, or Exact template, then click **Start**.

   For more information about the type of template to use to group resources, see Template types.
6. Search for a resource to get started. Enter a resource name, type, or tags.

   As you type, a drop-down menu is displayed with suggested search terms that exist in the topology service. If you select one of these, the search results are displayed underneath the search field, with the results listed under separate columns in the results table for **Resource name**, **Resource type**, and **Tags**.

   - For each resource, expand the row to display the properties and property values.
   - If you are creating a **Dynamic** or **Exact** template, you can select a resource to view its topology.
7. When you select a resource, a basic topology view centered around the resource as a seed is displayed in the *Topology Visualization* panel on the right.
8. If you are creating a **Tag-based template**, you can either search for the tags directly on the **Tags** tab or drill down into the resources on the **Resources** tab.
9. Enter a name for the template. The name must be unique. You can optionally enter a description.

   Restriction: The following special characters are not supported when specifying template names, custom group types, or tags:

   `. ? + * | { } [ ] ( ) " \ # @ & < > ~`

   If existing resources have names with these characters, you must change them before you can use them to define resource groups.
10. From the drop-down list, select the resource group type that is being defined. This determines what the resource groups generated by your template are going to model (for example, `cluster`, `network`, or `storage`). You can also define your own `Custom type`, which will be added to the list of available Resource group types. If you select Custom type, a new text box opens, where you type a name for your new resource group type.

- For **Tag** templates, provide a resource group name.
- For **Token** templates, no resource group name is defined, as groups are automatically named based on the rule tokens.
- For **Dynamic** and **Exact** templates, use name patterns that follow a defined topology. The seed name of the defined topology is substituted when a defined topology is generated, but you can define either a prefix, suffix, or both to create an appropriate name for each defined topology.

  Remember: Although an exact template type focuses on a single resource, it remains dynamic, which means as long as the seed resource itself remains unchanged, the defined topology can change through the removal or addition of connected resources or relationships.

11. Choose an icon to associate with the template using one of the following methods:

- From the **Icon** drop-down list:

  If you open the **Icon** drop-down list, all icons are listed by name in alphabetical order.
- From the **View all icons** button:

  If you click **View all icons**, all icons are displayed in alphabetical order. Click an icon to associate it with the template.
- From the **</> (Advanced mode)** button:

If you click **</>** , you can define a JavaScript function using the simple script editor to set a different icon based on a resource type's properties, such as tag or status. Ensure that the returned value is a string that matches the name of the icon to be displayed in the topology. Click **Example** for more guidance.

- From the Define new custom icon button:

  From here you can define a custom icon, which is automatically associated with the template when done.

12. Add a tag to be included for each defined topology. With tags, you can group the individually generated resource groups by matching them to resources.
13. Use the following information specific to the type of template you are building:

- For **Token** templates, use the right hand side of the screen to define a new rule.

  Under **Details**, enter a unique **Name** for the rule. Define the **Tokens** for the rule, then click **Add+** You can use variable substitution to combine more than one property value if required.

  Under **Conditions**, select observers to which this rule applies from the **Observer** name drop-down, or leave empty for all observers.

  Select providers to which this rule applies from the **Provider** name drop-down, or leave empty for all providers.

  Select resource types to which this rule applies from the **Resource type** drop-down, or leave empty for all resource types.

  Set the **Token** filter mode to either **Exclude** or **Include**, then define a filter (or filters) in the format of a regular expression. If you choose **Exclude**, tokens which match any one of the expressions will not be applied to resources. If you choose **Include**, only tokens which match one of the expressions will be applied to the applicable resources.

  Optionally, you can validate the filters using the Validate token filters button.

- For **Exact** templates, select the **Maintain displayed resource position** toggle to use the exact resource layout when generating a new resource group.

  Remember: For exact templates, you can also create a view with an exact layout in the Topology viewer and then use the context menu to create the template.

- For **Dynamic**, **Tag** and **Exact** templates, use the Search functionality to find a seed resource (or tag).

  As you insert your cursor into the Search field, the five most recent searches are listed, and can be selected. You can delete a recent search by clicking the delete button (X) to the right of the suggested search.

  If you type in a search term, suggested search results are listed, which contain results both from the topology database and from recent searches, with your entered search text highlighted.

  When you select a search result, the found resources are listed by 'Resource name', 'Type' and 'Tags'. You can expand a resource to display its properties and values, or click its name to render a basic topology view centered around the resource as seed. The topology is displayed on the right of the Template builder.

14. Depending on the type of template you have defined, you can **Preview resource groups** and refine the template further as required.

Note: Token templates only apply rules and create groups as new observations are received from their designated observers, and therefore **cannot** be previewed.

Tip: You can define a default layout for resource groups. When a resource group based on a template is selected, this default layout is used.

Note: It might take some time for large resource groups to display in the **Preview resource groups** section. Also, the previous limit of 5,000 no longer applies, but you might want to consider that to generate and maintain a large number of matches can place additional strain on your system resources.

15. Click **Save template**.

Results

Resource groups created from the templates can be viewed on the Resource management page.

Remember: Resource groups are updated automatically when the database is updated, and generated dynamically as they are displayed.

Application incidents

An incident is a collection of insights that are derived from different data sources such as logs, events, and alerts. Incidents can help you build understanding and help drive remediation.

Incidents represent the determination of an incident and are categorized by priority from 1 (high) to 5 (low).

Before you begin

Incidents are created automatically when the underlying data integrations, AI models, and processes are set up and running. To display an incident against an application, the application must be associated with the incident.

Note: If an issue or an alert occurs and is part of a resource or group of resources but occurs prior to an application being created, then the related incident will not appear in the application details view. It will however appear in the application hub incident count.

1. Configure a ChatOps integration. For more information, see ChatOps insight management
2. Define or edit an application. For more information, see Defining and managing applications.

The application needs to include the resource groups that are associated with the incident incidents.

3. Configure an integration to the data source that provides the incident data, such as for obtaining event or log data. For more information, see Configuring Integrations.

4. Create and train an AI model to process the data.

   For example:

   Log anomaly detection: You can train a log anomaly detection model to generate log anomalies, which can then be raised through an incident.

   Event grouping: You can group multiple events into an incident.

   Similar tickets: You can search for similar tickets in the context of an incident.

   You might need to enable the integration data flow to collect a sufficient quantity and quality of data from your data source for model training.

   For more information, see Managing AI models.

5. After the model training is complete, ensure that the integration data flow is enabled and set to collect live data for event, incident, or anomaly detection.

When you have the ChatOps integration set up, the data flow enabled and AI model training completed and working, incidents are created automatically when incidents occur.

Where incidents are associated with an application, the incidents are shown as summary cards on the details page for that application.

View incidents for an application

When you have integrations set up to begin creating incidents and an incident that is associated with a application, the incident can be viewed against that specific application:

1. Log in to the IBM Cloud Pak for AIOps console.

2. From the main navigation, click **Resource management**.

3. From the *Resource management* page, select the **Applications** tab and find the application which is associated with the incidents that you want to view. Complete either of the following steps:

   - If your application is set as a favorite, click **View details** on the summary card for the application in the *Favorites* section of the page.
   - For any application, find the application in the table of all applications by searching for it or browsing the table.

4. You can view all incidents for that application within the collapsible **Incidents** section on the right side of the page.

   If multiple incidents are associated with the application, you can use the **Search** field to find your incident, or filter the list of incidents by using the **Last updated**, **Priority**, or **Number of alerts** drop-down. You can also click the **Filter** icon to open a filter dialog, which you can use to filter the incidents at a more granular level by priority, status or time range.

For each incident listed in the **Incidents** sidebar, the displayed summary card includes the following information:

- Incident title that is generated based on the events
- Timestamp for when the incident was last **updated**
- Short description of the incident
- Assigned priority
- Status
- Number of alerts generated

Note: You can expand an incident to view more details by clicking anywhere on an incident card. The additional details for the incident include the following information:

- The application name and the link to the application within the *Resource management* page
- The incident Slack or Microsoft teams link, if defined
- Incident ID
- Incident owner
- User group
- Date and timestamp for when the incident was **created**
- Additional **Alert** details
- A **View incident** link

The additional Alert details include **Show history** and **Center to** buttons, which you can click to zoom into the affected resources in the Topology viewer to display the alerts in their historical context.

You can click the **View incident** link to view the same details for the incident.

If you are using a Slack or Teams integration, you can also click the chat link on this dialog to open the connected ChatOps service and work with the incident and related incidents in your ChatOps interface. Threads might be associated with the incident, which can include links to the individual alerts that comprise the incident and any updates to the incident. If the probable cause information is available, the links are highlighted. You can click the probable cause links to open the topology for the associated resource in the Topology viewer.

View relevant events shows the events that are grouped within the incident. You can click the event link to view the log source for that event.

If you click to **Search Recommended actions**, the similar tickets search dialog is opened to show you similar past tickets.

For more information, see ChatOps insight management.

Defining business criticality

You can define the levels of business criticality for your applications (services), resource groups and resources, and then view a list of them on the Business criticality page. These set definitions can then be assigned to an application (or resource/resource group).

Before you begin

As an administrator user, you define the business criticality values that can then be assigned to an application (or resource/resource group).

You do so using the *Business criticality* page, accessed from the *Topology configuration* page.

When you subsequently create or edit an application (or resource/resource group), you can choose from the defined options to set the business criticality. This allows you to dynamically set different alert severity or incident priority levels, depending on the business criticality you have set.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**.
3. From the **Settings** drop-down, click **Topology configuration**, then click **Configure** on the Business criticality card. The *Business criticality* page is displayed listing any existing criticality definitions in a table format by name, description, and criticality value, in sortable columns.

   Tips: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. To reload the items displayed, which can be useful if other users are customizing them, click **Refresh** (top right).

   From here, you can perform the following actions:

   - Sort the table of defined criticality definitions by column header.
   - Delete existing definitions by selecting one or more in the table and clicking **Delete business criticality definition(s)**. **Caution:** Deleting a criticality affects any resource, group or application to which this criticality has been assigned.
   - Edit an existing definition by selecting it in the table, then clicking **Edit business criticality definition**.
   - Create a new business criticality.
   - Start with the default criticalities (Tier 1, Tier 2, Tier 3).
4. Click the **Define criticality +** button to create a new criticality definition, click **Start with presets** (then **Add**), or click the pencil icon next to an existing definition to edit it.
5. Enter or edit a name (required), and description (optional).

   Tip: Use names that reflect your business needs, and that quickly identify the level of importance, for example `Tier 1`, `Level C`, and so on.
6. Enter a criticality value, either by typing a number between 1 and 100, or by using the **+** (plus) and **-** (minus) buttons to select a numerical value.

   Important: The higher the value you specify for the business criticality, the more important it is when assigned to an application (or resource/resource group).
7. Click **Save** (or **Define**). The business criticality definitions are saved and listed on the *Business criticality* page.

Criticality values can now be assigned to applications (or resources/resource groups) from the *Resource management* page or while viewing an application.

Running data administration routines

You can run data management routines from the *Data administration routines* page. You can also view details for any routines that have been run.

Before you begin

You must be an administrator user with `waiops_manage_topology_routines` permissions.

About this task

Long-running routines that perform certain data management activities are run by the system as and when required, and you can view details for these when they have run on the Data administration routines page. You can also run some of these manually when required, for example to re-index vertices for searching, or to delete all resources for a specific provider.

You can run the following routines. After defining the details specific to the routine, you specify the number of updates to be batched up, then run the routine to implement its specific function.

- **Deduplicate** - This routine removes duplicate resources from the system.
- **Delete provider resources** - This routine permanently deletes all the resources and resource groups supplied by a specific provider.

- **Encryption key change** - This routine updates passwords in observer jobs after the encryption key has been changed.
- **Rebroadcast** - This routine rebroadcasts vertices from the topology to force a re-indexing of the search database.
- **Resource status delete** - This routine improves system performance by deleting all status for a single resource that has an excessive amount of status assigned.
- **Status clear** - This routine clears obsolete status if there have been synchronization problems with the event management system.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**.
3. From the **Settings** drop-down, click **Topology configuration**, then click **Configure** on the Data administration routines card.

   The *Data administration routines* page is displayed listing existing routines that have been run. If user-defined display settings are in place, some routines may no longer be displayed. The table footer will indicate if this is the case.

   Tips:

   - To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text.
   - To reload the items displayed, which can be useful if other users are customizing them, click **Refresh** (top right).

   From here, you can perform the following actions:

   - Sort the table of defined criticality definitions by column header, which are name, type, status (ready or completed), last run, duration, operations (number), or items visited (number).
   - View a routine's parameters and parameter values when run last (by clicking the routine name).
   - Configure which data administration routines are displayed (by setting the display options).
   - Run a routine manually.
4. To define routine display settings, click the **Routine display settings** button (top right). The *Routine display settings* page opens, where you can set the following options:

   Age limit (days) Set the number of days from which a routine will be considered 'old' instead of 'recent'. The default is 30 days.

   Display options Next, select one or more of the following checkboxes to hide the selected routines based on your definitions of 'old' and 'recent':

   - Hide old routines that finished successfully
   - Hide recent routines that finished successfully
   - Hide old routines that were aborted
   - Hide recent routines that were aborted
5. To run a data administration routine manually, click the **Run a routine** button. The *Run a new routine* window opens displaying the available routines that you can run manually.
6. Select the routine to be run, then click **Next**. A window specific to the routine selected is displayed.
7. Define the details for the specific routine to be run.

   Deduplicate This routine removes duplicate resources from the system. Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Delete provider resources This routine permanently deletes all the resources and resource groups supplied by a specific provider. Select the provider from a drop-down list of existing providers. Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Encryption key change This routine updates the observer job passwords after the encryption key was changed. Enter the old key to be changed. Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Rebroadcast This routine rebroadcasts topology data to trigger re-indexing of the search database. Specify whether historic resources should be rebroadcast (**true** or **false**). Enter the broadcast IDs to be rebroadcast. You can add more than one. Choose whether the resource status should be rebroadcast (default is 'true'). Choose whether group and composite edges for resources should to rebroadcast (default is 'true'). Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Resource status delete This routine deletes all status for a single resource. Use the Search field to find the resource to be stripped of its status values. Set the minimum age of any status to be deleted. The default is 12 hours. Set a deletion limits (maximum). The default is 1.000 (one-thousand). Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Status clear This routine clears all status for the system. Set the minimum age of any status to be deleted. The default is 12 hours. Set a deletion limits (maximum). The default is 1,000 (one-thousand). Specify the number of updates to be batched up. The default is 1,000 (one-thousand).
8. Click **Run**.

When the routine has completed, its details will be displayed on the *Data administration routines* page.

Configuring topology user preferences

Set your personal preferences for how topologies are rendered.

About this task

The settings that can personalize are grouped into the following categories:

- Updates
- Layout
- Miscellaneous

Procedure

You can access the user preferences settings while viewing a topology or from the *Resource manager* page.

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, expand **Operate** and click **Resource management**.
3. Click the **Settings** icon that is next to the search field. Click **User preferences**.
4. Update the setting for any user preference setting that you want to change.
5. Click **Save** when you complete your changes and return to the topology view.

You must reopen the page before any changes to this user preference take effect.

Updates

These settings determine how and how often topologies are automatically updated.

- **Default auto refresh rate (seconds)**

  Set to `30`.

- **Maximum number of resources and relationships to load with auto refresh enabled**

When the set resource limit is reached, auto-refresh is turned off. The maximum value is `2000`. By default, the value is set to `500`.

Tip: If you find that the default value is too high and negatively impacts your topology viewer's performance, reduce this value.

- **Auto render new resources**
  - Enable this option to display new resources at the next scheduled or manual refresh as soon as they are detected. By default, this setting is disabled.
- **Remove deleted topology resources**
  - Enable this option to remove deleted resources at the next scheduled or manual refresh. By default, this setting is disabled.

Layout

These settings determine the default presentation of topologies. You can select the layout type to use for rendering a topology by default. In addition, for some layouts you can select to change the layout orientation for how the layout is displayed.

Tip: A change to a layout type is tracked in the URL (layout orientation is not tracked). You can manually edit your URL to change the layout type display settings.

- **Default layout type**
  - (Default) Use Layout 1
- **Layout 4 orientation**
  - (Default) Top to bottom
- **Layout 6 orientation**
  - (Default) Top to bottom
- **Layout 7 orientation**
  - (Default) Top to bottom
- **Layout 8 orientation**
  - (Default) Top to bottom

Miscellaneous

Other settings, such as displaying messages, screen reader support, and troubleshooting.

- **Information message auto hide timeout (seconds)**

  The number of seconds that information messages are displayed in the IBM Cloud Pak for AIOps console. The default is `3`.

  Tip: If you are using a screen reader, it may be helpful to increase this value to ensure that you do not miss a message.

- **Screen reader support for graphical topology**
  - You can enable the display for more Help text on-screen elements, which can improve the usability of screen readers. By default, this setting is disabled.
- **Client-side debug logging (on supported browsers)**

- Enhanced client-side ability to log problem diagnosis. If enabled, more debug output is generated, which you can use for defect isolation. By default, this setting is disabled.

Tip: Use this for specific defect hunting tasks, and then disable it again. If this setting is left enabled, the Topology viewer performance can be reduced.

- **Maintain zoom level on render configuration changes**
  - (Default) Disabled

Configuring topology elements

You can customize a number of IBM Cloud Pak for AIOps console elements for the Topology viewer, such as resource styles and icons. You can also create custom topology tools which users can access through a topology's context menu.

Tip: The Topology viewer loads all configuration settings into memory when it opens. If you change your topology configurations after opening the Topology viewer, you must reopen it before these changes take effect.

For more information on resource properties, status (and states), edges and so on, see the Topology API topics.

Configuring advanced topology settings

You can define advanced topology settings. Examples include the URLs for trusted sites, the required properties for tooltips or relationship type styles, or the maximum hop numbers a user can choose in the topology viewer.

Configuring topology tools

You can create custom topology tools, which users can then access from within a topology's context menu. Use this feature to access properties of a selected item, such as a resource or relationship, and run some customized functions within the context of that item.

Configuring resource types

You assign existing or new custom icons to particular resource types using the Resource types page. Here you can also apply further resource type style edits, such as adding custom labels to a resource type, and changing its shape, size, border and background.

Configuring relationship types

You can customize the styles and labels for specified relationship types using the Relationship types page. You can also delete existing, or create new relationship type styles.

Configuring custom icons

You can add custom icons for resources that are displayed by using the Custom Icons page.

Configuring rules

Rules help streamline topologies and conserve system resources, such as by merging different observer records of the same resource into a single composite resource, or by excluding specific changes from being recorded against a resource history.

Configuring advanced topology settings

You can define advanced topology settings to control how the topology UI works for all users.

You can change settings, such as the URLs for trusted sites or geolocation tile servers, the required properties for tooltips or relationship type styles, or the maximum hop numbers a user can choose in the topology viewer.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, expand **Operate** and click **Resource management**.
3. Click the **Settings** icon that is next to the search field. Click **Topology configuration**.
4. On the *Advanced setting* card, click **Configure**.
5. From the *Advanced topology settings* page, set or change any of the settings or values to meet your needs.

   The settings are included within these categories:

   - Topology rendering
   - Topology search
   - Topology tools
   - Resource map
6. After you have completed your changes, click **Save** to save your changes and return to the *Topology configuration* page.

Topology rendering

Within the *Topology rendering* section, you can set or change the following settings:

- **Default hop type for topology viewer**

  Choose the default hop type to be applied when viewing a resource in the Topology Viewer. The default setting is 'element to element'.

- **Maximum number of hops allowed**

  Select a value in the range 2 - 30 from the drop-down list. The default maximum hop count is `4``.

  Important: Ensure that you set a maximum hop count that does not overload your system data capacity when the maximum hop count is selected.

- **Historical changes threshold**

  This setting defines the maximum size of a topology based on the number of resources for which the historical changes indicators are enabled in delta mode.

  Set the maximum topology size by entering a value up to, but not exceeding **50**.

- **Include adjacent relationships on topological hops**

  Select whether to retrieve all relationships between resources within a topology by enabling or disabling the toggle. By default, this setting is enabled and all relationships are retrieved.

- **Properties required for tooltips or resource and relationship styling**

  To improve system performance only specific resource and relationship properties are loaded into the topology viewer. You can specify more properties to be retrieved by adding the properties to the table. These required properties can be uses in topology tooltips, custom status tools or to customize UI elements with JavaScript.

  To add a property, enter the property name within the text field and click **Add +**.

  After you have added a property, you can choose whether the property can be displayed in a tooltip. If you want to exclude the property from tooltips, clear the **Show in tooltip?** checkbox.

- **Resource types to block from topologies**

  Excluded resource types do not display in the Topology Viewer. Excluding resource types can optimize performance when complex topologies are being rendered. You can see the list of excluded types on the *Filter* toolbar of a topology visualization.

  To exclude a resource type from being displayed in the topology viewer, enter the resource type within the text field and click **Add +**.

  Note: You cannot exclude the seed resource of a topology.

  Tip: If an administrator unblocks a resource or relationship type while it is being viewed, the browser in which the topology viewer is displayed must be restarted for the unblocked types to appear.

- **Relationship types to block from topologies**

  Excluded relationship types do not display in the Topology Viewer. Excluding relationship types can optimize performance when complex topologies are being rendered. You can see the list of excluded types on the *Filter* toolbar of a topology visualization.

  To exclude a relationship type from being displayed in the topology viewer, enter the relationship type within the text field and click **Add +**.

- **Properties to hide in the resource details page**

  Excluded properties do not display on the *Resource details* page. In order to reduce visual clutter, you can hide properties that are of no interest to your operator users.

  If a property has been excluded from view by an administrator, it is removed from the tabular view, but remains visible in JSON format.

  To exclude a property from being displayed on the *Resource details* page, enter the property within the text field and click **Add +**.

Topology search

Within the *Topology search* section, you can define the following settings:

- **Add additional properties for filtering**

  By default, users can filter the resources, resource groups, or applications displayed on the *Resource management* page by using the 'entityTypes', 'name' and 'tags' properties. You can add additional properties to be available for filtering.

  Limitation: You cannot define filters by using properties with values that are sets.

  1. Type in a property name, then click **Add+**.
  2. Select the Operators to be available for the property from the Operators drop-down, then click **Add+** again.

  The added properties will be available when operator users define other filter properties on the Resource management page.

- **Include deleted resources in Search results**

  You can select to include or exclude an option in the topology viewer to include deleted resources within topology viewer search results.

  By default, the display of the option to include deleted resources is enabled. To disable the display of the option to include deleted resources, switch the toggle to **Disabled**..

Topology tools

Within the *Topology tools* section, you can add trusted websites that can be accessed through HTTP.

If you need to link to an HTTP address instead of HTTPS while writing a custom tool definition, and need to request data from a site that uses HTTP instead of HTTPS, then you can use the UI server as a proxy. The URL to the HTTP proxy takes the actual target HTTP site URL as a parameter. The proxy then sends the request itself, and returns the response back to the browser.

For security reasons, the HTTP proxy can access only the URLs defined by an administrator user as 'trusted sites'.

The **`trustedSites`** property values are a comma-separated list of the trusted sites to which the proxy server can link, and from which it can retrieve information.

Trusted sites operate with a 'starts with' condition.

To add a trusted site, complete the following steps:

1. Within the **Trusted website URL** table, enter the URL of the trusted site.

   For example, if you include the following URLs, the HTTP proxy permits requests for all target URLs that start with included trusted URL.

   - **`www.ibm.com`**
   - **`data-server.intranet.com:8080/network/statistics`**

   For instance, you can use the proxy for URLs, such as **`www.ibm.com/cloud`** or **`data-server.intranet.com:8080/network/statistics/1893713/info`**.

   You cannot use the proxy for URls such as **`www.ibm.co.uk`** or **`data-server.intranet.com:8080`**

2. Click **Add +**.

Resource maps

The geospatial mapping feature lets you plot resources on a Resource map at the required zoom level, and also includes the ability to filter resources according to their geospatial areas.

Important: To enable geospatial mapping, you must configure a tile server to provide map image tiles. Tile server providers are available with a range of licenses from free for limited use to paid for. Alternatively, you can deploy an on-site tile server. Any tile provider or mapping data organization you use requires an attribution credit on all maps displayed.

Tip: For more information and examples of configuring, using or creating a tile server, see the Switch to OpenStreetMap website.

To add a resource map site, complete the following steps:

1. Enter the URL for the map in the Map tile server URL template field.

   Example URL

   `https://tile.<example>.org/{z}/{x}/{y}.png`

   Where {z} is the zoom level, and {x} and {y} the tile coordinates.

   Tip: Variations on this basic format are possible. See the TileLayer section on the Leaflet site for more information.

2. Validate your map using the Test button (which attempts to fetch a single tile image with the three coordinate placeholders set to zero (0)).

   Note: If your server does not pass the test, the Save button remains unavailable.

3. Set server attribution.

   Server attribution is in the form of HTML and should be provided in your tile provider's documentation.

4. Set the parameters for the cluster threshold and the maximum pixel size.

   CAUTION: Accept the default settings unless advised otherwise by an IBM Support representative. Changing your clustering granularity and maximum pixel size affects the performance of your system.

   a. Set Map clustering thresholds (by number of resources).

   When a number of resources are located in close proximity, they are displayed on the Resource map as a cluster. The lower a clustering threshold is set, the coarser your clustering becomes and the more resources will be included when resources are clustered on the Resource map. This may improve system performance.

   - **Threshold 1**
     - Default is 10000.
     - Up to this number of resources, k-means clustering is used, which automatically determines the best cluster split of resources based on the distance between resources.
   - **Threshold 2**
     - Default is 30000.
     - Up to this number of resources, convex grid clustering is used, which splits the area where resources are into an X by X grid based on the map clustering pixel size, and then displays those resources within an appropriately shaped polygon.
   - **Threshold 3**

- Default is 50000.
- Up to this number of resources, the grid clustering is displayed as a standard shape determined by the cluster boundaries (rather than a bespoke polygon).

- **Threshold 4**
  - Default is 100000.
  - As for the previous threshold, except a larger grid size is used.
  - Above this number, grid clustering with a very large grid size is used.

b. Set Map clustering pixel size.

The default is 150. Increasing the maximum pixel size for clustering resources may improve performance at the cost of cluster resolution.

Once a map has been defined, it can be accessed from the Resource management and More details pages, as described here.

Configuring custom tools

You can create custom topology tools, which you and other users can then access and use from the context menu of a topology. Use this feature to access the properties of a selected item, such as a resource or relationship, and run some customized functions within the context of that item.

Before you begin

Ensure that you are proficient in JavaScript. You must use JavaScript to define custom tools.

About this task

Custom tools can be accessed through the right-click context menu in the UI. All users with permission to access topologies can access the tools, but only administrators with permission to manage topology tools can customize them.

Note: Some topology tools might fail due to browser restrictions, such as when the tool needs to load external content. Security measures for Firefox and Chrome, which prevent click-jacking and similar risks, can also prevent the loading of third-party content over HTTPS into an iframe. An iframe mechanism can be used by custom topology tools to display contextual information, and as such can be affected. As a workaround for Firefox browsers, users can select an option to load the content in a separate window.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, click **Resource management**, then access a topology.
3. In the navigation toolbar for a topology, click the **Settings** icon. From the drop-down menu, click **Topology configuration**. The *Topology configuration* page is displayed. The page includes cards for each category of topology elements and settings that can be configured or customized.
4. On the *Topology tools* card, click **Configure**. The *Topology tools* page is displayed, listing all of your defined tools in table format in sortable columns. These are the tool name, menu label, last update, the items the tool applies to, whether special conditions exist, and its description.

   From here you can create a new topology tool, or select existing tools in the table and edit or delete them.

   Tip: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. You can also click **Refresh** (upper right) to reload the items displayed, which can be useful if other users are customizing them.
5. Click **New** to create a new tool. The New topology tool page is displayed consisting of the expandable Details, Implementation, and Conditions sections.
6. In the **Details** section, enter the general details for your new tool:
   - **Name**

     Required. Enter a unique internal name for your tool. This name is used as an internal reference for identify individual tools.
   - **Menu label**

     Required. The menu label is the text that is shown in the context menu. This can be the same name as used by other tools, which is why the unique name is required.
   - **Description**

     Optional. A description to help administrative users record the tool's purpose. This text does not display in the context menu.
   - **Menu priority**

     Optional. The menu priority slider defines where in the context menu the tool is displayed. Available values are one to ten.

     For example, tools with a priority of two will be displayed higher in the menu than tools that have a priority of four.
7. In the **Implementation** section, define the JavaScript implementation for your tool.

   If needed, you can click **Example** to view example JavaScript to show you how to format your code. The code for your tool runs when the tool is accessed by an operator, such as when a user from the context menu of a topology.

To help you create tools, you have access to custom helper functions

8. In the **Conditions** section, select any conditions for using the tool.

   Select the resources, relationships, and status for which the tool applies. A user must select one of these items for the tool to display in the context menu.

   Use the **Applicable item type for tool definition** field to select between the types of topology items for which the tool is applicable:

   - **Resource**

     Select one or more resource types from the list displayed.

   - **Relationship**

     Select one or more relationship types from the list displayed.

   - **Resource and Relationship**

     Select one or more resource or relationship types from the combined list.

   - **Status**

     Select from the following possible states for which the tool will be available:

     - **Clear**
     - **Indeterminate**
     - **Information**
     - **Warning**
     - **Minor**
     - **Major**
     - **Critical**

   Depending on your selection, a number of checkboxes are displayed, which you can use to configure which resources, relationships or states are included. If you select **Resource**, **Relationship**, or **Resource and Relationship** (but **not** Status), a JavaScript editor is displayed, letting you define additional conditions for the display of the topology tools.

   The following relationship example creates a tool that shows only relationships that have source resources with the name 'kafkaService', or relationships that have target resources with 'entityType' properties that also include 'container':

   ```
   if (asmSourceProperties.name === "kafkaService" || asmTargetProperties.entityType.includes("container")) { return true; } else { return false;}
   ```

   The following relationships and resources example is for relationships and resources that have a property that is called 'kubernetes_namespace':

   ```
   if (asmProperties.kubernetes_namespace) { return true; } else { return false; }
   ```

   For each category, you can also select **All types** if you want the tool to be displayed for all types under the selected category. The tool also displays for any specific types under the selected category that is not listed.

   Remember: When you are creating status tools, the properties that you use in your JavaScript reference the properties for the status item. The properties that you use in your resource or relationship tools reference the properties for the resources or relationships.

9. Click **Save** to save your configuration and create the tool.

   Once you save your changes, you must close the Topology Viewer, and then reopen the UI to make the new tool available. Tools are also available on the context menu of a topology only when the configured conditions are met.

Custom helper functions

To help you create custom topology tools, you can use the following custom JavaScript helper functions:

asmProperties

Your tool implementation can access the properties of the relevant resource, relationship, or status through the **`asmProperties`** JavaScript object, which contains all the properties.

You can access the properties by using standard JavaScript, but you must protect against the property value not being present. For example, if you intend to use the property 'latitude', you must verify that it is present before using it. To do so, you can use the following check command:

```
asmProperties.hasOwnProperty('latitude')
```

If the property is present, the Boolean value true will be returned.

Status-related tool properties

For **Status**-related tool properties, you use JavaScript that is similar to the JavaScript that you use when creating resource or relationship tools. However, the properties that you must use in your status tool implementation, such as **`asmProperties`**, must reference the properties for the status item. For **Resource** or **Relationship** related tools, the implementation needs to instead reference the properties for the resources or relationships.

For example, if you use **`asmProperties.location`** in a status tool implementation, a corresponding **`location`** property must be in the status record. When creating status tools, the **`asmProperties`** object has a property that takes the form of an array that is called **`resources`**, which represents the resources in the topology with which this status is associated. Each item in the **`resources`** array is an object with properties that represent the properties of that resource. If a status is associated with two resources, the **`uniqueId`** property of the first of those two resources can be referenced in the JavaScript by using **`asmProperties.resources[0].uniqueId`**.

In addition, you can access the properties of the resource for which you are intending to use a status tool by using the **asmSourceProperties** object when scripting the status tool.

asmSourceProperties

You can access information about the source properties of any relationship or status that the custom tool is acting upon through the **asmSourceProperties** JavaScript object.

The following example shows the use of the helper function in a custom relationship stroke definition:

```
if (asmSourceProperties.myProp === 'high') {
    return 'blue';
} else {
    return 'black';
}
```

Note: The arrows on the topology indicate a relationship point from the source to the target.

asmTargetProperties

You can access information about the target properties of relationships that the custom tool is acting upon trough the **asmTargetProperties** JavaScript object.

asmFunctions

You can use a number of other helper functions, which are accessed from the asmFunctions object, which includes the following functions:

showConfirmationPopup(title, message, onOk)

Creates a confirmation dialog to allow the tool to confirm an action. This function takes a title and message, which is displayed on the dialog, and a function definition, which is run when the user clicks the **OK** button on the dialog.

showToasterMessage(status, message)

Opens a window toaster message with the appropriate status coloring and message.

showPopup(title, text)

Shows a dialog with a title and text body, which can be generated based on the properties of the resource or relationship.

Tip: With the **asmFunctions.showPopup** helper function, you can use markdown to create more sophisticated HTML pop-up windows. For more information about markdown syntax, consult a reputable markdown reference site.

showIframe(url)

Displays a dialog that fills most of the page, which wraps an iframe, showing the page of the URL. Use this function to embed more pages.

Restriction: You must add any websites that are referenced by the **url** parameter to a list of your trusted sites. For more information about adding sites to this list, see Defining advanced topology settings.

getResourceStatus(<resource_id>, <callback_function>, [<time_stamp>])

Use this function to request status information from a tool definition for a resource by using its **_id** parameter.

- **resource_id**

  Required. This value can be obtained from a resource through **asmProperties._id** and from a relationship by using **asmSourceProperties._id** or **asmTargetProperties._id**.

- **callback_function**

  Required. This function is called when the status data is collected from the topology service, with a single argument that contains an array of status objects.

- **time_stamp**

  Optional. A Unix millisecond timestamp to get the status from a set point in time.

The following example shows the status information of a source resource from a relationship context to the browser console log:

```
let printStatusCallback = function(statuses) {
    statuses.forEach(function(status) {
        console.log('status:', status.status,
                    'state:', status.state,
                    'severity:', status.severity,
                    'time:', new Date(status.time));
    })
}
asmFunctions.getResourceStatus(asmSourceProperties._id, printStatusCallback);
```

Tip

- The resource properties available in the UI are 'status', 'state', 'severity', 'time', 'id' and 'resource_id'.
- When the optional parameter **_include_status_severity** is set to true, the **_hasStatus** field will reflect the maximum and most severe severity of all of the status records related to a resource (such as **_hasStatus": "critical"**).

sendHttpRequest(url, options)

Use this function to send an HTTP or HTTPS request to a remote web server by using the IBM Cloud Pak® for AIOps backend server rather than the browser to avoid browser domain-blocking.

- **`url`**

  Required. The full URL of the remote site to be accessed, for example:

  ```
  https://data-svr-01.uk.com/inv?id=1892&offset=0
  ```

  Restriction: You must add any websites that are referenced by the **`url`** parameter to a list of trusted sites. For more information about adding sites to this list, see Defining advanced topology settings.

- **`options`**

  Optional parameters that you can include:

  - **`method`**

    The HTTP method to use. You can use the following methods:

    - GET (default)
    - POST
    - PUT
    - DELETE
  - **`headers`**

    An object that defines any required special request headers.

  - **`body`**

    A string that contains the body data for the request. Use POST and PUT requests only.

  - **`autoTrust`**

    A flag to indicate whether the remote web server can be automatically trusted. This flag is required when the website uses a self-signed SSL certificate with no CA, or uses a CA that is unknown to IBM Cloud Pak for AIOps. The value for the flag can be **`true`** or **`false`**. The default value is **`false`**.

  - **`onSuccess`**

    A callback function to run if the HTTP request succeeds. This function is passed the following three parameters:

    - Response text
    - HTTP status code
    - Response header
  - **`onError`**

    A callback function to run if the HTTP request fails. This function is passed the following three parameters:

    - Response text
    - HTTP status code
    - Response header

  The following sample JavaScript, shows the use of the **`options`** parameter:

  ```
  {
      method: 'GET',
      headers: {
          Content-Type: 'application/json',
          X-Locale: 'en'
      }
      body: '{ "itemName": "myData1" }',
      autoTrust: true,
      onSuccess: _onSuccessCallback,
      onError: _onErrorCallback
  }
  ```

Example tools

Use the following examples to help you create the JavaScript implementation for your own custom topology tools. This implementation needs to define the action that occurs when a user selects the tool from the context menu.

Example: Send HTTP GET request for external REST data

The following example uses the **`asmFunctions.sendHttpRequest(url, options)`** helper function to send an HTTP GET request for external REST data, which is then displayed in a dialog on the Topology viewer.

Reminder: You must add any websites that are referenced by the **`url`** parameter to a list of trusted sites. For more information about adding sites to this list, see Defining advanced topology settings.

```
// Get the unique id from the resource and use it to send a REST request
var extId = asmProperties.uniqueId;
if (!extId || extId === '') {
    asmFunctions.showToasterMessage('warning', 'The resource does not have a unique identifier.');
} else {
    var options = {
```

```
        method: 'GET',
        onSuccess: _onSuccess,
        onError: _onError
    };
    asmFunctions.sendHttpRequest('https://api.data.uk/resource/' + extId, options);
}

function _onSuccess(response, status, headers) {
    // Start to build text for the info dialog
    var output = 'Here are details for ' + asmProperties.name + ':\n\n';

    var data;
    try {
        data = JSON.parse(response);
    } catch (e) {
        data = {};
    }
    var props = data.additionalProperties;
    if (!props || props .length === 0) {
        asmFunctions.showToasterMessage('information', 'No information exists for this resource.');
    } else {
        for (var idx in props) {
            var prop = props[idx].key;
            var val = props[idx].value;
            if (!prop.endsWith('Id')) {
                output += props[idx].key + ' = ' + props[idx].value + '\n';
            }
        }

        // Show the output in an ASM popup dialog
        asmFunctions.showPopup('External Details', output);
    }
}

function _onError(response, status, headers) {
    if (!response || response === '') {
        asmFunctions.showToasterMessage('critical', 'HTTP request failed with status code ' + status);
    } else {
        asmFunctions.showToasterMessage('critical', 'HTTP request failed: ' + response);
    }
}
```

Example: Splunk integration

When you are configuring a custom tool, you can create a custom tool integration with Splunk. As an example, you can create an integration that allows you to search a Splunk repository through the context menu of a selected resource in the Topology viewer. For more information, see Example: Splunk integration.

Splunk integration example

With IBM Cloud Pak® for AIOps, you can add custom tools to the topology viewer, such as a custom tool integration with Splunk. The following example creates a custom tool for searching a Splunk repository. This tool can be accessed from the right-click context menu on the topology viewer when you are viewing a resource topology.

Note: You must install Splunk Enterprise and Splunk Forwarder before you can define a custom tool with the following example. The following example was tested with Splunk 8.1.0. For more information, see Splunk.

About this example

The following example code creates a custom tool that can be accessed from the right-click context menu. With this tool, you can use the properties of the resource that is selected in a topology when you click to open the tool to search a specific Splunk repository. This specific tool searches Splunk for logs that match the name of an IBM Cloud Pak for AIOps resource. This search can help you determine whether Splunk has registered any log information about the selected resource.

If you want to customize the following example code, you can change the values for the following properties:

- **`splunkUrl`**

  The base URL of your Splunk instance

- **`splunkRepository`**

  The Splunk repository you want to search

- **`searchValue`**

  The dynamically created **`searchTerm`** used in Splunk

In addition, you can use any IBM Cloud Pak for AIOps properties that are associated with the resource to create your tool. You can also define the tool response for when a search is unsuccessful.

Example

```
// Set the Splunk instance url
var splunkUrl = 'host:port';
```

```
// This will store the value that you want to search in Splunk
var searchValue = '';
// This will search results for defined period
//    0 = all time
// -15m = last 15 mins
// -24h = last 24 hours
//  -7d = last 7 days
var searchEarliest = "0";
// This flag is set if you find a valid search value
var foundValidSearchValue = false;
if (asmProperties && asmProperties.name) {
    searchValue = "\"" + asmProperties.name + "\"";
    foundValidSearchValue = true;
} else {
    var status = 'critical';
    var message = 'Unable to find name property for Splunk search request';
    asmFunctions.showToasterMessage(status, message);
}
// DO NOT MODIFY THE LINES BELOW
if (foundValidSearchValue) {
        var searchLatest = '';
        if(searchEarliest!="0") {
                searchLatest="now"
        }
    var encodedSearchValue = encodeURI(searchValue);
    var searchQuery = '/en-US/app/search/search?
q=search%20index%3D*%20'+encodedSearchValue+"&earliest="+searchEarliest+"&latest="+searchLatest;
        window.open(splunkUrl+searchQuery);
}
```

Configuring resource types

You can assign existing or new custom icons to particular resource types by using the Topology viewer *Resource types* page in the UI console. You can also apply further resource type style edits, such as adding custom labels to a resource type, and changing the resource type icon shape, size, border, and background.

You can define a custom icon as part of editing a resource type to have the icon immediately assigned to the selected resource type. However, if you want to create a number of icons without assigning them to specific resource types, or you simply want to edit or delete icons, use the *Custom icons* page. For more information, see Defining custom icons.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.
3. From the topology navigation toolbar, expand **Settings**, and click **Topology configuration**.
4. On the *Resource types* card, click **Configure**.

   The Resource Types page is displayed, which lists all existing resource types in table format in sortable columns. The table also displays the icons for the resource types and their names, categories (if defined), whether they are system icons or custom icons, and whether they have custom labels, styles or shapes. From here, you edit existing resource types, and then associate existing icons with them, and apply resource styling.

   Tips: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. You can click **Refresh** (top right) to reload the items displayed, which can be useful if other users are customizing them. You can select more than one resource type and reset them to their default values, or perform a batch edit.
5. To edit resource types, select them in the table and click **Edit resource type** (or **Batch edit resource types**). The *Configure relationship type* page is displayed.

   The *Edit Resource Type* page is displayed. You can toggle between the **Identification** and the **Styling** tabs. Use the options on these tabs to define the resource type.

   - Use the **Identification** tab to define the selected resource type's icon, label, and shape.
   - Use the **Styling** tab to define the selected resource type's border color, border pattern, background color, and resource display size.

   Restriction: You cannot change the name of an existing resource type.

   Tip: To quickly revert resource stype styles to their default, select them in the table, then click **Reset to defaults**.
6. Click **Save** when you are finished with your edits to save your changes and return to the *Resource types* page. The table displays your changes for the resource type and the associated icon.

Identification

On the **Identification** tab, define the selected resource type icon, label, and shape:

- Choose an icon to associate with the resource type by using one of the following methods:
  - **Icon** drop-down list

If you open the **Icon** drop-down list, all icons are listed by name in alphabetical order.

- **View all icons**

  If you click **View all icons**, all icons are displayed in alphabetical order. Click an icon to associate it with the resource type.

- **Quick assign**

  If an icon exists with the same name as the resource type, click the **Quick assign** button to select it without having to sort through all existing icons. This function is useful if you have added a custom icon, and are now assigning it to a resource type with the same name.

- **</>** (Advanced mode)

  Define a JavaScript function using the simple script editor to set a different icon based on a resource type's properties, such as tag or status. Ensure that the returned value is a string that matches the name of the icon to be displayed in the topology.

  Click *Example* for more guidance.

- **Define new custom icon**

  From here you can define a custom icon, which is automatically associated with the resource type when done.

  - Click the **Define new custom icon** button to display the Custom Icons page.
  - Click **New** to create a new icon. Alternatively, click the *edit* symbol to edit an existing icon. After you have selected or created an icon, the *Configure resource type* page is displayed. Use the following information to define the icon:

    Icon properties

    Each custom icon must have a name that uniquely identifies the icon when assigning it to a type.

    Remember: You cannot change the name of an existing icon. If you want an icon to have a different name, create a new icon, then delete the old one.

    Icons are defined inside an SVG editor, which runs an XML validation check. Each icon definition must be valid SVG XML with a viewBox, which is important to ensure scaling of the image.

    The SVG definition must include inline styling of the image, such as stroke color and fill color. Use of style classes is not advised, as it can cause visualization issues on some browsers. If style classes must be used, naming must be unique for each svg image to prevent class definitions from being overwritten.

    Optionally, each icon can be assigned to a category, so you can group icons of the same type or function together when displaying them in a list.

    Remember: You can also create custom icons from the Custom Icons page accessed through DASH, which is described in the Defining custom icons topic.

- Define the label for the resource type by editing the following fields:

  - Label

    By default the name of a property (**`asmProperties.name`**) is used as the label that is displayed in the topology. You can replace this by typing in a custom label for the resource type.

  - Label Maximum Length

    You can override the default label length of 15 characters to avoid truncating the displayed label.

    Warning: Avoid labels that are too long, as long labels can overlap in the topology. The maximum suggested label length is 20 characters.

- Choose a shape for the resource type from the **Resource Shape** drop-down list.

  The default shape for a resource in the Topology Viewer is a circle, or a square for a host server. You can change the shape of the resource type to one of the following shapes:

  - Circle
  - Square
  - Hexagon
  - Vertical hexagon

Styling

On the **Styling** tab, you can define the selected resource type border color, border pattern, background color, and resource display size:

- For the border color, enter a hex definition. The default border color is **`#171717`**.
- For the border pattern, the default pattern is `''`.
- For the resource background color, the default is **`#F3F3F3`** (light gray).
- For the size, the default is medium.

  You can further refine the size function by specifying how certain resource properties affect the size of the resource type displayed. For example, resource types can appear larger depending on the number of connections they have with other resources.

Configuring relationship types

You can customize the styles and labels for specified relationship types using the Relationship Types page. You can also delete existing, or create new relationship type styles.

You customize styles in two steps. First, you identify the relationship type to be customized, or create a relationship type. Then, you can customize the style elements for the relationship type. To customize the style, you can set the relationship line color, thickness, style, and label.

Any resource or relationship properties that are used in custom relationship styles must be added as required properties to the *Advanced topology settings*. If the properties do not exist in the advanced topology settings, you are prompted to add the properties when you save a new relationship style. For more information, see Advanced topology settings.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.
3. From the topology navigation toolbar, expand **Settings**, and click **Topology configuration**.
4. On the *Relationship types* card, click **Configure**.

   The *Relationship types page* is displayed and lists all of your existing customized relationship types in a table with sortable columns. The table also displays the last updated timestamp, and indicates whether a relationship type has a custom label, custom color, or custom width defined. From here, you can choose to delete or edit configurations. You can also create a relationship type configuration.

   Tips: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. You can click **Refresh** (top right) to reload the items displayed, which can be useful if other users are customizing them. You can select more than one relationship type and reset them to their default values, or perform a batch edit.
5. Select the action that you want to complete:
   - To delete a relationship type, select them in the table and click **Delete relationship settings**.
   - To revert to default settings, select an item in the table and click **Reset to defaults**.
   - To edit existing relationship types, select them in the table and click **Edit relationship type** (or **Batch edit relationship types**). The *Configure relationship type* page is displayed.
   - To create a new relationship type, click **New**. The *Configure relationship type* page is displayed.
6. Select the **Identification** tab on the *Configure relationship type* page.
   - Relationship Type

     Choose the relationship type that you want to configure from the dropdown list.

     Note: Only relationship types that exist in your topology database are listed.
   - Label Function

     Example of a JavaScript function to define a label for the relationship using the resource properties:

     ```
     return asmProperties.labelText;
     ```
7. Select the **Style** tab on the *Configure relationship type* page.

   Warning: If you use the advanced JavaScript format to define colors, do not define a numeric value instead of an actual color, as this will prevent the topology rendering from being completed.
   - Line Color Function

     Example JavaScript function to define link color:

     ```
     return 'blue';
     ```

     Default function:

     ```
     return '#171717';
     ```
   - Line Width Function

     Example JavaScript function to define line width for the relationship using the source resource properties:

     ```
     return asmSourceProperties.myProp > 10 ? '9px' : '1.5px';
     ```

     Default function:

     ```
     return '0.5px';
     ```
   - Line Pattern Function

     Example of a JavaScript function to define the line pattern:

     ```
     return '3 4 5';
     ```

     Default function:

```
return null;
```

8. Click **Save**.

   If any resource or link properties are not defined in the *Advanced topology settings*, you are prompted to save the properties now.

The relationship style and label for the specified relationship type is customized.

Style restrictions

- The line color (strokeWidth) property can have a maximum value of 9px. Any higher value can cause styling problems.
- The label (linkLabel) property must return a valid string to be displayed alongside the relationship.

For detailed SVG style definition information, see:

- Line color (stroke)
- Line width (stroke-width)
- Line pattern (stroke-dasharray)

Accessing properties for styling functions

When you are defining styles, you can access dynamic properties of the relationship or connected resources.

- To access the properties of resources, use the **asmProperties** JavaScript object.
- To access the properties of relationships, use the **asmSourceProperties** or **asmTargetProperties** JavaScript objects.

  Remember: The arrows indicating a relationship point from the source to the target.
- To access the highest open severity status from the source or target resource, use the **asmSourceProperties._hasStatus** or **asmTargetProperties._hasStatus** JavaScript objects. The following example uses the **_hasStatus** parameter to modify the relationship label:

```
if (asmSourceProperties._hasStatus || asmTargetProperties._hasStatus) {
    // object of all ASM's status severities
    let severityRank = {
        clear: 0,
        indeterminate: 1,
        information: 2,
        warning: 3,
        minor: 4,
        major: 5,
        critical: 6
    };
    let sourceSeverityRank = severityRank[asmSourceProperties._hasStatus] || 0;
    let targetSeverityRank = severityRank[asmTargetProperties._hasStatus] || 0;
    let labelString = asmProperties._edgeType;
    // Show highest Status on relationship label
    if(sourceSeverityRank > targetSeverityRank) {
        labelString += ': Status = ' + asmSourceProperties._hasStatus;
    } else {
        labelString += ': Status = ' + asmTargetProperties._hasStatus;
    }
    return labelString;
} else {
    // No status for source or target resources use plain label
    return asmProperties._edgeType;
}
```

Configuring custom icons

You can add custom icons for resources that are displayed in the IBM Cloud Pak for AIOps console using the *Custom Icons* page.

Before you begin

To access the *Custom Icons* page, you must have the required permissions.

About this task

Tip: Instead of using the following procedure, you can first select a specific resource type from the Resource Types page, and then define a custom icon for that specific resource type only. This is described in Editing resource type styles. However, if you intend to create a number of icons without assigning them, or simply want to edit or delete icons, use the following procedure.

Icon properties

- Each custom icon must have a name that uniquely identifies the icon when assigning it to a type. You cannot change the name of an existing icon. If you want an icon to have a different name, create another icon, then delete the old icon.
- Icons are defined inside an SVG editor, which runs an XML validation check. Each icon definition must be valid SVG XML with a **viewBox**, which is important to ensure scaling of the image.

The SVG definition must include inline styling of the image, such as stroke color and fill color. Use of style classes is not advised, as it can cause visualization issues on some browsers. If style classes must be used, naming must be unique for each svg image to prevent class definitions from being overwritten.

- Optionally, each icon can be assigned to a category, which allows you to group icons of the same type or function together when displaying them in a list.

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.
3. From the topology navigation toolbar, expand **Settings**, and click **Topology configuration**.
4. On the *Custom icons* card, click **Configure**. The Custom icons page lists the custom icons in table format by name, icon, category, last update, and size, in sortable columns.

   Tip: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. You can also click **Refresh** (top right) to reload the items displayed, which can be useful if other users are customizing them.
5. Click **New** to create an icon. Alternatively, select icons in the table to edit or delete them.
6. On the *New custom icon* page, enter the details for your custom icon:
   - Enter a unique name. Each icon must have a name, which uniquely identifies the icon when assigning it to a type.

     Remember: You cannot change the name of an existing icon. If you want an icon to have a different name, create an icon, then delete the old icon.
   - Optionally, enter a category name. Each icon can be assigned to a category. You can use categories to group icons of the same type or function together.
   - Enter the SVG XML to define the icon. Use the editor to enter valid XML.

     Each icon definition must be valid SVG XML with a viewBox, which is important to ensure scaling of the image. The SVG editor rejects any invalid XML entered.

     The XML editor includes a Preview area where the results of your edits display. The following conditions apply:

     - Height or width should not be specified.
     - Inline styles, such as stroke color and fill color, should be used and not style classes. Naming must be unique for each SVG image to prevent class definitions from being overwritten.
     - SVG namespace is required: `xmlns="http://www.w3.org/2000/svg"`

   Example: Use the following definition for the `disk` icon as guidance:

   ```
   <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 64 64">
      <ellipse style="fill-opacity:0;stroke:currentColor;stroke-width:12.12270069;" id="path4139" cx="33.627117"
   cy="32.949142" rx="16.803904" ry="17.210684"/>
      <circle cx="33.827423" cy="33.055576" r="3.3037829"/>
   </svg>
   ```
7. Click **Save**.

What to do next

You can assign your custom icons to particular resource types using the *Resource types* page. On that page, you can also apply further style edits to the resource types.

Configuring rules

Rules help streamline topologies and conserve system resources, such as by merging different observer records of the same resource into a single composite resource, or by excluding specific changes from being recorded against a resource history.

Before you begin

Ensure that you have the details to hand about the resources for which you intend to develop rules. For example, to create merge rules, you must know which resources exist as duplicate records before you can merge these into composites.

About this task

Resource filters: Resource filters let you use resource properties to either include or exclude all resources with these properties from having a rule applied to them. You can add more than one property as a filter. Resource filters cannot be set for history or event rules.

Tokens: Rules copy the values of properties to a token. Tokens are sets of strings that behave in ways specific to each rule type. You can combine several properties into a single token by using `${propertyName}`.

Token filters: An 'Include' token filter applies only to resources or groups having the listed tokens. You can use an 'Include' token filter simply to include (or exclude) specific properties, or extend its use with regular expressions to manipulate the input property.

Example use of tokens and token filters

Token	Include token filter	Input properties	Single output token value
host		host=example.ibm.com	example.ibm.com
${host}		host=example.ibm.com	example.ibm.com
${host}location-${location}		host=example.ibm.com,location=1	example.ibm.com location-1
host	host=~/^(.?)/..$/	host=example.ibm.com	example
${uniqueId}Location-x	uniqueId=~/^.(.)$/	uniqueId=anything	Location-X

Tip: See the Rules reference topic for additional information.

Rule types

You can use a number of different types of rules for different purposes. The rule type (**`ruleType`**) can be one the following rules.

Merge rules (mergeRule)

- Define a merge rule to populate the mergeTokens of resources matched by the rule to prevent duplicate records of the same resource from being displayed in the topology.
- A merge rule copies particular properties of a resource into the resource's mergeTokens property.

Match token rules (matchTokensRule)

- Define a match tokens rule to copy the value of another field to the matchTokens field so that it can be used to match events to resources.
- A match token rule populates the matchTokens of resources matched by the rule.

Tags rules (tagsRule)

- Define a tags rule to copy the value of another field to the tags field so that it becomes searchable in the UI.
- A tags rule populates the tags of resources matched by the rule.

  Tip: Any field that isn't indexed and can therefore not normally be searched for becomes searchable if copied to the tags property. For a list of indexed properties, see the 'Generic properties' table.

History rules (historyRule)

- Define a history rule to identify properties that change every observation, but that don't indicate an important change, for example a host's sysUpTime property.
- A history rule excludes properties from being retained in history, thereby saving resources by not maintaining detailed historical records of changes to these properties.

File enrichment rules

- Define a file enrichment rule in order to enrich resources as they are loaded into the topology service database with their geographical location.

Events filter rules (eventsFilterRule)

- Define an events filter rule to prevent status derived from observer jobs from generating events in the system.

Business criticality rules

- Define a business criticality rule to copy a particular property of a resource into the resource's businessCriticality property, which is then used to create a business criticality rule.

Accessing the Rules administration page

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.
3. From the topology navigation toolbar, expand **Settings**, click **Topology configuration**, and then open the *Rules administration* page.
4. Select the appropriate rules tab.

Defining merge rules

A merge rule copies particular properties of a resource into the resource's mergeTokens property. Resources which contain the same merge token will be merged so that they are treated as a single 'composite' resource. In this way, the topologies from different data sources can be connected.

About this task

Different observers deployed as part of the Agile Service Manager solution may record and then display the same resource as two (or more) resources. To prevent this, you create a merge rule that ensures that the separate records of the same resource share values in their tokens set, which then triggers the merge service to

create a single composite resource vertex. Composite, merged resources are displayed in the Topology Viewer as a single resource, which includes the properties of all merged resources. Merge rules are applied to a resource in an observer job before it is sent to the topology service.

Rules can be managed using the Rules REST API in the Merge Service. For each Agile Service Manager observer, merge rules control which tokens are considered merge tokens.

Procedure

1. On the *Rules administration* page, select the **Merge rules** tab.
2. Complete the **Details** section by defining the name, status, and tokens for the rule.

   a. Enter a name, which must be unique within the context of the tenant.

   You cannot change the name of an existing rule. If you want a rule to have a different name, create a new rule, then delete the old one.

   b. Toggle the rule status to be either **Enabled** or **Disabled**.

   When observer jobs are run, they apply only the rules that are in an enabled state.

   c. Define merge tokens for the rule, then click **Add+**.

   The tokens defined as part of a rule contain the list of shared resource parameter names that will become merge tokens for those resources to which the rule is applied. Merge tokens can be constructed using variable substitutions, which allow you to combine more than one property value in a token and also combine them with literal strings, as shown in this example.

   Important: The tokens are the shared elements that the duplicate records to be merged have in common.
3. Complete the **Conditions** section by selecting the observers, provider, and resource type for the rule, and by setting the token include / exclude switch.

   a. Select observers to which this rule applies from the Observer name drop-down. Leave empty to apply the rule to all observers.

   b. Select providers to which this rule applies from the Provider name drop-down. Leave empty to apply the rule to all providers.

   c. Select resource types to which this rule applies from the Resource type drop-down. Leave empty to apply the rule to all resource types.

   d. Set the Resource filter mode to either Exclude or Include, then define a filter by specifying a property. You can add more than one property as a filter.

   - **Exclude:** Resources with properties that match any of these filters will not have the rule applied to them.
   - **Include:** Only resources with properties that match any of these filters will have the rule applied to them.

   e. Set the Token filter mode to either Exclude or Include, then define a filter in the format of a regular expression. You can set more than one filter.

   - **Exclude:** Tokens which match any one of the expressions will not be applied to resources.
   - **Include:** Only tokens which match one of the expressions will be applied to the applicable resources.
4. Validate your filters using the **Validate token filters** button.
5. Click **Save**.

Examples

Default merge rule for the Docker Observer**

The following example is the default merge rule defined for the Docker Observer:

```
rules:
    - name: dockerId
      ruleType: mergeRule
      ruleStatus: enabled
      tokens: [ dockerId ]
      entityTypes: null
      observers: [ docker-observer ]
      providers: null
    - name: almExternalId
      ruleType: mergeRule
      ruleStatus: enabled
      tokens: [ externalId ]
      entityTypes: null
      observers: [ alm-observer ]
      providers: null
```

Note that the rules name in this example is **`dockerId`**, and that it applies only to instances of observers named **`docker-observer`**. The ruleType property here specifies the mergeRule rule type. This merge rule applies to all entity types and all providers and will copy the dockerId property into the merge tokens for all resources that have this property set.

Sample rules for merging resources

A composite resource has its own unique identifier, and the merged resources continue to exist separately. The following example shows the individual and composite resources when retrieved from the topology Resources API.

Resource one

```
https://<your_NASM_host>/1.0/topology/resources/ABC
{
  "_id": "ABC",
```

```
  "name": "host1",
  "propertyAbc": "This property only exists on ABC",
  "entityTypes": [ "host" ],
  "tokens": [ "host1MergeToken" ],
  "_compositeId": "XYZ"
}
```

The resource has an id of **ABC**, and the value of **compositeId** denotes the single, merged resource, which is **XYZ**.

Resource two

```
https://<your host>/1.0/topology/resources/DEF
{
  "_id": "DEF",
  "name": "host1",
  "propertyDef": "This property only exists on DEF",
  "entityTypes": [ "host" ],
  "tokens": [ "host1MergeToken" ],
  "_compositeId": "XYZ"
}
```

The resource has an id of **DEF**, and the value of compositeId denotes the single, merged resource, which is **XYZ**.

Composite resource

```
https://<your_NASM_host>/1.0/topology/resources/XYZ
{
  "_id": "XYZ",
  "name": "host1",
  "propertyAbc": "This property only exists on ABC",
  "propertyDef": "This property only exists on DEF",
  "entityTypes": [ "host" ],
  "tokens": [ "host1MergeToken" ],
  "_compositeOfIds": [ "ABC", "DEF " ]
}
```

The resource has an id of **XYZ**, and the value of **compositeOfIds** lists the ids of the merged resources, in this case **ABC** and **DEF**. The XYZ composite resource includes the properties from both of the merged resources.

Resource with variable substitutions and exclude list

```
{
  "name": "sysNameMatching",
  "tokens": [ "sysName", "${name}/${customField}"],
  "ruleStatus": "enabled",
  "entityTypes": [ "host", "server" ],
  "observers": [ "ITNM", "TADDM" ],
  "providers": [ "*" ],
  "customField": "string",
  "excludeTokens": [ "^asm-default.*"]
}
```

The **^asm-default.*** value set for excludeTokens ensures that any values that match the regular expressions are excluded. The merge token with the value of **${name}/${customField}** combine the **${name}** and **${customField}** properties using the **${}** syntax, and demonstrate how variable substitutions work.

Literal values are entered as they are in the merge token, which in this case is the / character.

To be backwards compatible, tokens consisting of a single value, as in the sysName example, are treated as variable substitutions, that is, as if they are ${sysName}.

Tip: You can find more information on merge rules at the following blog post: **https://medium.com/@ibmjdiaz/using-merge-rules-in-agile-service-manager-b66b36707b1c**

Defining match tokens rules

A match tokens rule copies the value of another field to the matchTokens field so that it can be used to match events (alerts) to resources.

About this task

A match tokens rule populates the matchTokens of resources matched by the rule (a simple example is included below the procedure).

Procedure

1. On the *Rules administration* page, select the **Match tokens rules** tab.
2. Complete the **Details** section by defining the name, status, and tokens for the rule.

   a. Enter a name, which must be unique within the context of the tenant.

   You cannot change the name of an existing rule. If you want a rule to have a different name, create a new rule, then delete the old one.

   b. Toggle the rule status to be either **Enabled** or **Disabled**.

When observer jobs are run, they apply only the rules that are in an enabled state.

c. Define merge tokens for the rule, then click **Add+**.

The tokens defined as part of a rule contain the list of shared resource parameter names that will become match tokens for those resources to which the rule is applied. Match tokens can be constructed using variable substitutions, which allow you to combine more than one property value in a token and also combine them with literal strings.

3. Complete the **Conditions** section by selecting the observers, provider, and resource type for the rule, and by setting the token include / exclude switch.

a. Select observers to which this rule applies from the Observer name drop-down. Leave empty to apply the rule to all observers.

b. Select providers to which this rule applies from the Provider name drop-down. Leave empty to apply the rule to all providers.

c. Select resource types to which this rule applies from the Resource type drop-down. Leave empty to apply the rule to all resource types.

d. Set the Resource filter mode to either Exclude or Include, then define a filter by specifying a property. You can add more than one property as a filter.

- **Exclude:** Resources with properties that match any of these filters will not have the rule applied to them.
- **Include:** Only resources with properties that match any of these filters will have the rule applied to them.

e. Set the Token filter mode to either Exclude or Include, then define a filter in the format of a regular expression. You can set more than one filter.

- **Exclude:** Tokens which match any one of the expressions will not be applied to resources.
- **Include:** Only tokens which match one of the expressions will be applied to the applicable resources.

4. Validate your filters using the **Validate token filters** button.

5. Click **Save**.

Examples

Default merge rule for the Docker Observer**

The following example is the default merge rule defined for the Docker Observer:

```
rules:
    - name: dockerId
      ruleType: mergeRule
      ruleStatus: enabled
      tokens: [ dockerId ]
      entityTypes: null
      observers: [ docker-observer ]
      providers: null
    - name: almExternalId
      ruleType: mergeRule
      ruleStatus: enabled
      tokens: [ externalId ]
      entityTypes: null
      observers: [ alm-observer ]
      providers: null
```

Note that the rules name in this example is **dockerId``, and that it applies only to instances of observers named** docker-observer``. The ruleType property here specifies the mergeRule rule type. This merge rule applies to all entity types and all providers and will copy the dockerId property into the merge tokens for all resources that have this property set.

Example

Match token rules

A match token rule allows the value of another field to be copied to the matchTokens field so that it can be used to match events to resources. It populates the matchTokens of resources matched by the rule.

Simple example of a matchTokensRule:

```
- name: matchRule
  ruleType: matchTokensRule
  ruleStatus: enabled
  tokens: [ name ]
  entityTypes: null
  observers: null
  providers: null
```

Resource with variable substitutions and exclude list

```
{
  "name": "sysNameMatching",
  "tokens": [ "sysName", "${name}/${customField}"],
  "ruleStatus": "enabled",
  "entityTypes": [ "host", "server" ],
  "observers": [ "ITNM", "TADDM" ],
  "providers": [ "*" ],
  "customField": "string",
  "excludeTokens": [ "^asm-default.*"]
}
```

The **^asm-default.*** value set for excludeTokens ensures that any values that match the regular expressions are excluded. The merge token with the value of **${name}/${customField}** combine the **${name}** and **${customField}** properties using the **${}** syntax, and demonstrate how variable substitutions work.

Literal values are entered as they are in the merge token, which in this case is the / character.

To be backwards compatible, tokens consisting of a single value, as in the sysName example, are treated as variable substitutions, that is, as if they are ${sysName}.

Defining tags rules

A tags rule copies the value of another field to the tags field so that it becomes searchable in the UI.

About this task

A resource's tags can be searched via the UI, and matching resources can be displayed. Resources with common tags can optionally be grouped using a tag-based template definition.

A tags rule populates the tags of resources matched by the rule. Examples are included below the procedure.

Tip: Any field that isn't indexed and can therefore not normally be searched for becomes searchable if copied to the tags property.

ITNM Observer tip: You can create custom tags on IBM Tivoli Network Manager, which the ITNM Observer can integrate into Agile Service Manager tags. For information on creating custom tags in Network Manager, see the 'A'dding name-value pairs to entities using the File finder' topic in the IBM Tivoli Network Manager IP Edition documentation.

Having added a custom tag in Network Manager, you can extract the value into an Agile Service Manager tag. The following example uses a regular expression to extract the value of the `Country` custom tag:

```
customTags=~/[\\{]Country=(.+?)[,\\}]/
```

Procedure

1. On the *Rules administration page, select the **Tags rules** tab.
2. Complete the **Details** section by defining the name, status, and tokens for the rule.
    1. Enter a name, which must be unique within the context of the tenant.

       You cannot change the name of an existing rule. If you want a rule to have a different name, create a new rule, then delete the old one.
    2. Toggle the rule status to be either **Enabled** or **Disabled**.

       When observer jobs are run, they apply only the rules that are in an enabled state.
    3. Define merge tokens for the rule, then click **Add+**.

       The tokens defined as part of a rule contain the list of shared resource parameter names that will become tags for those resources to which the rule is applied. Tag tokens can be constructed using variable substitutions, which allow you to combine more than one property value in a token and also combine them with literal strings.
3. Complete the **Conditions** section by selecting the observers, provider, and resource type for the rule, and by setting the token include / exclude switch.
    1. Select observers to which this rule applies from the Observer name drop-down. Leave empty to apply the rule to all observers.
    2. Select providers to which this rule applies from the Provider name drop-down. Leave empty to apply the rule to all providers.
    3. Select resource types to which this rule applies from the Resource type drop-down. Leave empty to apply the rule to all resource types.
    4. Set the Resource filter mode to either Exclude or Include, then define a filter by specifying a property. You can add more than one property as a filter.
        - **Exclude:** Resources with properties that match any of these filters will not have the rule applied to them.
        - **Include:** Only resources with properties that match any of these filters will have the rule applied to them.
    5. Set the Token filter mode to either Exclude or Include, then define a filter in the format of a regular expression. You can set more than one filter.
        - **Exclude:** Tokens which match any one of the expressions will not be applied to resources.
        - **Include:** Only tokens which match one of the expressions will be applied to the applicable resources.
4. Validate your filters using the **Validate token filters** button.
5. Click **Save**.

Example 1

This tags rule allows the value of another field to be copied to the tags field so that it becomes searchable in the UI. It populates the tags of resources matched by the rule.

Simple example of a tagsRule:

```
- name: tagRuleCustomProp
  ruleType: tagsRule
  ruleStatus: enabled
  tokens: [ name ]
  entityTypes: null
```

```
  observers: null
  providers: null
```

Example 2

The following example creates tags rules for the Docker Observer and will add **com.docker.compose.service** and **com.docker.compose.project** properties to the tags list, for entities of the type **container**:

```
curl -k -s -X POST --user $PROXY_USER:$PROXY_PASSWORD --header 'Content-Type: application/json' --header 'Accept:
application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -w "Response code: %{http_code}\n" -d '{
  "tokens": [
    "com.docker.compose.service",
    "com.docker.compose.project"
  ],
  "name": "tag-docker-container-service-project",
  "ruleStatus": "enabled",
  "ruleType": "tagsRule",
  "observers": [
    "docker_observer"
  ],
  "entityTypes": [
    "container"
  ]
}
' "https://${PROXY_HOST}/1.0/merge/rules"
```

To verify:

```
{
  "_createdAt": "2019-09-10T02:24:03.652Z",
  "_executionTime": 10,
  "_id": "YEaGjuCJT5-wCH-SAfCssA",
  "_modifiedAt": "2019-09-10T15:29:33.829Z",
  "_observedAt": "2019-09-10T15:29:33.829Z",
  "_startedAt": "2019-09-10T15:29:33.829Z",
  "architecture": "x86_64",
  "changeTime": 1568129373829,
  "com.docker.compose.config-hash": "6afc3ec12fd17d2b815df61e4ab276af6a27324a1487567f1bbba29a4a4bca1e",
  "com.docker.compose.container-number": "1",
  "com.docker.compose.oneoff": "False",
  "com.docker.compose.project": "asm",
  "com.docker.compose.service": "search",
  "com.docker.compose.version": "1.11.2",
  "createTime": 1568082243652,
  "dockerId": "dc39b5ea19cb79755e48a525bfbd2d479777464b1e80c708cb851da1b22fa952",
  "entityTypes": [
      "container"
  ],
  "matchTokens": [
      "asm_search_1",
      "dc39b5ea19cb79755e48a525bfbd2d479777464b1e80c708cb851da1b22fa952"
  ],
  "mergeTokens": [
      "dc39b5ea19cb79755e48a525bfbd2d479777464b1e80c708cb851da1b22fa952"
  ],
  "name": "asm_search_1",
  "observedTime": 1568129373829,
  "tags": [
      "asm_search_1",
      "asm",
      "search"
  ],
  "uniqueId": "dc39b5ea19cb79755e48a525bfbd2d479777464b1e80c708cb851da1b22fa952",
  "vcs-type": "git",
  "vendor": "IBM",
  "version": "1.0.10-SNAPSHOT"
}
```

Example 3

Resource with variable substitutions and exclude list:

```
{
  "name": "sysNameMatching",
  "tokens": [ "sysName", "${name}/${customField}"],
  "ruleStatus": "enabled",
  "entityTypes": [ "host", "server" ],
  "observers": [ "ITNM", "TADDM" ],
  "providers": [ "*" ],
  "customField": "string",
  "excludeTokens": [ "^asm-default.*"]
}
```

The **^asm-default.*** value set for excludeTokens ensures that any values that match the regular expressions are excluded. The merge token with the value of **${name}/${customField}** combine the **${name}** and **${customField}** properties using the **${}** syntax, and demonstrate how variable substitutions work.

Literal values are entered as they are in the merge token, which in this case is the / character.

To be backwards compatible, tokens consisting of a single value, as in the sysName example, are treated as variable substitutions, that is, as if they are ${sysName}.

Defining history rules

A history rule identifies properties that change every observation, but that don't indicate an important change, for example a host's **sysUpTime** property. A history rule allows you to exclude properties from being retained in history, thereby saving resources by not maintaining detailed historical records of changes to these properties.

About this task

A history rule populates the **historyExcludeTokens** field with field names.

When a resource is updated, the topology service checks if all the updates are on fields that are listed in **historyExcludeTokens**, and if they are, it updates the resource without creating history.

Procedure

1. On the *Rules administration* page, select the **History rules** tab.
2. Complete the **Details** section by defining the name, status, and tokens to be used for the rule.

   a. Enter a name, which must be unique within the context of the tenant.

   You cannot change the name of an existing rule. If you want a rule to have a different name, create a new rule, then delete the old one.

   b. Toggle the rule status to be either **Enabled** or **Disabled**.

   When observer jobs are run, they apply only the rules that are in an enabled state.

   c. Define a token for the rule, then click **Add+**.

   The tokens defined as part of a rule contain the list of shared resource parameter names that will become **history exclude** tokens for those resources to which the rule is applied. Merge tokens can be constructed using variable substitutions, which allow you to combine more than one property value in a token and also combine them with literal strings.
3. Complete the **Conditions** section by selecting the observers, provider, and resource type for the rule.

   a. Select observers to which this rule applies from the Observer name drop-down. Leave empty to apply the rule to all observers.

   b. Select providers to which this rule applies from the Provider name drop-down. Leave empty to apply the rule to all providers.

   c. Select resource types to which this rule applies from the Resource type drop-down. Leave empty to apply the rule to all resource types.
4. Click **Save**.

Defining file enrichment rules

A file enrichment rule provides additional resource information (via file) that is matched to specific resources via a specified resource identifier (the token).

About this task

The supplemental resource information provided via a file enrichment rule can be geolocation data provided as a GeoPoint file, or any other data provided as a more complex JSON file.

Using GeoPoint

Using JSON

Using a GeoPoint file

The GeoPoint format is specific to geolocation data. A file enrichment rule matches a specified resource identifier (the token) to the longitude and latitude (the geographical location) of the resource. Resources that have been enriched with a geographical location can be displayed on the topology resource map.

Before you begin

Remember: To enrich your resource records with geolocation data, you can use a file enrichment rule as described here, or you can load resource data with geolocation properties included, for example by using the File or REST observers.

Create a file that contains the location information for your resources.

The file enrichment rule uses the following parameters for GeoPoint:

```
{
"tokens" : ["clli"],
"fileName" : "example-geopoint.txt",
"format" : "geoPoint,
```

```
...
}
```

File format - The file you use must be in the TSV (tab-separated values) format. - The values on each line are 'token', 'longitude' and 'latitude':

```
token longitude latitude
```

Tip: CLLI (Common Language Location Identifiers) are supported and can be used as tokens.

Example

A file enrichment rule allows the value of a token to be used to map that resource to its location on a map.

Example TSV file

```
LONDON-CTR-01   -0.120226   51.509287
LONDON-CTR-02   -0.108514   51.485100
MCHSTR-CTR-01   -2.248665   53.482641
LVERPL-CTR-01   -2.996455   53.403925
```

Example rule

```
* name = serverBuildingLocations
* enabled = true
* token = buildingCode
* file = server-buildings.tsv
```

Example resource

```
{
  "uniqueId": "server-001",
  "name": "server-001",
  "entityTypes": [
    "server"
  ],
  "buildingCode": "LONDON-CTR-01"
}
```

Procedure

1. On the *Rules administration* page, select the **File enrichment rules** tab.
2. Complete the **Details** section by defining the name, status, token and file to be used for the rule.

   a. Enter a name, which must be unique within the context of the tenant.

   You cannot change the name of an existing rule. If you want a rule to have a different name, create a new rule, then delete the old one.

   b. Toggle the rule status to be either **Enabled** or **Disabled**.

   When observer jobs are run, they apply only the rules that are in an enabled state.

   c. Define a token for the rule, then click **Add+**.

   The token is used to uniquely identify the resource to which the rule is applied, against which a location is mapped using the data contained in the rule file. Only a single token is used to map a location, but you can construct it using variable substitutions, which lets you combine more than one property value if required.

   d. Specify the format of the enrichment file, in this case **GeoPoint**.

   Remember: You must create this file before you can create a new file enrichment rule.

   e. Specify the enrichment file to be uploaded.

   Under **File mode**, select one of the following options:

   - Upload local file: Either drag-and-drop or click and browse to choose a file.
   - Use server file: Specify the name of the file to be loaded. The file must have been uploaded previously. Do not specify a path.
3. Complete the **Conditions** section by selecting the observers, provider, and resource type for the rule, and by setting the token include / exclude switch.

   a. Select observers to which this rule applies from the Observer name drop-down. Leave empty to apply the rule to all observers.

   b. Select providers to which this rule applies from the Provider name drop-down. Leave empty to apply the rule to all providers.

   c. Select resource types to which this rule applies from the Resource type drop-down. Leave empty to apply the rule to all resource types.

   d. Set the Resource filter mode to either Exclude or Include, then define a filter by specifying a property. You can add more than one property as a filter.

   - **Exclude:** Resources with properties that match any of these filters will not have the rule applied to them.
   - **Include:** Only resources with properties that match any of these filters will have the rule applied to them.

   e. Set the Token filter mode to either Exclude or Include, then define a filter in the format of a regular expression. You can set more than one filter.

   - **Exclude:** Tokens which match any one of the expressions will not be applied to resources.
   - **Include:** Only tokens which match one of the expressions will be applied to the applicable resources.
4. Validate your filters using the **Validate token filters** button.
5. Click **Save**.

Results

Additional resource information added via the file enrichment functionality is displayed on the Resource details screen.

Using a JSON file

The JSON file format lets you enrich resource data with any custom properties required. A file enrichment rule using a JSON file matches a specified resource identifier (the token) as well as a file enrichment key (a second token) in order to provide additional data for specified resources.

Before you begin

Create a file that contains the additional information for your resources.

The file enrichment rule uses the following parameters for JSON:

```
{
"tokens" : ["uniqueId"],
"fileName" : "example-json.txt",
"format" : "json",
"fileEnrichmentKey":"cityName",
...
}
```

File format - The JSON file format requires separate tokens for the resource and for the file data.

Example

```
- name = fileEnrichmentKey
- token = uniqueId

{"name":"Ireland","capital":"Dublin","tags":["green","wet"]}
{"name":"UK","capital":"London","geolocation":{"geometry":{"type":"Point","coordinates":[0.00000,0.00000]}}}
{"name":"France","capital":"Paris","population":70000000}
{"name":"Germany","capital":"Berlin","entityTypes":["rave scene"]}
{"key":"Switzerland","capital":"Bern"}
```

Resources

```
V:{"uniqueId":"Ireland","entityTypes":["country"],"tags":["republic"]}
V:{"uniqueId":"UK","entityTypes":["country"]}
V:{"uniqueId":"France","entityTypes":["country"]}
V:{"uniqueId":"Germany","entityTypes":["country"]}
V:{"uniqueId":"Switzerland","entityTypes":["country"]}
```

Enriched resources

```
V:{"uniqueId":"Ireland","entityTypes":["country"],"tags":["republic","green","wet"],"capital":"Dublin"}
V:{"uniqueId":"UK","entityTypes":["country"],"capital":"London","geolocation":{"geometry":{"type":"Point","coordinates":
[0.00000,0.00000]}}}
V:{"uniqueId":"France","entityTypes":["country"],"population":70000000,"capital":"Paris"}
V:{"uniqueId":"Germany","entityTypes":["country","rave scene"],"capital":"Berlin"}
V:{"uniqueId":"Switzerland","entityTypes":["country"],"capital":"Bern"}
```

Procedure

1. On the *Rules administration* page, select the **File enrichment rules** tab.
2. Complete the **Details** section by defining the name, status, token and file to be used for the rule.

   a. Enter a name, which must be unique within the context of the tenant.

   You cannot change the name of an existing rule. If you want a rule to have a different name, create a new rule, then delete the old one.

   b. Toggle the rule status to be either **Enabled** or **Disabled**.

   When observer jobs are run, they apply only the rules that are in an enabled state.

   c. Define a token for the rule, then click **Add+**.

   The token is used to uniquely identify the resource to which the rule is applied.

   Remember: The `fileEnrichmentRule` you are creating uses this token to match the file to the resource.

   d. Specify the format of the enrichment file, in this case **JSON**.

   Remember: You must create this file before you can create a new file enrichment rule.

   e. Specify the enrichment file to be uploaded, and the enrichment file key.

   - Under **File mode**, either upload a local file (drag-and-drop or click and browse), or specify the name of the file to be loaded from a server (uploaded previously; do not specify a path).
   - Specify the enrichment file key. This identifies the property in each line of the enrichment file to be matched against the rule token.

     Remember: The `enrichmentFileKey` is the token used in the JSON file to perform the matching against the token chosen in step 2c.
3. Complete the **Conditions** section by selecting the observers, provider, and resource type for the rule, and by setting the token include / exclude switch.

a. Select observers to which this rule applies from the Observer name drop-down. Leave empty to apply the rule to all observers.

b. Select providers to which this rule applies from the Provider name drop-down. Leave empty to apply the rule to all providers.

c. Select resource types to which this rule applies from the Resource type drop-down. Leave empty to apply the rule to all resource types.

d. Set the Resource filter mode to either Exclude or Include, then define a filter by specifying a property. You can add more than one property as a filter.

- **Exclude:** Resources with properties that match any of these filters will not have the rule applied to them.
- **Include:** Only resources with properties that match any of these filters will have the rule applied to them.

e. Set the Token filter mode to either Exclude or Include, then define a filter in the format of a regular expression. You can set more than one filter.

- **Exclude:** Tokens which match any one of the expressions will not be applied to resources.
- **Include:** Only tokens which match one of the expressions will be applied to the applicable resources.

4. Validate your filters using the **Validate token filters** button.

5. Click **Save**.

Results

Additional resource information added via the file enrichment functionality is displayed on the Resource details screen.

Defining events filter rules

An events filter rule prevents the status derived from observer jobs from generating events in the system. You specify observers, providers and resource types to which the rule is applied.

About this task

You define events filter rules to prevent the unnecessary proliferation of system events, for example under testing conditions or any other scenarios where you do not require new events to be created for specific observers, providers or resource types.

Procedure

1. On the *Rules administration* page, select the **Events filter rules** tab.

2. Complete the **Details** section by defining the name and status for the rule.

   a. Enter a name, which must be unique within the context of the tenant.

   You cannot change the name of an existing rule. If you want a rule to have a different name, create a new rule, then delete the old one.

   b. Toggle the rule status to be either **Enabled** or **Disabled**.

   When observer jobs are run, they apply only the rules that are in an enabled state.

3. Complete the **Conditions** section by selecting the observers, provider, and resource type for the rule, and by setting the token include / exclude switch.

   a. Select observers to which this rule applies from the Observer name drop-down. Leave empty to apply the rule to all observers.

   b. Select providers to which this rule applies from the Provider name drop-down. Leave empty to apply the rule to all providers.

   c. Select resource types to which this rule applies from the Resource type drop-down. Leave empty to apply the rule to all resource types.

4. Click **Save**.

Defining business criticality rules

A business criticality rule uses a rule token, which becomes the business criticality value for those resources to which the rule is applied.

About this task

A token can be constructed from resource parameters using variable substitution, which allows you to combine more than one property value if required. Only one token should be specified.

A business criticality rule populates the `businessCriticality` property of resources matched by the rule.

Procedure

1. On the *Rules administration* page, select the **Business criticality rules** tab.

2. Complete the **Details** section by defining the name, status, and single token for the rule.

   a. Enter a name, which must be unique within the context of the tenant.

You cannot change the name of an existing rule. If you want a rule to have a different name, create a new rule, then delete the old one.

b. Toggle the rule status to be either **Enabled** or **Disabled**.

When observer jobs are run, they apply only the rules that are in an enabled state.

c. Define a token for the rule, then click **Add+**.

The single token specified as part of a rule becomes the business criticality value for those resources to which the rule is applied.

3. Complete the **Conditions** section by selecting the observers, provider, and resource type for the rule, and by setting the token include / exclude switch.

a. Select observers to which this rule applies from the Observer name drop-down. Leave empty to apply the rule to all observers.

b. Select providers to which this rule applies from the Provider name drop-down. Leave empty to apply the rule to all providers.

c. Select resource types to which this rule applies from the Resource type drop-down. Leave empty to apply the rule to all resource types.

d. Set the Resource filter mode to either Exclude or Include, then define a filter by specifying a property. You can add more than one property as a filter.

- **Exclude:** Resources with properties that match any of these filters will not have the rule applied to them.
- **Include:** Only resources with properties that match any of these filters will have the rule applied to them.

e. Set the Token filter mode to either Exclude or Include, then define a filter in the format of a regular expression. You can set more than one filter.

- **Exclude:** Tokens which match any one of the expressions will not be applied to resources.
- **Include:** Only tokens which match one of the expressions will be applied to the applicable resources.

4. Validate your filters using the **Validate token filters** button.

5. Click **Save**.

Rules reference

Use the following reference information when constructing rules.

Token referencing rules

The rule mechanism supports a number of key use-cases, for example:

```
- Event correlation
- Merging data from multiple data sources
- Tagging resources
```

Rules extract data in a common or standardized way to support such use cases. Data structures can be complex, for example they may include nested key-value pairs in maps and lists, and the rule processing mechanism supports the extraction of data from such structures into a variable known as a 'token'. These tokens are used when applying the rules in order to modify the resource records being processed.

For example, you may wish to extract the value of a nested property and combine it with a literal string that you provide and treat that combination as a single resource tag.

Given the following input:

```
{{{key1=value1, customTags={field1_2=level1, field1_1={field2_2=level2, field2={field3=
{nestedKey=level3}}}}}}}
```

The following token definitions result in the shown result:

Table 1. Tokens and results

Token	Result
`key1`	`[value1]`
`${key1}`	`[value1]`
`customTags`	`[{field1_2=level1, field1_1={field2_2=level2, field2={field3={nestedKey=level3}}}}]`
`${customTags}`	`[{field1_2=level1, field1_1={field2_2=level2, field2={field3={nestedKey=level3}}}}]`
`${customTags->field1_2}`	`[level1]`
`${customTags->field1_1}`	`[{field2_2=level2, field2={field3={nestedKey=level3}}}]`
`${customTags->field1_1->field2_2}`	`[level2]`
`${customTags->field1_1->field2}`	`[{field3={nestedKey=level3}}]`
`${customTags->field1_1->field2->field3}`	`[{nestedKey=level3}]`
`${customTags->field1_1->field2->field3->nestedKey}`	`[level3]`
`${customTags->field1_1->field2>field3->nestedKey}`	`[]`
`${customTags->unknown_key}`	`[]`
`${unknownKey->unknown_key}`	`[]`
`${customTags->->unknown_key}`	`[]`
`${customTags->->field1_1}`	`[]`
`${customTags->}`	`[]`
`${->customTags}`	`[]`

Token	Result
`${>customTags>}`	`[]`
`${key1}and${customTags->field1_2}`	`[value1-and-level1]`

Table 2. Changes

Token	Revious result	New result
`{{${whatever}:${name}`	`[[foo, boo]:dinoop]`	`[foo:dinoop, boo:dinoop]}}`
`{{${whatever}:${name}:${place}`	`[[foo, boo]:dinoop:india]`	`[foo:dinoop:india, boo:dinoop:india]}}`
`{{${name}:${whatever}`	`[dinoop:[foo, boo]]`	`[dinoop:foo, dinoop:boo]}}`
`{{${place}:${whatever}:${name}`	`[India:[foo, boo]:dinoop]`	`india:foo:dinoop, india:bar:dinoop, india:zoo:dinoop}}`
`{{${whatever}:newlist`	`[[foo, boo]:newlist]`	`[foo:newlist, boo:newlist]}}`

Table 3. No changes

Token	Revious result	New result
`{{name`	`[dinoop]`	`[dinoop]}}`
`{{${name}`	`[dinoop]`	`[dinoop]}}`
`{{${place}:${name} }}`	`{{[india:dinoop]`	`[india:dinoop]}}`
`{{${age}:${name}`	`[30:dinoop]`	`[30:dinoop]}}`
`{{${whatever}`	`[[foo, boo]]`	`[[foo, boo]]}}`
`{{whatever`	`[boo, foo]`	`[boo, foo]}}`
`{{mymap`	`[{key1=val1}]`	`[{key1=val1}]}}`
`{{${mymap}`	`[{key1=val1}]`	`[{key1=val1}]}}`
`{{${mymap}:${name}`	`[{key1=val1}:dinoop]`	`[{key1=val1}:dinoop]}}`
`{{${whatever}:${newlist}`	`[[foo, boo]:[abc, xyz]]`	`[[foo, boo]:[abc, xyz]]}}`
`{{${whatever}:${mymap}`	`[[foo, boo]:{key1=val1}]`	`[[foo, boo]:{key1=val1}]}}`

Managing incidents and alerts

IBM Cloud Pak® for AIOps enables you to identify health issues across your application, services, and network infrastructure on a single management console. A **Incident list** displays each incident's priority, status, and description. Drill down into an incident to see the probable cause alerts, alert topology, and associated automations.

The Alert Viewer brings together alert details, alert severity, and alert filtering. Some alerts in the console are presented in alert groups, based on analytics. Using the alert list you can explore why these alerts were grouped together, and this can further help with alert resolution.

For more information about incidents and alerts, see the following topics:

- About events, alerts and incidents
- Incidents
- Alerts

About events, alerts and incidents

IBM Cloud Pak® for AIOps represents the state of managed entities using the following three concepts:

- **Events**

  An event is a record containing structured data summarizing key attributes of an occurrence on a managed entity, which might be a network resource, some part of that resource, or other key element associated with your network, services, or applications.

  An event might or might not indicate something anomalous and is a point-in-time, immutable statement about the entity in question.

- **Alerts**

  Alerts represent an ongoing anomalous condition against a single managed entity. Unlike events, alerts might evolve over time as the condition changes. Alerts have a start and an end time. The creation and evolution of alerts are informed by events.

- **Incidents**

  Incidents represent the context around an issue which is currently severely impacting operations. This includes all alerts that are related to the issue and information about how the affected resources are related. The creation and evolution of incidents are informed by alerts.

Alert creation

Alerts are created when one or more events indicate an anomalous condition. This is determined based on the **`type.eventType`** field of an event being set to **`problem`**. When such an event is received by the system, if no alert already exists for the condition, a new alert is created.

Where multiple events occur, indicative of the same condition on the same resource, the events update an existing open alert where one exists rather than creating a new one. This is determined based on events having the same values for the **`resource`** and **`type`** fields as the event that originally created the alert.

If a matching event occurs with a **`type.eventType`** set to **`resolution`**, the alert changes to the **`clear`** state. It stays in this state for 2 minutes after the last update. During this time, if another matching **`problem`** event occurs, the alert re-opens. If however, no more matching events occur, the alert enters the **`closed`** state. Once an alert is in the **`closed`** state, it can no longer be re-opened and any subsequent events create a new alert.

Note: Alerts that are closed are purged from the active store after 2 minutes.

Alert correlation

IBM Cloud Pak® for AIOps automatically correlates your alerts to determine what alerts are likely to share a common cause. This is determined based on a combination of the following mechanisms:

- **Scope-based correlation** - Any alerts which have the same value for the **`resource`** field are correlated.
- **Temporal correlation** - The system continually analyzes past alerts to determine which alerts tend to frequently co-occur. When these alerts occur together again, they are correlated.
- **Topological correlation** - Any alerts which refer to resources within the same Resource group are correlated.

For all of these mechanisms, the correlation only occurs where the last related alert was created within the past **15 minutes**. If an alert is created more than 15 minutes after the last alert, they are not correlated.

Incident creation

Incidents are created when you enable one of the two preset default incident creation policies in the Policies UI, or create your own custom incident creation policy. The preset incident creation policies are disabled by default.

Note: If you are upgrading a deployment of Cloud Pak for AIOps from version 3.2.1 or version 3.2.0, the "Default incident creation policy for high severity alerts" remains enabled by default. This is to retain the same behavior as before the upgrade.

IBM Cloud Pak® for AIOps determines which of your alerts were caused by the same problem. If a relevant incident already exists for a given problem, a new incident is not created. An existing incident is deemed relevant if it was created due to any other alert which shares a common cause with the alert in question. If a relevant incident does already exist and is currently open, the alert is added to the incident as an additional *trigger alert*. An incident's trigger alerts are defined as any alert that either caused the incident to be created, or would have caused creation had an incident not already existed. An incident takes the name of its first trigger alert.

When an incident is created, any alerts sharing the same cause as the incident's trigger alerts are automatically added as *contextual alerts*. This list is kept up to date as subsequent related alerts are detected.

In a incident overview (or a in a ChatOps notification), you can also find alerts that are ranked as *probable cause alerts* by Cloud Pak for AIOps. That is, the alerts that are most likely to be the probable cause of the incident. The distinction between *probable cause alerts* and *trigger alerts* is that the trigger alert(s) indicated that the problem is severe enough to warrant an incident (and therefore triggered its creation), but they did not necessarily cause the issue the incident represents. The probable cause alert (which could be a contextual alert) might have caused the incident, as it caused the underlying issue. Without the trigger alert(s), the incident is not created, but other parts of the issue (including the cause) might still be there.

The state of the incident is linked to the state of the incident's trigger alerts. If all trigger alerts within an incident enter the **`clear`** state, the incident enters the **`resolved`** state, and is closed 2 minutes after the last update. Likewise, an incident can become unresolved (that is, go back to in progress) if any of the triggered alerts become open again. This can happen when a new problem alert is received before the incident was closed.

Incidents

The **Incidents** tab displays information such as incident's priority, status, description, and when it was opened. From here, you can change the status or priority of an incident, and also manage the assignees. Drill down into any incident in the list to see the probable cause alerts, alert topology, and any associated automations. When you click an incident in the table, you can see more details in the side panel.

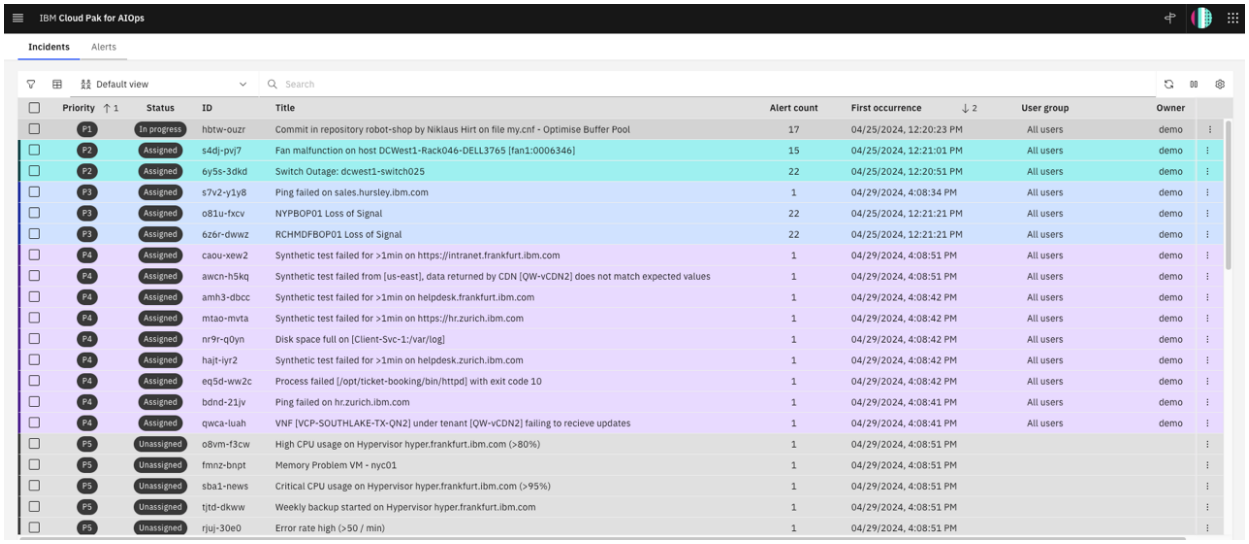


Figure. Incidents UI

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Incidents**.
    - Click the menu overflow icon of any incident (the three dots at the end of the row) to **Change priority** (from P1 - highest priority to P5 - lowest), **Change status** (Unassigned, In progress, On hold, or Resolved), **Change title**, or **Manage assignees**.
    - Click on an incident's **Title** text to open the incident's **Overview** page.

Side panel

Click in the row of an incident you are interested in to open a **side panel** on the right-hand side of the table. It shows the incident's **Priority** (Priority 1 - highest priority through to Priority 5 - lowest) and current **Status** (Unassigned, In progress, On hold, or Resolved). You can click the options menu at the top of the side panel (three vertical dots) to change the incident's status, priority, or manage the assignees.

The **Incident ID**, the linked **ServiceNow** incident number, the linked **Slack** message and Slack reactive channel name, and when the incident was **Opened** are all displayed in the side panel. Click **Investigate** to go to the incident **Overview** page.

If the incident is assigned, the side panel shows the **Group** and **Owner** on the incident. If the incident is not yet assigned, click **Assign** to assign a group or owner. To change a current assignment, click **Edit**.

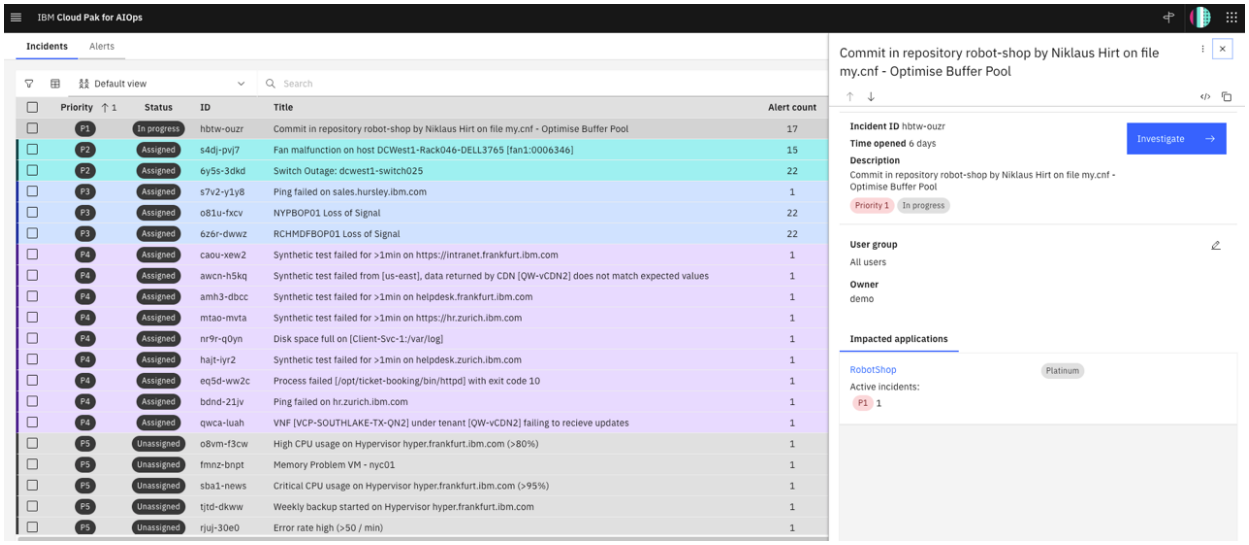


Figure. Incident Side panel

Applications that are impacted by the alerts that are part of the incident are listed under **Impacted applications**. The number of **Active incidents** that the application is impacted by is displayed. These incidents are categorized by **Priority level**. The importance of each impacted application to the business is also shown by a **Business criticality** indicator on the right of the side panel. Applications are sorted based on priority. You can click an application name to launch out to the application in **Resource management**.

The following table lists the columns present in the **Incidents** tab.

Table. Incident attributes

Column	Description
Priority	Indicates the impact level of an incident on customer. Priority levels range from **P1** (highest priority) to **P5** (lowest).
Status	An incident can have a status of **Unassigned**, **Assigned**, **On hold**, **In progress**, or **Resolved**.
Description	A description of the incident. Hover the cursor over the text in the **Description** column to display a link to the Incident overview.
Time open	Time period since the incident was created.
User group	Group assigned to the incident.
Owner	User assigned to the incident.

The following icons are used on the incident list:

Table. Incident icons

Icon	Name	Description
	Search	Click this icon to search for text in any of the columns displayed in the table.
	Filter	Click the filter icon to filter the incidents in the table based on any of the following criteria: • Priority • Status • Impacted applications • User groups • Owner • First occurrence
	Refresh	Refresh the incident list to view all the latest incidents at the current point in time.

For more information about incidents in Cloud Pak for AIOps, see the following topics:

- Incident overview

- Searching and filtering incidents
- Managing multiple incidents

Incident overview

The incident **Overview** functions as a report on the best course of action by presenting the top probable causes, as well as similar past resolution tickets, associated automations to remediate the alerts, and a host of other information.

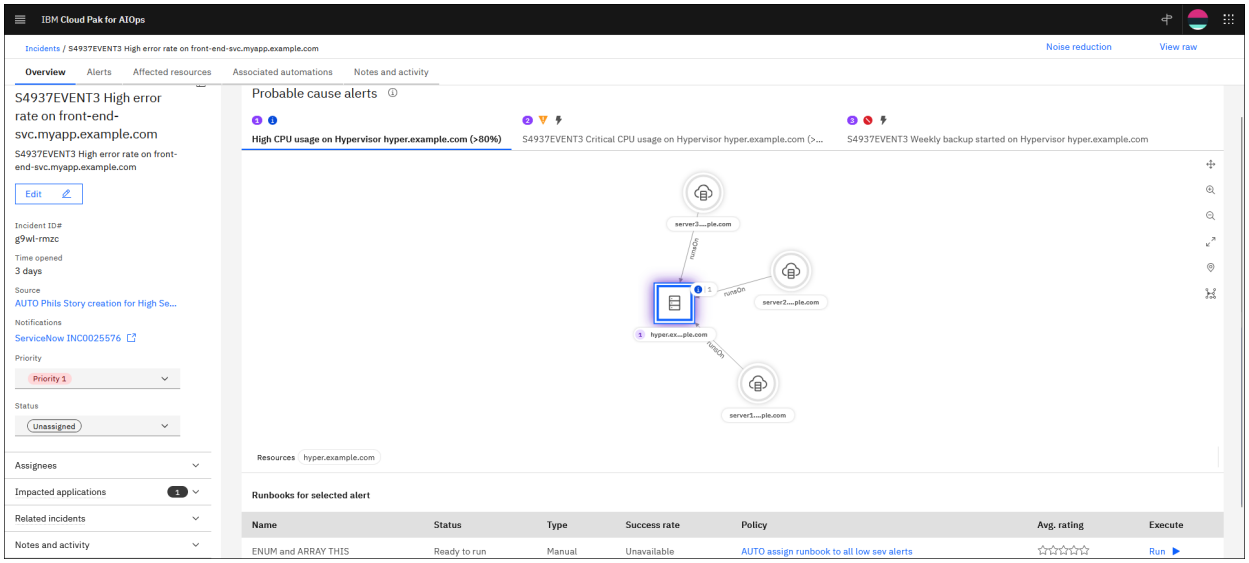


Figure. Incident overview

From a ChatOps integration (Slack, Microsoft Teams, or ServiceNow), you can launch into an incident's overview from the **incident title link** in the notification. From within Cloud Pak for AIOps, complete the following steps to navigate to an incident's overview page.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Incidents**.
3. From the **Incidents** tab, select an incident from the list and click the text in the **Title** column.

The **Noise reduction** indicator (upper right) shows how Cloud Pak for AIOps reduces the number of IT events that your operations staff must evaluate. The Noise reduction percentage shows the reduction in the number of events that are correlated and grouped into alerts that are included in each incident.

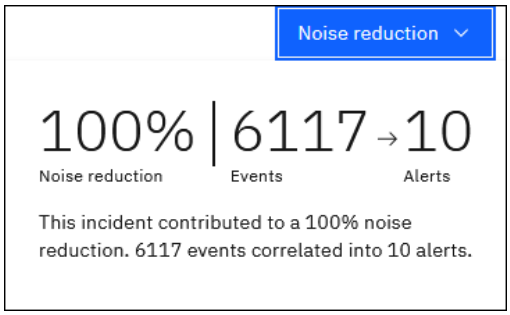


Figure. Noise reduction

Next to Noise reduction, you can click **View raw** if you want to view the incident data in raw JSON format.

The following sections are presented on the incident **Overview** page:

- Probable cause alerts
- Alert topology
- Collapsible side panel
- Probable alert automations
- Similar past resolution tickets

For more information about the **Alerts** tab and **Topology** tab, see Alert Viewer and Viewing a topology.

Probable cause alerts

Based on the topology that is associated to an incident, and the classification of the alerts, the alerts that are most likely to be the probable cause of the incident are presented here. Probable cause alerts show the top three most likely cause alerts, ranked in order of probability of being the cause of the incident.



Figure. Probable cause alerts

The following information is displayed for each probable cause alert:

- **Probable cause ranking** 1 2 3

  Probable cause assigns a ranking to all of the alerts in the incident. Alerts are ranked in order of likelihood of being the cause of the incident.

- **Severity indicator**

  Indicates the severity of the alert. The following severity icons are used:

  - Critical
  - Major
  - Minor
  - Warning
  - Informational
  - Indeterminate

- **Trigger alert**

  Denotes alerts that are defined as trigger alerts. That is, an alert that either caused the incident to be created, or would have caused creation had an incident not existed.

- **Summary**

  A description of the alert.

Note: Where probable cause cannot be identified, the section **Top Alerts** appears instead, listing the top 3 alerts for the incident ranked in order by severity, first occurrence, trigger alerts, and finally by the availability of associated automations.

Select any of the probable cause or top alerts to see topology information for that alert.

To get more details about a probable cause alert, click **View alert** below the topology diagram. This brings you to the Alert Viewer, with the relevant alert selected and the **Alert details** displayed.

Alert topology

Select a probable cause alert (or top alert) from the list. If the resource on which the alert occurred can be located in the network topology system, a one hop topology map is displayed for this alert, centered on the resource on which the alert occurred.

Restriction: For an incident topology to be generated from the alert, at least one topology group that is enabled for correlation **must** have been added to the alert.

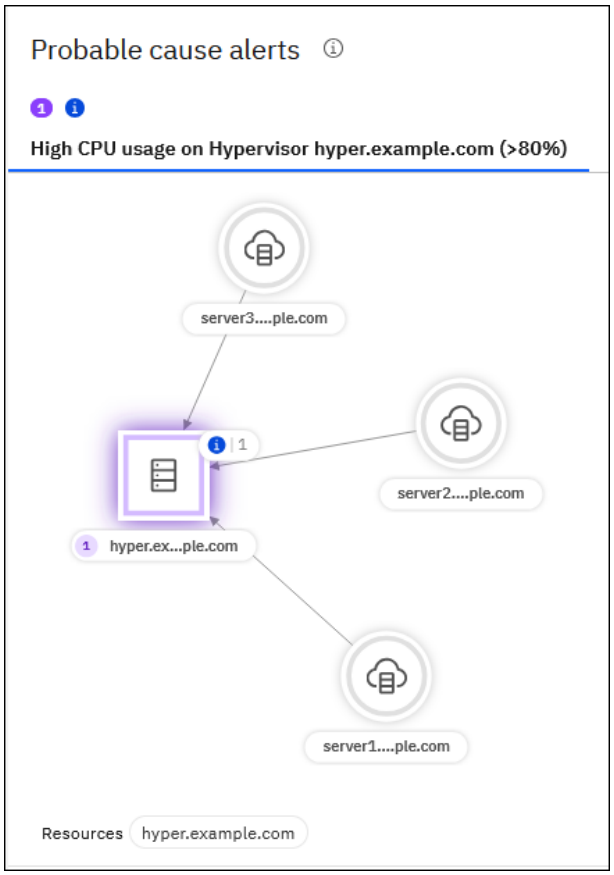


Figure. Alert topology

To investigate the incident topology, click **View topology** below the topology diagram. If the topology group condition is met, then this action opens the **Topology** tab, which provides a holistic view of the relationships between different infrastructure and application components.

For each resource associated with the incident, you can see the number of alerts, severity indicator, and probable cause ranking. This information can be used to rapidly identify the root cause and the impact of the issue on your environment before acting.

Side panel

The side panel is displayed by default on the Incident overview page. You can click **Edit** in the side panel to change the incident name or description.

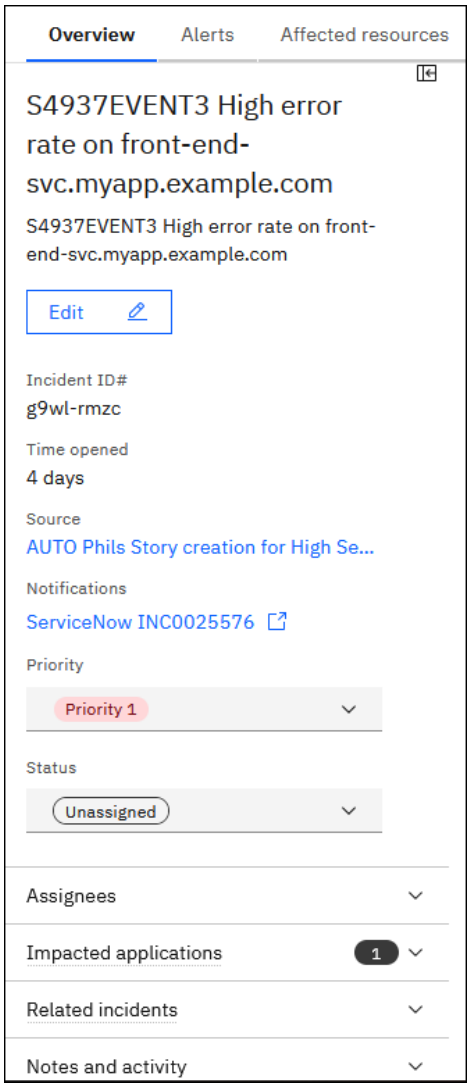


Figure. Incident Side panel

The following incident information is displayed in the Overview side panel:

- *Incident ID* - A unique number that is assigned to the incident
- *Time opened* - How long the incident has been opened for.
- *Source* - The IBM Cloud Pak for AIOps policy that created the incident. Click the link to locate that policy on the Policies UI.
- *Notifications* - When available, the linked **ServiceNow** incident number.
- *Priority* - Priority level that indicates the impact level of an incident on the customer. Priority 1 is the highest priority, through to Priority 5 the lowest. You can change an incident's priority level from the drop-down menu in the side panel.
- *Status* - Unassigned, In progress, On hold, or Resolved. You can change an incident's status from the drop-down menu in the side panel.

The overview **side panel** also contains the following sections:

- **Assignees**

  If the incident is assigned, the **Group** and **Owner** of the incident are listed under Assignees. If the incident is not yet assigned, expand **Assignees** to assign a group or owner.

- **Impacted applications**

  Applications that are impacted by the alerts that are part of the incident are listed under Impacted applications. A number indicates how many applications have been impacted. Click the arrow to expand the list of applications. The importance of each impacted application to the business is shown by a **Business criticality** indicator on the right of the side panel. The number of **Active incidents** that the application is impacted by is also displayed. These incidents are categorized by **Priority level**. Applications are sorted based on priority. You can click an application name to launch out to the application in **Resource management**. If there are more than five applications, you can click **View all** to open a modal containing the complete list. You can sort the applications in the table by clicking on each column header within the modal table. To filter the list of impacted applications, enter a search string in the **Search** box within the modal and start typing the name of the application you are looking for. As you type, the list displays only applications that contain the typed text. As with the applications displayed in the side panel, clicking an application's name opens that application.

- **Related incidents**

Any incidents that share contextual alerts are listed in the **Related incidents** section of the side panel. Related incidents can provide more context when you are resolving an incident. A number indicates how many related incidents have been found. Click the arrow to expand the list of incidents. The priority and status of each incident is displayed, and the number of associated alerts. Related incidents are sorted based on priority. You can click an incident's title text to go to the incident's **Overview** page. If there are more than five related incidents, you can click **View all** to open a modal containing a complete list. You can sort the incidents in the table by priority, status, and alert count. To filter the list of incidents, enter a search string in the **Search** box within the modal and start typing the name of the incident you are looking for. As you type, the list displays only incidents that contain the typed text. As with the incidents displayed in the side panel, clicking the title text opens that incident's **Overview** page.

- **Notes and activity**

  A timeline of major events that have occurred in the lifecycle of an incident. Click View all notes and activity to go to the **Notes and activity** tab.

  To collapse the side panel, click the arrow at the top of the panel beside the incident name.

Probable alert automations

The overview page for an incident displays the top three probable alerts for an incident. Runbooks that are associated with those alerts are visible in a table below the selected alert's topology. Select an alert to display the associated runbooks.

Similar past resolution tickets

It can be helpful to review the details of similar past resolution tickets to help determine the best course of action. Click **Add tickets to incident** button to open the window.

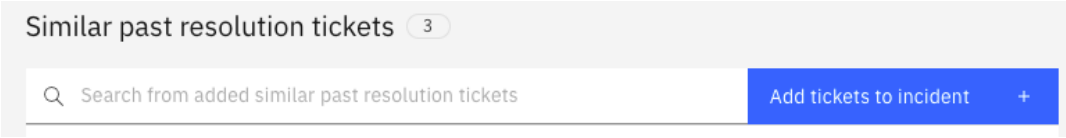


Figure. Similar past resolution tickets window

When the *Add tickets to incident* window opens you can search, select tickets from the past, and click **Add to incident** button to add the selected tickets to this incident. The following diagram shows the capability to add tickets to an incident.

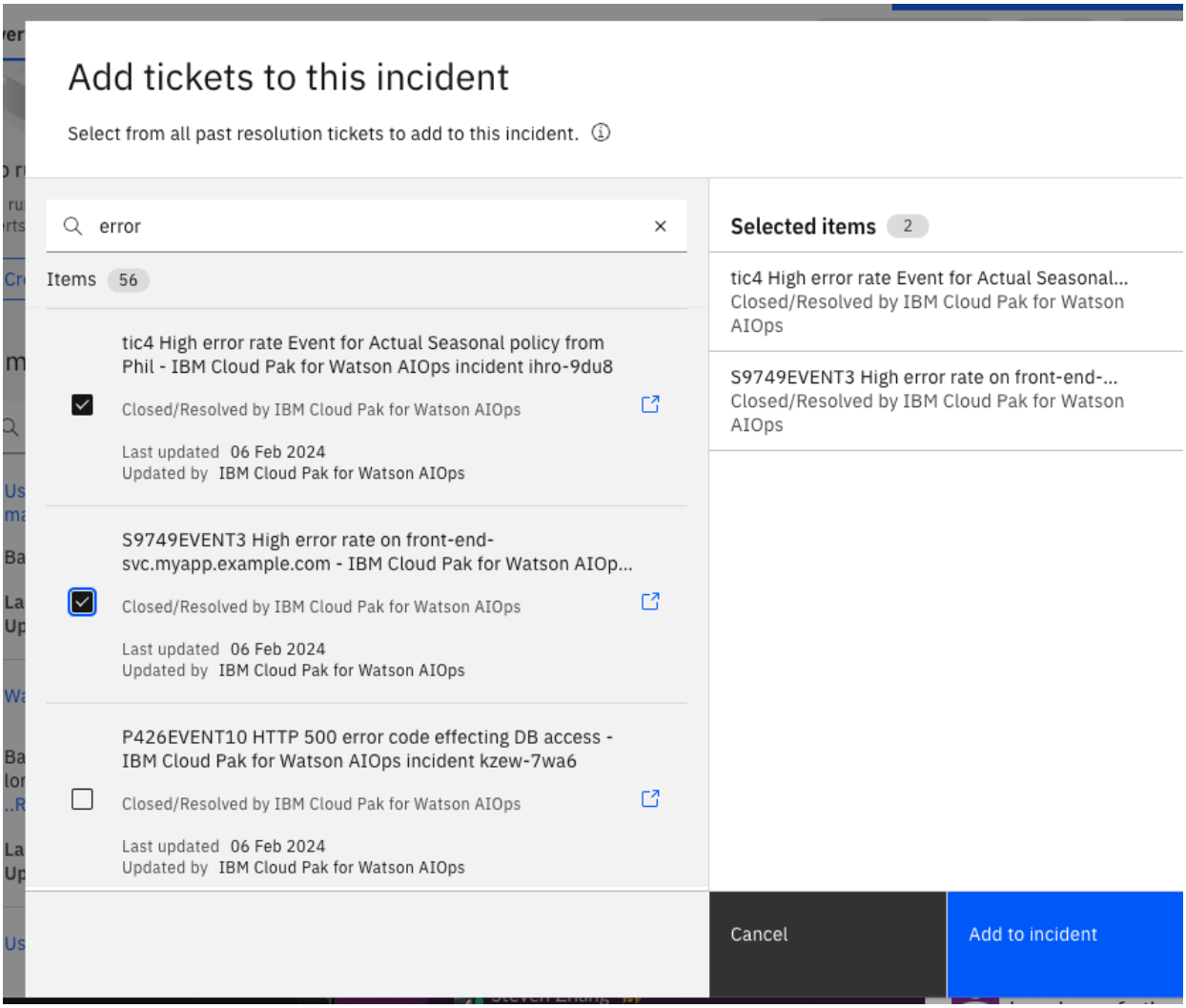


Figure. Add tickets to the incident by using the Add to incident button

The following prerequisites must be met for the data to show in the ticketing insights window.

1. Setup ticketing integrations. For more information about Integrations, see Integrations.
2. Run Similar tickets training. For more information about setting up Similar tickets training, see Setting up training for similar tickets.
3. Enable the *default story query similar incidents service* policy. To enable this policy, go to **Automations** > **Policies**. Set the state of this policy to *enabled*.
4. A minimum of five qualifying tickets are needed. **Qualified tickets are closed**. For more information about the required attributes for similar tickets, see Qualifying tickets.

Insights are used to display any similar past resolution tickets that are associated with the incident, together with a description of each ticket. You can remove the past resolution tickets from an incident. You can click the **dismiss (-)** button to remove the past resolution ticket. The following diagram shows the ability to remove tickets from an incident.

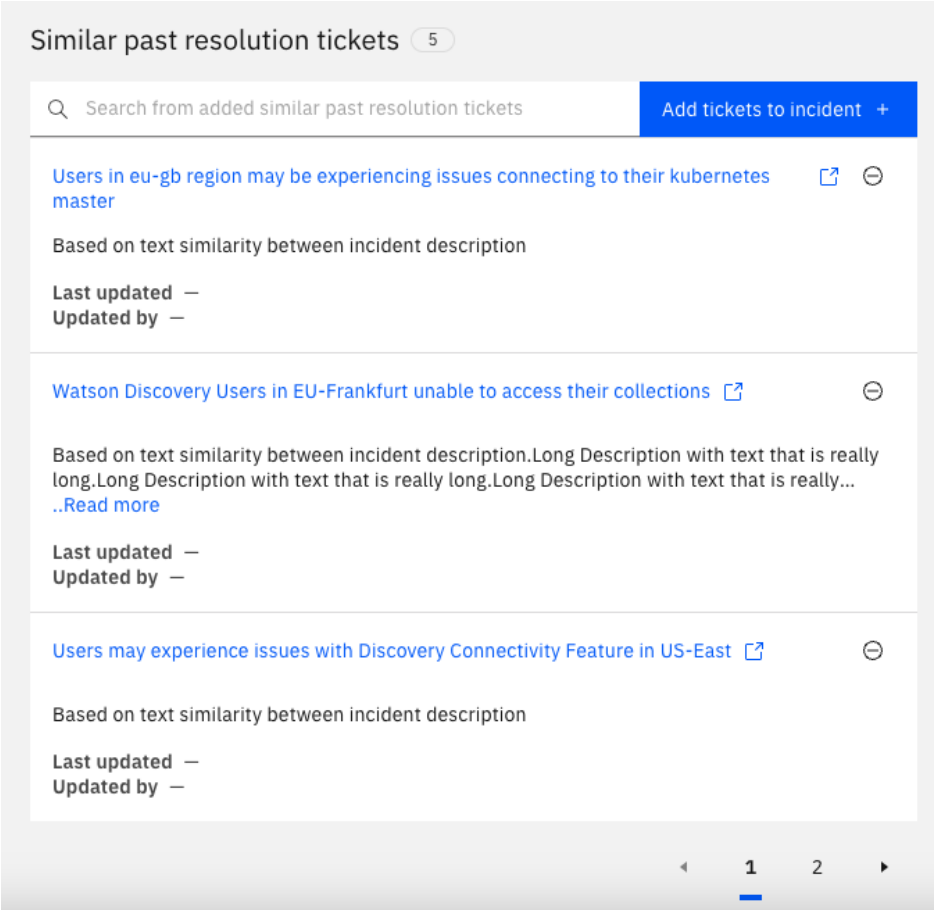


Figure. Remove tickets to the incident by using the dismiss button

Click the title of a past-resolution ticket to open the linked ticket. For example, the ticket can be a ServiceNow ticket.

You can navigate to the following **tabs** from the incident overview:

Alerts

Click the **Alerts** tab for a list of all the alerts that are associated to the incident.

Affected resources

Shows the relationships of impacted resources. Note, this view is only populated when an incident has an associated topology.

Associated automations

See an aggregate of automations associated with alerts within the incident. For a given automation, the table displays various details and you can navigate to the associated policy, alert, or run the automation.

The following information is displayed for each recommended runbook:

- **Status**

  Successfully executed, Failed, Cancelled, In progress, or Completed.

- **Type**

  Indicates the Runbook type: Manual or Automated.

- **Success rate**

  The success rate is calculated by using the number of successful and unsuccessful executions of the runbook.

- **Policy**

  Shows the Cloud Pak for AIOps policy that assigned the runbook to the selected alert. Click the policy name to locate that policy on the Policies UI.

- **Average rating**

  Operations analysts can provide feedback about the quality of a runbook including a rating, which is displayed here (five stars is the top rating).

- **Associated alert**

  Click to view the alert that is associated to the runbook in the Alert Viewer.

- **Execute**

  Click **Run** to run the runbook.

Notes and activity

A timeline of major events that have occurred in the lifecycle of an incident. The following events are logged to the Activity timeline:

- When an incident is created.
- Incident is assigned to new owner or team.
- An alert has been associated with the incident.
- Incident state changes. For example, the incident has been set to **On hold** from **In progress**, or marked as **Resolved**.
- Changes to the incident title or description.
- User added comment.

To add a comment to the incident timeline, type a comment in the field that is provided and click **+** Add comment. A maximum of 250 characters are supported. The comment is stored in the timeline in chronological order with the other entries.

For more information, see the following pages:

- About events, alerts and incidents
- Viewing a topology
- Runbooks
- Policies

Searching and filtering incidents

Search for incidents by name, or filter the list based on selected critria.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Incidents**.

- To search for an incident in the table:
  - Click anywhere in the **Search** field.
  - Start typing a description of the incident in the search field. For example, type "failed" and the system will show in the table all incidents that contain the text "failed" in the description.
- To filter the list of incidents in the table:
  - Click **Filter** to open the filter side panel.
  - Select from the following filters for incidents:
    - Priority: P1 - highest priority, through to P5 - lowest.
    - Status: Unassigned, Assigned, In progress, On hold, or Resolved.
    - User groups: Find user groups to filter by.
    - Owner: Find owners to filter by.
    - Time open: under 1 hour, 1 hour to 1 day, 1 day to 7 days, 7 days to 15 days, 15 days to 30 days, 30 days and above.
  - Click **Done** to close the Filter window.

3. To clear all filters, click **Filter** and then **Clear all**.

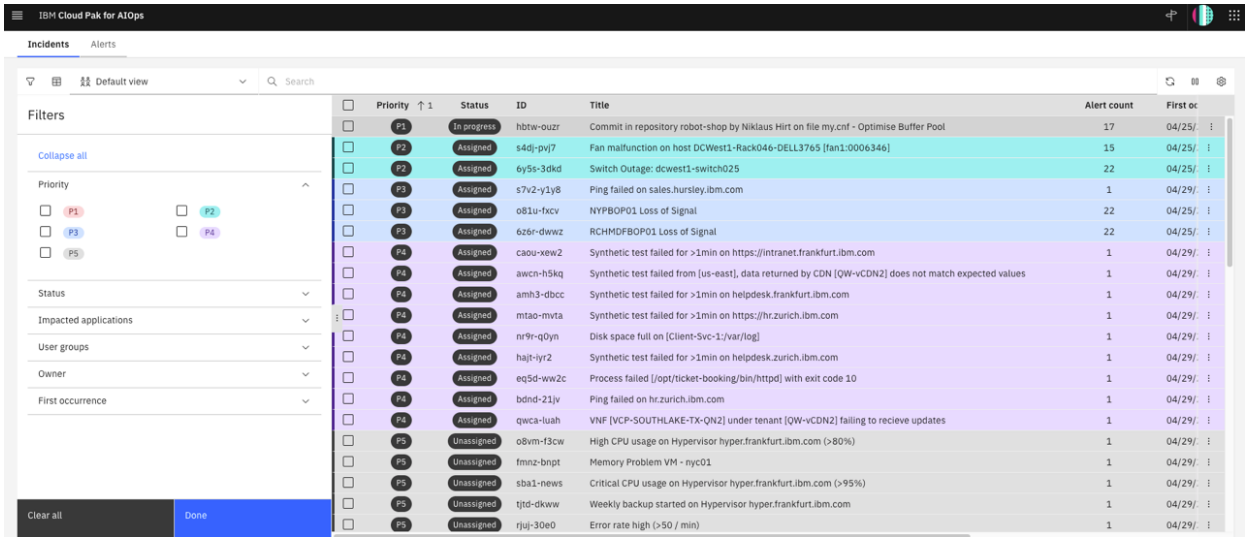


Figure. Search and filter incidents

To save a new filter

1. Apply the filters preferences you want and then Click **Save as a new filter**.

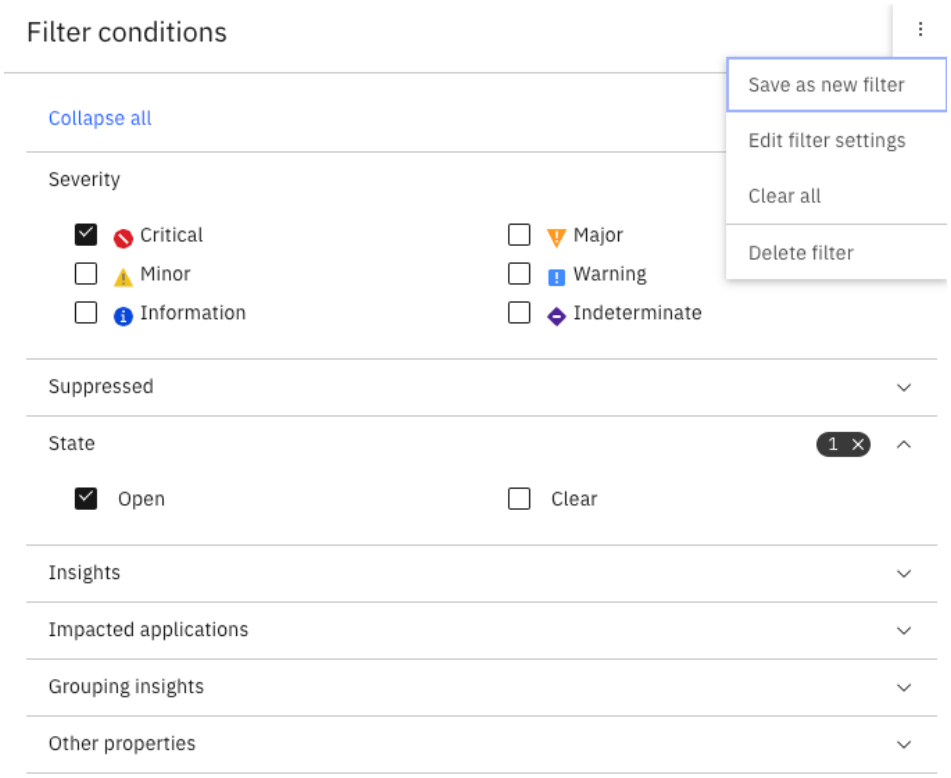


Figure. Save as new filter dialog

Note:: Alternatively, if you are an administrator who has *manage profiles* priviledges had enabled, you can set Role Based Access Control (RBAC) functionality that allows you see and click the **Save as a new restriction filter** option, thus applying certain restriction filters, incidents in this case.

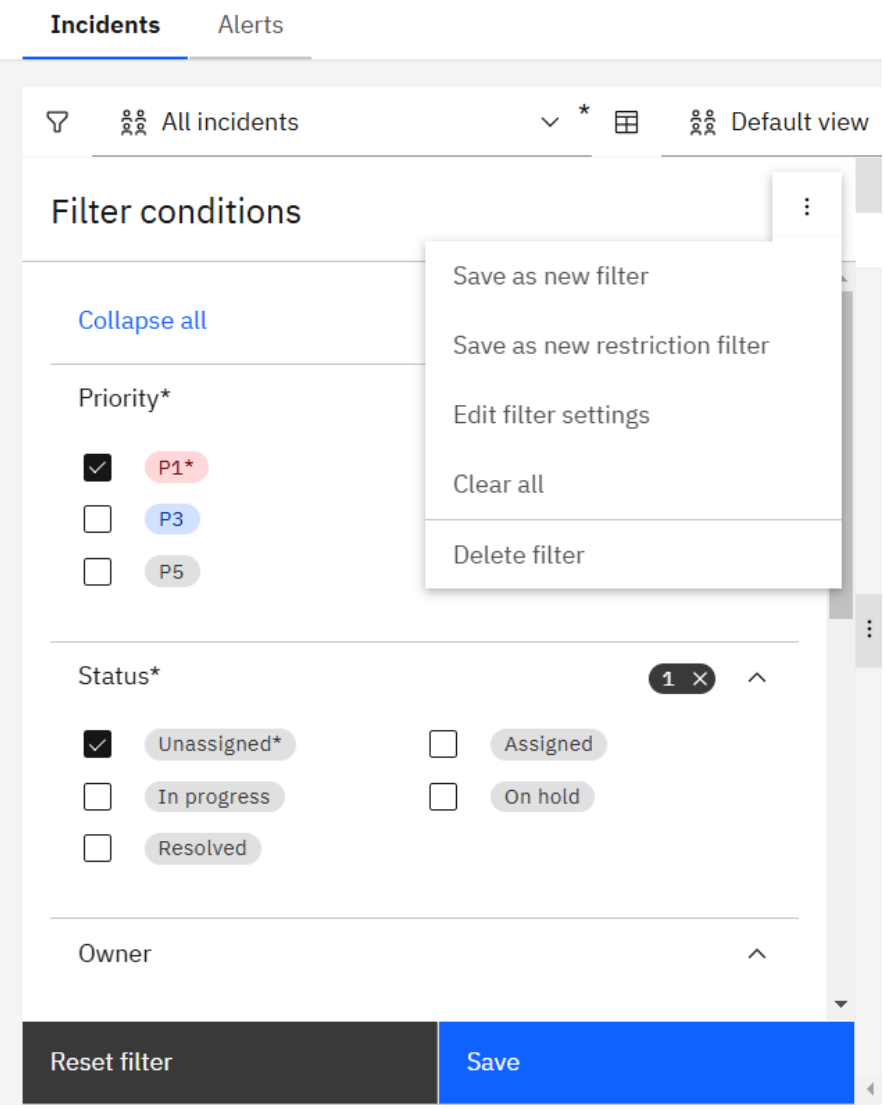


Figure. Save as new restriction filter

The *Save a new restriction filter* dialog that opens is slightly different to the regular *Save a new filter dialog* window also in terms of fields to fill in, as shown below.

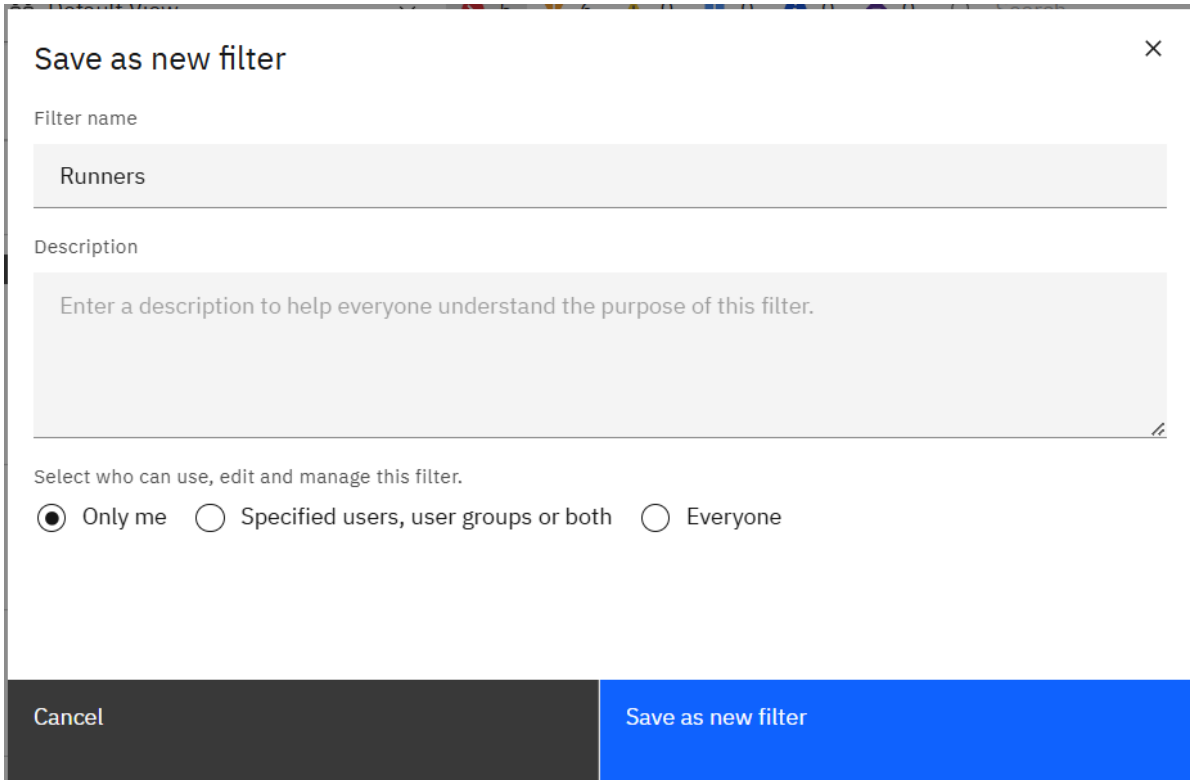


Figure. Incidents save a new filter

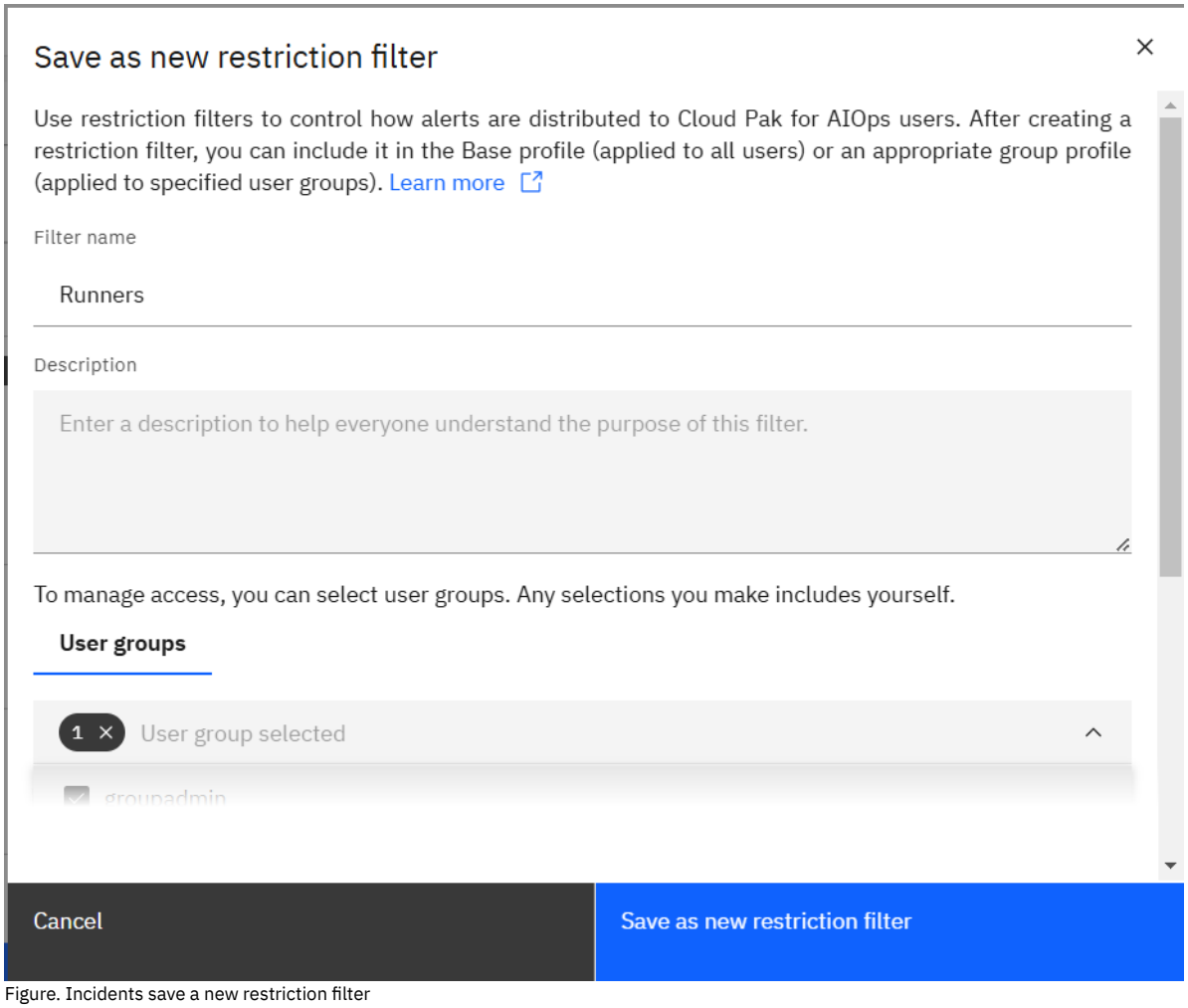


Figure. Incidents save a new restriction filter

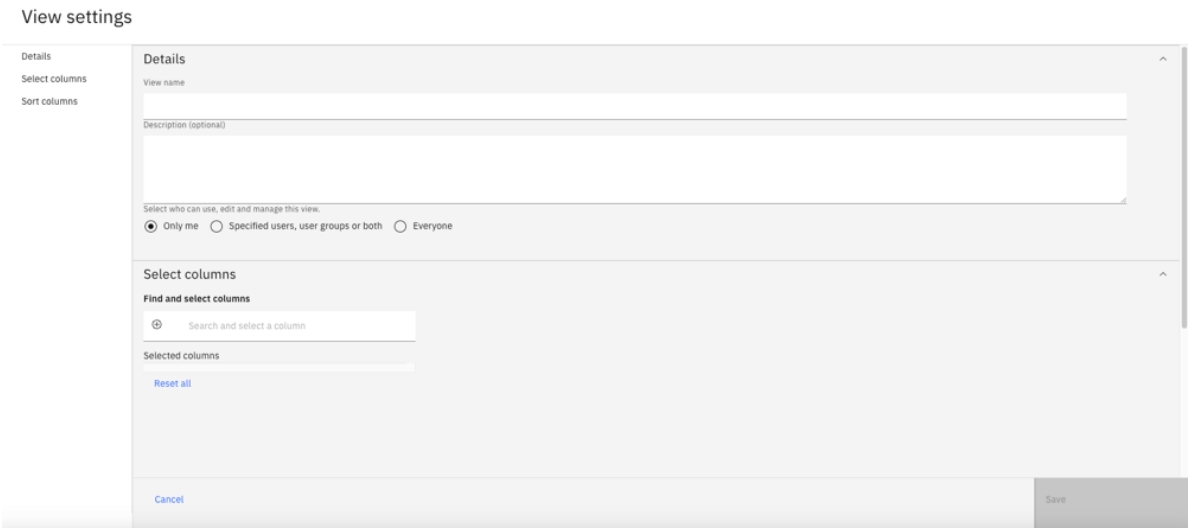
Note: A Role Based Access Control (RBAC) restriction filter can be applied to a user that filters the data they see, but they can additionally apply their own filter conditions on top of that RBAC filter.

Creating views

Create custom views that are dynamically applied to the Incident list. The views determine what information is displayed from the available incident data.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Incidents**.
3. On the toolbar, click **View configuration** > **Create a new view**.
4. Configure the following properties for a custom view:

- **Details**
  - View name
  - Description
  - Select who can use, edit, and manage this view:
    - *Only me*
    - *Specified users, user groups, or both*
    - *Everyone*



View settings - Details

- **Select columns**

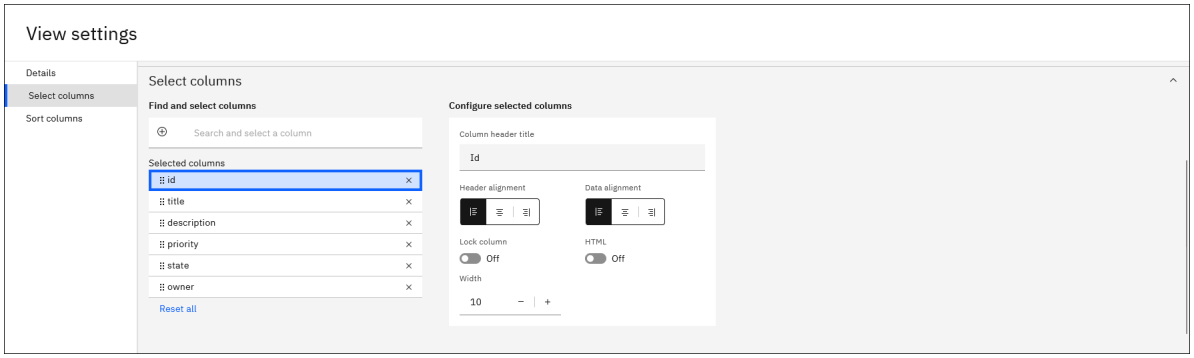
  Select from incident attributes that are predefined for Cloud Pak for AIOps and common to most incidents. Click anywhere in the field provided to display the variable catalog. Move the cursor over the attributes to display the descriptions. You can also type the name of the property that you are looking for and the system will show in the drop-down list all incident properties that contain the typed text. You can drag and drop the columns to determine the order of the columns appearing in the Incident list.

  Important: Select at least one column for each view.

  - **Configure selected columns**

    Configure the following properties to customize the columns that display incident data:

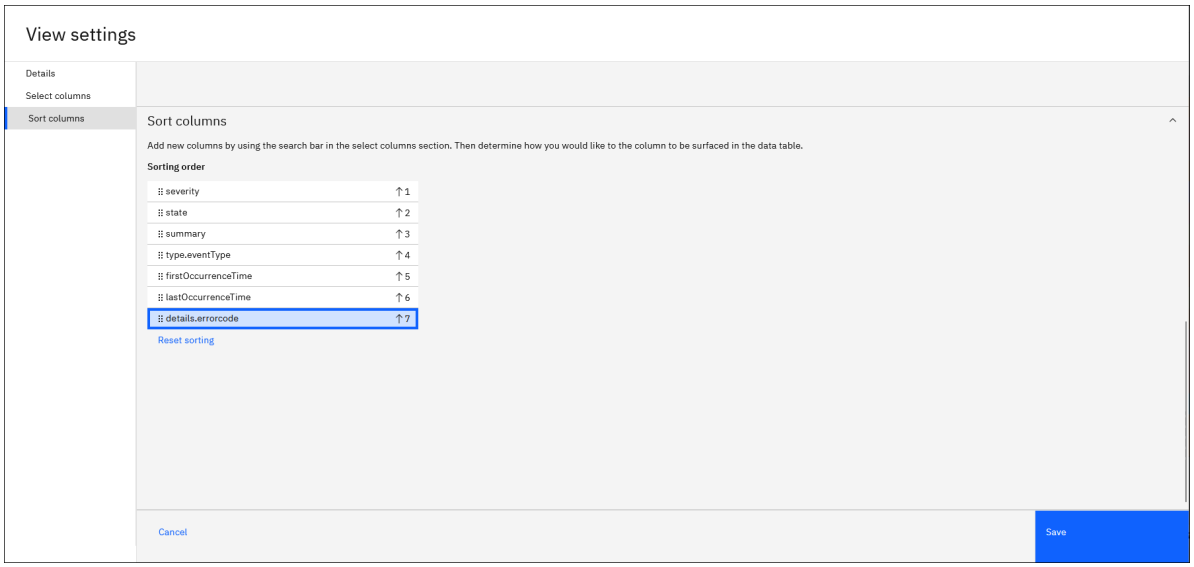
    - *Column header title*: type a new title for the column or keep the default.
    - *Header alignment*: set the position of the title text in the column heading.
    - *Data alignment*: set the position of the incident data in the columns.
    - *Lock column*: locked columns are always displayed when you scroll an incident list horizontally. Locked columns appear at the left side of the view.
    - *HTML*: enable clickable links in the Incident list cells.
    - *Width*: type a numeric value in the Width field to change the width of a column.



View settings - Select columns

- **Sort columns**

  Add new columns in the Select columns section. Then, in the Sort columns list, single click on a column item to set ascending order. Click the column twice to sort in descending order. Drag and drop columns to prioritise sorting across columns.

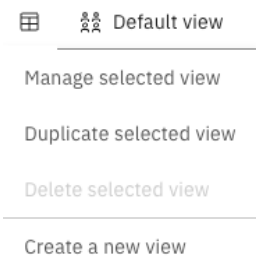


View settings - Sort columns

What to do next

- **Manage selected view**

  1. From the view list on the incident page, select the view that you want to edit. The incident list updates with the view properties.
  2. Click **View configuration** > **Manage selected view**.



View configuration

  3. Edit the **View settings** as required.
  4. Click **Save** to save the view.

- **Duplicate selected view**

  1. From the view list on the incident page, select the view that you want to copy. The incident list updates with the view properties.
  2. Click **View configuration** > **Duplicate selected view**.

     A copy of the view is displayed. By default, the view name is set to "Copy of *view name*."

  3. Edit the **View settings** as required.

4. Click **Save** to save the view.

- **Delete selected view**

  1. From the view list on the incident page, select the view that you want to delete. The incident list updates with the view properties.
  2. Click **View configuration** > **Delete selected view**. This action cannot be undone. Once the view has been deleted, your view is updated based on the first view in the list.
  3. Click **Delete view** to confirm deletion.

Setting user preferences for the Incidents list

You can configure the appearance and behavior of the Incidents list via the **Settings**.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Incidents**.
3. On the upper right corner of the page, click **Settings**.

   The following preferences can be applied to the **Incidents** list:

   - **Row size**

     Select from the following table row sizes:

     - Very small
     - Small
     - Medium
     - Large
   - **Incident details launch type**

     Selecting an incident in the table opens the incident details for that incident. These can be displayed in a **Side panel** or in a separate **Pop-up window**.
   - **Incident details should open on:**

     - Double-click
     - Single-click
   - **Incident activity entries sorting**

     - Newest at the top
     - Newest at the bottom
   - **Row coloring**

     Change how incident priority is depicted in the table by applying a different color background for each priority level.
   - **Table auto-refresh**

     Enable auto-refresh to have the incident list automatically updated at predescribed intervals.
   - **Table auto-refresh interval (s)**

     Auto-refresh interval in seconds.
4. Click **Save** to apply your changes to the incident list.

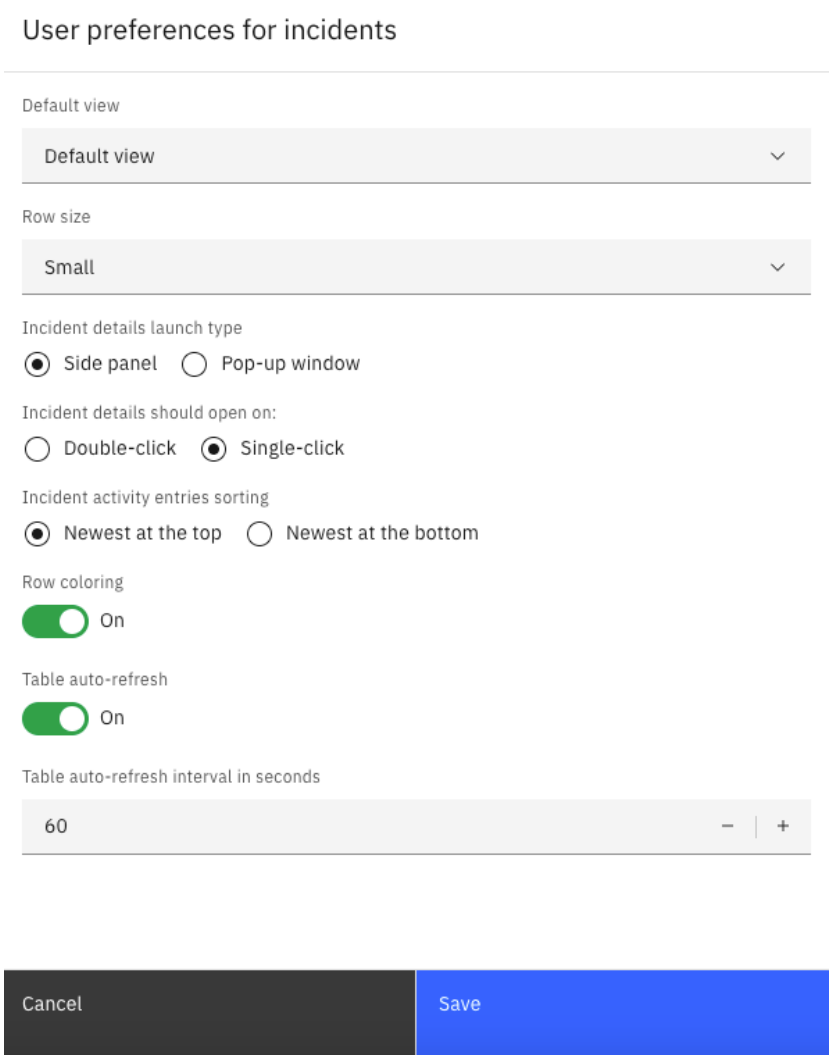


Figure. Incident preferences

Managing multiple incidents

You can multi-select incidents and apply actions to all of your selections at one time.

Procedure

1. Select multiple incidents from the table by using the checkboxes at the beginning of the table rows.

   Tip: You can select all of the incidents in the table by clicking the checkbox in the table header beside **Priority**.

2. A banner above the table displays the number of incidents that are selected, together with a menu of options. The following actions can be applied to multiple incidents at the same time:

   - Change priority: select from a range of **P1** (highest priority) to **P5** (lowest).
   - Change status: to **In progress**, **On hold**, or **Resolved**.
   - Manage assignees: assign a **Group**, **Owner**, and **Collaborators** to the incidents.

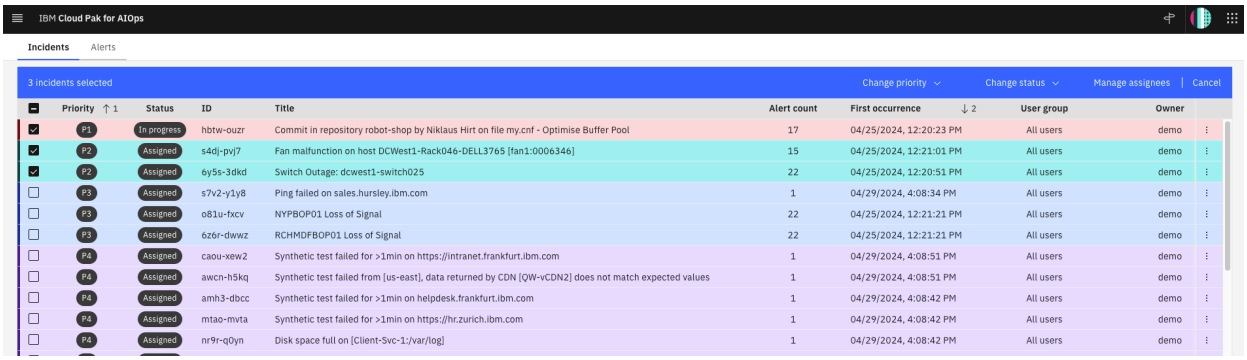


Figure. Managing multiple incidents

Alert Viewer

Alerts are displayed in the Alert Viewer. From a ChatOps integration (Slack or Microsoft Teams), you can launch into an Alert Viewer that shows the alerts that are associated to a specific incident. The numbers on the columns represent sort order. By default, alerts are sorted in the table by **Severity**. Where severity is the same, alerts are sorted by **First occurrence**. Each increasing sort order only sorts equivalent values. So once all values in a column sort order are unique (not equivalent), subsequent sort orders are ignored.


Figure. Alert viewer

There is also an alert list that displays all of the alerts found by IBM Cloud Pak® for AIOps. Follow the steps below to navigate to the main Alert Viewer.

Viewing alerts

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.

The following table lists alert attributes.

Table. Alert attributes

Attribute	Description
Severity	Displays the severity of the alert. The possible values that can display in the Severity column are 6: Critical, 5: Major, 4: Minor, 3: Warning, 2: Informational, 1: Indeterminate.
Business criticality	Set in **Resource management** to define the importance of an application, resource group, or resource to the business.
State	An alert has one of the following three states: **Open** - Always has a severity other than `clear` to indicate an active issue that can require your attention. **Clear** - Working as expected, also has a severity of `clear`. **Closed** - No longer active or relevant, a deleted event. Also has a severity of `clear`.
Ranking	Probable cause assigns a ranking to all of the alerts in the incident. Alerts are ranked in order of likelihood of being the cause of the incident.
Summary	Provides more detail about the alert. For possible log anomaly summaries, see Log anomaly summaries.
Event Type	Describes the type of alert, for example, Utilization, Anomaly, System status, or Threshold breach. For possible log anomaly alert types, see Log anomaly alert types.
Sender	Identifies what, such as a log anomaly, that sent the alert. Different senders can result in different alert attributes.
Resource	The source from which the alert originated. This attribute can be a device name or hostname, service name, or application.
First occurrence	Displays the date and time in which the alert first occurred. The date format shown is based on your browser locale settings. Refer to the browser help for instructions on display format for dates, times, and numbers.
Last occurrence	Displays the date and time in which the alert last occurred. The date format shown is based on your browser locale settings. Refer to the browser help for instructions on display format for dates, times, and numbers.
Runbooks	A number in this column indicates the number of runbooks that are associated to an alert. Click the number to display the runbook details in the side panel. From here, you can see more information, preview, and run the runbook.

Attribute	Description
Topology	If the resource on which an alert occurred can be located in the network topology system, a large dot ● is presented in the alert's Topology column. Click the ● to display a topology map for this alert, centered on the resource on which the alert occurred.
Seasonal	Alerts that occur within a seasonal time window. Alerts that are not part of an alert group, that have associated seasonality, have a circle ○ visible in this column. Alerts that are part of an alert group, that have associated seasonality, have a dot ● in this column. Click the circle or dot to display more information in the **Seasonality** section of the **Alert details** panel.
In incident	Indicates if the alert is associated to an incident.
Trigger *	Denotes alerts that are defined as trigger alerts. That is, an alert that either caused the incident to be created, or would have caused creation had an incident not already existed. An incident takes the name of its trigger alert.
Suppressed	A 'Yes' in this column indicates that the `alert.suppressed` flag is set to `true` for the alert. The alert is still present in the system and can be viewed in the Alert Viewer, but the alert is filtered out of the view by default.
Golden signal	A label based on the value of `alert.type.classification` that describes the type of the event. For example, utilization, system status, threshold breach, and so on. Golden signal labels are divided into the categories of **Effect**: latency, error, availability. **Cause**: saturation, traffic. **None**: information.

*Only displayed on alert lists within the context of an incident.

Click the **Correlation information** icon to enable the Correlation column and display three more icons under this column. If the alert is part of an alert group, the type of analysis that is used to generate the group is indicated in these columns.

Table. Alert groups

Correlation column	Description
Temporal group	Details of a temporal group in which this alert is involved.
Scope-based group	Details of a scope-based group in which this alert is involved.
Topological group	Details of a topology group in which this alert is involved.

For more information, see Displaying analytics details for an alert group.

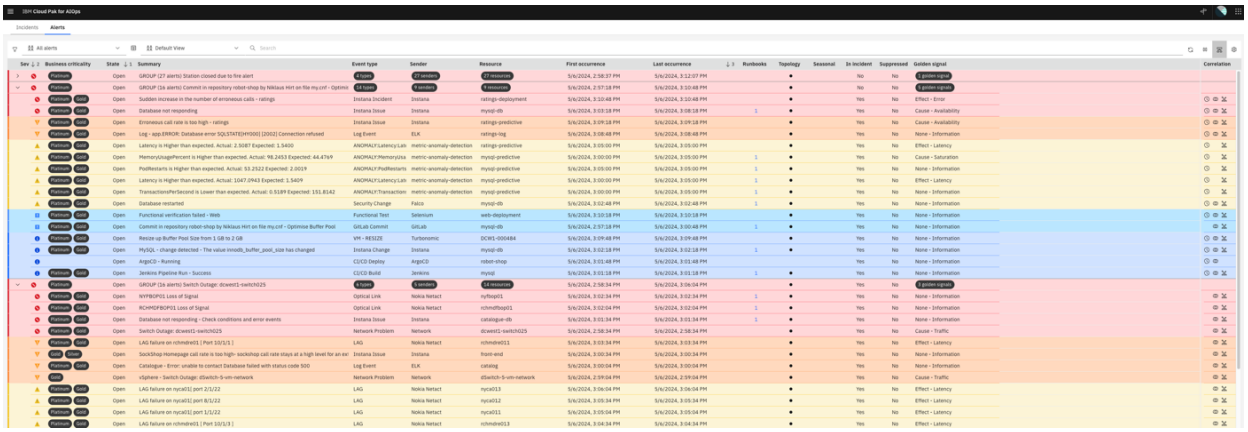


Figure. Correlation information

To view more detailed information about the alert list count, click the information icon at the bottom of the table.

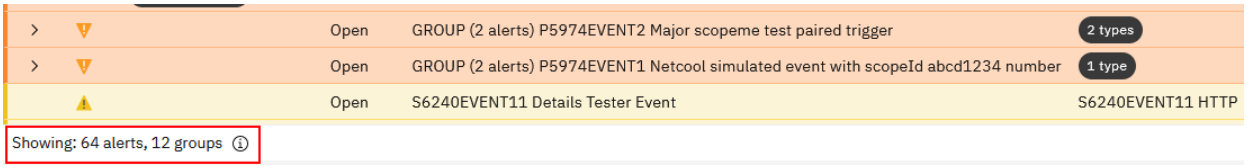


Figure. Alert count

The following alert list information is displayed:

- **Showing** (alerts): The number of alerts that are shown in the alert list.
- **Showing** (groups): The number of alert groups that are shown in the alert list.
- **Grouped**: The number and percentage of alerts that are part of a group shown in the alert list.
- **Limit**: The limit for the number of alerts you can view in the alert list at one time. This value is set by an administrator.
- **Total matched**: The number of alerts identified in the system that match the current saved filter. This can be higher than the number shown when there is a limit set to improve performance.
- **Total active**: The total number of alerts in the system that haven't been archived. Historical alerts are archived for AI training.

Alert anomaly detection

IBM Cloud Pak for AIOps analyzes alert counts over time to learn about the normal pattern of behavior to detect situations such as alert storms. If an anomalous alert rate is detected, an alert is raised and displayed as a metric anomaly in the Alert Viewer. When you click this alert, the Alert details panel depicts the expected values against the actual values.

Icons used in the Alert Viewer

Use this information to understand the purpose of the different icons used in the Alert Viewer.

Table. Icons used in the Alert Viewer

Icon	Name	Description
	Critical alert	A service-affecting condition has occurred, and corrective action is immediately required. For example, a system or device has gone out of service and needs to be restored.
	Major alert	A service-affecting problem occurred. Corrective action is urgently required. For example, a severe degradation occurred in the capability of a system or device and full capability must be restored.
	Minor alert	A non-service affecting problem occurred; take corrective action to prevent alerts of higher severity. For example, a problem occurred on a system or device but it does not impair the capacity or performance of the device.
	Warning alert	Potential or impending problems were detected. Further investigation is needed to prevent alerts of higher severity.
	Information alert	Alerts that are for information purposes only. Further investigation might be needed.
	Indeterminate alert	The severity level cannot be determined.
	Search	Click this icon to search for text in any of the columns displayed in the table.
	Correlation information *	Click this icon to display alert grouping information.
	Filter	Click this icon to filter the alerts in the table based on one of the following criteria: severity, enrichment, grouping, state, or alert type.
	Refresh	Refresh the Alert Viewer to view all the latest alerts at the current point in time.
	Temporal *	Based on alert history, these alerts tend to occur within a short time of each other.
	Scope-based *	Scope-based alerts occur within a configurable time window on an administrator-defined scope, such as a location, service, or resource.
	Topological *	Topological alerts occur on resources within a predefined section of your network topology.
	Trigger alert *	Denotes alerts that are defined as trigger alerts. That is, an alert that either caused the incident to be created, or would have caused creation had an incident not existed.
	Seasonal alert	Denotes a seasonal occurrence of an alert.
	Seasonal alert	Denotes a non-seasonal occurrence of a seasonal alert.

*Displayed only on alert lists associated to an incident.

Alert details

The **Alert details** side panel is accessed in the Alert Viewer. From here, you can quickly clear alerts or access additional information about the alert properties.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. For the complete list of Cloud Pak for AIOps alerts, click **Operate** > **Alerts**.

   For the alerts associated to a specific incident, click **Operate** > **Incidents** > ***Incident Title*** > **Alerts**.

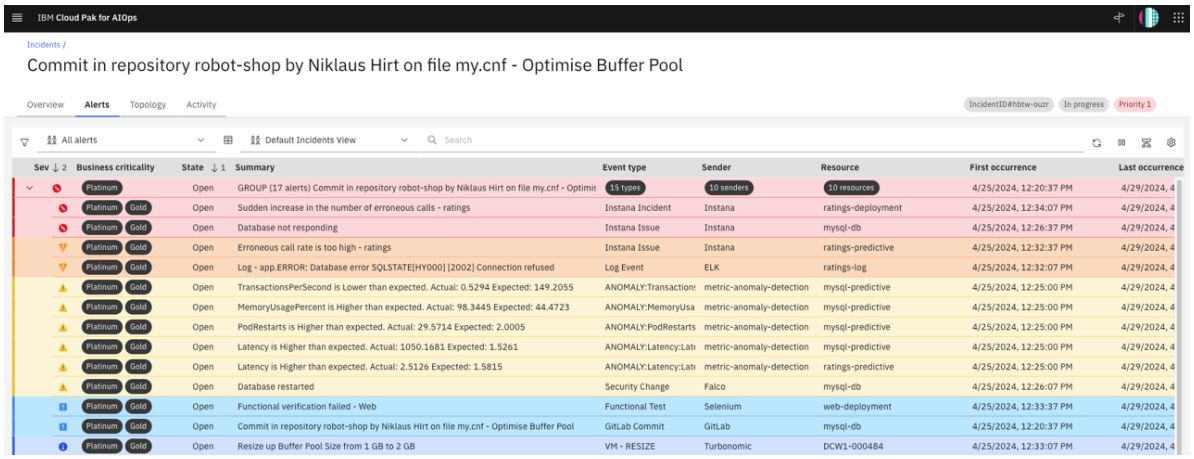


Figure. Incident alerts

3. Double click an alert in the table to display the **Alert details** side panel for that alert. The **Information** section is open by default in the side panel.

   Tip: To clear the selected alert, scroll to the end of the side panel and click **Clear alert** and then click **Clear** to confirm.

To display **Alert details** in a new window, click the pop-up icon in the side panel.

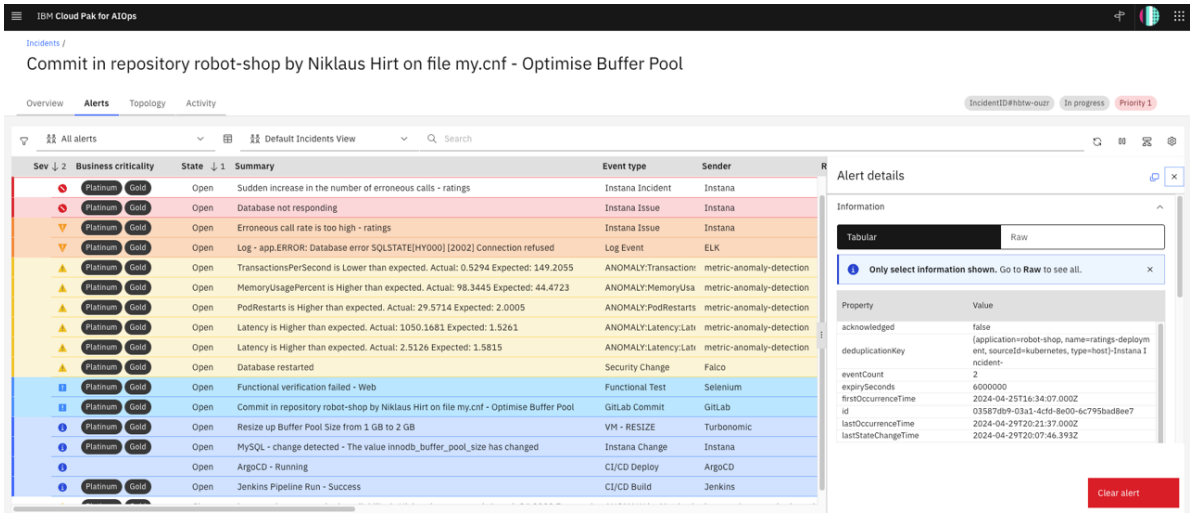


Figure. Alert side panel

Depending on the alert selected, **Alert details** can contain the following sections:

- Information
- Activity
- Incidents
- Anomaly insights
- Runbooks
- Topology
- Log anomaly
- Metric anomaly
- Recommended actions
- Correlation information

Information

Under Information, alert properties can be viewed in two different formats:

- **Tabular**

  Alert properties are displayed in a table format of **Property** - **Value** pairs.

- **Raw**

  Raw data in a JSON format. Click **Copy** to copy the alert data to the clipboard.

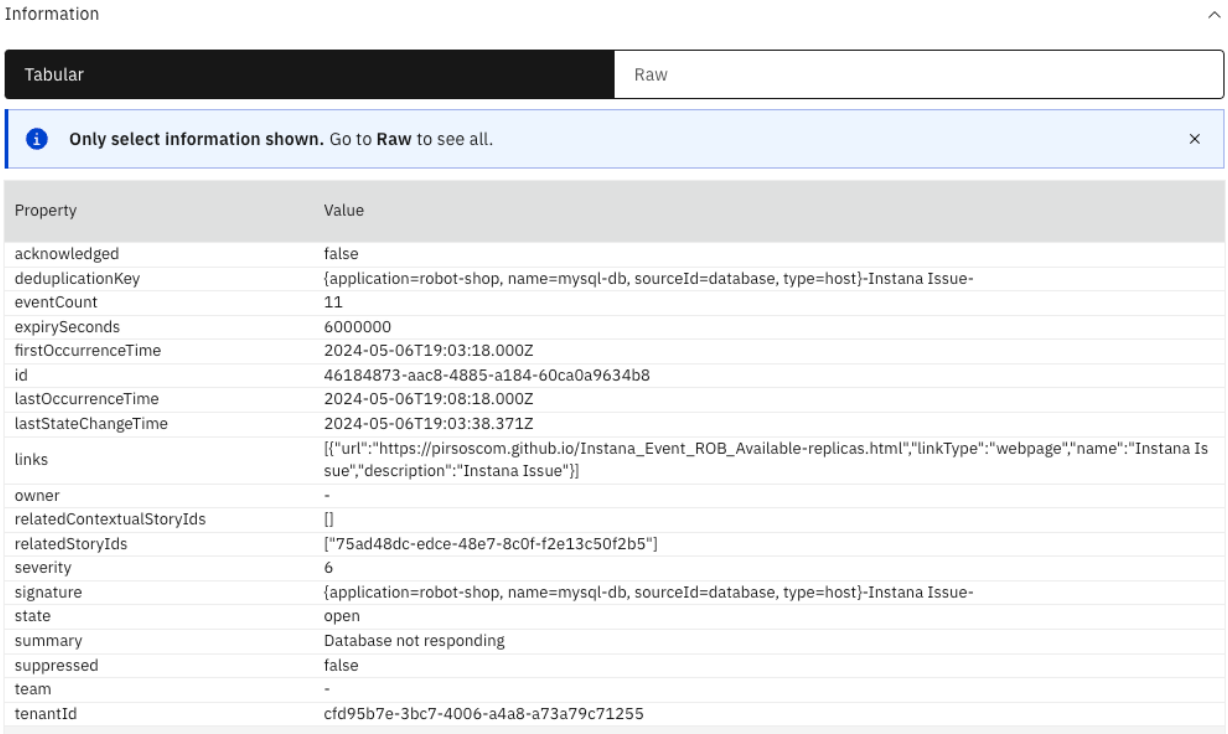


Figure. Alert information

Activity

A timeline of major events that have occurred in the lifecycle of an alert. The following events are logged to the Activity timeline:

- When an alert is created
- Alert is assigned to new owner or team
- Alert is acknowledged/de-acknowledged
- Alert is suppressed/de-suppressed
- Alert state changes (open or clear)
- Alert severity changes
- User added comment

To add a comment to an alert at any time, type a comment in the field that is provided and click **+** Add comment. A maximum of 250 characters are supported. The comment is stored in the timeline in chronological order with the other entries.



Figure. Alert activity

Incidents

If you are viewing an incident's alert list, the Alert details side panel displays a link to the Incident overview. You can also see the incident's priority, status, and the number of alerts in the incident.

Anomaly insights

In the Alerts table, if you select an alert that contains anomaly insights, the Alert details panel contains an Anomaly insights section. This section shows Property, which is the Key Performance Indicator (KPI) type, and Value, which is the name, for each anomaly insight. Use this information to understand the anomaly and why an alert was generated. You can also use this information to create a policy. For example, you can suppress the alert or associate it with a runbook. For more information, see Supported Anomaly insights properties.

Runbooks

The following information is displayed for each runbook in **Alert details**:

- **Status**

  Successfully executed, Failed, Cancelled, In progress, or Completed.

- **Success rate**

  The success rate is calculated by using the number of successful and unsuccessful executions of the runbook.

- **Average rating**

  Operations analysts can provide feedback about the quality of a runbook including a rating, which is displayed here (five stars is the top rating).

- **Type**

  Indicates the Runbook type: Manual or Automated.

- **Policy**

  Shows the Cloud Pak for AIOps policy that assigned the runbook to the selected alert. Click the policy name to locate that policy on the Policies UI.

Click the menu overflow icon (the three dots) and click **Preview** to view the runbook or click **Run** to re-run the runbook.

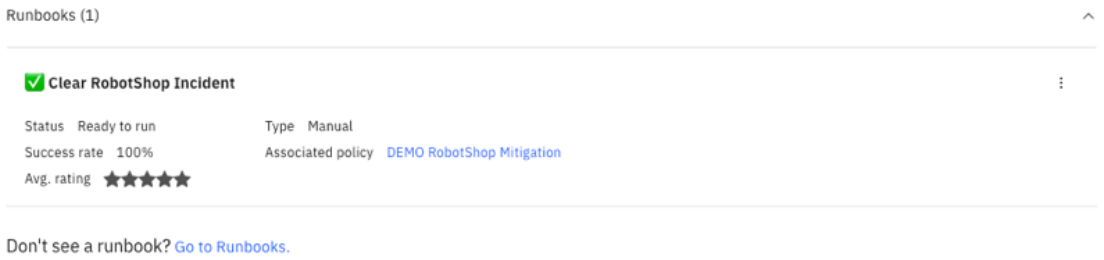


Figure. Runbooks

Topology

If the resource on which an alert occurred can be located in the network topology system, you can collapse the **Information** section in the side panel to view the **Topology** section. For more information, see Displaying alert topology.

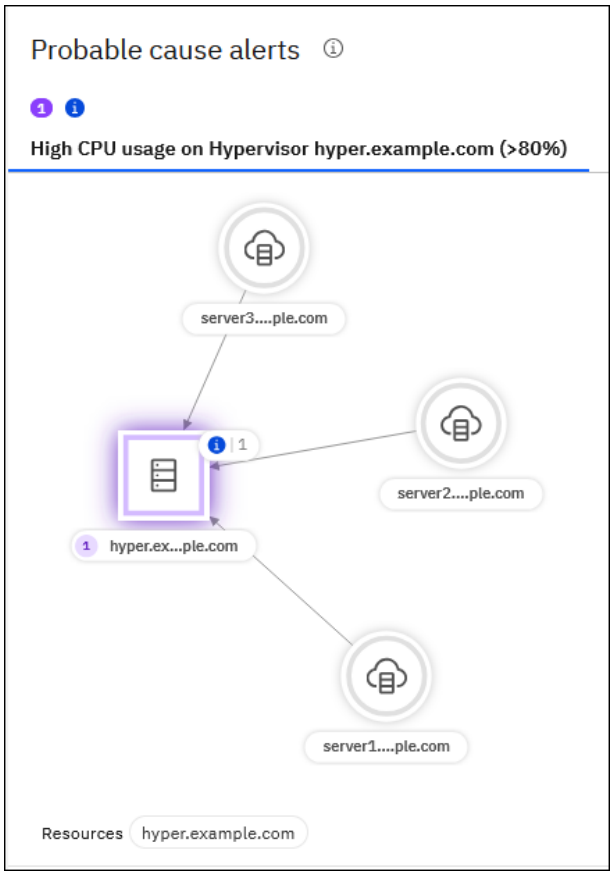


Figure. Alert topology

Log anomaly

When you select a log anomaly alert in the table, the **Alert details** panel contains a **Log anomaly details** section. For more information, see Log anomaly alerts.

Domain-specific log anomaly alert details

Select a domain-specific log anomaly alert, such as one pertaining to IBM MQ or WebSphere, in the **Alerts** table. The **Alert details** panel contains a **Log anomaly** details section with the following information:

Table. Alert detail explanation

Detail	Explanation
Message code	Every domain-specific log includes a message code, and message codes are summarized in an alert. This message code is the most significant domain-specific message code for the current alert.
Short explanation	The title of the log anomaly is based on the message code.
Detailed explanation	The description of the log anomaly is based on the message code.
Category and Subcategory	The log anomaly is assigned to a category and subcategory based on the message code.
Message codes and frequencies	A log anomaly alert consists of multiple logs, and each log has a message code. The frequency shows how often a certain message code appears in the logs that make up the log anomaly alert.

Metric anomaly

Default alert expectations are calculated by the Metric Manager based on the input data. If an anomalous alert rate is detected that is either higher or lower than expected, an alert is raised and displayed as a `Metric anomaly` in the Alerts table. When a metric anomaly alert is selected in the table, the **Alert details** panel contains a **Metric anomaly details** section and depicts the expected values against the actual values. For more information, see Metric anomaly details.

Recommended actions

For domain-specific log anomaly alerts, up to three links for resolving the alert are included. Each link consists of a title and a short description of the resolution. These links are based on the message code and past resolution tickets that are associated with the selected alert.

Correlation information

If you are viewing an incident's alert list, and the selected alert is part of an alert group, the Alert details side panel displays information about the underlying temporal, topological, or scope-based groups.

- **Scope-based grouping**

  The scope-based grouping section contains the following information:

    - These alerts are found to share a cause as they all occurred within the same scope and period of time. The scope defines the properties that alerts must share in order to be grouped. It can be set in a scope-based grouping policy or by the scope-based grouping AI algorithm.
    - **Scope-based policy associated to this alert**. Click the link to go directly to the associated policy in the Policies UI.

- **Topological grouping**

  The topological grouping section contains the following information:

    - Name of the resource defined in the topology management service, on which this topology group is based.
    - Pane showing the resources in the topology on which this topology group is based. Click the resource to displays the relationship between that resource and neighboring resources. Right-click the resource to display the context menu.

- **Temporal correlation**

  The temporal correlation section contains the following information:

    - Date and time of first instance of this group.
    - Total number of historical instances of this group. For details of when these instances occurred and how many alerts occurred in each instance, see the Group instance heatmap.
    - Average time in seconds that this group instance lasted.
    - Time-based heatmap showing recent historical period in days with a gray square for each day. Each darker square indicates a day on which there was at least one group instance. Hover over the square to see details of this group instance.

Working with alerts and menu customization

Actions for alerts can be accessed via the right-click context menu.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.
3. Right-click any alert (or multiple alerts) in the Alert Viewer and select from the following actions:

- **Acknowledge**

  Acknowledge an alert when you want to work on that alert. The acknowledged status is recorded as `true` in the Alert details.

- **De-acknowledge**

  De-acknowledge an alert if you are no longer working on it. The acknowledged status is recorded as `false` in the Alert details.

- **Update severity**

  Select from the following alert severities:

    -  Indeterminate
    -  Informational
    -  Warning
    -  Minor
    -  Major
    -  Critical

- **Suppress**

  When an alert is suppressed in Cloud Pak for AIOps, the `alert.suppressed` flag is set to `true`. The alert can still be viewed in the Alert Viewer, but the alert is filtered out of the view by default.

- **De-suppress**

  Changes the `alert.suppressed` flag from `true` to `false`. The alert is not filtered out of the view by default.

- **Take ownership**

  Take ownership of an alert if you want to work on resolving that alert.

- **Assign owner**

  Use this command to assign an alert to another user. That user then becomes the alert owner. You must select a user group the owner belongs to.

- **Assign user group**

  Use this command to assign an alert to a group.

- **Clear**

Change the alert state to Clear.

- **Quick Filter**

  Use quick filters to display alerts that match a selected criteria. For more information, see Creating quick filters.

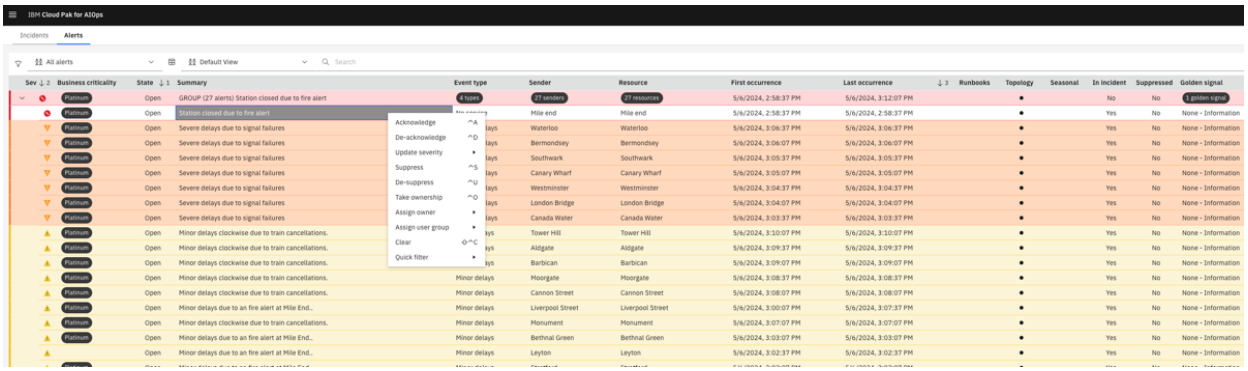


Figure. Alert actions

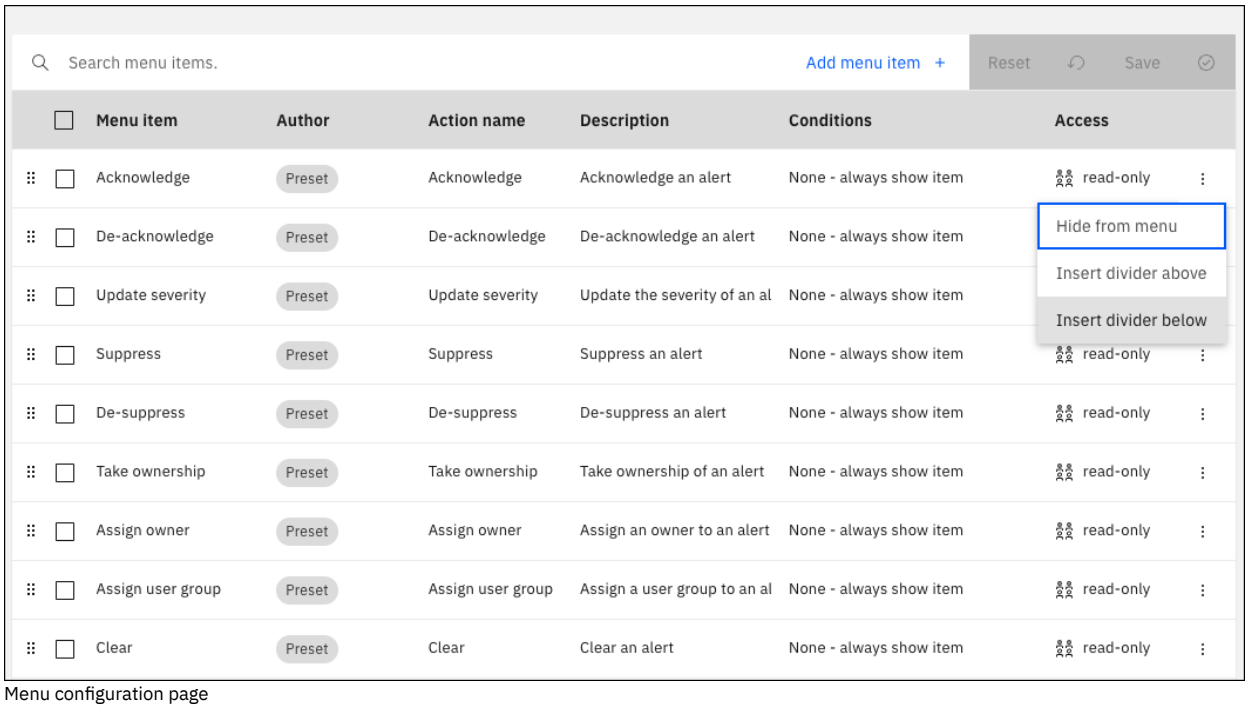
Menu customization

You can personalize the arrangement of right-click menu items. The arrangement is only visible to the user customizing the menu. That is, each user can define how their own menu looks. Users can control the following settings:

- Menu items are hidden or visible
- The order of menu items within the menu
- Grouping menu items with dividers

To access menu configuration in the Alert Viewer, click > **Configure alerts contextual menu**.

The menu configuration page represents the layout of the menu. From here, you can click and drag menu items to re-arrange the order. You can also click the options menu (three vertical dots) to hide or show menu items, and group menu items with dividers.



Menu configuration page

If a user has access to shared menu items, initially these items are shown as disabled in the menu configuration page. To enable shared menu items, select **Show in menu** from the options menu.

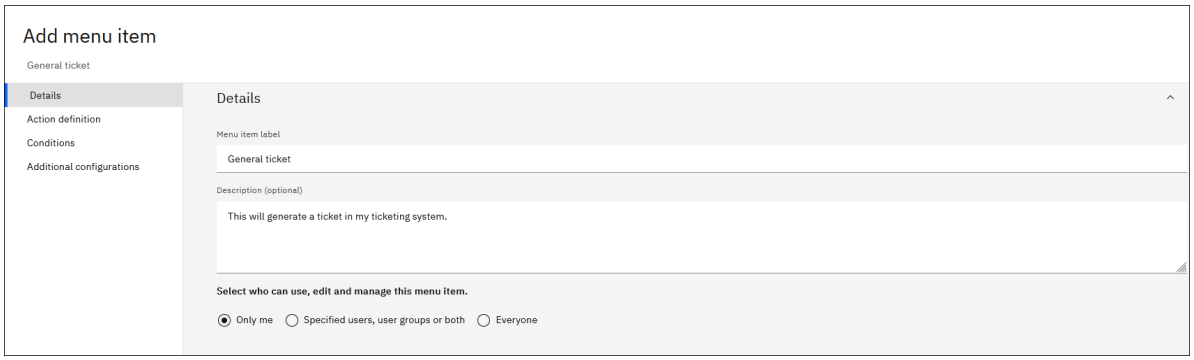
Add menu items

To add custom menu items for use in your personal menu, or to share with everyone, click **Add menu item +** from the menu configuration page.

Configure the following properties for a custom menu item:

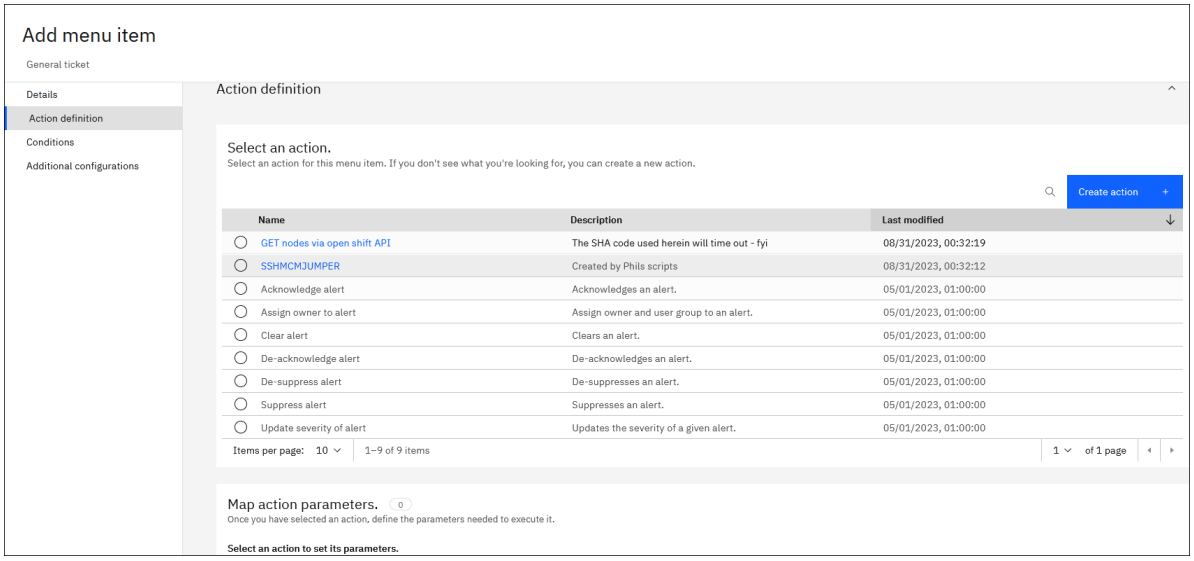
- **Details**
  - Display name

- Description
- Select who can use this menu item (**Only me** or **Everyone**).

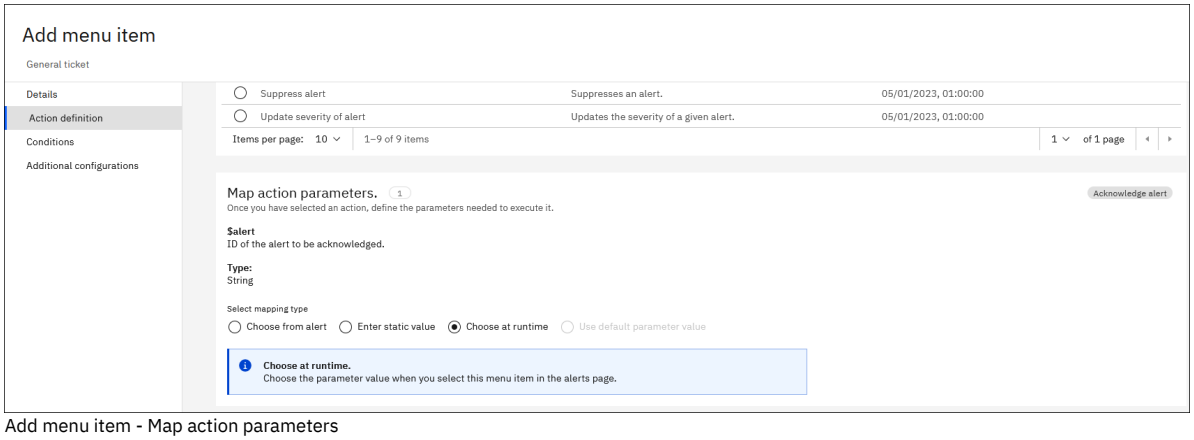


Add menu item - Details

- **Action definition** and parameter-mapping that includes the following options:
  - Choose from alert: based on the selected alert(s), set the desired properties as the action parameter value. If you want to minimize the scope to a singular key within the alert's details, enter a string value that matches a key from an alert's details. For example, if you enter "`field1`" in the Property field, it is understood as "`alert.details.field1`". **Note**: The permitted characters for the "`details`" field are A to Z, a to z, 0-9, and the underscore symbol (_).
  - Static value: set a fixed value for a parameter.
  - Default value: use the default value specified in the Automation action definition.
  - Choose at run time: input the value at the time the menu item is run.



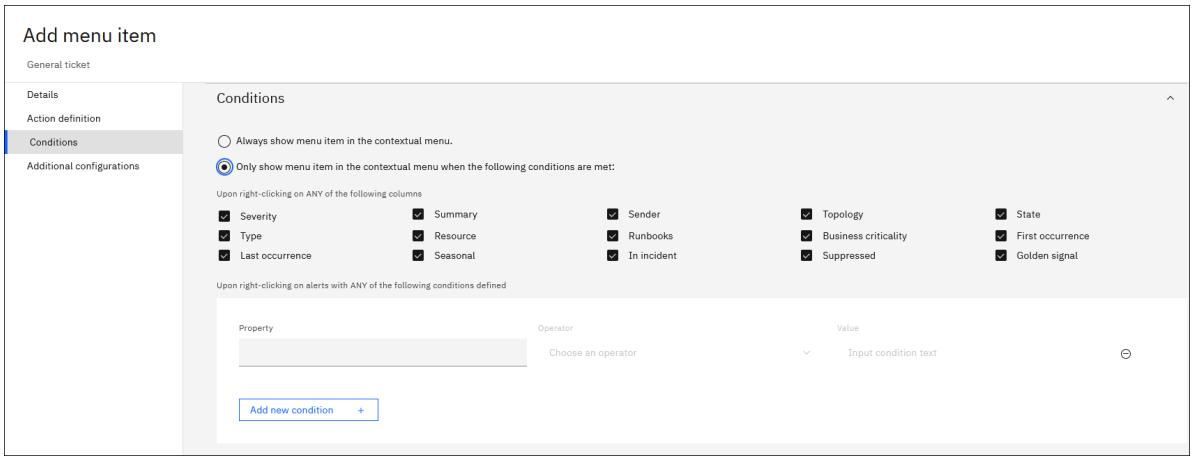
Add menu item - Action definition



Add menu item - Map action parameters

- **Conditions** to define when a menu item should show on an alert right-click action:
  - Columns: select specific columns for which the menu item will surface.
  - Property: select from alert attributes that are predefined for Cloud Pak for AIOps and common to most alerts. If you want to minimize the scope to a singular key within the alert's details, enter a string value that matches a key from an alert's details. For example, if you enter "`field1`" in the Property field, it is understood as "`alert.details.field1`". Then, enter the string value in the **Value** field. **Note**: The permitted characters for the "`details`" field are A to Z, a to z, 0-9, and the underscore symbol (_).

- Operator: select a comparison operator from this list. The range of comparisons available is determined by your selection in the Property field.
- Value: the fields (or free-form string value) that appear here are dependent on the options selected in the Property and Operator lists.



Add menu item - Conditions

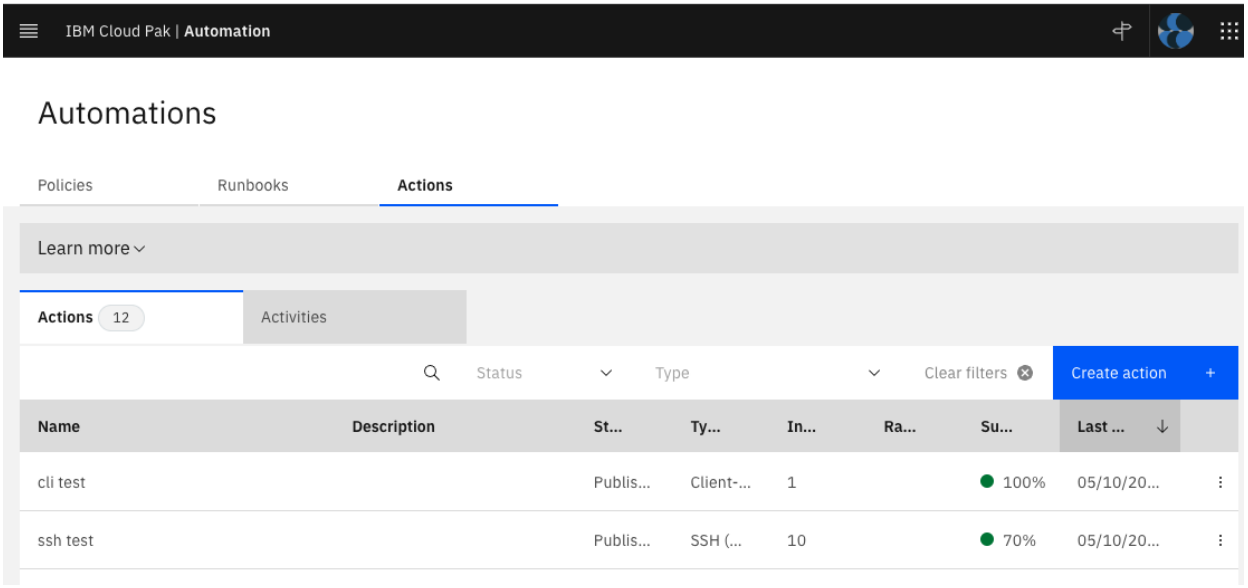
- **Additional configurations** to define the alert activity entry. This is the message that will appear in an alert's activity once this menu item has been selected to act on the alert. To change the default entry, edit the message and preview the output.



Add menu item - Additional configurations

Implementation

The actions that are associated with custom menu items are implemented as Automation Actions.



Automation Actions

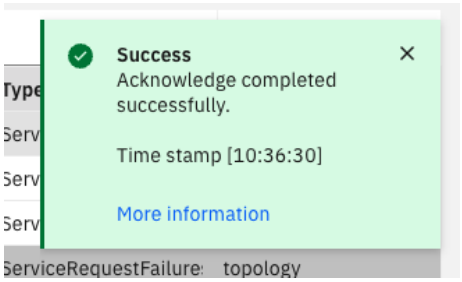
Actions are one of the following types:

- HTTP
- SSH
- Client-side

One benefit of using the Automation action framework for implementation is that these actions can be used independenly of the right-click menu, either stand-alone or within a runbook. For more information, see Managing actions.

Menu item execution

Upon clicking a menu item, the actions will run in the background for each selected alert if there are no parameter mappings specified as **Choose at run time**. When all actions are completed, a notification is displayed with results for each selected alert.



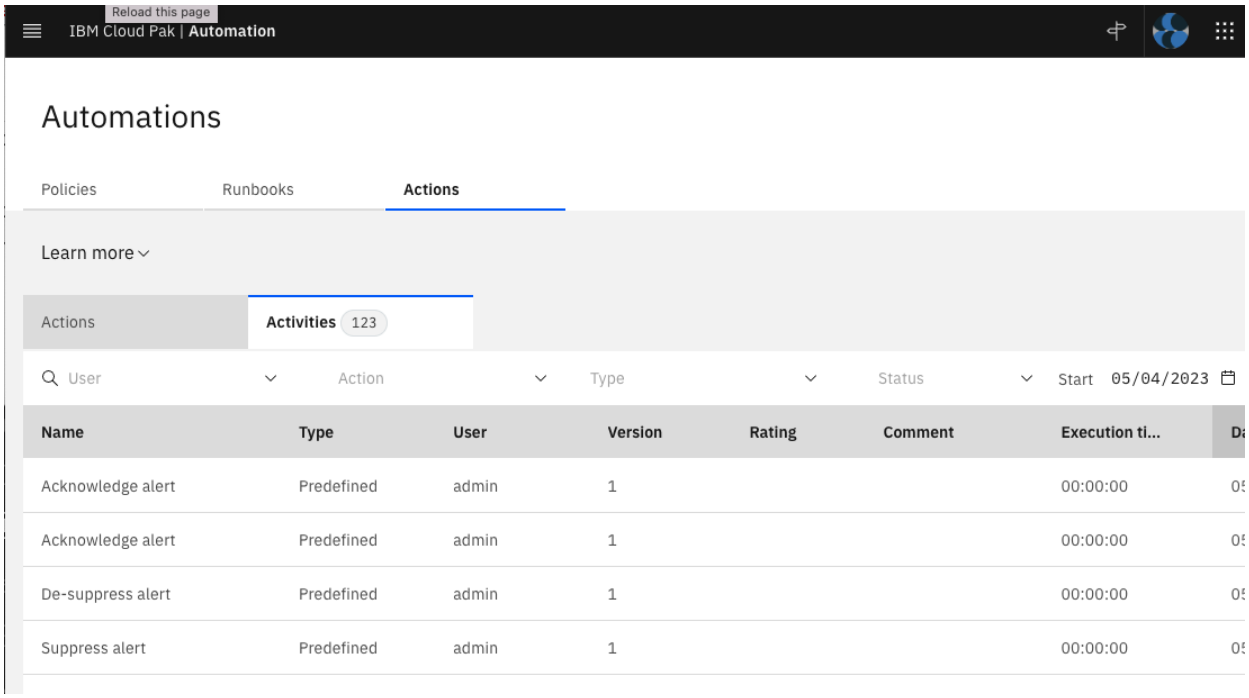
Menu item notification

Click the **More information** link in the notification to show the raw output from the action execution, which can be copied with the copy icon .



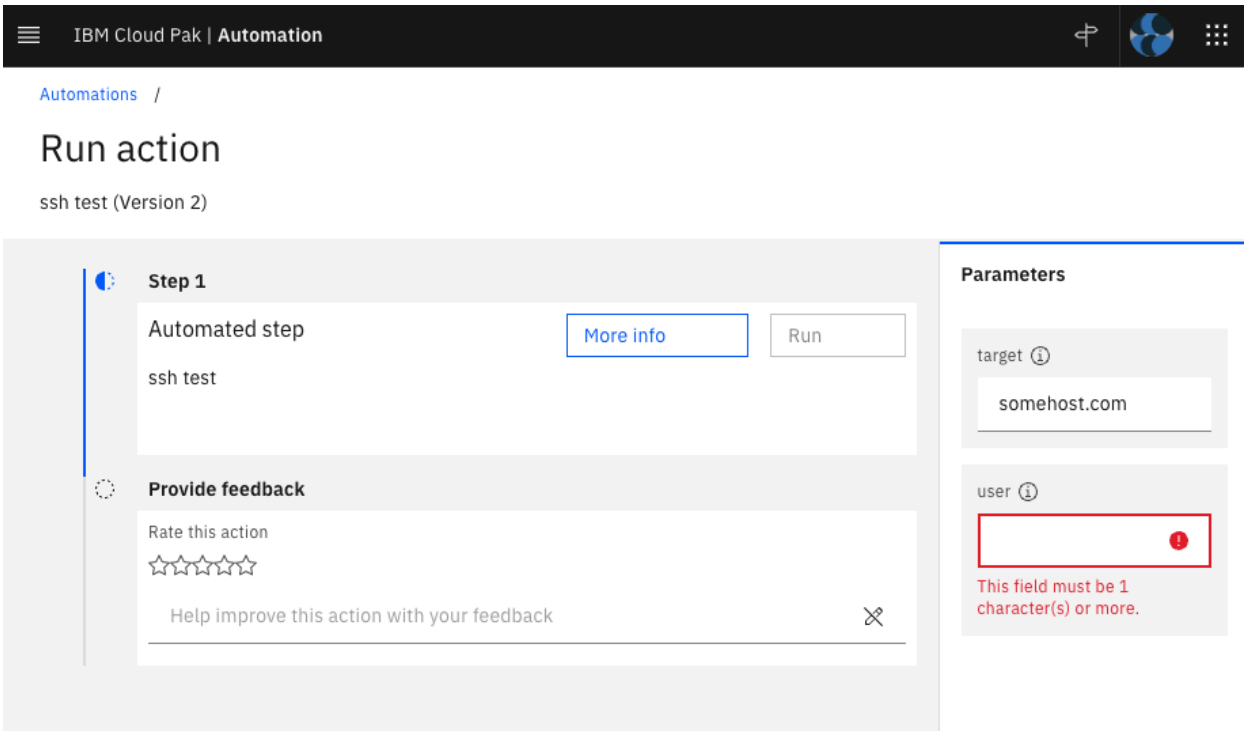
More information

An audit of all action executions is available on the Automations > Actions > **Activities** page.



Activities page

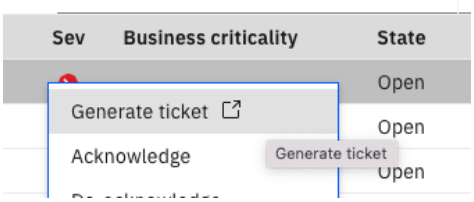
If a menu item has a parameter mapping specified as **Choose at run time**, the user is directed to the **Run action** page to complete the parameter input and run the action when clicking a menu item.



Run action

In this scenario, when multiple alerts are selected the alert properties are combined into a comma-delimited list. The automation action can then process this list as required.

For menu items using an action of type Client-side, if there are no parameter mappings of **Choose at run time**, the parameter values are mapped to URL query parameters and a new browser tab opens with the resolved URL when running the menu item. These menu items are differentiated by the "launch" icon. Note, menu items of this type only work for single alerts. If multiple alerts are selected, properties of the context menu alert are used to resolve parameter mappings.



Launch icon

Access control

Access control is applied at the menu item level. The menu items available for a user's personal menu consist of items the user has authored themselves, or items set with "Everyone" access. Preset menu items are read-only for everyone. Custom menu items can be updated by anyone with access to the menu item.

Searching and filtering alerts

Search for alerts by name, or filter the list based on selected criteria. You can also save a filter configuration for future use.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.

Search alerts

To search for an alert in the table:

- Click anywhere in the **Search** field.
- Start typing a description of the alert in the search field. For example, type "error" and the system will show in the table all alerts that contain the text "error" in the summary.



Figure. Search bar

Note: A search term entered in the **Search** text field is not part of a saved filter. Only the configuration applied in the **Filter conditions** side panel is part of the saved filter.

Predefined alert filters

All Alerts are displayed in the alert list by default. Click the Down chevron icon to select from the following predefined filters:

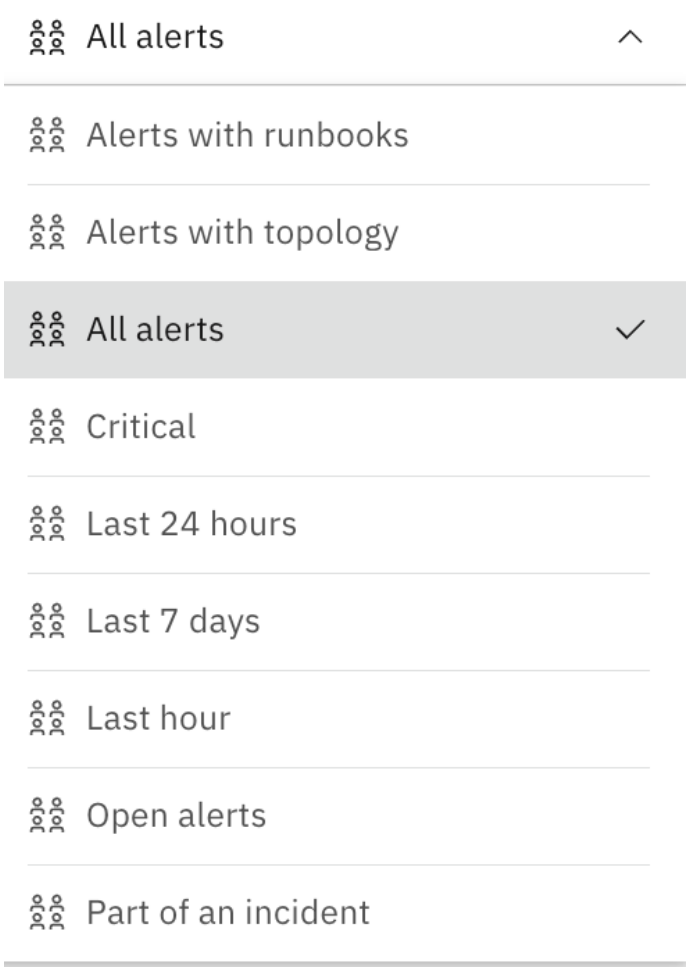


Figure. Predefined filters

Customized alert filters

1. Select an existing filter from the drop-down menu or use the default All Alerts.
2. Click the Filter icon to open the filter side panel.

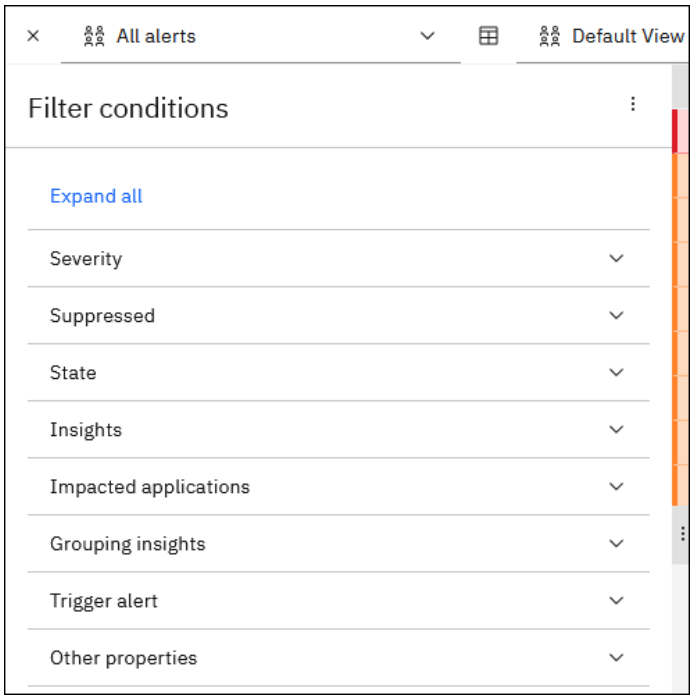


Figure. Alert filters

3. Apply filters to the alert list (or apply additional criteria to an existing filter). Select from the following filters for alerts:

- **Severity**:
  - Critical
  - Major
  - Minor
  - Warning
  - Informational
  - Indeterminate
- **Suppressed**:
  - Yes
  - No
- **State**:
  - Open
  - Clear
  - Closed
- **Insights**:
  - Part of an incident
  - Runbooks available
  - Enrichment
- **Impacted applications**:

  Find impacted applications to filter by.
- **Grouping insights**:
  - Temporal
  - Scope-based
  - Topological
- **Other properties**

  You can create more personalized filters by using alert properties. Expand the **Other properties** section and complete the fields as follows:

- **AND and OR**: when adding conditions, you can join multiple condition types by using the AND and OR operators. The AND operator means that alerts are matched only if all of the individual conditions are true. The OR operator means that alerts are matched if any of the individual conditions are true. The default behavior for alert filter conditions is AND.
- **Property**: select from alert attributes that are predefined for Cloud Pak for AIOps and common to most alerts.
  - If you want to minimize the scope to a singular key within the alert's details, enter a string value that matches a key from an alert's details. For example, if you enter "`field1`" in the Property field, it is understood as "`details.field1`". Then, enter the string value in the **Value** field. **Note**: The permitted characters for the "`details`" field are A to Z, a to z, 0-9, and "_" (underscore).
  - You can also access additional properties of `alert.sender`, `alert.resouce`, and `alert.type` by typing your custom property in the Property field. For example, to filter alerts where the `alert.sender.customProperty` = "custom", type `sender.customProperty` and select "`Property: sender.customProperty`". The only permitted special character for custom properties is "_" (underscore).

    For more information, see Examples of policy conditions mapped to alert JSON.

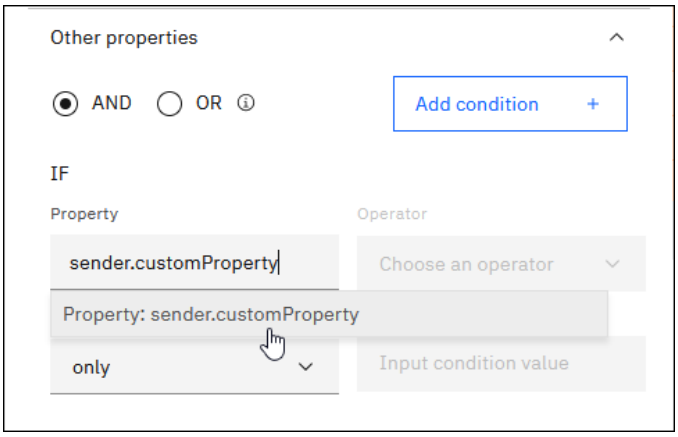


Figure. Custom property filter

- **Operator**: select a comparison operator from this list. The range of comparisons available is determined by your selection in the Property field.
- **Value**: the fields (or free-form string value) that appear here are dependent on the options selected in the Property and Operator lists.

To update an existing filter

1. After you have modified the existing filter criteria, click **Save**.
2. A **Filter updated** message is displayed to confirm that your changes have been saved.

Note: An asterisk (*) is displayed next to the filter name if there are unsaved changes to the selected filter. An asterisk is also displayed when another user has modified the current filter. Unsaved filters are only applied to the already fetched alerts. Click **Reset filter** in the filter side panel to get the latest changes to the filter and have them applied to the alert list.

To save as a new filter

1. After you have applied the filters, click the three vertical dots in the **Filter conditions** side panel to open the list of options.
2. Click **Save as a new filter**.

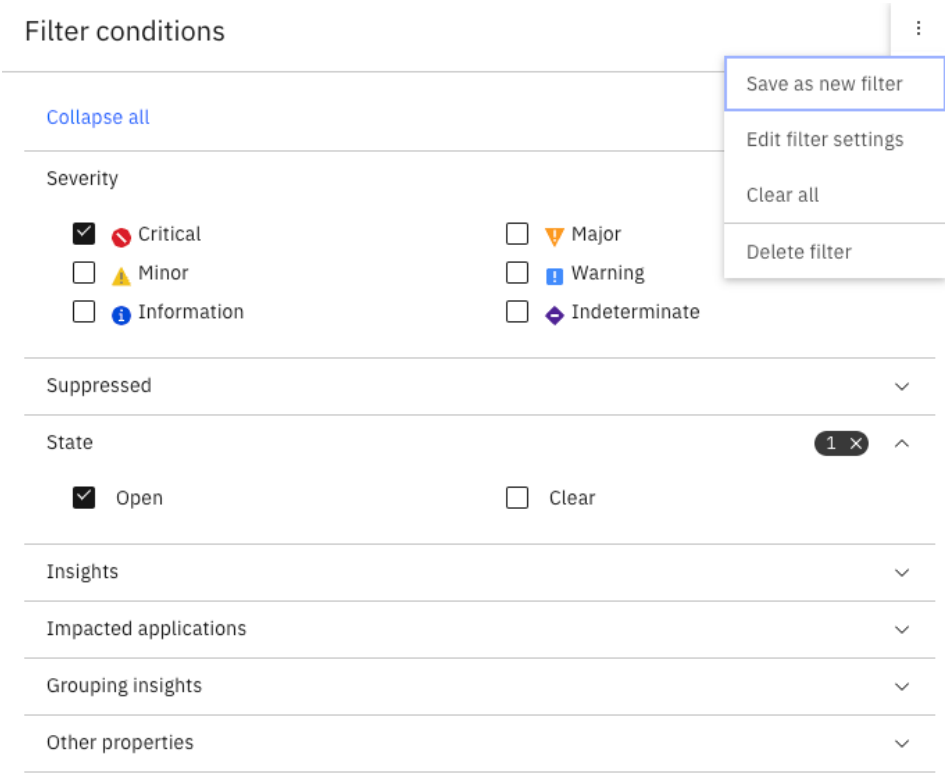


Figure. Filters side panel

3. Enter a **Filter name** and **Description**.

   Note: Filter names must be unique across different categories of filters, whether a normal filter or a restriction filter set by an administrator. If a chosen filter name is already in use for any type of filter, an error message is displayed to say the filter name already exists.

   Select who can use this filter from the following options:

   - **Only me**
   - **Specified users, user groups, or both**: to manage access, you can select users, user groups, or both. Any selections you make includes yourself. After you select the users or user groups who can use the filter, you must specify their level of access:
     - **Can use**: users can see the filter in the drop-down list and apply it to the list of alerts.
     - **And edit**: users can use, edit, and save the filter.
     - **And manage**: users can use, edit, and manage the filter name, description, and access control.
   - **Everyone**: additionally specify if everyone can **Use this filter** or **Edit this filter**.

To clear changes that have been applied to a filter, click **Reset filter** in the filter side panel or **Clear all** from the list of options (three vertical dots). From the options list you can also **Delete filter** and **Edit filter settings**.

To save as a new filter

1. After you have applied the filters, click the three vertical dots in the **Filter conditions** side panel to open the list of options.
2. Click **Save as a new filter**.

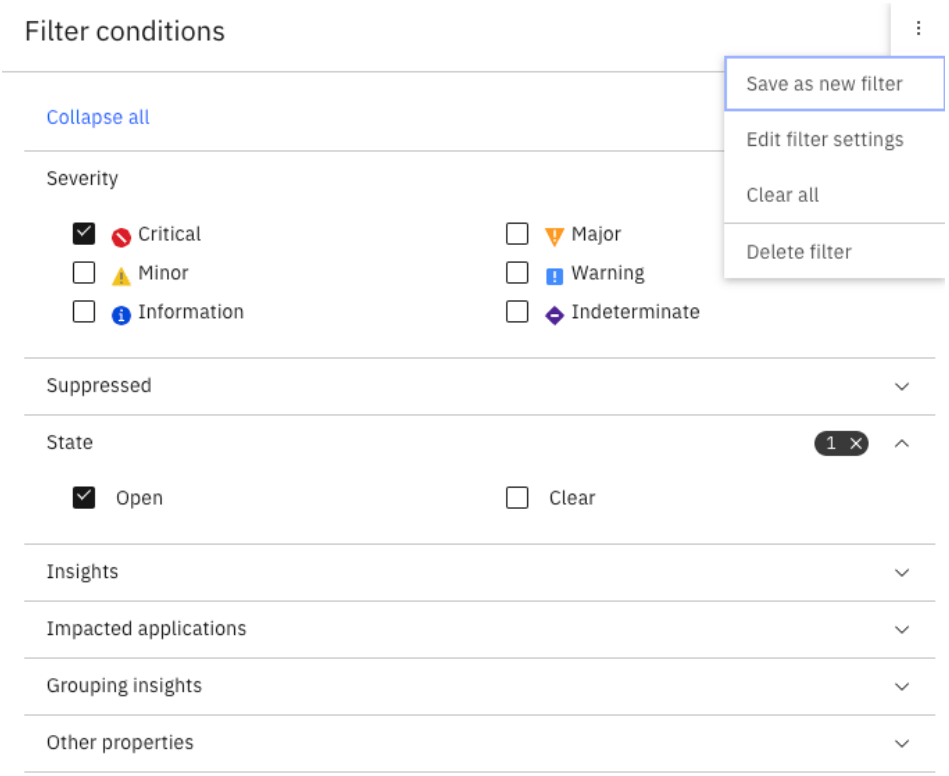


Figure. Save as new filter dialog

Note:: Alternatively, if you are an administrator who has *manage profiles* priviledges had enabled, you can set Role Based Access Control (RBAC) functionality that allows you see and click the **Save as a new restriction filter** option, thus applying certain restriction filters to data types and resources, alerts in this case. The *save a new restriction filter dialog* window is slightly different to the *Save a new filter dialog* also.

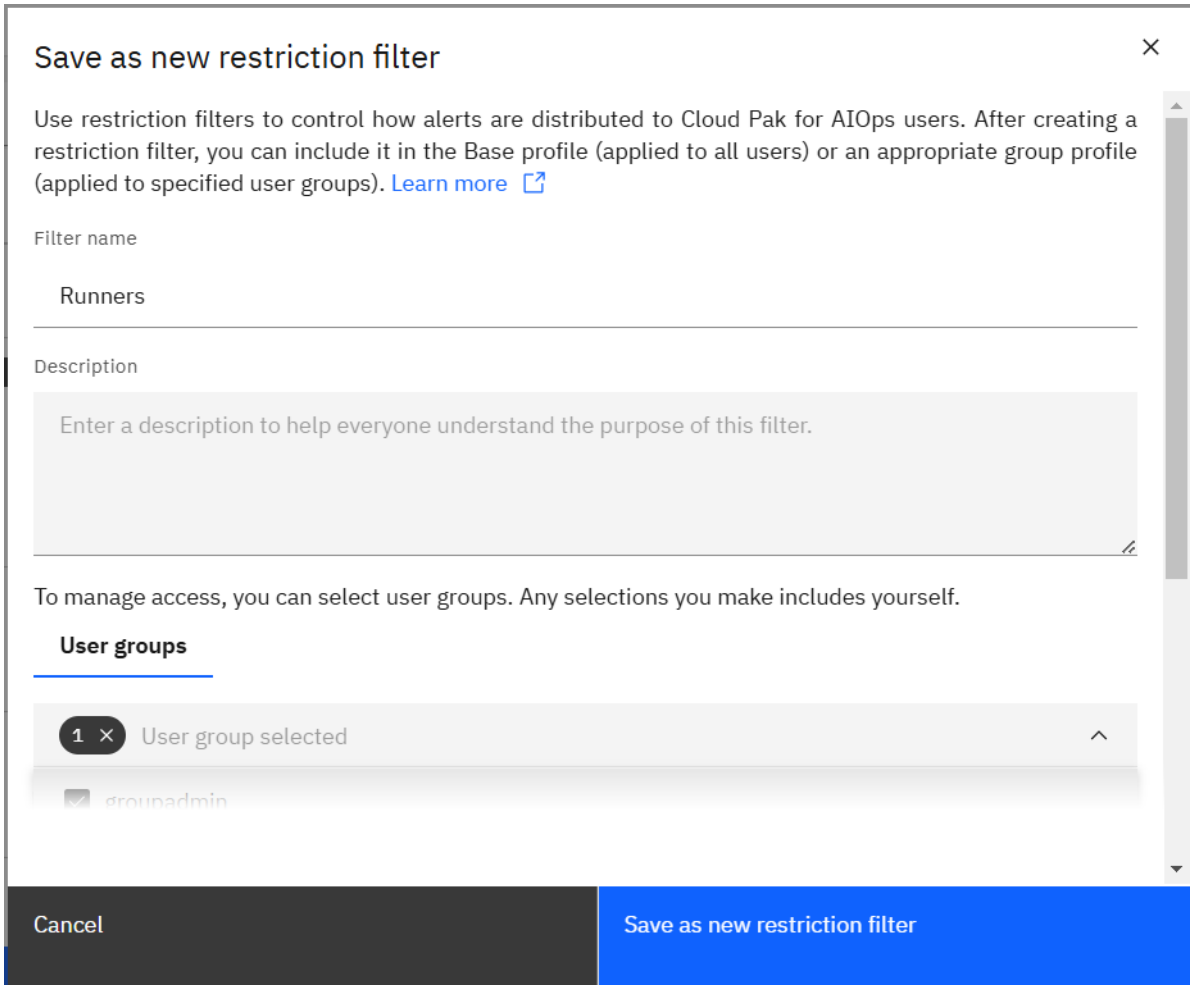


Figure. alerts save as new restriction filter

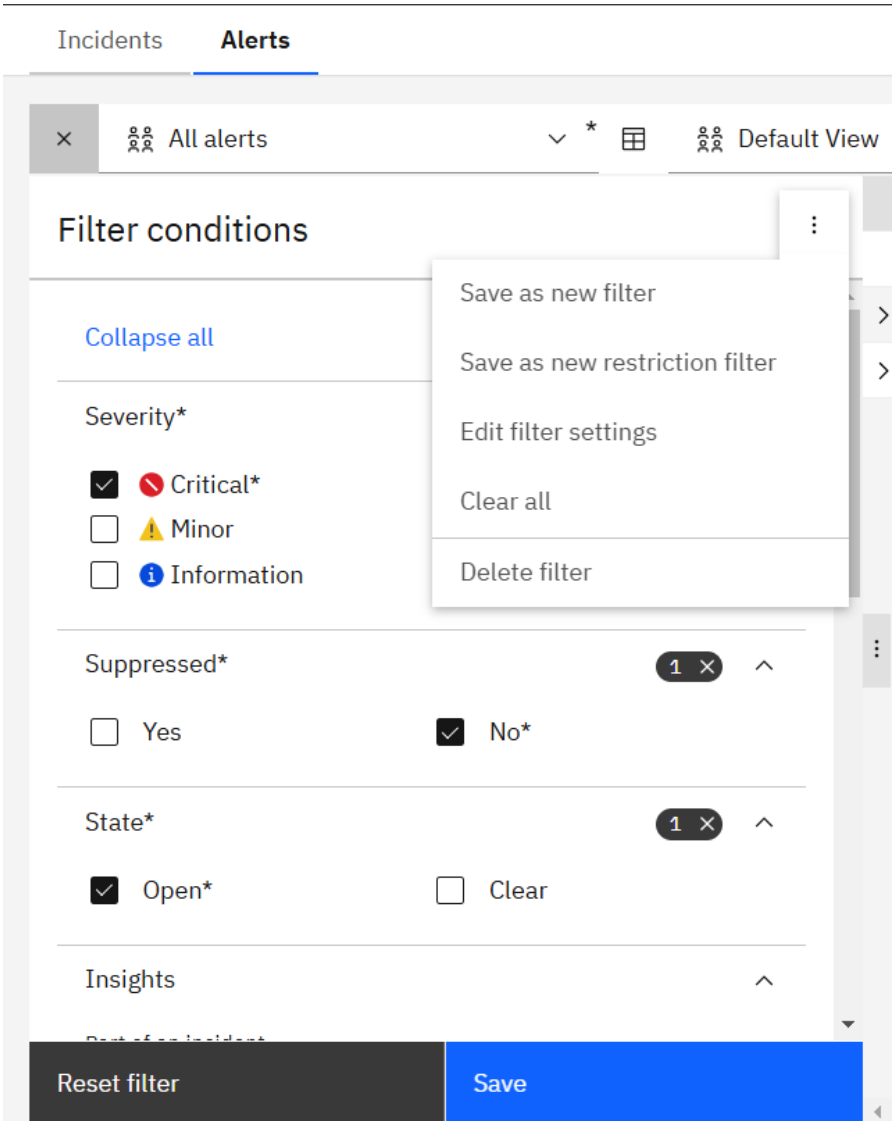


Figure. Filters side panel

3. Enter a **Filter name** and **Description**.

   Note: Filter names must be unique across different categories of filters, whether a normal filter or a restriction filter set by an administrator. If a chosen filter name is already in use for any type of filter, an error message is displayed to say the filter name already exists.

   Select who can use this filter from the following options:

   - **Only me**
   - **Specified users, user groups, or both**: to manage access, you can select users, user groups, or both. Any selections you make includes yourself. After you select the users or user groups who can use the filter, you must specify their level of access:
     - **Can use**: users can see the filter in the drop-down list and apply it to the list of alerts.
     - **And edit**: users can use, edit, and save the filter.
     - **And manage**: users can use, edit, and manage the filter name, description, and access control.

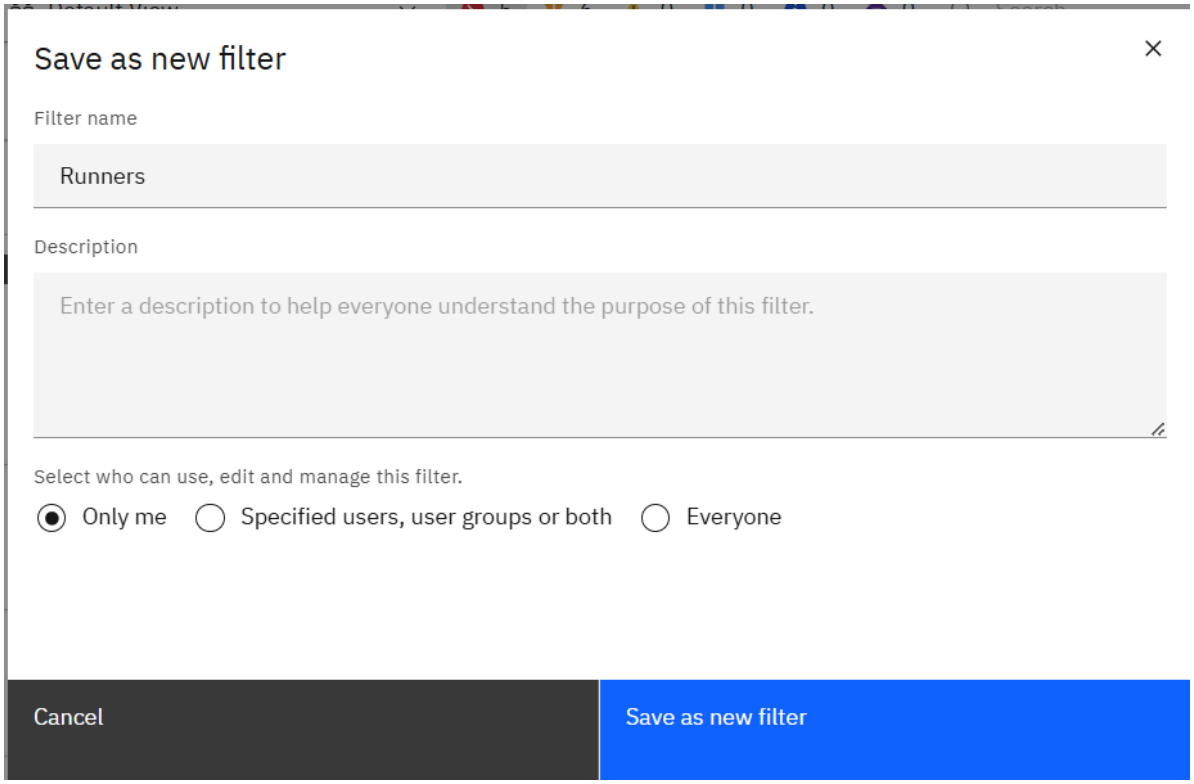


Figure. alerts save as new filter

- **Everyone**: additionally specify if everyone can **Use this filter** or **Edit this filter**.

To clear changes that have been applied to a filter, click **Reset filter** in the filter side panel or **Clear all** from the list of options (three vertical dots). From the options list you can also **Delete filter** and **Edit filter settings**.

Managing multiple alerts

You can multi-select alerts and apply actions to all of your selections at one time.

Procedure

1. To multi-select alerts that are not next to each other in the Alert Viewer, press and hold down the Ctrl key (Command key on Mac) and click the alerts that you want to select.
2. If you want to multi-select alerts that are next to each other in the table, you don't have to click each alert. First, press and hold down the Shift key. Then, click on the top alert and bottom alert in your selection. All the alerts between the first and last are also selected.
3. Actions for alerts can be accessed via the right-click context menu. For more information, see Working with alerts and menu customization.

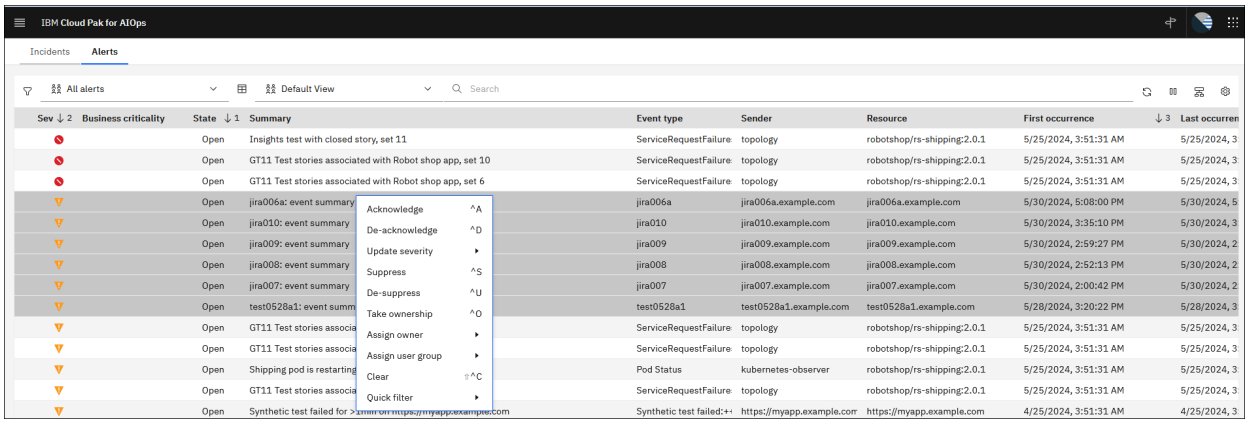


Figure. Managing multiple alerts

Creating views

Create custom views that are dynamically applied to the Alert Viewer. The views determine what information is displayed from the available alert data.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.
3. On the toolbar, click **View configuration** > **Create a new view**.

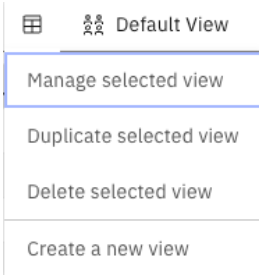


Figure. Create a view

4. Configure the following properties for a custom view:

- **Details**
  - View name
  - Description
  - Select who can use, edit, and manage this view:
    - *Only me*
    - *Specified users, user groups, or both*
    - *Everyone*



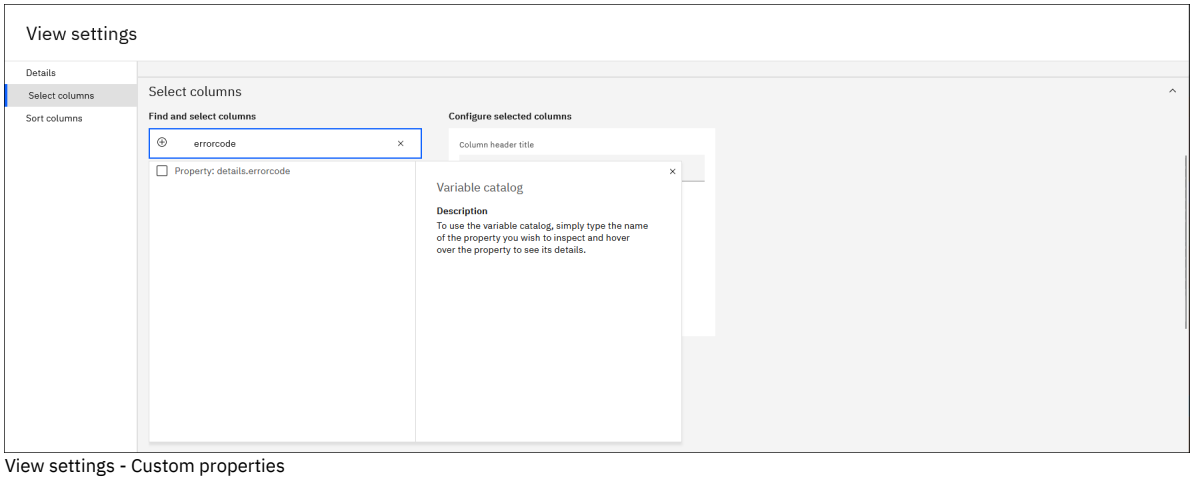
View settings - Details

- **Select columns**

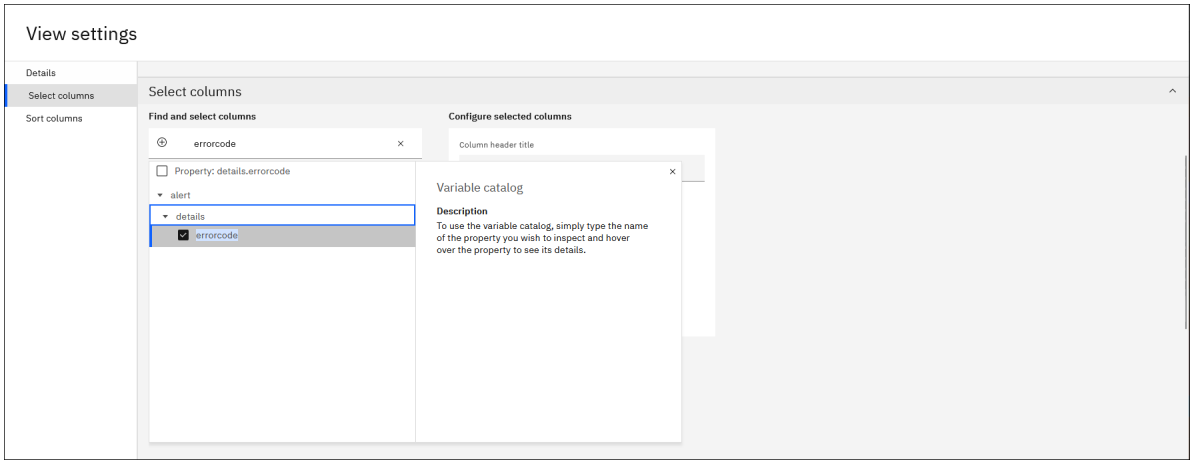
  Select from alert attributes that are predefined for Cloud Pak for AIOps and common to most alerts. Click anywhere in the field provided to display the variable catalog. Move the cursor over the attributes to display the descriptions. You can also type the name of the property that you are looking for and the system will show in the drop-down list all alert properties that contain the typed text. You can drag and drop the columns to determine the order of the columns appearing in the Alert Viewer.

  - **Columns from custom properties**

    Columns can also be added based on custom properties that have been mapped to the alert inside the **`details`** property. For example, if you have mapped some data to **`details.errorcode`** you can then utilise that property as a column in the Alert Viewer. Simply type the property name, **`errorcode`** in this example, and select the resulting property with the "**`details.`**" prefix:

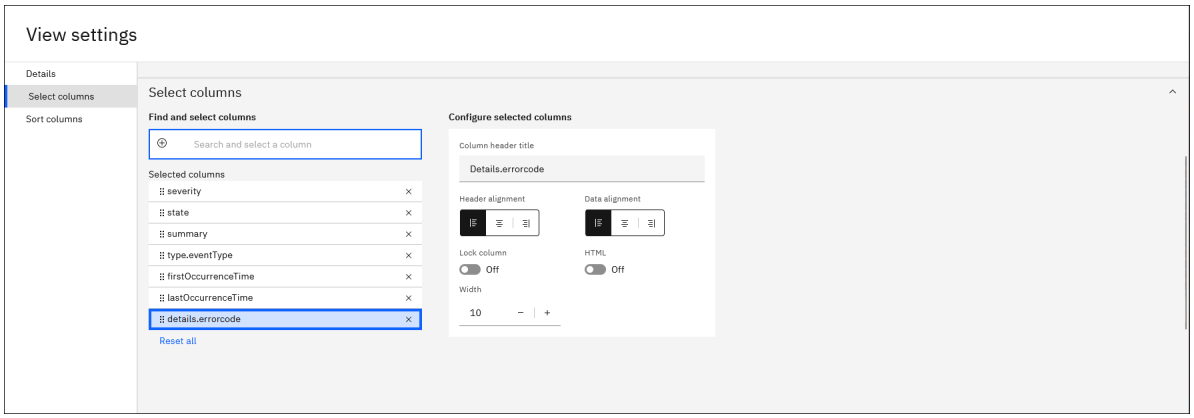


View settings - Custom properties



View settings - Custom properties

Close the catalog and you will find the property added at the bottom the list:



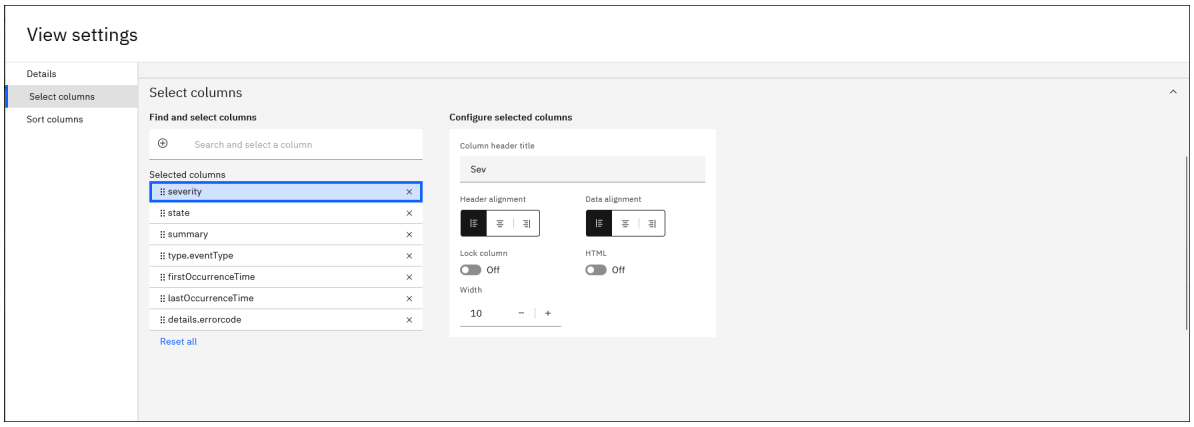
View settings - Custom properties

Important: A view must contain the **ID** and **Summary** columns.

- **Configure selected columns**

  Configure the following properties to customize the columns that display alert data:

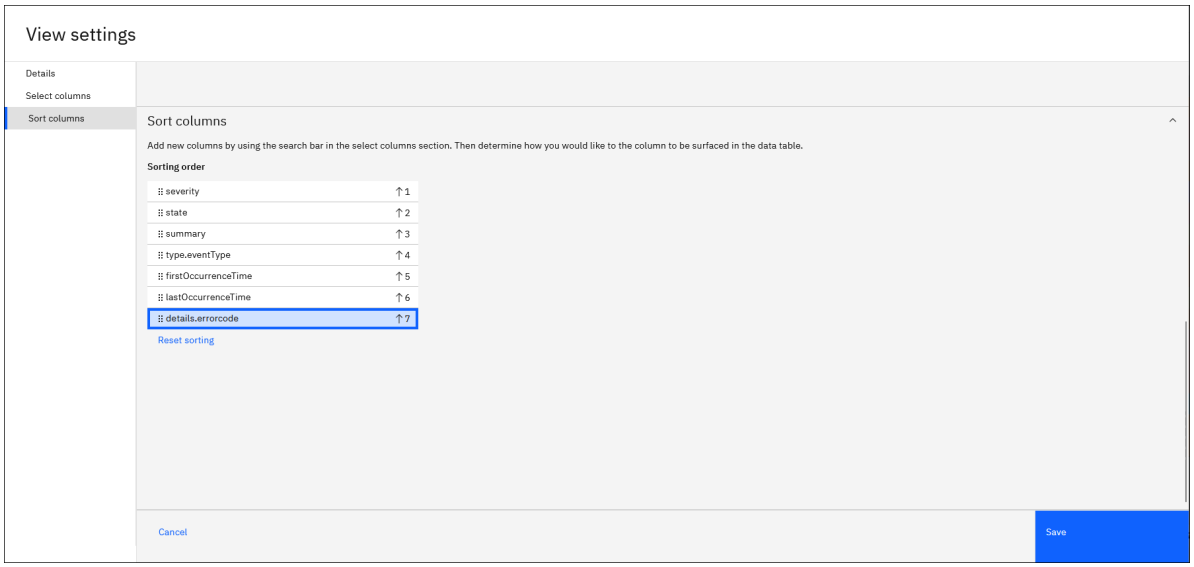
  - *Column header title*: type a new title for the column or keep the default.
  - *Header alignment*: set the position of the title text in the column heading.
  - *Data alignment*: set the position of the alert data in the columns.
  - *Lock column*: locked columns are always displayed when you scroll an alert list horizontally. Locked columns appear at the left side of the view.
  - *HTML*: enable clickable links in the Alert Viewer cells.
  - *Width*: type a numeric value in the Width field to change the width of a column.



View settings - Configure selected columns

- **Sort columns**

  Add new columns in the Select columns section. Then, in the Sort columns list, single click on a column item to set ascending order. Click the column twice to sort in descending order. Drag and drop columns to prioritise sorting across columns.



View settings - Sort columns

What to do next

- **Manage selected view**
  1. From the view list in the Alert Viewer, select the view that you want to edit. The alert list updates with the view properties.
  2. Click **View configuration** > **Manage selected view**.
  3. Edit the **View settings** as required.
  4. Click **Save** to save the view.
- **Duplicate selected view**
  1. From the view list in the Alert Viewer, select the view that you want to copy. The alert list updates with the view properties.
  2. Click **View configuration** > **Duplicate selected view**.

     A copy of the view is displayed. By default, the view name is set to "Copy of *view name*."
  3. Edit the **View settings** as required.
  4. Click **Save** to save the view.
- **Delete selected view**
  1. From the view list in the Alert Viewer, select the view that you want to delete. The alert list updates with the view properties.
  2. Click **View configuration** > **Delete selected view**. This action cannot be undone. Once the view has been deleted, your view is updated based on the first view in the list.
  3. Click **Delete view** to confirm deletion.

Creating quick filters

You can use quick filters as a fast way of displaying alerts that match a selected criteria. For example, you can quickly display only those alerts that occurred at the same time as, or before, a selected alert. A quick filter is applied via the right-click context menu. Quick filters when applied are merged into the filter side panel together with the base filter selection.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.
3. Right-click on any column or cell in the Alert Viewer, then select **Quick Filter** and choose from the following operators:
   - Equals
   - Does not equal
   - Contains
   - Does not contain

   The **Insights** columns for **Part of an incident**, **Runbooks**, and **Topology** can be filtered using the **Contains** operator.

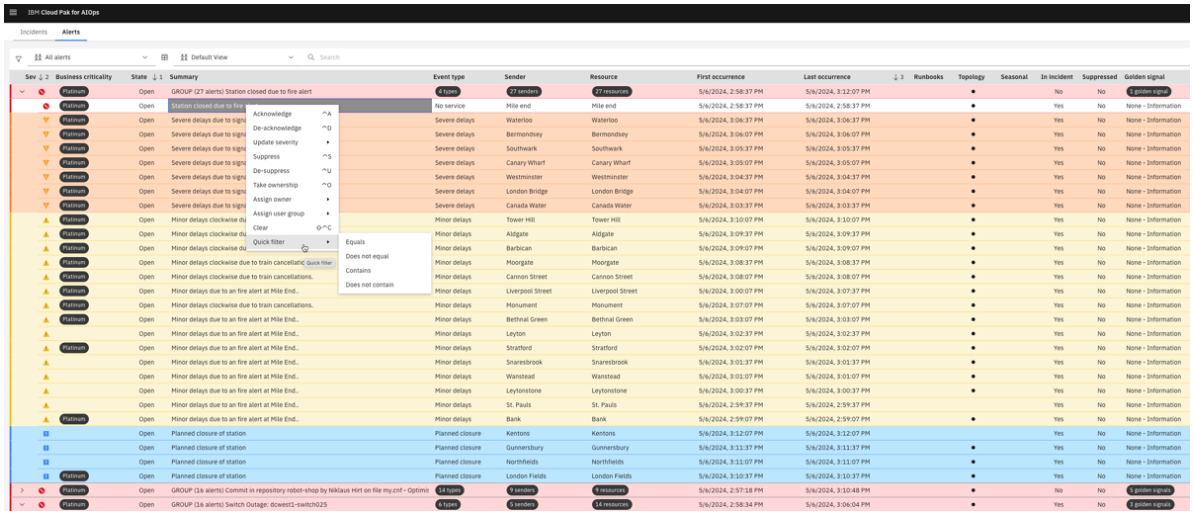


Figure. Quick filter option

Results

The alerts are filtered to only those that meet the filter condition. The applied filter condition is displayed in the filter side panel.

Related links

Menu customization

You can personalize the arrangement of right-click menu items. The arrangement is only visible to the user customizing the menu. That is, each user can define how their own menu looks.

Using custom fields

Use the custom fields associated with alerts in IBM Cloud Pak for AIOps.

Alerts in Cloud Pak for AIOps contain a set of predefined attributes that cover the general use cases. For more information about predefined attributes, see Alert and Event schemas.

In some cases you might want to add custom fields to alert properties, which are specific to your processes. For example, a retail store might want to keep track of a store id, or a communication service provider might want to track the location of a cell tower. These custom fields can be used for mapping incoming data, determining which policies to execute, or as part of automation actions. You can view custom fields alongside the predefined fields in Alert Viewer.

You can use custom fields in integrations that allow JSON mapping. These integrations include IBM Netcool Operations Insight ObjectServer and Webhook.

For more information about IBM Netcool Operations Insight ObjectServer integration, see Creating IBM Netcool Operations Insight ObjectServer integrations.

For more information about Webhook integration, see Creating Generic Webhook integrations.

If you want to use custom fields when you map incoming data (within an integration), then you need to define **`details`** field in the JSON. Add key and value pairs for your custom fields within the **`details`** object.

Use the following format for the custom fields:

```
{
 "details": {
   "myCustomField": "value"
 },
}
```

Following sample example shows the usage of custom fields within IBM Netcool Operations Insight ObjectServer integration:

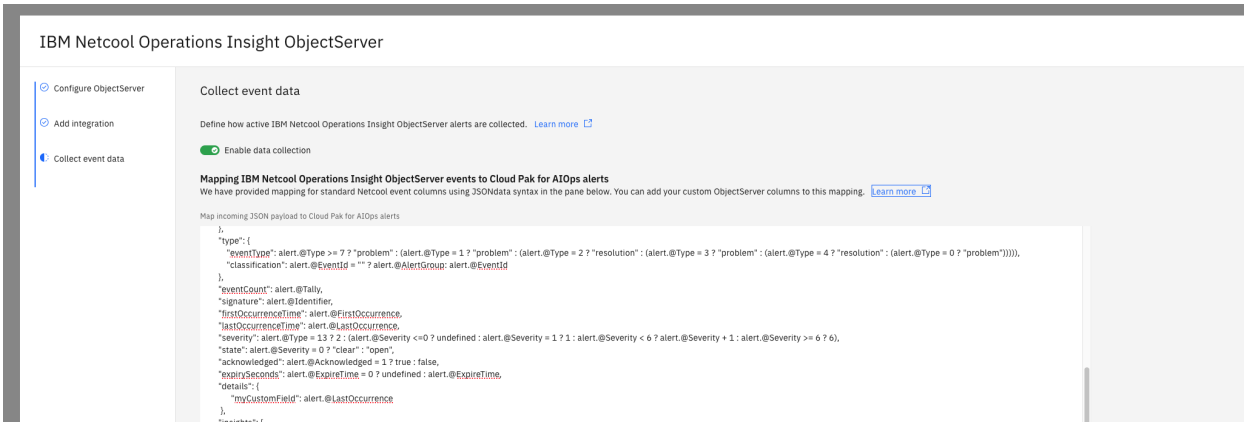


Figure. Custom fields

For more information about displaying custom fields in Alert Viewer, see **Columns from custom properties** section in Creating views.

For more information about usage of custom fields when you create or customize a policy, see Policy editor.

Setting user preferences for the Alert Viewer

You can configure the appearance and behavior of the Alert Viewer via the **User preferences** ⚙.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.
3. On the upper right corner of the page, click ⚙ > **User preferences**.

   The following preferences can be applied to the Alert Viewer:

   - **Default filter**

     Set a default filter for the alert list from the predefined or customized alert filters available in the drop-down list. For more information, see Searching and filtering alerts.

   - **Default view**

     Set a default view for the alert list from the predefined or customized views available in the drop-down list. For more information, see Creating views.

   - **Default view for alerts in incidents**

     Set a different default view for alerts in incidents.

   - **Default filter for alerts in incidents**

     Set a different default filter for alerts in incidents.

   - **Row size**

     Select from the following table row sizes:

     - Very small
     - Small
     - Medium
     - Large

   - **Alert details launch type**

     Selecting an alert in the table opens the alert details for that alert. These can be displayed in a **Side panel** or in a separate **Pop-up window**.

   - **Alert details should open on:**

     - Double-click
     - Single-click

   - **Alert activity entries sorting**

     - Newest at the top
     - Newest at the bottom

   - **Row coloring**

     Change how alert severity is depicted in the table by applying a different color background for each priority level.

   - **Table auto-refresh**

Enable auto-refresh to have the alert list automatically updated at predescribed intervals.

- **Table auto-refresh interval (s)**

  Auto-refresh interval in seconds.

- **Warning prompts**

  Turn on or off.

- **Select child alerts on parent alert right-click**

  Turn on or off.

4. Click **Save** to apply your changes to the Alert Viewer.

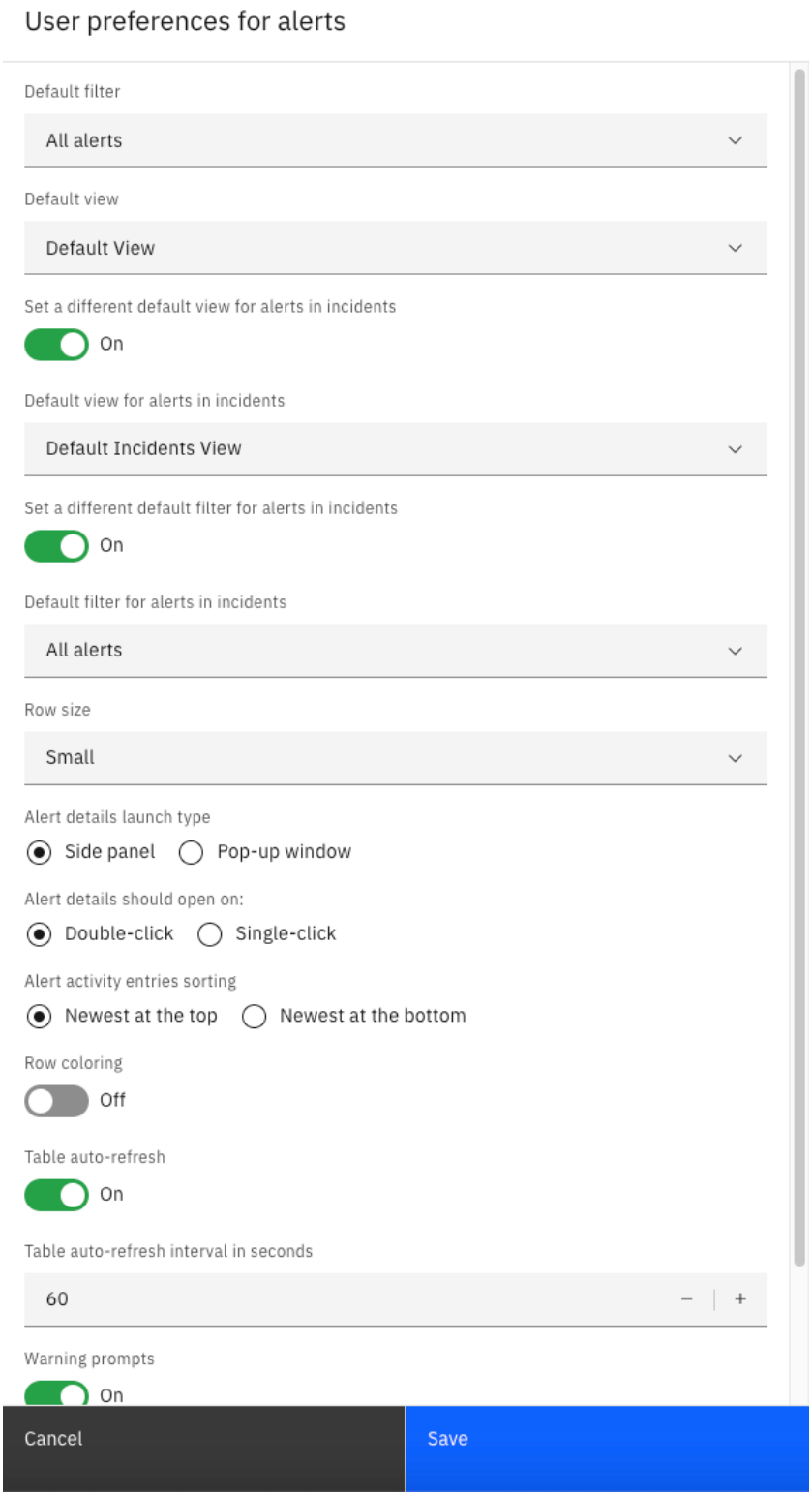


Figure. Alert preferences

About alert groups

An alert group is a group of two or more alerts that Cloud Pak for AIOps has correlated together because the underlying analytics have determined that these alerts belong together.

Alerts can be added to an alert group because of one or more of these factors:

Table. Factors for alert groups

Correlation type	Example
Temporal groups: based on alert history, these alerts tend to occur within a short time of each other.	A "Latency" alert on a server is regularly followed by a "Ping response time high" alert on that same server. These alerts are grouped into a temporal subgroup.
Topological groups: the alerts occur on resources within a predefined section of your network topology.	If there is a predefined section of the network that groups together a specific switch and all the nodes that depend on that switch, then any alerts occurring on that specific switch or the nodes connected to it are grouped together. These alerts are grouped into a topological subgroup.
Scope-based groups: the alerts occur on a user defined scope.	An administrator defines a `ScopeID` based on properties that define the scope for how alerts are grouped. Any alerts that match the scope and occur within a default time window are then automatically grouped together. An example would be where an alert storm occurs on the `london145.acme.com` server. All of the alerts in that storm will be grouped together as they match the `alert.resource.location= london145.acme.com`, and they occur within the default time window. These alerts are grouped into a scope-based subgroup.



Figure. Alert groups

Displaying an alert group

Expand an alert group to display the alerts that were correlated together within the group.

An alert group contains two or more alerts, which are correlated together by the underlying analytics. The group can include **Temporal** groups, **Topological** groups, and **Scope-based** groups.

1. Identify an alert group within the table.

   You can identify a group using the following signs:

   - It has a Down chevron icon in its row in the table.
   - By default, it has a summary that includes the words: GROUP (n alerts), where n is the number of the alerts in the group.

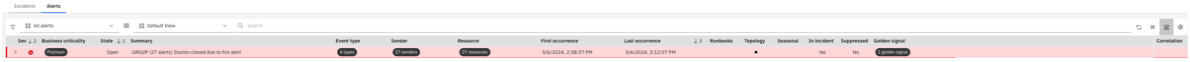
   

   Figure. Alert groups

2. Open the alert group by clicking the Down chevron icon in its row in the table.

   The group's alerts are now displayed under the parent alert.

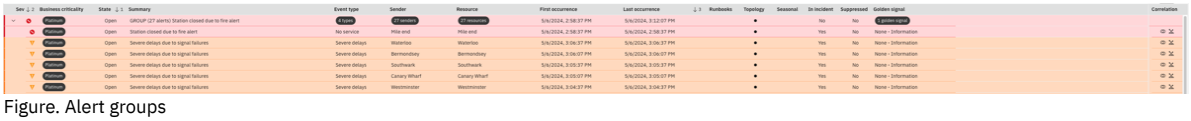
   

   Figure. Alert groups

Displaying alert topology

If the resource on which an alert occurred can be located in the network topology system, then a large dot ● is presented in the alert's Topology column. Double-click the large dot ● to display a topology map for this alert, centered on the resource on which the alert occurred.

Procedure

1. Double-click the large dot ● in the Topology column for the alert of interest.

A side panel containing multiple information sections opens on the right side of the table. The Topology section is open and provides the following information:

- **Resource**

  Name of the resource on which the alert of interest occurred.

- **Begin time**

  Date and time when the topology for this resource was last updated.

- **Observed time**

  Date and time when the alert of interest was observed on this resource.

- **Topology pane**

  Shows a one hop topology map for this alert, centered on the resource on which the alert occurred.

2. To investigate the topology further, click the **More information** link, which is located at the end of the side panel's Topology section.

   This action opens the Topology viewer, which displays the same topology map, with the full topology display experience. For more information on how to use the Topology viewer to explore topology, see the related link at the bottom of this topic.

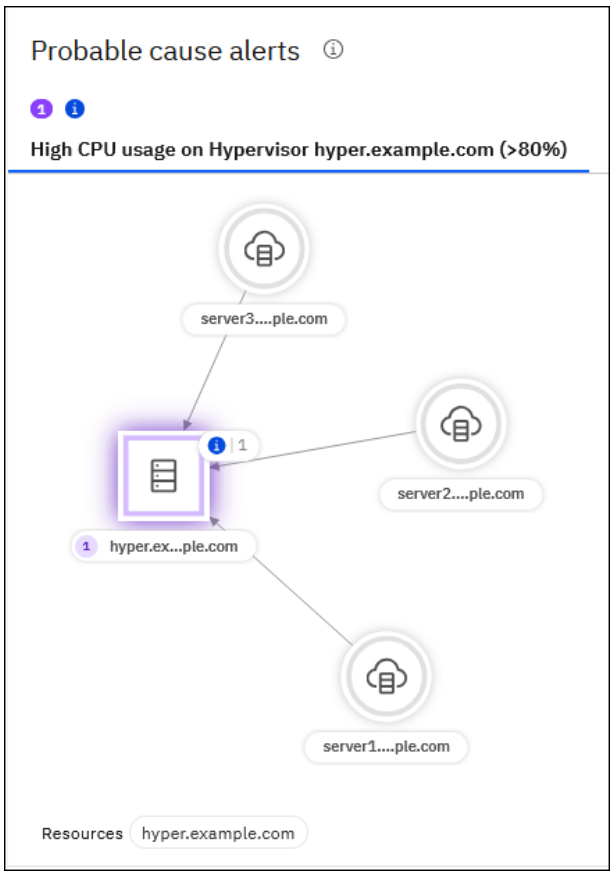


Figure. Alert topology

Note: From an incident's Alert Viewer, the Topology viewer opens in the **Topology** tab on the same page. From the general Alert Viewer, the Topology viewer opens in a separate browser tab.

Related links

Viewing a topology

You can use the interactive Topology viewer tool to drill into the individual resource details and status, enable a timeline to view changes over time, and more.

Displaying alert seasonality

Alerts can occur within a seasonal time window. For example, an alert might tend to occur on a certain day of the week or a certain time of the day. Alerts that tend to occur within a seasonal time window are highlighted in the Alert Viewer using a circle ○ or a circle with a check mark ⊘ in the alert's **Seasonal** column. Click these icons to find out when the alert tends to occur.

Examples of seasonal time windows include the following times:

Hour of the day

`Between 12:00 and 1:00 pm`

Day of the week

`On Mondays`

Day of the month

`On the 3rd of the month`

Day of the week at a specific hour

`On Mondays, between 12:00 and 1:00 pm`

Day of the month at a specific hour

`On the 3rd of the month, between 12:00 and 1:00 pm`

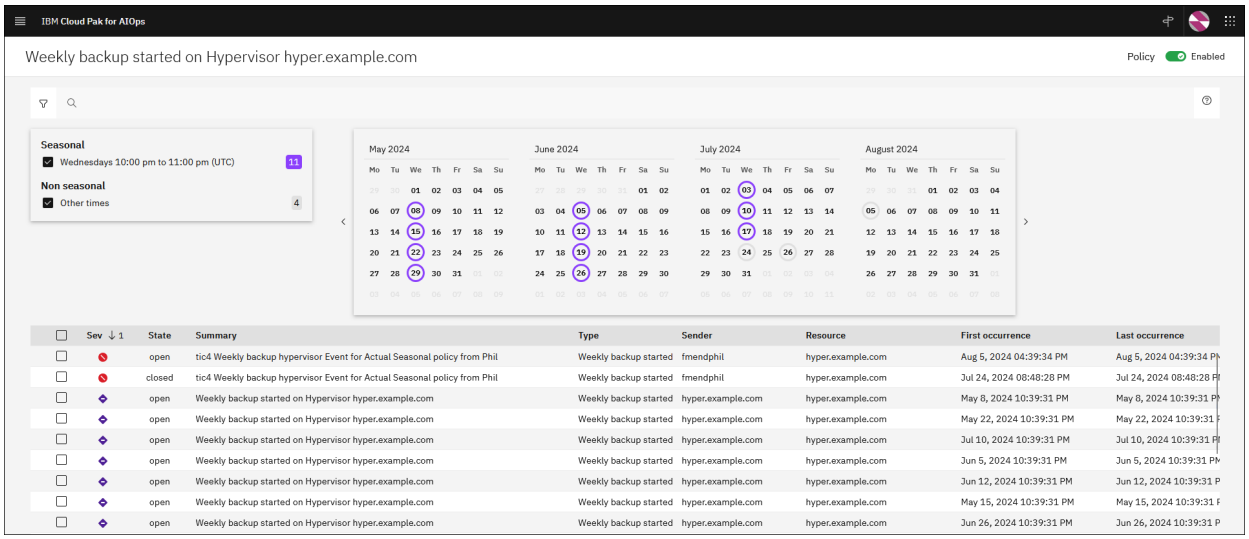
Procedure

1. Identify alerts that are seasonal.
   - Alerts that are not part of an alert group, that have associated seasonality, have a circle visible in the **Seasonal** column.
   - Alerts that are part of an alert group, that have associated seasonality, have a circle with a check mark in the **Seasonal** column, but you must first open the alert group in order to see the alert. Open the alert group by clicking the Down chevron icon in the parent alert group row.
2. Click the circle or dot to display more information on the alert of interest.

   A side panel that contains multiple information sections opens in the table. The **Seasonality** section is open and displays one or more seasonal time windows for the selected alert.
3. Click the Down chevron icon next to each seasonal time window to see a visual representation of the time window.

   For example, if the underlying algorithm has identified that the selected alert tends to occur between 12 noon and 1:00 pm, then this alert will have a seasonal time window as follows, and this will be displayed using a clock-based representation:

   `Between 12:00 and 1:00 pm`
4. Click the **More information** link at the end of the side panel's **Seasonality** section to launch out to the **Seasonality details** page.



Seasonality details

This page provides a calendar view of all of the historical alerts that contributed to the selected seasonal alert. One or more seasonal time windows are listed in the top left pane. Select a seasonal time window to filter the view to show just historical alerts that contributed to that time window. The **Seasonality details** page contains the following sections:

- **Toolbar**
  - Search: searches alert data in all event group instances that are shown on this page.
  - Filter: filters the alerts that are shown by severity level:
    - Critical
    - Major
    - Minor
    - Warning

- Informational
- Indeterminate

- **Seasonal time windows**

  One or more seasonal time windows are listed together with a number indicating how many historical alerts occurred within this time window during the historical period analyzed. The color of the square is the color code for the respective seasonal time window. Select a seasonal time window to filter the Calendar and the Historical alert table to show just historical alerts that contributed to that time window.

- **Calendar**

  A calendar of the historical period analyzed (the last three months or more) showing days on which the historical alerts of interest occurred, and colored with the color code for the respective time window. Click one or more days in the calendar view to filter the table of alerts details to the selected date.

- **Historical event table**

  Table listing the historical alerts during the historical period analyzed.

Displaying temporal correlation

The system continually analyzes past alerts to determine which alerts tend to frequently co-occur. When these alerts occur together again, they are correlated.

The temporal details page shows the alerts that make up a temporal group, together with all of the historical instances of the group. You can navigate into these instances to view detailed timelines of each instance. You can also view alert details.

Procedure

1. Identify alerts that are temporal in the Alert Viewer. You can click the Filter icon to filter by Grouping insights.
2. Click the **View correlation** icon to display the **Correlation** column in the table.
3. Clicking the temporal icon in this column opens the sidebar, with the **Temporal correlation** section open. This section contains the following information:

- **First group instance**

  Date and time of first instance of this group.

- **Total group instances**

  Total number of historical instances of this group. For details of when these instances occurred and how many alerts occurred in each instance, see the Group instance heatmap.

- **Average instance duration**

  Average time in seconds that this group instance lasted.

- **Group instance heatmap**

  Time-based heatmap showing recent historical period in days with a gray square for each day. Each darker square indicates a day on which there was at least one group instance. Hover over the square to see details of this group instance.

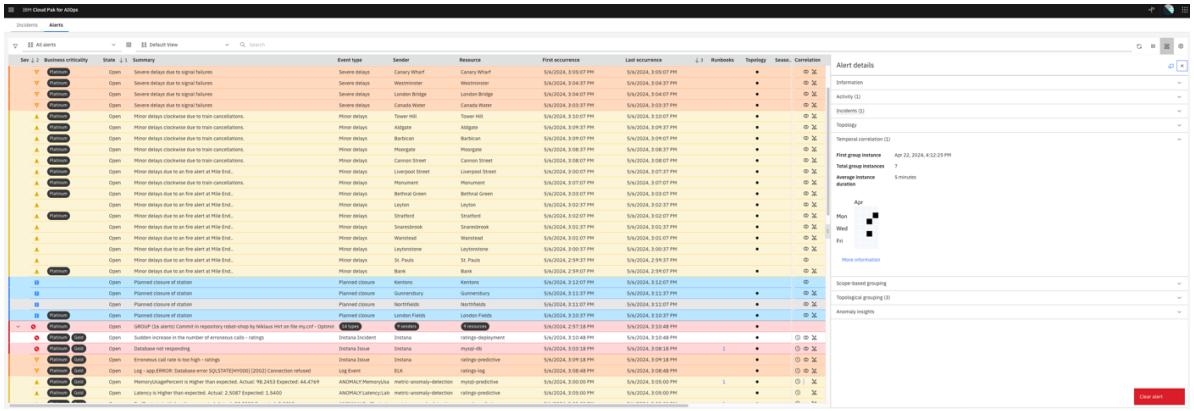
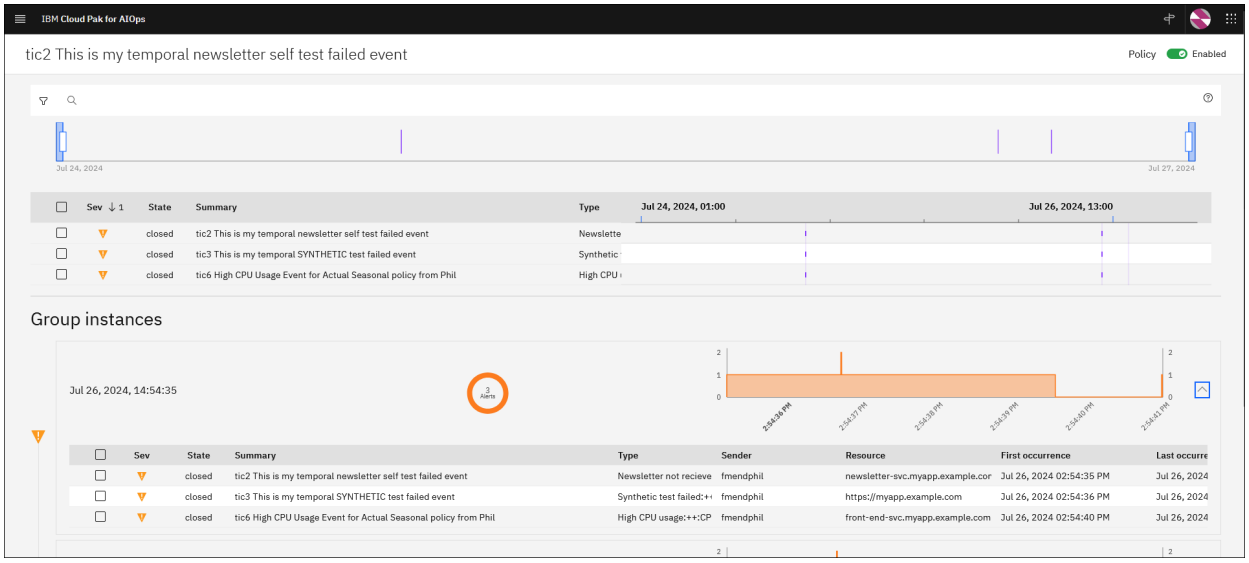


Figure. Temporal correlation section

4. Click the **More information** link at the end of the side panel's **Temporal correlation** section to launch out to the temporal details page.



temporal correlation details

The temporal details page shows the alerts that make up a temporal group, together with all of the historical instances of the group. You can navigate into these instances to view detailed timelines of each instance. You can also view alert details. The temporal details page contains the following sections:

- **Toolbar**
  - Search: searches alert data in all event group instances that are shown on this page.
  - Filter: filters the alerts that are shown by severity level:
    - Critical
    - Major
    - Minor
    - Warning
    - Informational
    - Indeterminate
- **Overview timeline**

  Displays alert group instances over time and controls the display of alert group instance data on the rest of the page. By default the time range sliders are open sufficiently to show data on all alert group instances. Modify the time range by either clicking and dragging over the desired range inside the timeline, or by dragging the sliders to the desired range. The rest of the screen updates accordingly.
- **Alert group instance timeline**

  Displays all of the alerts that have historically participated in instances of this temporal alert group. The instance map provides a graphical view over time of when the various instances have occurred.
- **Alert group instance details**

  Displays the following information for each alert group instance:
  - Start date and time of event group instance: indicates the first occurrence value of the first alert in the alert group instance.
  - Distribution of event severity values: pie chart providing a visual indication of the alert severity values. Hover over the pie chart for more details.
  - Sparkline: chart of alert occurrence over time.
  - Duration of event group instance: duration of the alert group instance, in text.
  - Down chevron icon: click the down chevron icon ⌄ to see an alert table showing column details for each alert in this group instance.

Log anomaly alerts

Log anomaly alerts have a **Sender** attribute of **Log Anomaly** if you use natural language processing and statistical baseline methods. If you use log anomaly detection - golden signals, the **Sender** attribute is **log-metric-anomaly-detection**.

Types

For log anomaly alerts, the **Type** attribute shows which of the log anomaly models detected the anomaly:

- Natural language anomaly
- Statistical baseline anomaly

Summaries

The **Summary** attribute of an alert provides more detail about why an anomaly was detected.

A summary might look similar to the following example:

```
Logs containing errors for component: QMA with a primary message code of AMQ9665E. Evidence includes: domain-specific entities.
```

- The **Evidence includes** statement can show the following values:
  - `domain-specific entities`, for example, message codes
  - `embeddings (statistical baseline)`
  - `error entities`, for example, HTTP error codes and exceptions
  - `patterns + embeddings (natural language)`
- A **primary message code**, such as `AMQ9665E`, is included only for alerts where an anomaly is detected in logs that originate from a domain-specific source, such as IBM MQ or WebSphere.

Categorization of log anomaly alerts

During the training phase, templates are extracted based on the input logs. Models are trained to represent the normal behavior of a component or service in an incident-free window. In the inference, any incoming new live log data is matched against the templates that were extracted during training, and the distribution for a fixed window is computed. Any new logs, error logs, or deviation from the normal distribution from training might be flagged as an anomalous window and generate a log anomaly event.

A template is classified as an error by using keyword matching with a predefined error dictionary and error symptom words.

Alerts that are generated for log anomalies come in two types:

- Alerts with the `Logs containing errors for` prefix are log anomaly alerts that are generated when an error template is included in the alert. Templates include internal lists of error keywords. For example, a template can be classified as `Logs containing errors for` when it includes the keywords `expires`, `failure`, or `incomplete`.
- Alerts with the `Abnormal behavior in the logs for` prefix are log anomaly alerts that are generated when no error templates are included in the alert.

Explainability

An alert is generated when a log anomaly is detected. The metadata for an alert contains fields and information that can help you understand the identified log anomaly.

See the following alert field descriptions to better understand reported log anomalies:

- `description`: If error templates were found in the ten-second window and if the count of these templates is greater than `0`, these templates are captured in the `description` field. If error templates are not present after an internal ranking algorithm is applied, the top five templates that have a count value greater than `0` are included in this field.
- `details`: This field includes the `text_dict` field, which contains critical fields, such as `template_ids` and `template_list`. This field includes the sorted templates and entities based on an internal ranking algorithm that corresponds to the log anomaly. This ranking algorithm prioritizes error templates and low occurrence error templates. It also prioritizes error entities over other types of entities. Based on the ranking, only the top 5 templates are shown.

Microsoft Teams and Slack integrations

If a Microsoft Teams or Slack integration is set up, the alerts are also sent to those integrations. The same alert data is parsed and shown in Microsoft Teams or Slack.

- The Microsoft Teams or Slack UI parses and displays the templates that are associated with the alert.
- Use the **preview logs** feature to see a sample of the logs that are associated with a template.
- When you click **Attach log message patterns** on the Microsoft Teams or Slack UI, you receive all anomalous and normal logs in the ten-second window time range for the resource, also known as the application component, that is associated with the alert.
- For more information, see Configuring Slack applications for integration and Creating a Microsoft Teams integration.

High risk alerts from change risk

When a new, incoming change request is evaluated as high risk, an alert is posted to the IBM Cloud Pak® for AIOps Platform UI. It can be included in the Events report and can be grouped into an incident. By reporting these alerts, high risk insights can be used for incident avoidance and mitigation.

A change risk high risk alert can be identified in the Alert Viewer when the **Sender** column for an alert contains `change-risk`. So a high risk alert resembles this example:

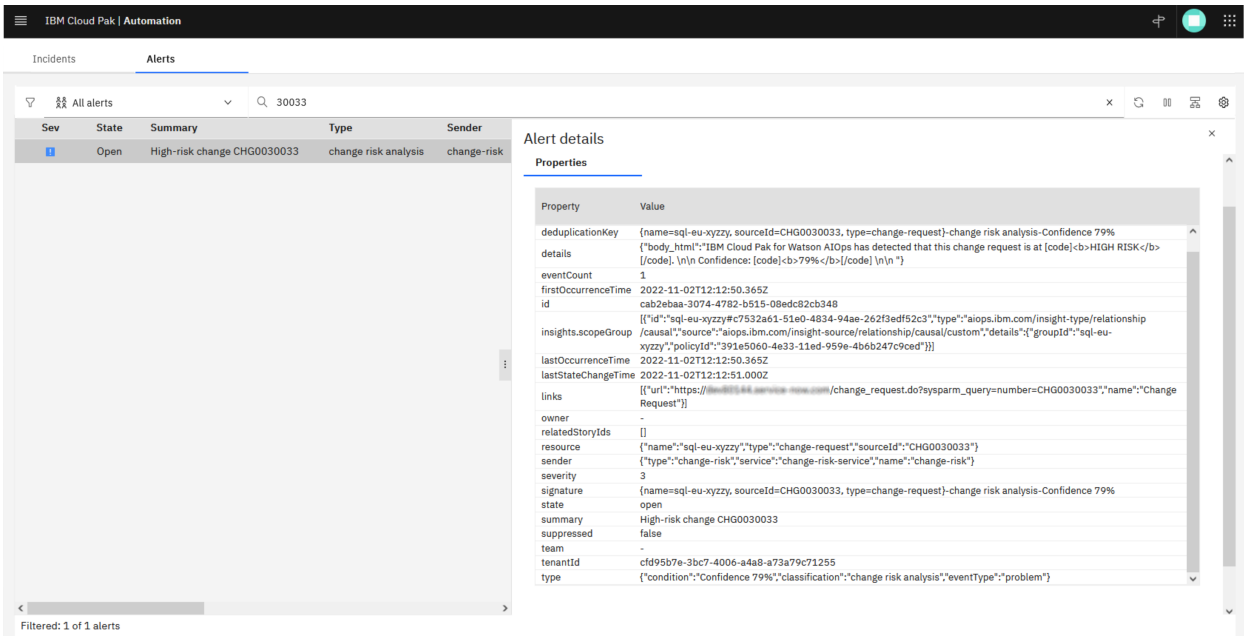


Figure. Alert viewer high risk

High risk in Events example:

```
id: UUID (str)
 sender: {
     service: "change-risk-service"
     name: "change-risk",
     type: "change-risk"
 }
 resource: {
     name: str, optional - Service Now CMDB ID (if present)
     sourceId: str - Change number (e.g. "CHG12345")
     type: "change-request"
 }
 type: {
     classification: "change risk analysis"
     condition: str - Confidence level (e.g. "Confidence 94%")
     eventType: "problem"
 }
 severity: 3 (int)
 summary: str - Change reference (e.g. "High-risk change CHG1")
 details: {
     body_html: str - HTML-marked English description of the change, related changes, and related incidents.
 }
 occurrenceTime: datetime (str, UTC, ISO 8601 with timezone) - Time the risk evaluation was processed
 expirySeconds: 0 (int)
 links: [
     {
         name: "Change Request",  (e.g. 'Change Request')
         url: str (e.g. 'https://example.service-now.com/change_request.do?sysparm_query=number=CHG1')
     }
 ]
 insights: [] (empty list)
```

Viewing metric anomaly details

In **Incidents and alerts**, alerts can appear in the Alert Viewer table to indicate metric anomaly detection. More detailed information on these alerts, and a Key Performance Indicator (KPI) timeseries chart, can be viewed by opening the *Alert details* panel.

KPI: A KPI is made up of a resource, which is a physical entity such as a network interface card, and a metric. For example, if a server has 2 network interface cards and you want to analyze data for 10 metrics on each network card in the server, the total number of KPIs is 20. KPIs are often interchangeably referred to as metrics, so a metric can sometimes refer to a measurement that is taken for a type of device - for example **`CPU_Context_Switches`**, and sometimes it can refer to an instance of that metric - for example **`CPU_Context_Switches`** on server **`ABC.acme.com`**.

Alert count metric anomalies: Default alert expectations are calculated by the metric manager capability based on the input data. If an anomalous alert rate is detected that is either higher or lower than expected, an alert is raised and displayed as a **`Metric anomaly`** in the Alerts table. When selected, the *Alert details* window depicts the expected values against the actual values. If required, an administrator can disable this specific functionality by editing the **`feature-flags-configmap.yaml`** file, and setting the **IBM_IR_AI_ALERT_COUNT_METRICS_ENABLED** flag to **`false`**.

Procedure

1. In the Alert Viewer (**Incidents and alerts** > **Alerts** tab) click anywhere on the row of a metric anomaly alert (identified by ANOMALY displayed in the *Type* column in the table).

2. An *Alert details* side panel opens. By default, the *Metric anomaly details* chart displays, showing the properties of the alert and associated values.
3. The chart plots the activity of the anomaly over a 2-day period (the two days up to the date in the date field).
4. You can increase the viewing size of the chart within the panel:
    - Click the 3 vertical dots in the gray tab on the left side of the panel. Then, drag the tab to the left.
    - Select a **Graph start date** and **Graph end date** to view data for the selected date range.

Understanding the chart

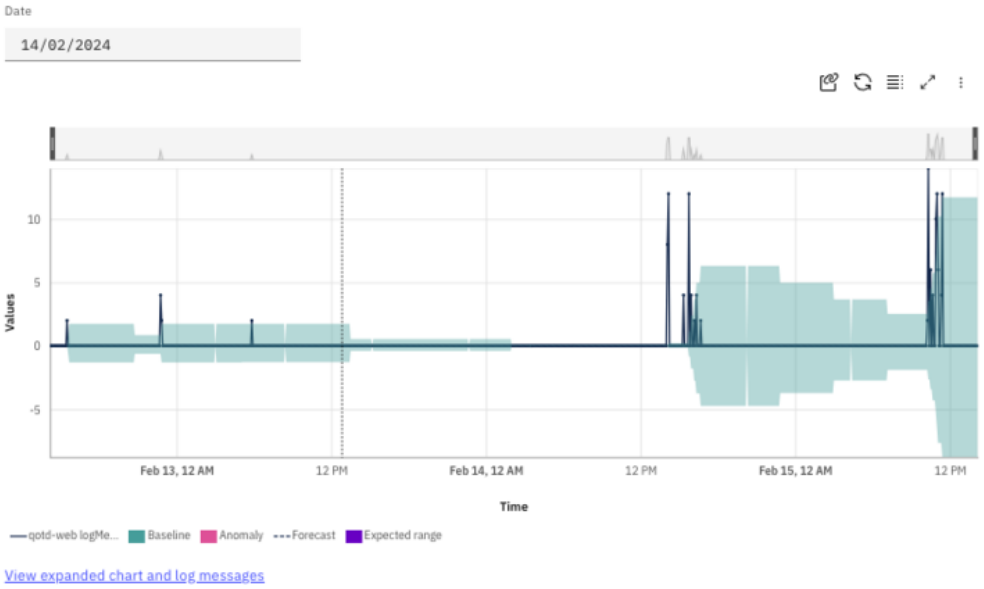


Figure. 2-day chart

- The unbroken black line that is labeled with the name of the metric resource that is plotted on the chart represents the values of the selected metric over time.
- A green-shaded area (labeled *Baseline*) represents the acceptable operating range of that anomaly.

  Note: The default KPI Baseline is always that of the primary metric. To change to another Baseline, from related alerts or additional metrics, click the radio button (blank circle) attached to another KPI. Making this Baseline adjustment also changes the KPI that is forecast. If you select multiple metrics, you can choose to view only one Baseline from all of the KPIs that are in the related alerts and additional metrics tabs.
- A red-shaded area or red vertical line (labeled *Anomaly*) indicates where the metric is anomalous: one or more of the anomaly detection algorithms determined the metric was not behaving in accordance with the learned normal behavior.
- A purple area in the chart (labeled *Expected range*), with a dashed line (labeled *Forecast*) indicates forecasted metric anomalies. For more information, see Metric anomaly forecasts.

  Note: Sometimes the purple range is not shown, or is only partially shown. This is because the range becomes too wide to show it. The dashed line denoting the forecast will still be shown unless there is some reason that the forecast is not produced, for example if there are too many gaps in the timeseries, or if data availability is too low.

Tooltip: Hover the cursor over any part of a time-series graph to view tooltip information that is related to multiple anomalies at a particular point in time.

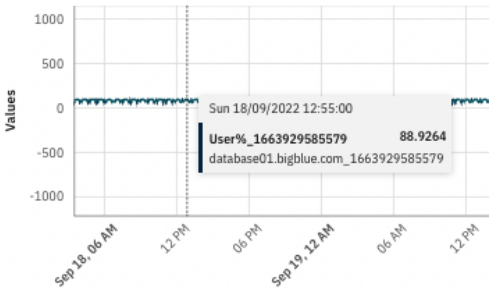


Figure. Tooltip

Chart Icons

Icon	Name	Description
	Copy link	Copies URL link of chart to clipboard.

Icon	Name	Description
	Show as table	Tabular representation of anomaly, including occurrence times and values, which can be downloaded as a CSV or JSON file.
	Make fullscreen	Maximizes size of chart to fit onto full page.
	More options	Allows chart to be exported as a CSV, JSON, PNG, or JPG file. PNG and JPEG options are only available when you use the Chrome browser.

Zooming in on the timeline

On the main chart, place and click the "+" "cursor at the beginning of the time period you want to focus on, then drag it along the timeline, to your chosen endpoint. The main chart adjusts to the selected timeline.

For a more controlled zoom, use the zoom scale (miniaturized chart on top of the main chart) to focus on a particular date range. Select and move the vertical slider black bars, left or right:



Figure. Metric anomaly chart zoom

Expanding the timeline of the chart

- For metric anomaly detection alerts, click **View expanded chart** to change the 2-day timeline in the side panel to a larger, 7-day timeline chart.
- For **Log Anomaly - Golden Signals** metric anomalies, click **View expanded chart and log messages**.

The **Metric anomaly details** page opens.

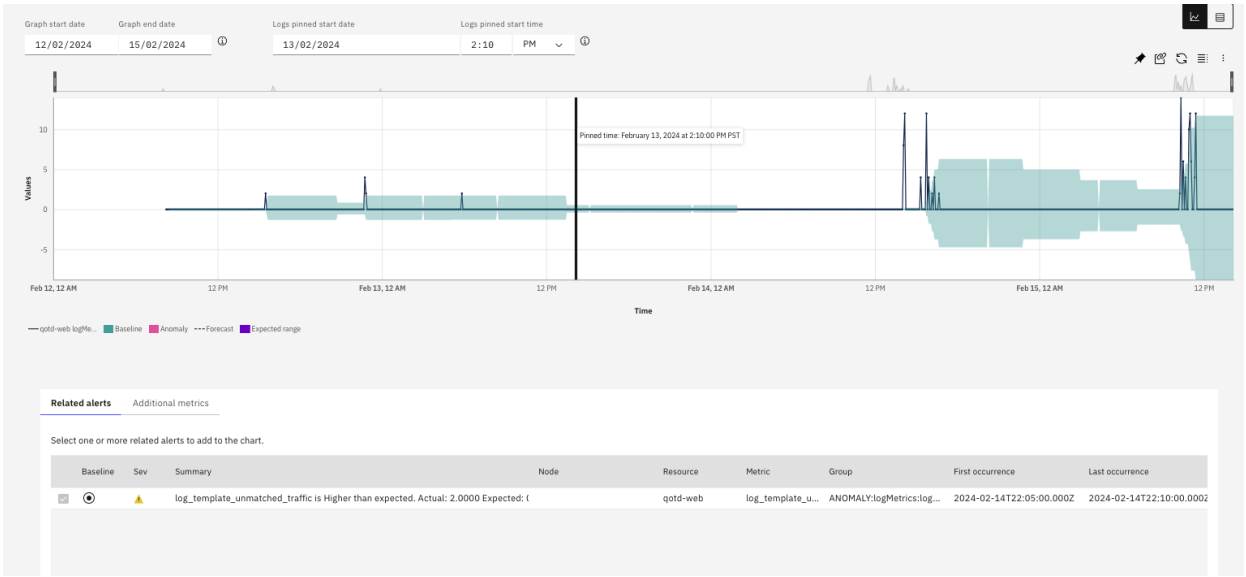


Figure. The metric anomaly expanded chart with the Related alerts and Additional metrics tabs

Metric anomaly graph

From the **Metric anomaly details** page for **Log Anomaly - Golden Signals** metric anomalies, the **Metric anomaly graph** section includes a **Graph view** and a **Logs view**.

- Both the **Graph view** and **Logs view** include filters to specify the date and time you want to display.
  - Select a **Graph start date** and **Graph end date** to view data for the selected date range.
  - Select a **Logs pinned start date** and a **Logs pinned start time** to specify a pinned time in the graph. Click **Pin logs start time** or **Unpin logs start time** to pin or unpin the specified time. Pinning can help you see how the logs in the **Logs view** correspond to the graph in the **Graph view**.
- The **Graph view** is displayed by default and shows the timeline. Baseline data is omitted where the value on the graph is zero.
- Click the **Logs view** to view a table of log messages that correspond to the alert and template ID.
  - To filter the logs that are shown, specify the date and time or search for keywords.
  - Download the logs by clicking **Export to CSV file**.
  - If the alert was derived from live logs that can be found in the log aggregator, you can click **Launch log aggregator** to open a log aggregator at the pinned start time.

Related alerts table: If alerts related to the primary metric are detected, then a tab for *Related alerts* appears in the expanded view of the chart. When this tab is selected, a table of related alerts appears. Up to 4 related alert metrics can be added to the chart as KPIs, in addition to the main metric KPI. For more information, see Related alerts table.

Additional metrics

Use the *Additional metrics* tab to add up to another 3 KPIs to the Metric anomaly details chart.

1. Click **Add metric**.
2. Select an item from each field: *Metric group*, *Metric*, and *Resource*.
3. When selection is complete, the timeline of the KPI is added to the chart.
4. Repeat steps 1-3 up to 2 more times.

To modify a selection, remove the current item by clicking × in the *Metric group*, *Metric*, or *Resource* fields. Then, click inside the empty field, and a menu displays all available items for that field.

To delete a KPI, click × in the *Metric group* field or the ⊖ icon.

Note: Entering part of a string in a field filters the menu so that it displays only the items that contain that element. In the following example, in the *Resource* field, entering 02 brings up all resources that contain the string 02.

Related topics

- Metric anomaly forecasts
- Viewing multiple metrics on the metric anomaly chart

Metric anomaly forecasts

Forecasting can indicate when an alert can become critical and help avoid an outage. Use the metric anomaly details chart to view forecasts for a metric anomaly.

Procedure

Open the metric anomaly details chart, as normal (see Viewing metric anomaly details, for more information).

The forecasted trend is the area of the chart that has a dashed (--) line rather than an unbroken (–) line, which indicates historical data. The dashed line might have a purple "confidence interval" around it that shows the likely range of values for the forecast. In some cases, this confidence interval is not shown due to a high level of error that would result in large confidence intervals.

To get more a detailed view of where historical data ends and forecasted data begins, expand the chart by

- clicking and dragging the gray tab with the three vertically aligned dots on the side panel
- or clicking the maximize icon
- or clicking **View expanded chart**

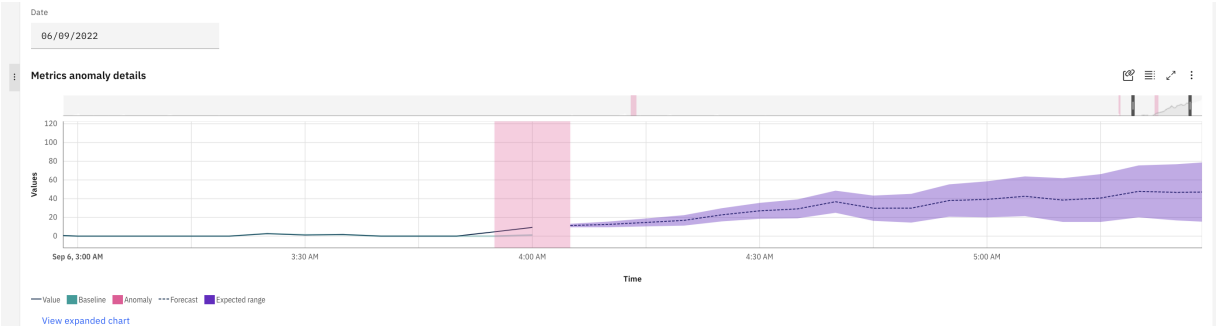


Figure. Expanded and zoomed

When in an enlarged view, you can continue to zoom in on the chart, as normal.

Viewing multiple metrics on the metric anomaly chart

Add multiple metrics (KPIs expressed in a timeseries) to a metric anomaly chart by using the *Related alerts* table or the *Additional metrics* tab. **Note**: The Related alerts table is available when the alert is associated with an Incident.



Figure. Related alerts tab

Procedure

Two methods are available to view the *Related alerts* table. Use the Alert viewer or the Incident viewer.

- To access from Alert viewer
  1. In the **Alerts** tab, open the side panel that contains the metric anomaly details. If multiple alerts are grouped or in an Incident, they are shown in the **Related Alerts** table. For more information, see Viewing metric anomaly details.
  2. Click the **View expanded chart** link beneath the chart.
  3. In the expanded view, any alerts (metric and nonmetric) related to the selected alert are listed in the *Related Alerts* table.
- To access from Incident viewer
  1. In **Incidents and alerts**, click the **Incidents** tab.
  2. For the Incident metric you want to investigate, hover over the text in the *Title* column and click the link.
  3. The Incidents *Overview* page opens. Click a metric anomaly in the *Probable cause alerts* or *Top alerts* (when probable cause ranking is not available) column and the metric anomaly chart opens.



Figure. Metric anomaly chart

  4. Click ⧉ to open the chart in its own window.
  5. If related alerts are present, they are listed in the *Related alerts* table. Depending on your screen size, you might have to scroll down to view this table.

Related alerts table

Use the checkboxes in this table to enable or disable Key Performance Indicators (KPIs) on the chart.

Note: A KPI is made up of a resource, which is a physical entity such as a network interface card, and a metric. For example, if a server has 2 network interface cards and you want to analyse data for 10 metrics on each network card in the server, the total number of KPIs is 20. KPIs are often interchangeably referred to as

metrics, so a metric can sometimes refer to a measurement that is taken for a type of device - for example **`CPU_Context_Switches`**, and sometimes it can refer to an instance of that metric - for example **`CPU_Context_Switches`** on server **`ABC.acme.com`**.

The checkbox of the main KPI is enabled by default and cannot be deselected. The Baseline radio button is also selected by default, but it can be deselected by clicking the radio button (circle icon) of another KPI.

Up to 4 KPIs from the *Related alerts* table can appear on the timeseries chart at the same time.

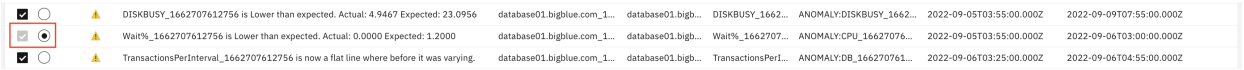


Figure. Multi-Metric chart 2

Note: You cannot apply multiple baselines to a chart. Only one baseline can appear at any one time.

Tooltip: Hover the cursor over any part of a time-series graph to view tooltip information that is related to multiple anomalies at a particular point in time.

First occurrence

When a nonmetric alert is selected on the *Related Alerts* table, a blue vertical line or bar can display, if it is in the same timeline as the metric data displayed. The line indicates the first occurrence of this alert in the selected time period.

Additional metrics

To find and add up to 3 more KPIs to the chart, do the following.

1. Click the *Additional metrics* tab.
2. Find a particular metric by selecting an available value from each field, *Metric group*, *Metric*, and *Resource*.
3. When selection is complete, the KPI is added to the chart.

To modify a selection, remove the current item by clicking ✕ in the *Metric group*, *Metric*, or *Resource* fields. Then, click inside the empty field and a drop-down displays all available items for that field.

To delete a KPI, click ✕ in the *Metric group* field or the ⊖ icon.

Note: Entering part of a string in a field filters the drop-down menu so that it displays only the items that contain that element. In the following example, in the *Resource* field, entering 02 brings up all resources that contain the string 02.



Normalization

The different scales of different metrics can make it difficult to compare them when viewing absolute values together. For example, some metrics can have values of less than 100, while others can have values of more than 10 million. Normalization is used to accommodate comparison, so the scale-bar for the y-axis is not shown. To view absolute values of a metric, uncheck selected metrics until all remaining metric time-series are on the same scale, with the y-axis visible.

Note: If a tooltip is applied to the chart, the absolute value displays, not the normalized value.

Related topics

- Viewing metric anomaly details chart
- Metric anomaly forecasts

Using Metric search to create a timeseries chart

Visualize one or more KPIs, with or without an anomaly, in a Key Performance Indicator (KPI) timeseries chart.

Note: A KPI is made up of a resource, which is a physical entity such as a network interface card, and a metric. For example, if a server has 2 network interface cards and you want to analyse data for 10 metrics on each network card in the server, the total number of KPIs is 20. KPIs are often interchangeably referred to as metrics, so a metric can sometimes refer to a measurement that is taken for a type of device - for example **`CPU_Context_Switches`**, and sometimes it can refer to an instance of that metric - for example **`CPU_Context_Switches`** on server **`ABC.acme.com`**.

Procedure

1. Go to the Cloud Pak for AIOps console.
2. In the **Overview** menu, click **Metric search**.

3. On the **Metric search** page, refine your search to a specific KPI by selecting options from the *Metric group*, *Metric*, and *Resource* menus. The date that is selected represents the endpoint of the chart. KPI.

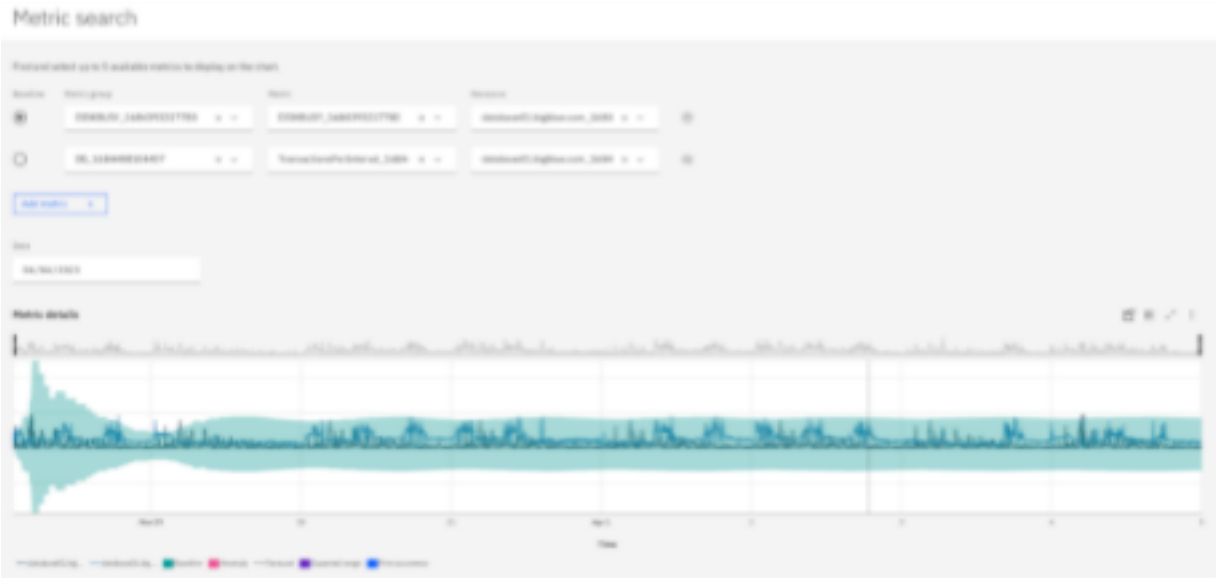


Figure. Metric search panel

Adding more KPIs to the chart

View up to 5 KPIs on the same timeseries chart.

Assuming that your first-searched metric is plotted on the chart as the primary KPI.

1. Click **Add metric**.
2. Select an item from each field: *Metric group*, *Metric*, and *Resource*.
3. When selection is complete, the KPI timeseries is added to the chart.
4. Repeat steps 1-3 up to 3 more times.

To modify a selection, remove the current item by clicking × in the *Metric group*, *Metric*, or *Resource* fields. Then, click inside the empty field and a drop-down displays all available items for that field.

To delete a KPI, click × in the *Metric group* field or the ⊖ icon.

Note: Entering part of a string in a field filters the drop-down menu so that it displays only the items that contain that element. In the following example, in the *Resource* field, entering 02 brings up all resources that contain the string 02.



Understanding the chart

- The chart displays 7 days of data by default. You can zoom out to see all historical data, including historical data for KPIs that you added to the chart.
- An unbroken black line that is labeled with the name of the KPI that is plotted on the chart represents the values of the selected KPI over time.
- A green-shaded area (labeled *Baseline*) represents the acceptable operating range of that KPI.

Note: The default KPI Baseline is always that of the metric first searched. To change to another Baseline, click the radio button (blank circle) attached to another KPI. Making this Baseline adjustment also changes the KPI that is forecast.

- A red-shaded area or red vertical line (labeled *Anomaly*) indicates where the metric is anomalous: one or more of the anomaly detection algorithms determined the metric was not behaving in accordance with the learned normal behavior.
- A purple area in the chart (labeled *Expected range*), with a dashed line (labeled *Forecast*) indicates forecasted metric anomalies. For more information, see Metric anomaly forecasts.

Related topic

- Viewing metric anomaly details chart
- Metric anomaly forecasts

Blocking, unblocking, and purging metrics

Learn how to add a resource to a blocklist so that it is not processed by metric anomaly detection. You can remove a resource from a blocklist to resume processing metric data. You can also purge all metric data from listed resources.

Blocking metrics

1. Find the metric group that you want to block and click the overflow menu.
2. Select **Block resource**. The **Resource added to blocklist** notification appears.

Unblocking metrics

1. To unblock a resource, click **View resource blocklist**. The **Resource blocklist** window appears.
2. For the resources that you want to unblock, select **Remove from blocklist**.
3. Click **Save**. The resources are unblocked.

Purging metrics

1. Find the metric group that you want to delete and click the overflow menu. Select **Purge metric data**. The **Purge metric data from selected resource** window appears.
2. Confirm purge and click **Purge**. The metric data is purged.

Purge metrics from the **Resource blocklist** window:

1. Click **View resource blocklist**. The **Resource blocklist** window appears.
2. Click **Purge all metric data from listed resources**.
3. Click **Save**. The **Purge metric data from selected resource** window appears.
4. Confirm purge and click **Purge**. The metric data is purged.

Displaying analytics details for an alert group

On alert lists associated to an incident, click the option to show the correlation information to understand why these alerts were grouped together. This option displays the underlying temporal, topological, and scope-based groups.

Procedure

1. In the Alert Viewer, click the Click the **View correlation** icon to display the **Correlation** column in the table.
   - **Temporal group column**

     Based on alert history, the alerts in this column that are marked with a large dot ● tend to occur within a short time of each other.
   - **Scope-based group column**

     The alerts in this column that are marked with a large dot ● occur within a configurable time window on an administrator defined scope, such as a location, service, or resource.
   - **Topological group column**

     The alerts in this column that are marked with a large dot ● occur on resources within a predefined section of your network topology.
2. Click a dot ● to see more details on any of these sub-groups.

Temporal group column

Clicking a dot ● in this column opens the sidebar, with the **Temporal correlation** section open. This section contains the following information:

- **First group instance**

  Date and time of first instance of this group.
- **Total group instances**

  Total number of historical instances of this group. For details of when these instances occurred and how many alerts occurred in each instance, see the Group instance heatmap.
- **Average instance duration**

  Average time in seconds that this group instance lasted.
- **Group instance heatmap**

Time-based heatmap showing recent historical period in days with a gray square for each day. Each darker square indicates a day on which there was at least one group instance. Hover over the square to see details of this group instance.

Scope-based group column

Clicking a dot ● in this column opens the sidebar, with the **Scope-based** section open. This section contains the following information:

- Group duration

  Duration of the scope-based group. For example: "These alerts were found to share a cause as they all occurred on the same resource within **N** minutes of each other." **N** being the configured time.

Topological group column

Clicking a dot ● in this column opens the sidebar, with the **Topological** section open. This section contains the following information:

- **Topology group name**

  Name of the topology defined in the topology management service, on which this topology group is based.

- **Topology**

  Pane showing the resources in the topology on which this topology group is based. You can perform the following actions on the topology:

  - Click the resource to displays the relationship between that resource and neighboring resources. The relationships are displayed in text on the lines connecting the resources. Examples of relationships include: runsOn, members, exposes.
  - Right-click the resource to display the context menu. For more information about the actions available from the context menu, see see Viewing a topology.

Viewing incidents that are displayed in outgoing integrations

View incidents in outgoing integrations.

When a new anomaly is detected, IBM Cloud Pak for AIOps generates an incident. If an outgoing integration is configured, the incident is sent to ServiceNow, Slack, or Teams.

- In ServiceNow, an incident is created, and a link to the incident is added to the IBM Cloud Pak for AIOps incident and to the Slack or Teams message if those integrations are also configured.
- In Slack or Teams, the IBM Cloud Pak for AIOps incident appears as a message, and a link to the message is added to the IBM Cloud Pak for AIOps incident.

Viewing a ServiceNow incident

1. See the incident section.

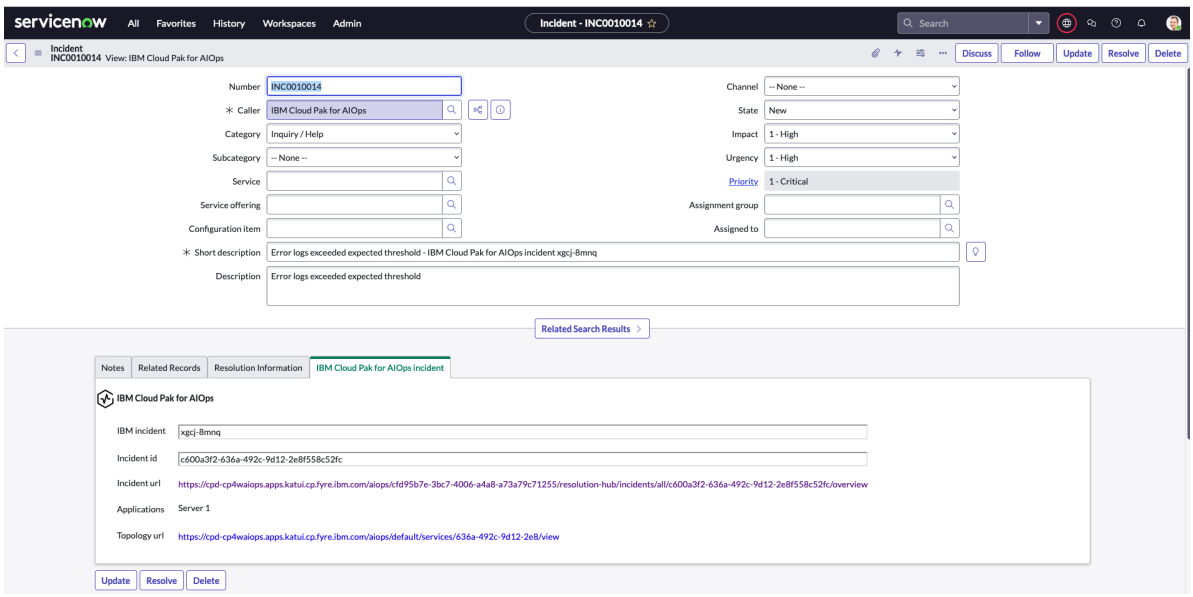


Figure. ServiceNow incident overview

Number: The incident number identifies the incident across IBM Cloud Pak for AIOps, Slack, and ServiceNow.

Caller: The caller is the user who was used to set up the ServiceNow Integration in IBM Cloud Pak for AIOps.

Impact and Urgency: Impact and urgency are set to map the severity of the incident to the priority field of the incident in ServiceNow.

Short description and **Description**: These values are set to the title and long description of the incident. You can change the values in the ServiceNow ticket as well, but the changes are not sent back to IBM Cloud Pak for AIOps.

2. See the IBM Cloud Pak for AIOps section.



Figure. IBM Cloud Pak for AIOps incident details in ServiceNow

Incident number and ID: The incident number and ID are used to identify the incident in IBM Cloud Pak for AIOps.

Incident URL: The incident URL is a link to the incident in the IBM Cloud Pak for AIOps UI. Click the URL to open the IBM Cloud Pak for AIOps UI to the corresponding incident.

Applications: This value lists the names and URLs of the applications that the incident is associated with. If no applications are detected, **`N/A`** is shown.

Topology URL: This URL links back to the IBM Cloud Pak for AIOps UI to show topology details.

3. See the IBM Cloud Pak for AIOps incidents tab.



Figure. The IBM Cloud Pak for AIOps incidents tab in ServiceNow

This information is the same as in the IBM Cloud Pak for AIOps section. Click the incident number link to go to the IBM Cloud Pak for AIOps table and view all the IBM Cloud Pak for AIOps incidents in ServiceNow.

4. See the IBM Cloud Pak for AIOps alerts tab.

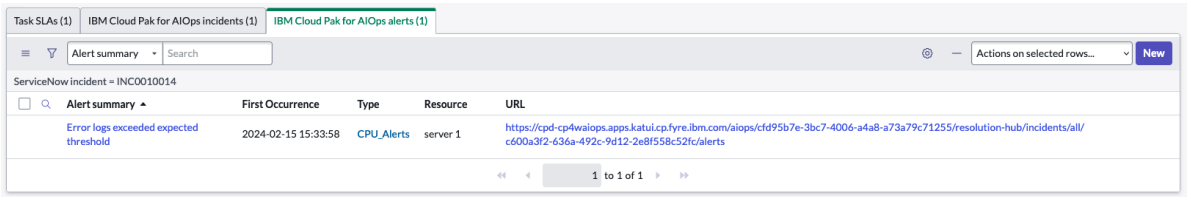


Figure. IBM Cloud Pak for AIOps alerts tab in ServiceNow

IBM Cloud Pak for AIOps incidents are made from a collection of alerts. View alerts that contributed to this incident in the IBM Cloud Pak for AIOps alerts tab.

Event enrichment

Event enrichment is the process of enhancing alerts or incidents by updating their standard attributes or populating custom attributes such as, adding, updating, or removing them. This process is driven by user-defined logic, allowing for tailored enrichment to meet specific needs.

Event enrichment in IBM Cloud Pak for AIOps is done by using Netcool/Impact. For more information about event enrichment, see Event enrichment.

For more information about how to develop an event enrichment solution by using Netcool/Impact, see Event enrichment tutorial.

Event mapping and alert severity

The following table identifies the mappings from IBM Netcool Operations Insight **`alerts.status`** table columns to the IBM Cloud Pak for AIOps event that is sent to lifecycle for processing.

Creating a Netcool integration is a prerequisite to synchronize your event data into IBM Cloud Pak for AIOps. For more information, see Creating Netcool integrations.

The second table represents the differences in alert severities between IBM Cloud Pak for AIOps and IBM Netcool Operations Insight.

Event mappings from IBM Netcool Operations Insight to IBM Cloud Pak for AIOps

IBM Cloud Pak for AIOps property	IBM Netcool Operations Insight column	Comment
id	N/A	IBM Cloud Pak for AIOps generated identifier for event.
occurrenceTime	LastOccurrence	
summary	Summary	
severity	Severity	Severity is remapped to IBM Cloud Pak for AIOps scale.
type.classification	[EventId + ':++:' +] AlertGroup	If no EventId is provided, only the value from AlertGroup is used. The separator (:++:) is only present if EventId contributed to the classification value and indicates the extent of EventId and AlertGroup in the string.
type.eventType	Type	
sender.service	Agent	
sender.name	Manager	If the Manager column is not defined, it is set to unknown.
resource.name	Node	If the Node column is not defined, it is the value of NodeAlias column if one has been defined.
resource.hostname	Node	If value is hostname, or NodeAlias if hostname and Node are not defined.
resource.ipaddress	NodeAlias	If the value is an IP address, otherwise not set.
resource.location	Location	
resource.sourceId	BSM_Identity	
resource.service	Service	
resource.port	PhysicalPort	
resource.physicalslot	PhysicalSlot	
resource.physicalcard	PhysicalCard	
resource.interface	LocalPriObj or LocalRootObj	If the value starts with "ifEntry." or "ospfIfEntry.", the trailing interface details will be mapped. Example: "ifEntry.eth0". **Note**: Where values exist for both LocalPriObj and LocalRootObj, the LocalPriObj value will be used, as it is the primary object element of topology. If it has no value, but LocalRootObj is populated, then this will contribute to resource mapping information.
resource.component	LocalPriObj or LocalRootObj	If the value starts with "entPhysicalEntry.", the trailing component details will be mapped. Example: "entPhysicalEntry.name" **Note**: Where values exist for both LocalPriObj and LocalRootObj, the LocalPriObj value will be used, as it is the primary object element of topology. If it has no value, but LocalRootObj is populated, then this will contribute to resource mapping information.
resource.scopeId	ScopeID	
expirySeconds	ExpireTime	
details	alerts.details	The set of details that are stored in details table for event as name-value pairs.

Differences between alert severities in IBM Cloud Pak for AIOps and IBM Netcool Operations Insight

Alert	IBM Netcool Operations Insight severity	IBM Cloud Pak for AIOps severity	Comment
Critical	5	6	
Major	4	5	
Minor	3	4	
Warning	2	3	
Informational	N/A	2	No informational level in IBM Netcool Operations Insight.
Indeterminate	1	1	
Clear	0	N/A	IBM Cloud Pak for AIOps alerts have a state property that indicates an alert is open, clear, or closed. Allows alerts original severity to be maintained.

Managing ChatOps insights

The insights produced by IBM Cloud Pak® for AIOps surface in ChatOps. Whether you use Slack or Microsoft Teams as your ChatOps interface, the experience is consistent with the IBM Cloud Pak for AIOps Platform UI.

For more information about setting up ChatOps integrations, see:

- Creating Microsoft Teams integrations
- Creating a ServiceNow integration
- Creating Slack integrations

Prerequisites

When you configure your ChatOps integration, you must set up two types of channels for your insight flow

- A *reactive* channel for your incidents and alerts
- A *proactive* channel for notifications (from change risk)

Verifying your ChatOps integration

After you create your ChatOps integration, you can use the *reactive* channel to verify that the integration is up and running, by tagging the bot. For example, `@<bot_name>` hello.

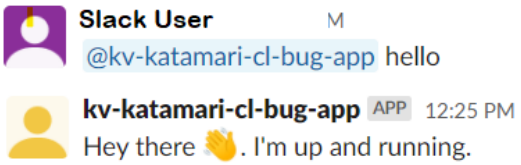


Figure. Slack hello

To get the botName, see Configuring Slack or Configuring Microsoft Teams for more details.

In Microsoft Teams, when you enter **`hello`** the response is **`I'm sorry, I don't understand the message`**. This response indicates that although it does not understand the message, the message was received, so the integration is verified.

Welcome tour

In Slack, after you are added to the *reactive* channel, you will receive a welcome message.

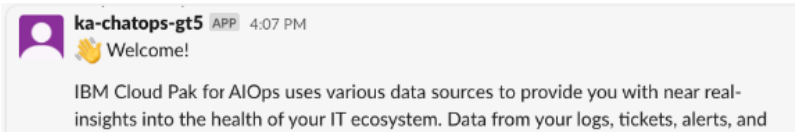


Figure. Slack message

Choose one of the following options:

- *Take tour* Provides an example of incident notification details.
- *Close welcome* Closes the welcome message.
- *Learn more* Links to additional information in the IBM Cloud Pak for AIOps documentation.

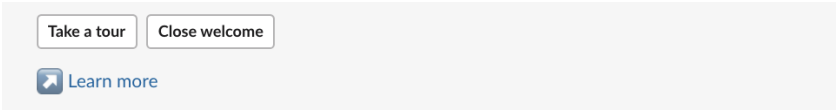


Figure. Slack welcome

In Microsoft Teams, when you receive the welcome notification, select one of the following options to continue:

- *Take tour* Provides an example of incident notification details.
- *Learn more* Links to additional information in the IBM Cloud Pak for AIOps documentation.

Click Teams in the sidebar to access the ChatOps *reactive* channel, anytime.

For more information about working with your insights from your ChatOps integration, see the following topics:

- Managing Incidents In ChatOps
- Probable cause
- Incident Resources
- Recommended runbooks
- Recommended actions
- Viewing alerts in ChatOps
- Change risk in ChatOps

Managing Incidents In ChatOps

Incidents appear in the *reactive* channel of your ChatOps interface. They provide the context for an issue that is severely impacting operations. Included are all alerts that are relevant to the issue, and information about how the affected resources relate to one another. The creation and evolution of incidents is informed by alerts.

Learn more about events, alerts and incidents.

Note: For reasons of convenience and consistency, on this page and on related pages, icons and screen grabs are taken from a Slack implementation of ChatOps.

ChatOps Icons

Icon	Description
	Priority
	Critical Alert (Sev 6)
	Major/Minor Alert (Sev 5 or 4)
	Warning (Sev 3)
	Information/Indeterminate (Sev 2 or 1)
	Link to Platform UI or other online resource
	Probable cause
	Alerts
	Trigger alert
	Recommended runbooks
	Recommended actions
	Incident resources
	Similar past resolution tickets
	Log anomaly

Incident statement

Every incident on a ChatOps interface has an Incident statement area that provides a quick overview of a potential problem. The statement includes a description of the incident, its priority, and when the incident was triggered. The status reflects the current state of the incident and can be set by the SRE.

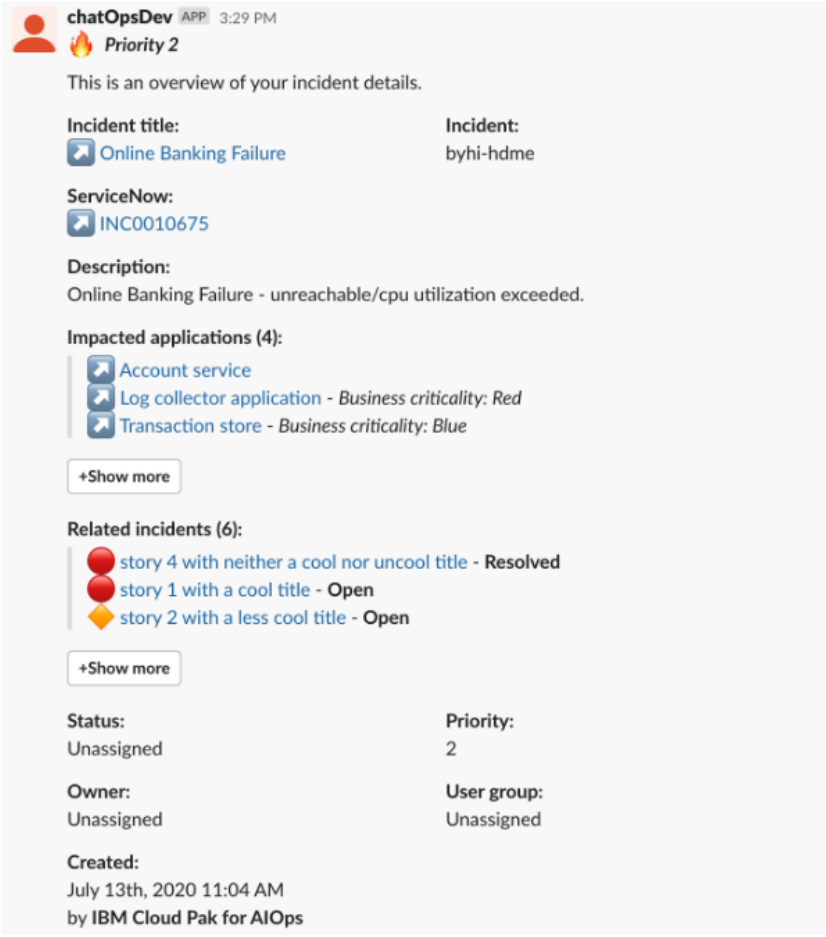


Figure. Incident statement

The Incident statement contains overall information relevant to an incident, including:

Table 1. Incident detection information

Field name	Description
Priority	Priority level that indicated the impact level of an incident on customer.
Incident title	A title that describes the incident. Set as a summary of the first trigger alert, unless it is modified by the user.
Incident	A unique number that is assigned to the incident, also shown within the Platform UI.
ServiceNow	The linked ServiceNow incident number. For more information about configuring a ServiceNow integration, see Configuring the ServiceNow App for IBM Cloud Pak for AIOps.
Description	A summary of the incident, including number of alerts and log anomalies.
Impacted applications	List of applications impacted by incident. Access the applications directly that use the hyperlinks.
Status	*Open*: Incident is open and unassigned *In progress*: Incident is assigned and is being worked on. *Resolved*: Incident is resolved. *Closed*: Incident is closed.
Owner	The email registered to the Slack/Microsoft Teams accounts assigning themselves to the incident. It is recommended that this email matches the email that is registered in the Platform UI.
Created	Date and time that the incident is created.
Updated	Last date and time that the incident is updated.

Updating an incident

At the end of each incident entity, there are options to assign and update the incident.



Figure. Update incident

Click *Edit* to revise incident title, description, group, or priority level. Users can also self-assign the incident and mark it in progress or resolved.

Self-assign and Set to in-progress

The incident owner is set by clicking *Self-Assign* or *Set-in-progress*. When either option is selected, incident ownership is immediately set to the current user. **Note:** It is not currently possible to assign other users as owners.

Tagging users in Slack

When you *Set to in-progress* an incident, you can also tag other users by adding their names to the *Users to notify* field. However, to be tagged, users must be part of the workgroup (**Note:** If even one of those tagged is not in the workgroup, then tagging fails for all users). When tagged, the user is automatically added to the channel and receives notifications to the incident.

Incident resolution and closure

Where an incident status is *Resolved*, it is eventually moved to *Closed* status, in accordance with retention policy period.

Note: When an incident has the status *Resolved* or *Closed*, the functions in ChatOps are limited and no action or link is provided for that incident.

For more information about working with insights from ChatOps, see the following topics:

- Probable Cause
- Incident resources
- Recommended runbooks
- Recommended actions
- Viewing alerts in ChatOps
- Change risk in ChatOps

Probable cause in ChatOps

When an incident is created or updated in response to an issue, IBM Cloud Pak® for AIOps automatically determines which of the related alerts were the likely cause of that issue. For more information, see About probable cause ranking.

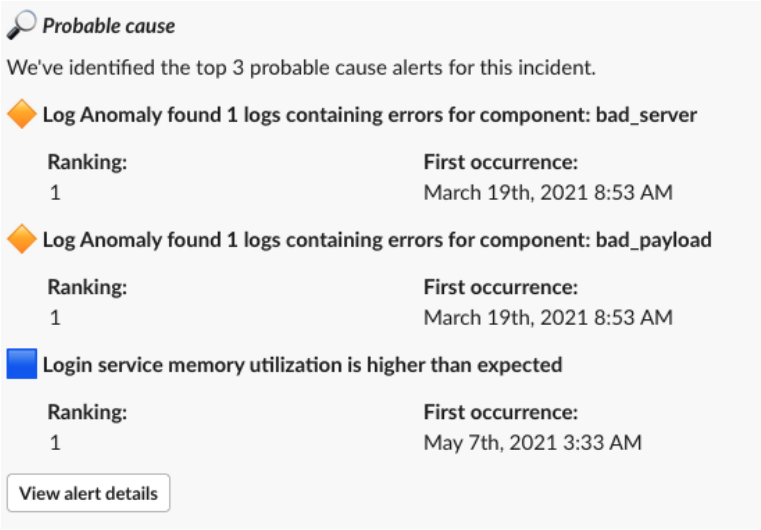


Figure. Probable cause

The *probable cause* section in ChatOps displays the top three alerts, ranked from 1 (highest) to 3 (lowest) in order of probability of being the cause of the incident. To find more information, including severity levels of these alerts, click **View alert details**.

Where *probable cause* alerts cannot be ranked because of similar ranking values, a link is available to the Alert Viewer in the IBM Cloud Pak for AIOps console, where all probable causes are listed.

Note: Where *probable cause* cannot be identified, the section **Top Alerts** appears instead, listing the top three alerts for the incident ranked by severity and first occurrence. A modal can be opened which shows the expanded alert details of this section.

Next steps

For more information on creating access tokens for the Probable Cause Customization APIs, see Probable Cause Customization API.

Incident resources

The topology view in ChatOps' *Incident resources* provides a holistic view of the relationships between different infrastructure and application components.

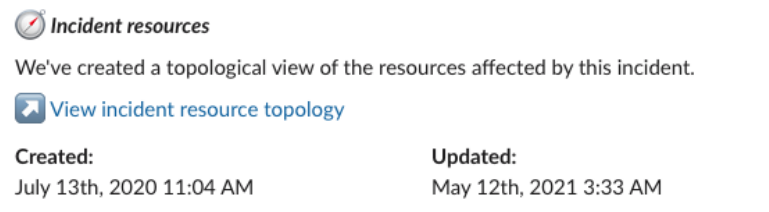


Figure. Incident resources

It can be used to rapidly identify the root cause and the impact of the issue on the wider environment, before any action is taken, avoiding interruption to business-critical activities.

Under *Incident resources*, click **View incident resource topology** to go directly to the Platform UI where you can access topological details about individual entities within an incident.

Note: *Incident resources* is not visible where no application can be associated with the incident.

Associated automations

Runbooks are automated solutions that can be triggered to resolve certain issues. They can be created, edited and updated in IBM Cloud Pak for AIOps.

When an incident surfaces in ChatOps, you can access runbooks associated with this incident. Click the *View Runbook* automations link. This will take you directly to the Platform UI where more information is available about the associated automations:

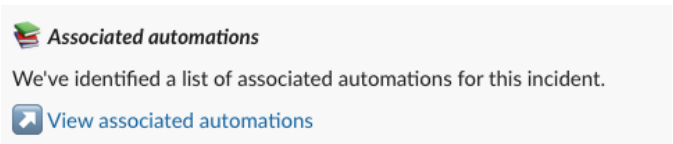


Figure. Associated automations

For more information on runbooks in IBM Cloud Pak for AIOps, see Runbooks.

Recommended actions

ChatOps displays the top three *Recommended actions* based on past resolution tickets that are associated with the incident presented. If you have similar ticket training that is completed for ServiceNow (see Setting up training for similar tickets), you can also search for other past tickets.

Insights and context from your training materials are used to suggest paths to remediation. Use the hyperlinks to go directly to IBM Cloud Pak for AIOps Platform UI.

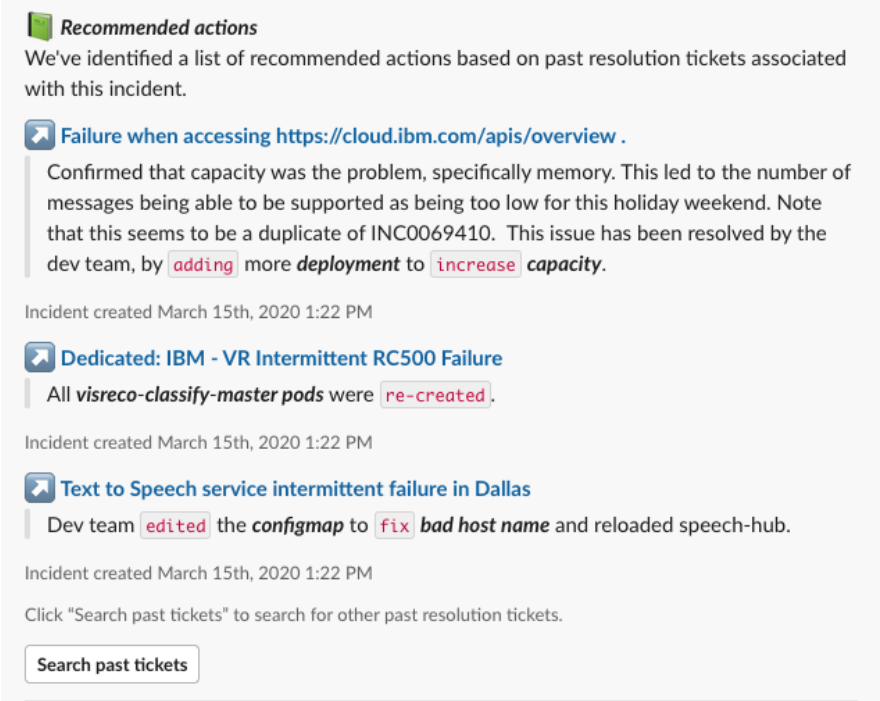


Figure. Recommended actions

Searching recommended actions

Site Reliability Engineers (SRE) can search past tickets in *Recommended actions* to zero-in on likely solutions from similar past tickets.

Search steps

1. Open the *Recommended actions* modal.
2. Click **`Search past tickets`** for any event.
3. Enter search parameters in the *Search past resolution tickets* dialog box.

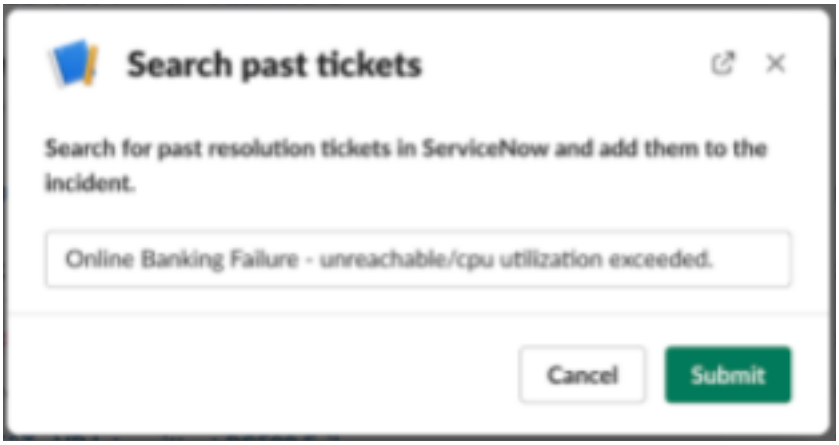


Figure. Search action dialog

Click **Done**. A list of similar tickets appears, with a description for each search result, and details of the action that was taken. You can click the title to go to the issue, or add the issue to the incident by clicking **Add to Incident** .

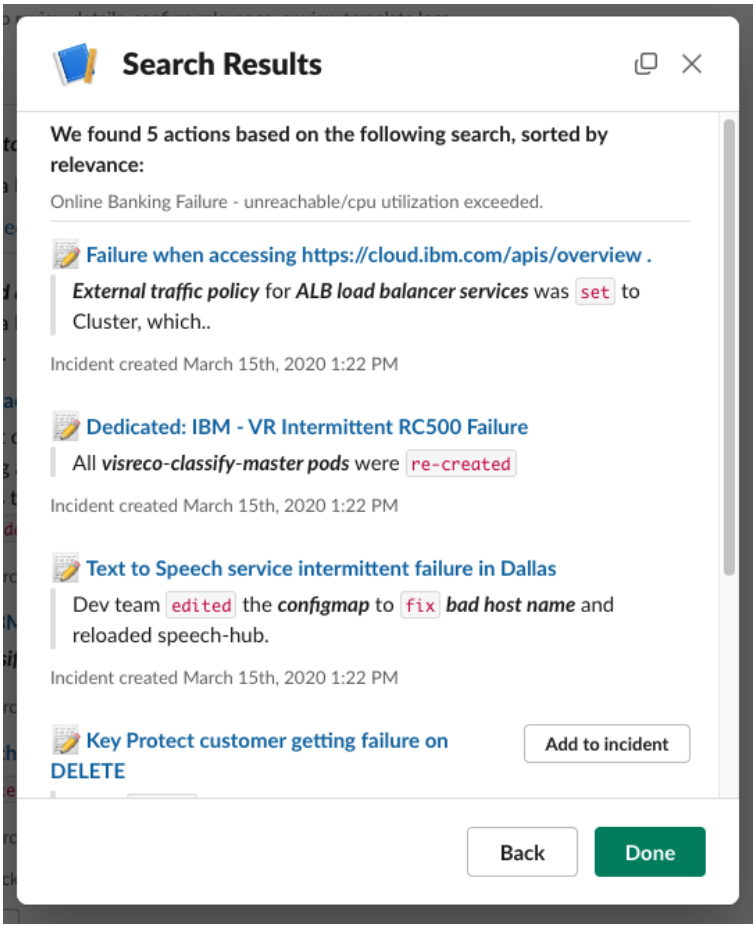


Figure. Search results

Viewing alerts in ChatOps

The *Alerts* area in ChatOps provides details and traceable evidence about alerts. View details of potentially relevant alerts and log anomalies to help you better understand the incident, determine the cause for the alerts, and confirm a diagnosis. Alerts are grouped based on three algorithms: Scope-based, Temporal, and Topological.

- View basic Alerts information
- View detailed information on Alerts
- Investigating log anomalies
- Domain-specific log anomalies

View basic Alerts information

1. Open ChatOps in Slack or Microsoft Teams.
2. Select an incident from your incident (*proactive*) channel, then scroll down to the *Alerts* section of the Incident. This entity presents alerts that are associated with the incident, including their severity level. Click **View in Alert Viewer** in ChatOps to access the *alerts* in IBM Cloud Pak® for AIOps Platform UI.

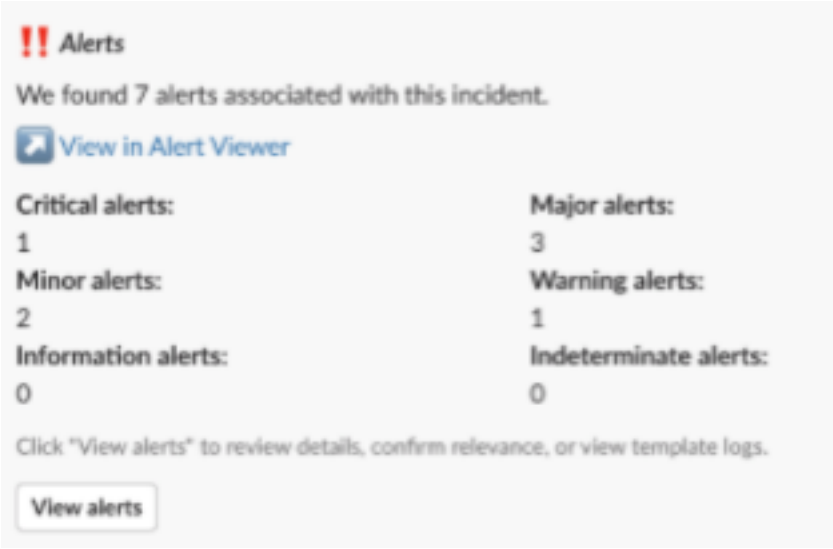


Figure. Alerts

View detailed information on Alerts

In the *Alerts* section of your incident, click **View alerts**. The *View alerts* modal opens.

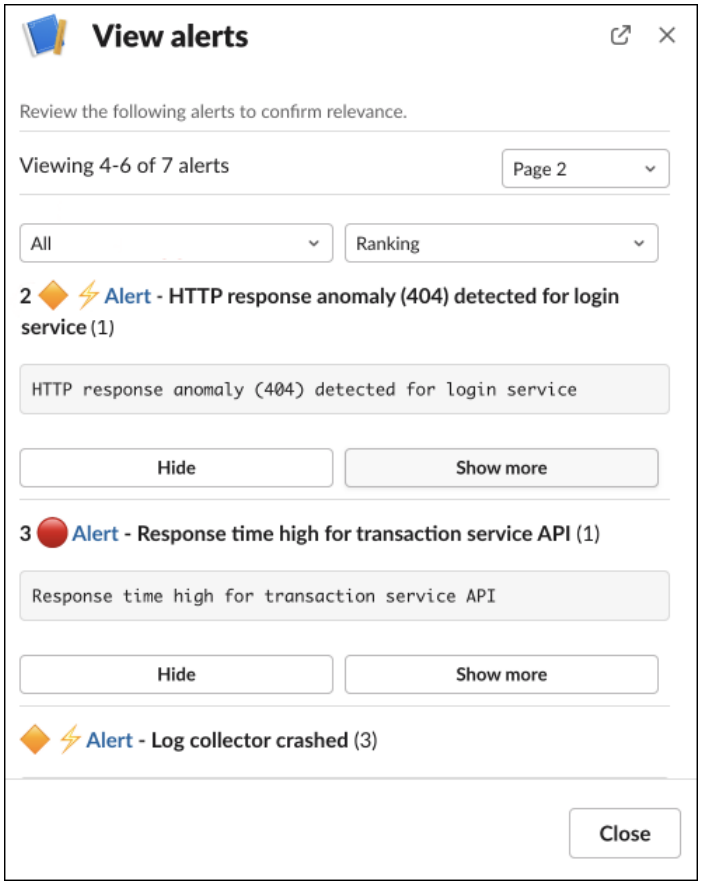


Figure. View alerts

- Use the two dropdown menus to filter alerts by type of alert, severity, status and rank.
- If necessary, scroll down to find all alerts associated with the incident.
- An alert listing can have a number and icon label that is attached to it.

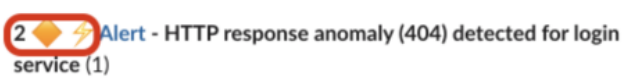


Figure. Icons label

These labels indicate probable cause ranking, severity of alert and if the alert is a trigger alert - indicated by a lightning icon. **Note:** These labels are present only if the alert is a trigger alert and/or if it is a probable cause of the incident - or both.

- You can hide any listed alerts for a specific incident entity by clicking **Hide** for that alert. Click **Done** when you are done hiding alerts. **Note**: To see alerts that you hid, select the entity, and click **Unhide**.
- For more information on an alert, click **Show more** on the **View alerts** modal to view the alert's *Status*, *Source*, *First occurrence*, *Last occurrence* and *Resource*, *Ranking* and *Severity*.

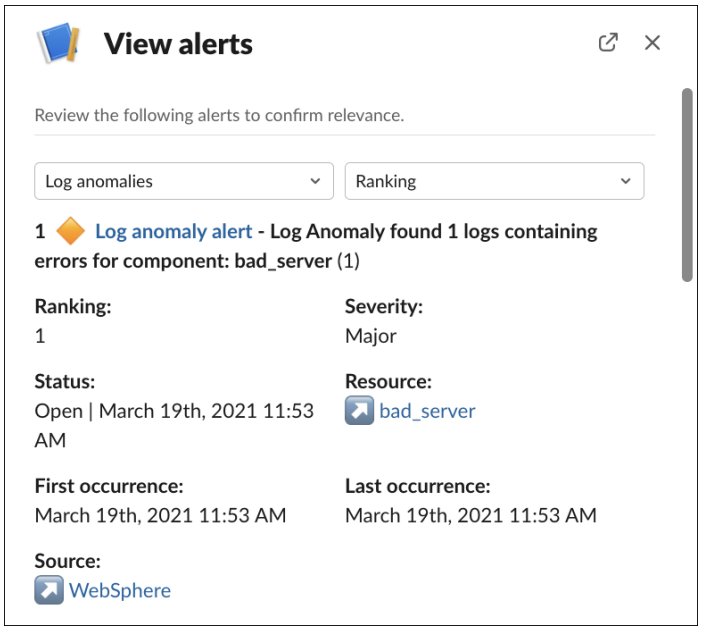


Figure. Show more

Investigating log anomalies

Based on an analysis of your log data performed during AI model training, templates are built that model each type of log message identified. Live log data is analyzed with respect to these templates and any surprises are reported as log anomaly alerts. Here are some examples of log anomaly alerts:

- A log message was expected but is not present in the log data within a time range.
- A log message was not expected but is present in the log data within a time range.
- A log message occurred with a higher or lower frequency than expected within a time range.

Select *Log anomalies* from the dropdown menu in the *View alerts* modal to view the templates that are used to identify anomalous behavior in your log files.

To view the entire log of a log anomaly (the log for all of the templates), click **Attach template logs**. This action attaches the entire log to your Slack channel where you and your collaborators can view the log in its entirety.

This window also reports surprises associated with individual log anomaly templates. For each surprise, the following information is provided.

- Log anomaly **Template or label**. If this log message has never been seen before, one of the following labels will appear:
  - `Unknown_error`: the log message has not been seen before and it contains error information.
  - `Unknown_normal`: the log message has not been seen before but it does not contain error information.
- **Actual**: The number of times this log message actually occurs in the log data within a time range.
- **Expected**: The number of times this log message is expected to occur in the log data within a time range.
- **Preview log**: Click this button to view the actual occurrences in your log. The *Log preview* window shows a maximum of 3000 characters.

  Note: In Microsoft Teams, logs are saved to Microsoft Sharepoint and can be shared from there.

Example 1

Template or label	Actual	Expected
`stdout F {"name":"@pod75wd5abd11","__in":<NUM>, "hostname":"backend-456b7373c9efgh","pid":<NUM>,"module": "announceCycle/agentHostLookup","level":<NUM>,"msg":"Extremely unusual message for system`	0	3.12

In this example, three log messages matching the template were expected in a given timeframe, but there were no log messages present in the log data. You need to investigate why the expected log messages did not appear. **Note** Expected_count is shown as a float with two digits after the decimal point. This value is calculated based on the statistical analysis described in Workflow of natural language log anomaly detection and Workflow of statistical baseline log anomaly detection.

Example 2

Template or label	Actual	Expected
`Unknown_error`	2	0

In this example, a log message that has never been seen before was present in the live log data, twice within the time range. This unexpected log message also contains error information so should be investigated with urgency.

Domain-specific log anomalies

If an alert is identified as a log anomaly from a specific domain, such as IBM MQ or WebSphere, you can scroll to the (**View Alerts** > **Show more**) modal to find more details about the anomaly. View *Message Code*, *Explanation*, *Categories*, *Frequency of related message codes* and *Recommended actions*, which can contain documentation links that might help resolve the alert.

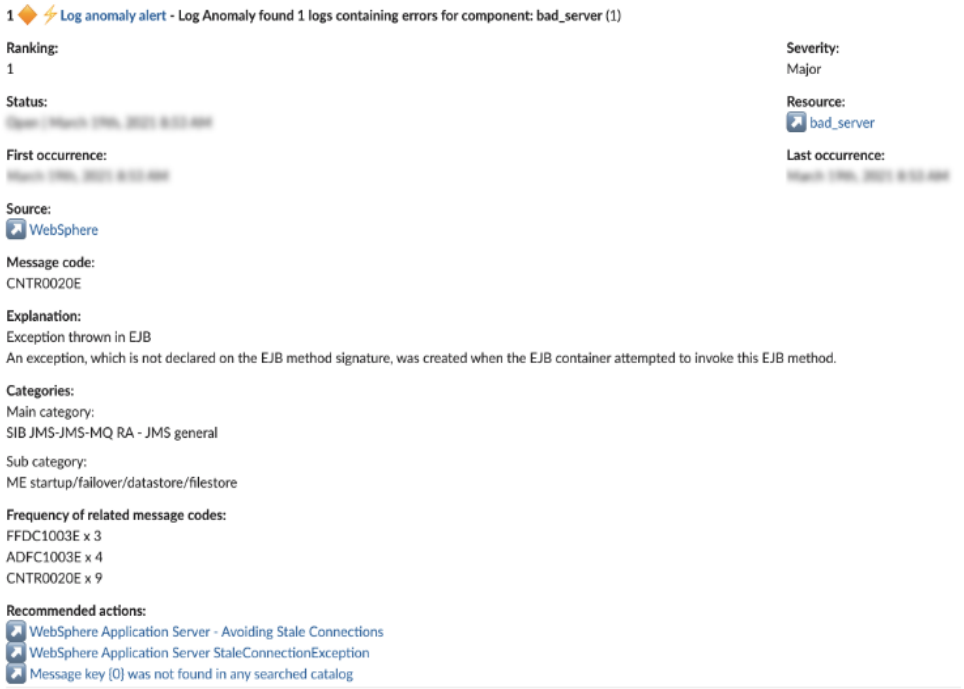


Figure. WebSphere

For more information, see Domain-specific log anomaly alert details.

Change risk in ChatOps

Change risk assessments, which can be notified in ChatOps, provide you with a *Confidence* percentage that represents the confidence level in the AI's prediction of the impact of a change. This percentage is relative to the risk associated with a new or updated ServiceNow change request ticket.


Chatops Change Risk
Figure. Chatops change risk

The *Confidence* percentage is calculated based on the change risk AI model and the data that you feed into it. Change can be evaluated based on any type of change, whether it is to code, or to configurations, or to data. If you have historical change request tickets for that type of change, IBM Cloud Pak® for AIOps can provide a confidence percentage for it.

NOTE: Feedback options (thumbs up and thumbs down) appear under the change risk assessment, to allow users to provide input on the assessment and the relevancy of related change requests. Providing feedback helps improve the quality of the change risk model.

Prerequisites

To implement change risk notifications:

- In ServiceNow, any change needs to be recorded in the Change Request Ticket Fields
- In ServiceNow, any deviation from an expected standard of operation must be documented in an incident ticket field

Change Request Ticket Fields

Change request tickets in ServiceNow record detailed information about a change, such as the reason for the change, the priority, the risk, the type of change, and the change category. This information is used to calculate a confidence percentage.

Required fields for change request tickets expect a `str` data type value and include the following fields:

- *sys_id*
- *number*
- *assigned_to*
- *sys_created_by*
- *sys_domain*
- *business_service*
- *type*

- *state*
- *short_description*
- *impact*
- *reason*
- *justification*
- *description*
- *backout_plan*
- *close_code*
- *close_notes*

For more information about change request fields, see Change Request in the ServiceNow documentation.

Six incident fields in ServiceNow are scanned for the presence of change ticket numbers: " *description*", "*short_description*", "*work_notes*", "*caused_by*", "*close_notes*", "*comments_and_work_notes*".

Incident Ticket Fields

You create an incident record to document a deviation from an expected standard of operation. The following list describes required fields for incident tickets:

- *number*
- *opened_at*
- *short_description*
- *description*
- *work_notes*
- *caused_by*
- *rfc*
- *close_code*
- *close_notes*
- *closed_at*
- *cmdb_ci*
- *active*
- *assignment_group*
- *comments_and_work_notes*
- *impact*
- *problem_id*
- *priority*
- *severity*
- *state*
- *urgency*

Note: All fields expect an `str` data type value, except "*work_notes*", which expects a `List[str]` data type.

For more information on incident ticket fields, see Incident Request in the ServiceNow documentation.

Linking Change Tickets and Incidents

Without the correct linkage information generated from these fields, the change risk assessment logic can fail or provide incomplete results. Also, if no linkage is found, the training is marked as unsuccessful or aborted.

When you determine the linkage between the change tickets and the incidents, you also need to look at data interdependence, such as that of the start and end dates. Set the Change Start Date to earlier than the Incident Start Date because the incident occurs after the changes are made.

For more information about change management in ServiceNow, see Change management in the ServiceNow documentation.

In order for you to receive change risk notifications, you must satisfy the following requirements:

1. Set up ServiceNow and your ServiceNow integration
2. Set up your ChatOps integration
3. Train your change risk AI model

Managing automations

By creating automations in IBM Cloud Pak® for AIOps, you can proactively set up policies, runbooks, and actions to detect and remediate events, both current and future.

Policies are rules that contain conditions and a set of actions that can be manual or automated. With policies, you can automatically suppress alerts, group alerts based on scope, assign runbooks to alerts, and promote alerts to an incident.

When you create a runbook, you can define the required steps to resolve a situation. If you don't have any runbooks, you can create a new one or load some sample runbooks.

Actions in runbooks are the collection of several manual steps into a single automated entity. An action improves runbook efficiency by automatically performing procedures and operations.

For more information about automations, see the following topics:

- Policies

- Runbooks
- Actions

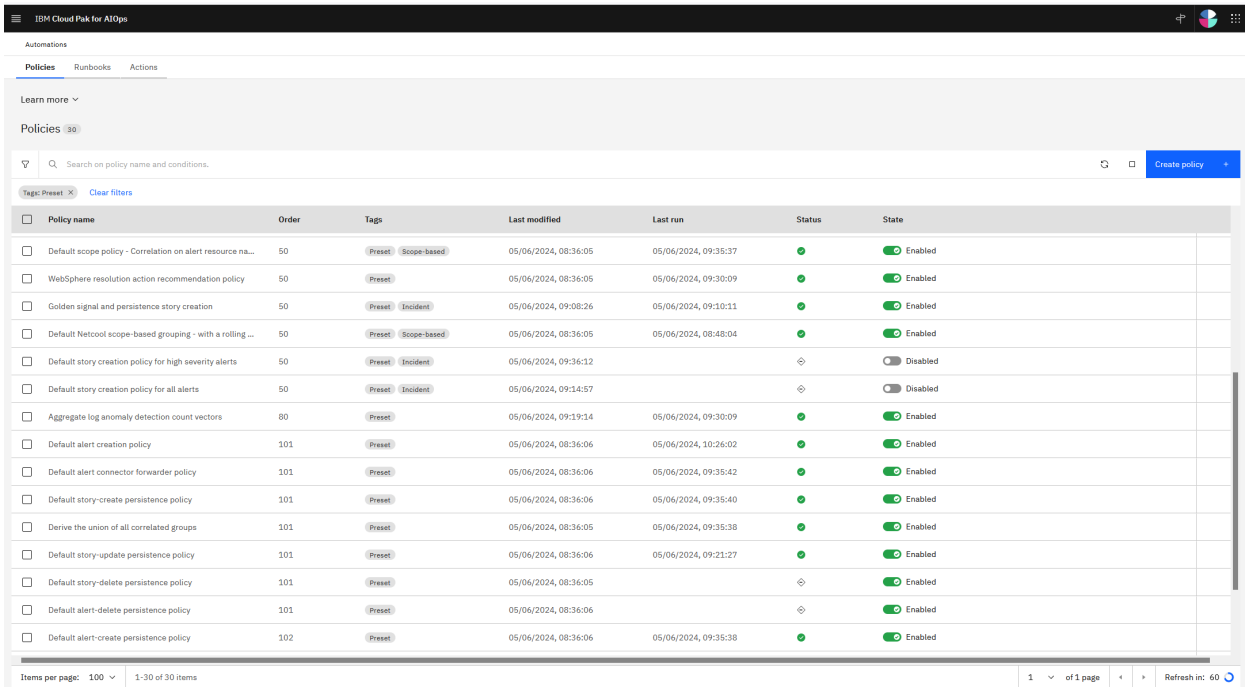
About policies

Policies are rules that contain multiple condition and action sets. They can be triggered to reduce noise by suppressing alerts or grouping alerts together, automatically promote alerts to incidents, assign runbooks to remediate alerts, and invoke IBM Tivoli Netcool/Impact policies to take actions on your alerts.

Cloud Pak for AIOps policies are created in the Policy editor, where you can build condition sets based on Alert properties and Alert insights.

The policies table displays all of the Cloud Pak for AIOps policies that are preset, created by you and your team, or generated by AI analytics. You can enable or disable these policies, create new ones, and edit existing policies. When you click a policy in the table, you can see more details in the side panel. The list of policies is split by pagination, use the navigation arrows beneath the table to move between the next or previous page of policies. To customize the number of rows that are displayed in the table, click the **Down chevron** icon ⌄ next to the **Items per page**.

By default, policies are sorted in the table by **Order** ascending (policies with a lower number are processed first). Where execution order is the same, policies are sorted by **Last run** descending (the most recent policies are displayed first in the table).



Policy UI

Note: Preset policy names and internal action references to "story" or "stories" are synonymous with incidents. Incident triggers and condition references supported in the "Invoke IBM Tivoli Netcool/Impact" policy template are stored as story. For policy search based on conditions, use "story" instead of "incident". For example, to find incident.title in a policy search, use story.title. These internal references and policy names have no bearing on actual behavior and are for internal processing purposes only. Any references to "story" in preset policy names, or other policy details or specifications is not an error, and is equivalent to incident in all respects.

Policies table columns

Policies table columns

Column	Description
Policy name	A name to identify the policy. The default name of a policy is automatically created based on the alert summary field, and specifically on the alert with the highest severity that has ocurred most frequently. To customize a policy name, point the cursor over the name and click Edit. Preset policy names cannot be customized.
Order	Determines the order in which the policy is executed. Execution order is set at policy creation time, on a sliding scale of 1-100. Policies with a lower number are processed first. This is especially important for incident policies. Set the execution order lower for your most important incident policies. Values outside of 1-100 are reserved for suppression, some analytics, and preset policies that handle system needs. **Note:** Actual execution order is a combination of trigger type and execution order. So when the table is sorted by execution order, you might see later **Last run** times on higher priority policies. This is because the processing for different triggers occurs at different times.

Column	Description
Tags	Tags are used to identify the different types of policies. Cloud Pak for AIOps policies are tagged with the following policy types: • Analytics: policies generated by AI analytics. Analytics policies can be viewed, enabled, disabled, and deleted from the Policies table. • Impact: a Cloud Pak for AIOps policy that invokes a IBM Tivoli Netcool/Impact policy, through a IBM Tivoli Netcool/Impact integration. By default, IBM Tivoli Netcool/Impact policies are invoked without input or transformation. Editing options are available in the IBM Tivoli Netcool/Impact policy template. • Incident: policies to create actionable incidents from alerts to improve insights into your issue. • Preset: preset policies define the core behavior of event and alert processing. Preset policies can be viewed, enabled, or disabled from the Policies table. • Runbooks: policies that assign a runbook to alerts for easier automated resolution. • Scope-based: policies that group alerts based on common properties and time of occurrence to identify when issues might be related. • Seasonal: policies that group alerts together that occur within a seasonal time window. • Suppression: suppress alerts that do not need attention to reduce noise and improve team focus. • Temporal: policies that group alerts together that occur within a short time of each other. • User: policies that can be edited by the user. All Runbook and Incident policies, even if tagged as Preset, are also tagged with User. User policies can be viewed, enabled, disabled, modified, and deleted from the Policies table.
Last modified	Displays a timestamp of the last modification to the policy. The date is expressed as month, day, year. The time is expressed as hours:, minutes:, seconds. The time also indicates whether AM or PM. For example, 3/9/2022, 4:45:17 PM. Enabling or disabling a policy, renaming or editing a policy, and adding a comment from the **Journal** tab in the side panel will all update the last modified time stamp.
Last run	Specifies the date and time on which a policy was last run. The date is expressed as month, day, year. The time is expressed as hours:, minutes:, seconds. The time also indicates whether AM or PM. For example, 3/9/2022, 4:45:17 PM. **Note:** If an alert meets the incident-creation conditions of multiple policies, only one incident is created. However, all policies that proposed an incident, and the system incident creation policy will have the same **Last run** time.
Status	The icons in this column indicate one of the following execution states for a policy: • Policy successful, no recent failures • Policy has not been attempted • Policy successful, but has had recent failures • Policy failed on last attempt
State	Denotes if a policy is Enabled or Disabled. You can click the toggle button to change a policy state. Disabled policies do not take any action against incoming alerts. You can also enable or disable a policy in the policy details side panel, or directly within a seasonality or temporal policy details page. **Note:** Disabling preset policies will prevent some event and alert processing features from working.

Note: A yellow execution status will remain for 8 hours in the event where a policy fails, but then works. Irrespective of how many times the policy works in that 8-hour period. If a policy is "flapping", that is alternating between failing and working, the execution status flips between a yellow and red status.

For more information about policies, see the following topics:

- Searching and filtering policies
- Policy details
- Default incident creation policies
- Default Netcool scope-based grouping
- Creating policies
- Policy editor
- Refreshing the policies table
- Managing multiple policies
- Renaming policies

The following policy templates are available:

- Suppress alerts
- Group alerts based on scope
- Assign a runbook to alerts
- Promote alerts to an incident
- Invoke IBM Tivoli Netcool/Impact

Searching and filtering policies

Text entered in the search field searches for the characters in the policy names and the policy conditions. Note, some conditions are internal, so the search might return policies that you are not expecting. You can also filter the list based on the policy type or current state.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.

To search for a policy in the table:

Click anywhere in the **Search** field and enter a search term, for example, type *hostname*. The system shows table with all the policies that contain *hostname* in the policy names and the policy conditions.

Note: Searches are not case-sensitive.

To filter the list of policies in the table:

Click the Filter icon to open the filter side panel.

- Select from the following **Tags** to filter by policy type:
  - Analytics
  - Impact
  - Dashboard
  - Incident
  - Preset
  - Runbooks
  - Scope-based
  - Seasonal
  - Suppression
  - Temporal
  - User
- Filter policies by **Status**:
  - Policy successful, no recent failures
  - Policy successful, but has had recent failures
  - Policy failed on last attempt
  - Policy has not been attempted

  Note: Failed policies turn red. In the event where a policy fails but then works, a yellow execution status will remain for approximately 8 hours. Policies with a red or yellow status will move back to green after approximately 8 hours.
- Filter policies by **State**:
  - Disabled
  - Enabled
- Filter policies by **Triggers**:
  - Before an alert is created
  - After an alert is created
  - After an alert has been updated
  - After an incident is created
  - After an incident has been updated
  - System triggers

  Within a filter type, multiple selections are summed together. For example, if you select both **Runbooks** and **Incident** under the **Tag** filter, you will see the list of policies that contain **either** of these tags.

  Between filter types, the AND operator is used. For example, if you select **Incident** under the **Tag** filter and **Enabled** under the **State** filter, you will see the list of policies that contain **both** of these conditions.

3. Click **Apply** to apply your selected filters to the policies table.
4. Click **Clear filters** above the table header row to clear all searches and filters.

Note: Preset policy names and internal action references to "story" or "stories" are synonymous with incidents. Incident triggers and condition references supported in the "Invoke IBM Tivoli Netcool/Impact" policy template are stored as story. For policy search based on conditions, use "story" instead of "incident". For example, to find incident.title in a policy search, use story.title. These internal references and policy names have no bearing on actual behavior and are for internal processing purposes only. Any references to "story" in preset policy names, or other policy details or specifications is not an error, and is equivalent to incident in all respects.

Policy details

When you click a policy of interest in the table on the Policy UI, a side panel containing multiple information sections opens on the right-hand side of the table. Use the up and down arrows to navigate through the list of policies. Where permissions allow, you can Edit and Delete or enable and disable some policies here. You can also archive alert seasonality and temporal grouping policies that are not in use, or further investigate the justification for their creation directly from the side panel.

The four tabs that are displayed in the side panel are **Details**, **Journal**, **Specification**, and **Source**.

- **Details**:

This tab displays the complete set of column data for the selected policy, including familiar fields, such as **State**, **Status**, **Tags**, **Created by**, **Last modified by**, **Last run**, and **Execution status**. The **Last failure** is the timestamp that the selected policy last failed. **Number of failures** indicates how many times the policy has failed, and **Last failure details** is the error message. Click **Copy** to copy Last failure details to the clipboard.

The **Matched** number is incremented by 1 if one or more actions is proposed or performed by a policy. However, when errors occur during policy execution, matched might not be incremented. Therefore, you might see failures but the **Matched** count indicates 0. Matched may only be incremented upon successful policy execution.

Note: In a case where an alert meets the incident-creation conditions of multiple policies, only one incident is created. However, all policies that proposed an incident, and the system incident creation policy will have the same **Last run** times. Each of these policies will also have their **Matched** counts incremented by one.

- **Journal**:

  The Journal tab displays any comments added by operators to record manual changes or updates to the policy. Comments are shown in chronological order. The policy creation date is also displayed.

- **Specification**:

  Shows you the key information about your conditions and policy. Each condition set of a policy is detailed in the Specification tab, as well as what is created by the policy.

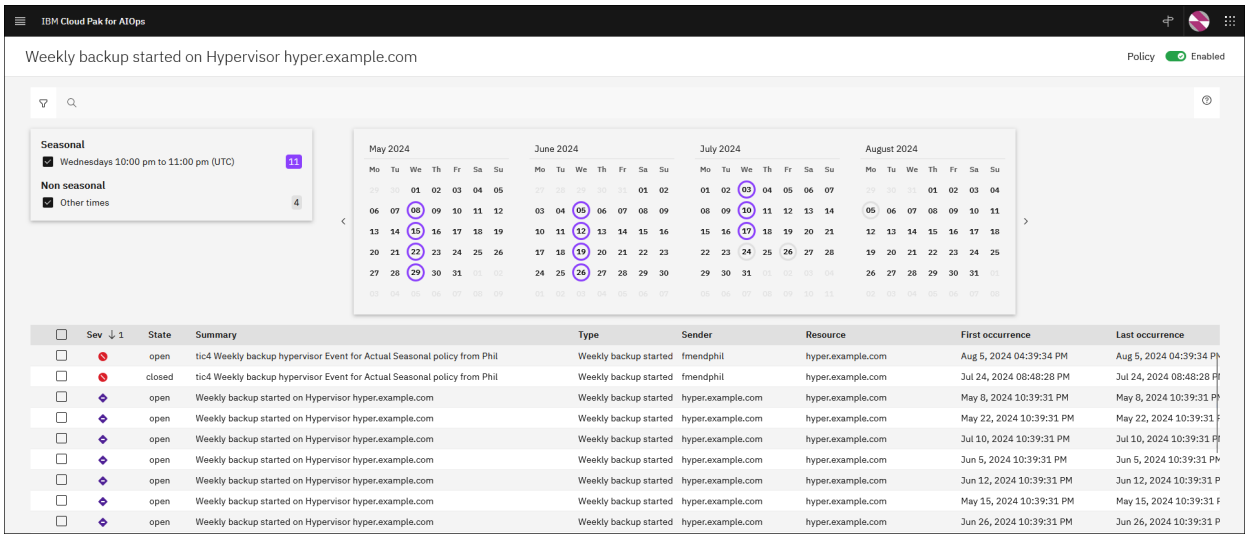
- **Source**:

  View a policy in its raw form in either JSON or YAML format, depending on your requirements. Click **Copy** to copy the policy data to the clipboard.

Note: Preset policy names and internal action references to "story" or "stories" are synonymous with incidents. Incident triggers and condition references supported in the "Invoke IBM Tivoli Netcool/Impact" policy template are stored as story. For policy search based on conditions, use "story" instead of "incident". For example, to find incident.title in a policy search, use story.title. These internal references and policy names have no bearing on actual behavior and are for internal processing purposes only. Any references to "story" in preset policy names, or other policy details or specifications is not an error, and is equivalent to incident in all respects.

Seasonality policy details

The seasonality details page provides a calendar view of all of the historical alerts that contributed to the selected seasonal policy. One or more seasonal time windows are listed under the filter icon. Select a seasonal time window to filter the view to show just historical alerts that contributed to that time window.



Seasonality details

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Filter the table to display **Temporal** and/or **Seasonal** policies, or search for a policy by entering a search term in the field provided.
4. Locate the policy of interest and do one of the following to launch out to the policy details page:
   - Click the options menu (three vertical dots) at the end of the policy's table row and then click **View justification**.
   - Click anywhere in the policy's table row and then click **Justification** in the side panel displayed.

The seasonality policy details page contains the following sections:

- **Header**:

- Policy name
- Toggle button to change the policy state to enabled or disabled. Disabled policies do not take any action against incoming alerts.

  Note: Users must have the *Edit operational policies* permission to enable or disable policies from the seasonality policy details page. The toggle button for policy state will be disabled without this permission. For more information, see Roles and permissions.

- **Toolbar**:
  - Search: searches alert data in all event group instances that are shown on this page.
  - Filter: filters the alerts that are shown by severity level:
    - Critical
    - Major
    - Minor
    - Warning
    - Informational
    - Indeterminate
- **Seasonal time windows**:

  One or more seasonal time windows are listed together with a number indicating how many historical alerts occurred within this time window during the historical period analyzed. The color of the square is the color code for the respective seasonal time window. Select a seasonal time window to filter the Calendar and the Historical alert table to show just historical alerts that contributed to that time window.

- **Calendar**:

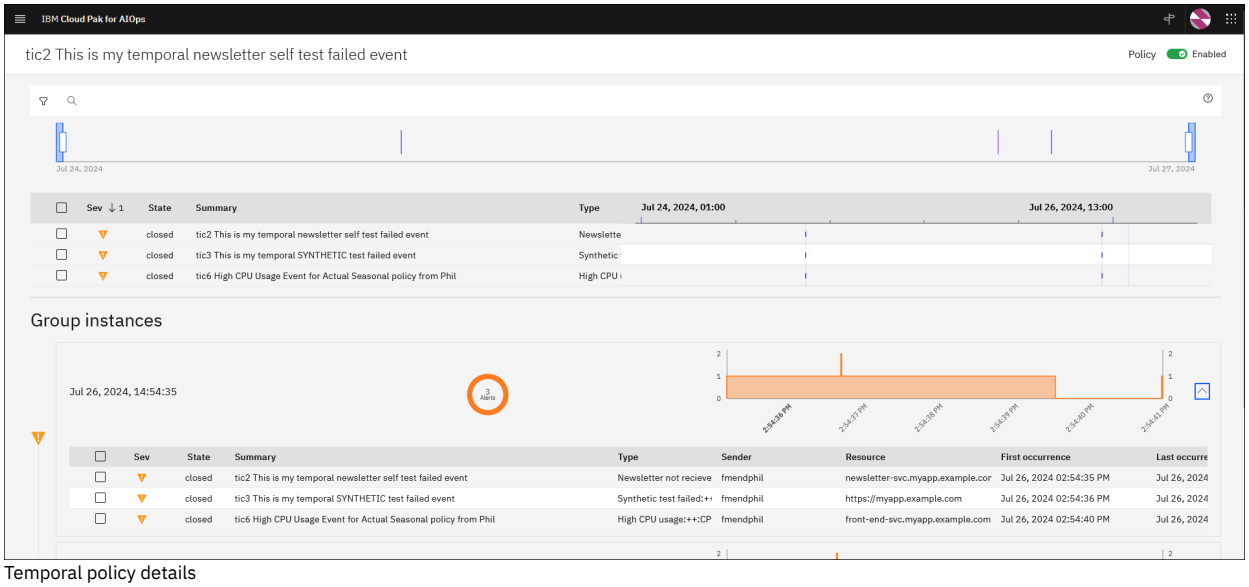
  A calendar of the historical period analyzed (the last three months or more) showing days on which the historical alerts of interest occurred, and colored with the color code for the respective time window. If you hover over any indicated day in the calendar, a tooltip displays a breakdown of the seasonal alerts for that day.

- **Historical event table**:

  Table listing the historical alerts during the historical period analyzed.

Temporal policy details

The temporal policy details page shows the alerts that make up a temporal group, together with all of the historical instances of the group. You can navigate into these instances to view detailed timelines of each instance. You can also view alert details.



Temporal policy details

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.

3. Filter the table to display **Temporal** and/or **Seasonal** policies, or search for a policy by entering a search term in the field provided.
4. Locate the policy of interest and do one of the following to launch out to the policy details page:
   - Click the options menu (three vertical dots) at the end of the policy's table row and then click **View justification**.
   - Click anywhere in the policy's table row and then click **Justification** in the side panel displayed.

The temporal policy details page contains the following sections:

- **Header**:
  - Policy name
  - Toggle button to change the policy state to enabled or disabled. Disabled policies do not take any action against incoming alerts.

    Note: Users must have the *Edit operational policies* permission to enable or disable policies from the temporal policy details page. The toggle button for policy state will be disabled without this permission. For more information, see Roles and permissions.
- **Toolbar**:
  - Search: searches alert data in all event group instances that are shown on this page.
  - Filter: filters the alerts that are shown by severity level:
    - Critical
    - Major
    - Minor
    - Warning
    - Informational
    - Indeterminate
- **Overview timeline**:

  Displays alert group instances over time and controls the display of alert group instance data on the rest of the page. By default the time range sliders are open sufficiently to show data on all alert group instances. Modify the time range by either clicking and dragging over the desired range inside the timeline, or by dragging the sliders to the desired range. The rest of the screen updates accordingly.
- **Alert group instance timeline**:

  Displays all of the alerts that have historically participated in instances of this temporal alert group. The instance map provides a graphical view over time of when the various instances have occurred.
- **Alert group instance details**:

  Displays the following information for each alert group instance:
  - Start date and time of event group instance: indicates the first occurrence value of the first alert in the alert group instance.
  - Distribution of event severity values: pie chart providing a visual indication of the alert severity values. Hover over the pie chart for more details.
  - Sparkline: chart of alert occurrence over time.
  - Duration of event group instance: duration of the alert group instance, in text.
  - Down chevron icon: click the down chevron icon to see an alert table showing column details for each alert in this group instance.

Default incident creation policies

IBM Cloud Pak® for AIOps provides two preset (or default) incident creation policies. For incident creation to occur, you must enable one of these default policies, or create your own custom incident creation policy. These policies are disabled by default.

Default incident creation policy for high severity alerts

This policy proposes an incident based on a Major or Critical severity alert. The policy contains one condition set and action.

Condition set 1

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	greater or equal	only	5-Major

The following action is taken: Create a **Priority 1** incident.

Default incident creation policy for all alerts

This policy proposes an incident based on any alert. The policy contains the following condition sets and actions.

Note: In incident policies, only the actions from the first matching condition set are executed.

Condition set 1

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	equal to	only	1-Indeterminate

The following action is taken: Create a **Priority 5** incident.

Condition set 2

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	equal to	only	2-Information

The following action is taken: Create a **Priority 4** incident.

Condition set 3

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	equal to	only	3-Warning

The following action is taken: Create a **Priority 3** incident.

Condition set 4

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	equal to	only	4-Minor

The following action is taken: Create a **Priority 2** incident.

Condition set 5

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	greater or equal	only	5-Major

The following action is taken: Create a **Priority 1** incident.

Default Netcool scope-based grouping - with a rolling window of 15 minutes

IBM Cloud Pak® for AIOps provides a preset (or default) Netcool scope-based grouping policy. Provided that the `ScopeID` field in your IBM Netcool Operations Insight system is mapped to the `alert.resource.scopeId` field in Cloud Pak for AIOps, this policy uses that value to create scope-based groupings. The policy is enabled by default.

If scope-based grouping is used in an existing Netcool environment, you need to disable `correlation_triggers` groups in the ObjectServer infrastructure, and the correlation needs to done at the Cloud Pak for AIOps layer.

Condition set

If this condition is met:

Table #1. Condition set

Property	Operator	Matches	Value
Value of: alert.resource.scopeId	not empty		

The following action is taken: **Create a scope-based grouping**.

ScopeID

The ScopeID are the properties that define the scope for how alerts are grouped.

The ScopeID in this policy is **alert.resource.scopeId**.

Time window

This default policy has a 900-second **Rolling** time window. That is, 900 seconds is the threshold that needs to pass with no further alerts, after which the system stops including alerts in the group.

Incident Creation based on Persistence and Golden Signal

BETA *Incident creation based on persistence and golden signal* is available only for technology preview, not for production usage.
Incident prioritization uses the golden signal policy and the alert suppression policy to filter out incidents when the observed alerts contain a golden signal and are not suppressed. The policy suppresses alerts that do not need attention.

An incident is only created when the following criteria are met:

1. Events persist for a significant duration, leading to an unsuppressed alert.
2. The golden signal that is associated with the alert is of the type `EFFECT`.

Prerequisites

Configuration prerequisites

To enable the X in Y algorithm, use the following steps.

1. Log in to your Red Hat OpenShift Container Platform cluster as an administrator where Cloud Pak for AIOps is installed.
2. Create or update the ConfigMap with the `IBM_IR_AI_X_IN_Y_PREVIEW_ENABLED` field. Set the value of this field to `true`. A sample YAML file can resemble the following example:

   ```
   apiVersion: v1
   data:
     IBM_IR_AI_X_IN_Y_PREVIEW_ENABLED: "true"
   kind: ConfigMap
   metadata:
     name: feature-flag-configmap
     namespace: <namespace>
   ```

   Where `<namespace>` is the namespace where IBM Cloud Pak for AIOps is installed.
3. If you are not running these steps before installation or upgrading, delete the `ibm-ir-ai-operator-controller-manager` pod. When the pod is deleted, OpenShift automatically creates and starts a new pod.

Data prerequisites

Historical alert data is used in the training process to learn values of X and Y, which are later employed for creating XinY alert suppression policies.

1. In the Cloud Pak for AIOps console, click **Operate** > **AI model management** and find the *Alert suppression XinY policies* tile. If this tile does not appear, delete the `aimanager-aio-ai-platform-api-server` pod using the OpenShift console. When the pod is deleted OpenShift automatically creates and starts a new pod.

   For more information, see The Alert suppression XinY policies tile does not appear in AI model management.
2. Click the tile and follow the steps to train the model and then deploy it. For more information, see training a model.

This instantiates the policy and adds it to the policy list in **Automations**.

Note: To create effective Alert Suppression policies, a minimum of 7 days' worth of historical alert data is needed. This data can be supplied either during the training phase or continuously gathered from the alerts that are received after integrating your application with IBM Cloud Pak for AIOps.

About this task

After you train your model with historical data, you are ready to proceed to using the incident creation based on persistence and golden signal policy.

Procedure

1. Go to the **Automations** page and click the **Policy** tab to see all available policies.
2. Click the **Toggle filters** icon and select **Suppression** from the list.
3. Any policies that have suppressions are listed. Find the policy that contains the prefix *X-in-Y-alert-suppression*. If it is not there, try clicking the refresh icon, to update the policy list.
4. Click the *X-in-Y-alert-suppression* policy to open a sidebar information panel. In the panel, slide the **Status** switch to **Enabled**.
5. Next, close the panel, click the *Tag* header column, clear *Suppression*, and check the *Incident* box.
6. Among the incident policy listings are the *Golden Signal and Persistence incident creation* policy. Click this policy to open the sidebar panel.
7. In the sidebar panel, set the **Status** switch to **Enabled**.
8. The *Golden Signal and Persistence incident creation* must be the only *Incident* tagged policy that is enabled. If other *Incident* policies are enabled, then click them, in turn, and disable them in the sidebar panel.

You now have a policy that combines the two policy values when creating incidents. Now when alerts come in, they must fulfill the conditions of the two policies to be promoted as incidents:

- They need to contain a Golden Signal `Error`

- The X and Y values must be outside the allowed range set when the model was trained – for example, where the model has the value of 12 events within 60 minutes, an anomaly only surfaces when the events are 12 or above.

Creating policies

Click the **Create policy** button and choose from a set of templates to help you get started.

See the following topics for more information about creating policies:

- Suppress alerts
- Group alerts based on scope
- Assign a runbook to alerts
- Promote alerts to an incident
- Invoke an IBM Tivoli Netcool/Impact policy

For a detailed breakdown of the policy editor, see Editing policies.

Suppress alerts

Some alerts might not be important for monitoring your organization's environment. For alerts that do not need to be viewed or acted on, use suppression policies to reduce noise for your operations teams by suppressing unwanted alerts. When an alert is suppressed in Cloud Pak for AIOps, the `alert.suppressed` flag is set to true. The alert is still present in the system and can be viewed in the Alert Viewer, but the alert is filtered out of the view by default. Incident creation and runbook policies do not run for suppressed alerts. However, scope-based grouping policies continue to run so there is a scenario where suppressed alerts can be included in incidents.

Enabling X in Y alert rate suppression

Many alerts can represent problems where performance thresholds were breached, for example, a switch that reaches its maximum bandwidth capacity during a peak period. These temporary threshold breaches are often expected, and the business policy might be to ignore them unless the breach is sustained for a prolonged but specific period. However, if such alerts are sustained beyond these specified time periods, operators must take action. In this classic X in Y alert rate scenario, the operator is only interested in alerts that occur more than X times in Y minutes. When the alert rate breaches the threshold, the described alerts should not be suppressed. Until the threshold is breached, the alerts do not need to be acted upon.

In suppression policies, you enable X in Y alert rate suppression under actions in the policy editor. Select **Suppress the described alerts until:**. Then, specify your values for X (number of alerts) and Y (minutes).

Attention: Please be aware about alert states that contribute to the X count. The X value is incremented for deduplicated problem and resolution alerts. Therefore, if a deduplicated alert flaps 10 times (alternating between open and clear) "X" would contain a value of 20 (not 10 for just the problem alerts).

Limitations

While the suppression policy can be applied to all alerts, and all alerts can use the Alert property condition to identify when to suppress an alert, use of the Alert insight conditions have certain limitations.

- Anomaly insights are available on all metric anomaly alerts and on certain log anomaly alerts.
- Only a subset of log anomaly alerts identify an exception or error code.
- The statistical baseline log anomaly algorithm typically identifies these exceptions or error codes.

About this task

In this example, preproduction, development, and test environments are monitored by the same monitoring tools as the production environments. The development and test environments naturally have ongoing changes that might trigger various warnings and notifications all the time. These warnings and notifications are relayed to Cloud Pak for AIOps as alerts. However, these alerts do not require action as they are from resources that change constantly as part of the preproduction work. The monitoring tools are usually set up to avoid sending such alerts to the operations teams. However, if such alert information is sent to Cloud Pak for AIOps, for example, due to a misconfiguration, then the operations team might face unnecessary noise.

To save your operations team from being distracted by alerts from such environments, you can create a suppression policy to suppress alerts that come from the monitored resources that make up these environments.

Important:

- Unlike other Cloud Pak for AIOps policy templates, suppression policies contain a single condition set.
- The supported **Anomaly insights** properties for suppression policies are displayed in the tree view of the **Property** field. To display the available options, click **Add condition** > **Alert insight** and place the cursor in the Property field. For more information, see Supported Anomaly insights properties.

Example

1. Click the navigation icon to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.

3. Click **Create policy**.
4. Click the **Suppress alerts** tile.
5. Enter a name in **Policy name**, for example, the name can be "Suppress events from preproduction environment". You can also add an explanation of the policy in **Description** to help you and others understand the purpose of the policy, for example, "Suppress alerts from hosts in the preproduction environment to prevent incidents from being created for development and test hosts".

   User-created suppression policies are always the first user policies to be run. The order is set to 0 and cannot be changed.
6. To define the conditions for alerts to activate alert suppression, click **Add condition** and select **Alert property** or **Alert insight** attributes. For this example, Alert property is used.
7. For each attribute, select values for **Property**, **Operator**, **Matches**, and **Value**. For **Alert insights** attributes, also select an **Insight type** value, such as **Anomaly** or **Business criticality**.

   For this example, select **`alert.resource.hostname`** from the **Property** list .You can type **`hostname`**, and the system shows in the property list all alert properties that contain the text **`hostname`**. Then, select **`alert.resource.hostname`**.
8. From the **Operator** list, select **starts with**.
9. From the **Matches** list, select **any of**.
10. In the **Value** field, enter **`dev`** and select **`String: dev`**. In the same field, enter **`test`** and select **`String: test`**.

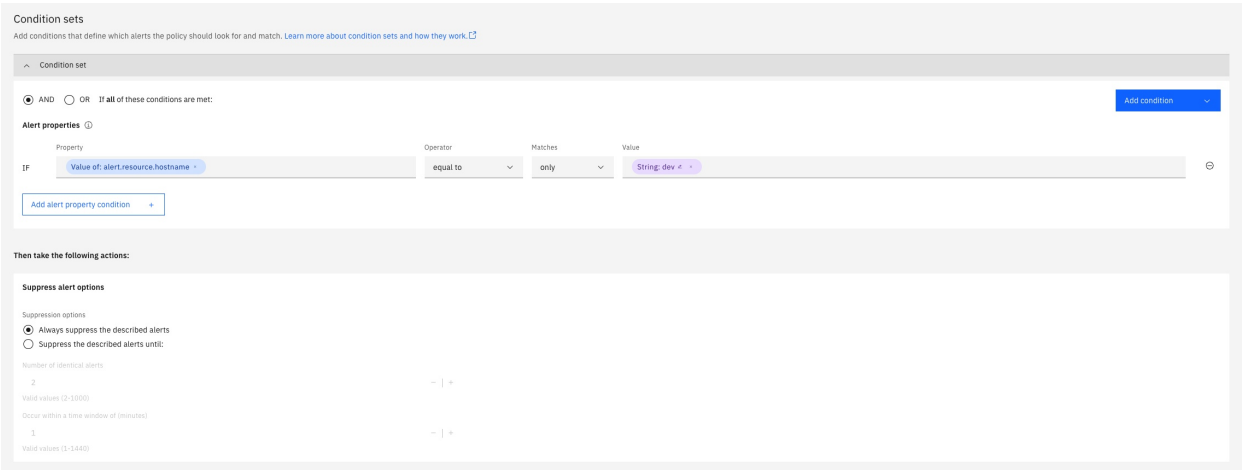


Figure. Supression policy

11. Select **Always suppress the described alerts** or **Suppress the described alerts until** and specify the number of identical alerts or how many alerts occur within a time window of minutes.
12. Click **Create policy**.

New and updated policies can take up to 2 minutes to take effect.

Note: Suppression occurs on the alert level, not the event level. If a suppression policy for a specific condition is enabled, any matching alerts are suppressed. If the policy is removed later, any future matching alerts are not suppressed, and any existing alerts are not unsuppressed. If the suppressed alert is not closed, for example, if the problem is still active, any future events are deduplicated into the suppressed alert until the problem is resolved.

Scenario where alerts can be both suppressed and part of an incident

Suppressed alerts are not typically included in incidents. However, take a scenario where a group of alerts are subject to a scope-based policy. One alert in the group is not suppressed, and also matches the conditions of an incident policy. Then it is possible for the other alerts to show suppressed and in an incident. They are added to the incident because the scope grouping makes them contextual alerts for that incident.

In this image the one alert which is not suppressed is subject to the same scope-based policy as all the suppressed alerts. If that alert also matches an incident policy, then the other "suppressed" alerts are all included in the incident. This is expected as scope policies still run for suppressed alerts. On the Alert Viewer then it is possible if these circumstances occur for alerts to be both suppressed and part of an incident.

In incident	Suppressed
Yes	Yes
Yes	Yes
Yes	No
Yes	Yes
Yes	Yes
Yes	Yes

Suppressed alerts and part of an incident

Suppress seasonal alerts

Generate a new suppression policy to suppress the in-season alerts identified by an analytics-created seaonal policy. After the suppression policy is created, you can edit it to add conditions if needed.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**. The Policies table is displayed.
3. Click the Filter icon to open the filter side panel.
4. Select the **Seasonal** tag and click **Apply**.
5. In the Policies table, click anywhere in the row of the analytics-created seaonal policy that you want to generate a suppression policy for.
6. In the side panel displayed, click the plus icon "**+**" beside **Generate suppression policy**.
7. Enter a name for your policy that is 1-120 characters. Leading and trailing blank spaces will be removed from the policy name if provided.
8. Click **Generate policy**.

The new suppression policy is created and available in the Policies table. You can edit the policy to add conditions if required.

To see how many suppression policies or other policies reference a seasonal policy ID, complete the following steps:

1. In the Policies table, select the seasonal policy of interest.
2. If one or more related policies are found, click the search icon for **All related policies**.

The list displayed is generated by searching all policy names and conditions for the policy ID of the selected seasonal policy. You can click **Clear filters and show policies in table** to launch out to the related policies in the Policies table.

Suppress seasonal alerts using the policy templates

You can also create your own suppression policy for seasonal alerts by using the policy template and condition builder.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Suppress alerts** tile.
5. Enter a **Policy name** and optionally a **Description**.
6. Suppress seasonal alerts by making the following selections in a condition set:
    1. Click **Add condition** and select **Alert insight**.
    2. Set the Insight type to **Seasonal**.
    3. From the Property drop-down list, select `insight.details.policyId`.
    4. Set the Operator to **equal to**.
    5. Set Matches to **only**.
    6. Set the Value to the seasonal policy ID.

For more information about alert seasonality, see Displaying alert seasonality.

Group alerts based on scope

Group alerts based on common properties and time of occurrence to identify when issues might be related. When related alerts are grouped based on their scope, you can view the details in the Alert Viewer.

For example, a scope-based grouping policy might be to group events together that occur at the same physical geographical location within a 5 minute window. The scope can be any attribute that makes sense in the context of the correlation: the same application group, business unit, or physical building.

Note: There can be multiple scope-based policies each with different time windows.

IBM Cloud Pak for AIOps includes two scope-based grouping policies:

- Group alerts that have the same resource name (host name), and use a 15 min rolling window (default).
- Group alerts that come from a Netcool environment where the *ScopeID* field is set, based on its value, and use a 900 second rolling window.

The default scope-based policy in Cloud Pak for AIOps is a correlation on alert.resource.name constrained by a rolling window of 15 minutes. By using this template, you can override the default scope-based groupings.

Note: Alerts in Cloud Pak for AIOps can have multiple scopes and be members of more than one scope-based group at once. If groups have common events, these groups are automatically merged in Cloud Pak for AIOps into super groups.

About this task

In this example, you want to scope alerts by their cluster name and by their location.

Example

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Group alerts based on scope** tile.
5. Enter a name in **Policy name**, for example, the name can be "Sales region 2 scope policy". You can also add an explanation of the policy in **Description** to help you and others understand the purpose of the policy, for example, "Scope alerts from the sales region 2 cluster by location".
6. Set the **Execution order** to higher.
7. Define the following conditions for alerts that will activate scope-based alert grouping, based on the ScopeID that is specified:
    1. Click **Add condition** and select **Alert property**.
    2. From the Property drop-down list, select **alert.resource.cluster**. You can type "cluster" and the system will show in the property drop-down list all alert properties that contain the text "cluster", which in this case is two properties.
    3. From the Operator drop-down list, select **equal to**. From the Matches drop-down list, select **only**.
    4. In the Values field, enter **SALES.REGION2.CLUSTER** and select **String:SALES.REGION2.CLUSTER**.
8. Add alert properties, strings, or both to create the ScopeID for this group. These properties define the scope for how the alerts are grouped. If **`alert.details`** is selected in the property field, a secondary input box is displayed underneath. The **Details name** field is an optional input where you can minimize the scope to a singular key within the alert's details. For example, if you enter **`specificKey`** in the Details name field, it is understood as **`alert.details.specificKey`**. You must enter a string value that matches a key from an alert's details.

    Note: For ScopeID you can send only specific alert details by name and not the entire **`alert.details`** object. You must enter a string value in the **Details name** field.

    In this example, the two alert fields in **ScopeID** are separated with string ":". This will group all alerts from the cluster (SALES.REGION2.CLUSTER), but sub-divide them based on their locations (for example, Dallas, New York).

    Tip: To reorder the ScopeID fields, click on the Draggable icon ⠿ within a 'Value of' or 'String' field and move it to the desired location in the **ScopeID** field.

    For this example specify the following ScopeID:

    1. Click anywhere in the properties field and select **alert.resource.cluster** (or type "cluster" and select it from the options displayed).
    2. In the same field, enter a colon **":"** and select **String::**.
    3. Finally, click in the field again and select **alert.resource.location** (or type "location" and select it from the options displayed).

    Example scope-based policy

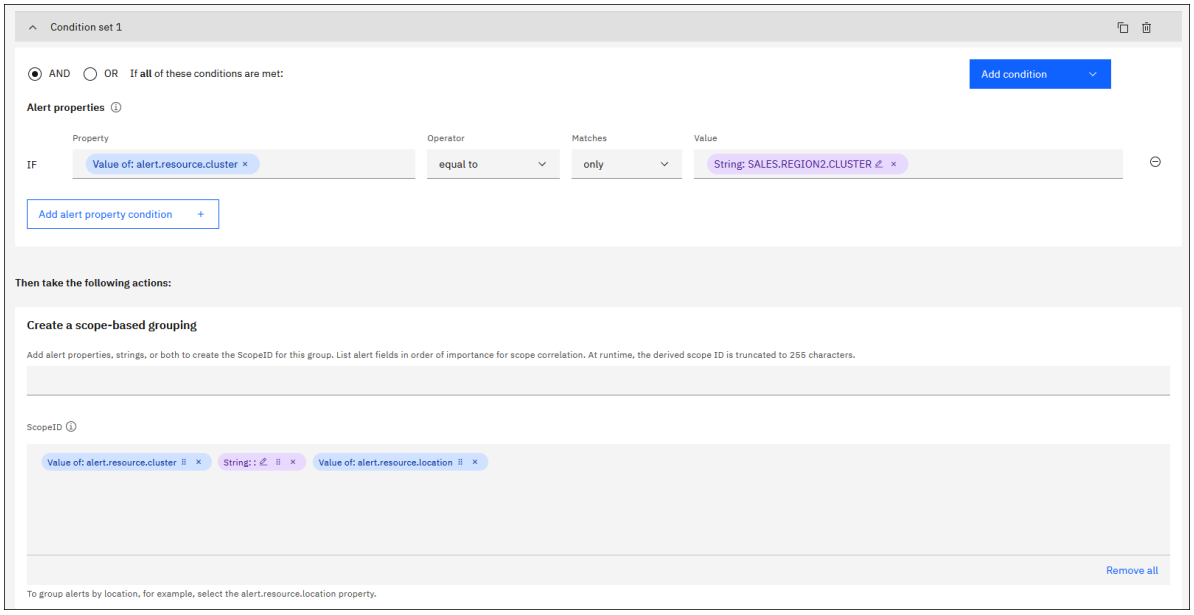


Figure. Scope-based policy

9. Specify a **Time window** (in seconds) to group alerts that match the scope. The minimum value is 10 seconds. The maximum recommended value should not exceed 3600 seconds (1 hour).

    Select the time window **Type**:

    - **Rolling**: The time window value is the threshold number of seconds that need to pass with no further alerts, after which the system stops including alerts in the group.
    - **Fixed**: The time window value is a set amount of time during which incoming alerts are included in the group.

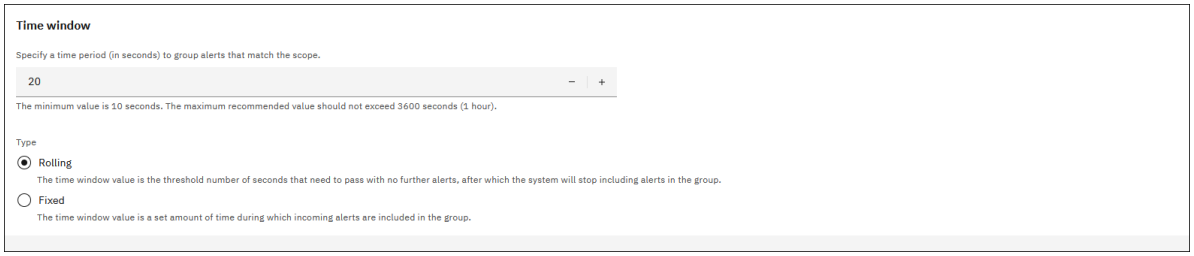


Figure. Time window for scope-based policy

10. Click **Create policy**.

Scope-based alert grouping is activated for alerts that match the condition, based on the ScopeID specified.

New and updated policies can take up to 2 minutes to take effect.

Assign a runbook to alerts

Assign a runbook to alerts for easier automated resolution.

Important: in runbook policies, the actions from every matching condition set are executed. Policies are triggered upon alert creation only.

About this task

In this example, you want to have a runbook take action when you receive alerts warning that your disk space is filling up. The runbook could delete the contents of the `/tmp` directory to free up space. You can create a policy to use this runbook every time alerts warn of your disk space becoming full.

Note: This example assumes you have a runbook that is called **Clear tmp directories**. For information about setting up runbooks, see Runbooks.

Example

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Assign a runbook to alerts** tile.
5. Enter a name in **Policy name**, for example, "Clear tmp directories". You can also add an explanation of the policy in **Description** to help you and others understand the purpose of the policy, for example, "Policy to assign a runbook to alerts that indicate that disk is full due to /tmp filling up".
6. Set the **Execution order** to 50.
7. Define how the policy is triggered to assign a runbook to alerts. Once triggered, it will look for alerts that match the conditions you specify in order to take action. The policy triggers are when an alert is created, updated, or both. Updated means the alert state changes. For example, if an alert changes from Severity 1 to Severity 5. In this example, select **An alert is created**.
8. Define the following conditions for alerts that will assign a runbook to alerts:
    1. Click **Add condition** and select **Alert property**.
    2. From the Property drop-down list, select **alert.type.classification**. You can type "class" and the system will show in the property drop-down list all alert properties that contain the text "class", which in this case is only alert.type.classification. From the Operator drop-down list, select **equal to**. From the Matches drop-down list, select **only**. In the Values field, enter **Disk Full** and select **String:Disk Full**.
    3. Click **Add alert property condition**.
    4. From the Property drop-down list, select **alert.severity**. From the Operator drop-down list, select **greater or equal**. From the Matches drop-down list, select **only**. In the Values field, select **5-Major**.
    5. Click **Add alert property condition**.
    6. For the third condition, select **summary** from the Property drop-down list. From the Operator drop-down list, select **contains**. From the Matches drop-down list, select **only**. In the Values field, enter **/tmp** and select **String:/tmp**.
9. In **Assign Runbook(s)**, select one or more runbooks from the list to assign. In this example, select the **Clear tmp directories** runbook.
10. Review the list of parameters that are required for each runbook selected. You must decide how the values of the runbook parameters are filled by creating a parameter mapping. Select from the following mapping options:

    - **Choose from alert**: Select the name of an alert field from the tree view, its value will be passed to the runbook parameter. For string variable type parameters only, you can select `alert.details` as the alert property. The **Details name** field is an optional input where you can minimize the scope to a singular key within the alert's details. For example, if you enter `specificKey` in the Details name field, it is understood as `alert.details.specificKey`. You must enter a string value that matches a key from an alert's details.

      Note: For runbook parameters you can send only specific alert details by name and not the entire `alert.details` object. You must enter a string value in the **Details name** field.

      Important: The parameter **Type** is displayed on the UI. If this option is selected, you should select alert fields that contain matching types of data. For example, strings vs integers.

- **Enter static value**: A fixed value for the parameter at runtime.
- **Choose at runtime**: After launching the runbook from the incident page, if this option is selected, you must provide a value at runbook execution time. Sometimes the default value as set in the runbook is prefilled for the parameter, in this case just edit the value at execution time.
- **Use default parameter value**: If the runbook defines a default value for this parameter, you can use that value. This option is grayed out if no default was set in the runbook.

For this example, select **Use default parameter value**.

Example runbook policy

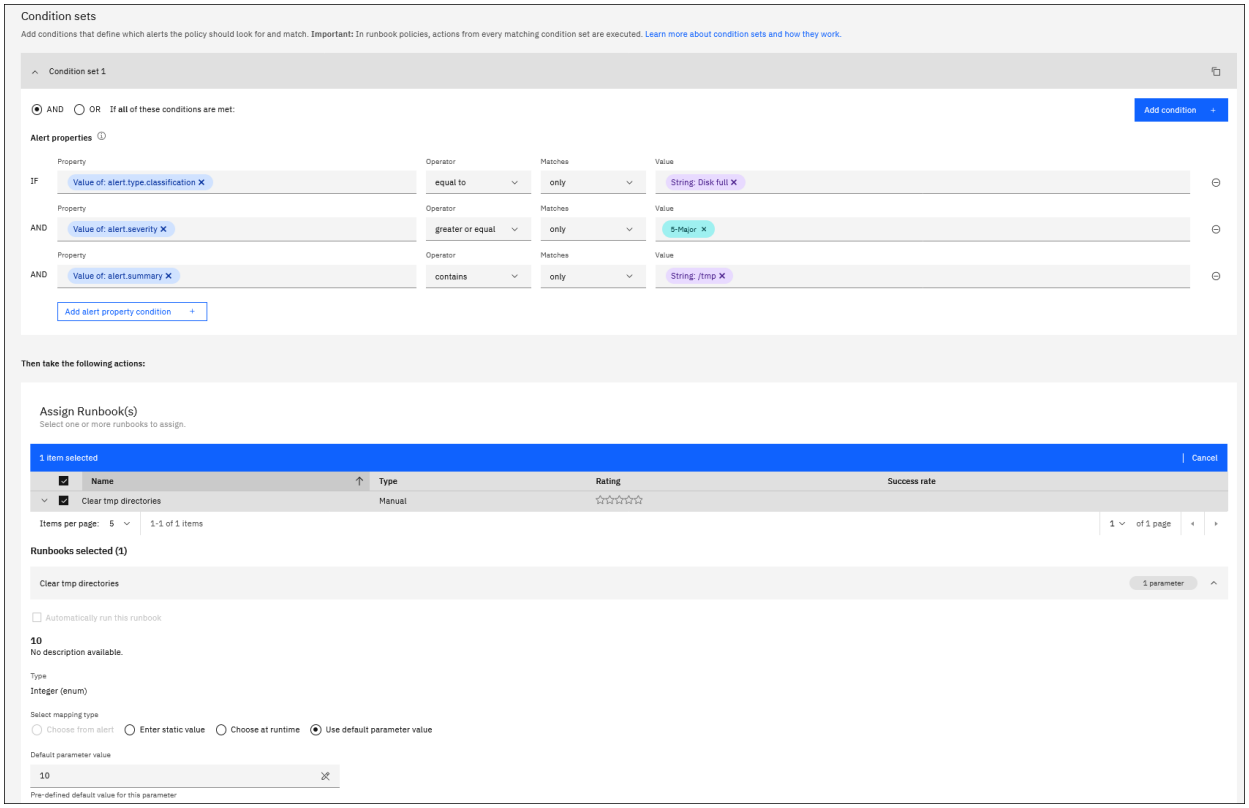


Figure. Supression policy

11. Click **Create policy**.

New and updated policies can take up to 2 minutes to take effect.

Note: If more than one runbook policy is triggered for the same alert and the same runbook is assigned, the policy parameter selections from the last matching condition set or policy are passed to the runbook.

Promote alerts to an incident

Create actionable incidents from alerts to improve insights into your issue. You can then automatically assign incidents to a user group and owner. You can also send notifications about the incident through ChatOps.

Important: In incident policies, only the actions from the first matching condition set are executed. Policies are triggered upon alert creation only.

About this task

In this example, all high priority incidents that include alerts at or higher then a defined business criticality level are automatically assigned to the Db2 admin group with a notification sent to the group's team leader of such incidents.

There are three preset business criticalities available in Cloud Pak for AIOps:

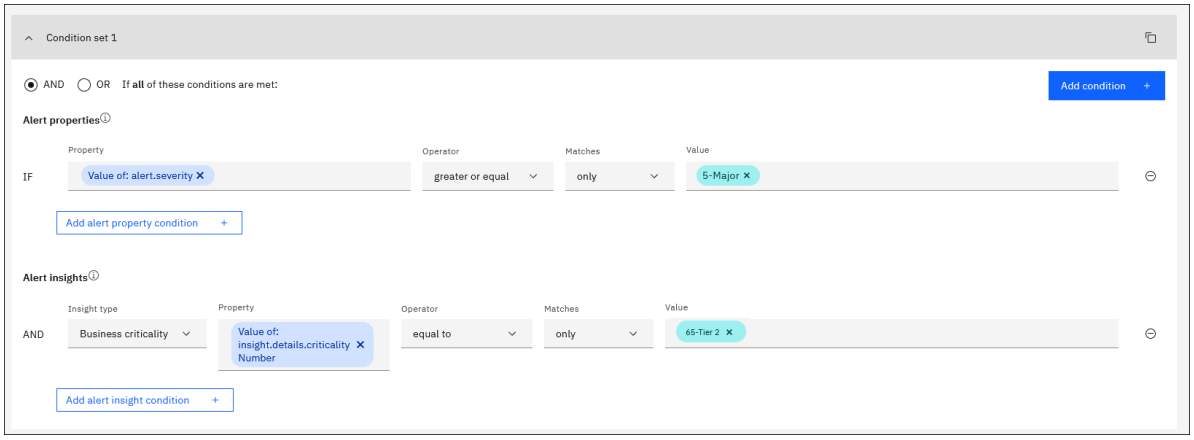
Table. Preset business criticalities

Name	Description	Criticality score
Tier 1	Highest priority	100
Tier 2	Medium priority	65
Tier 3	Lowest priority	25

Note: Preset business criticalities can be edited after they are added to your environment. For more information, see Defining business criticality.

Example

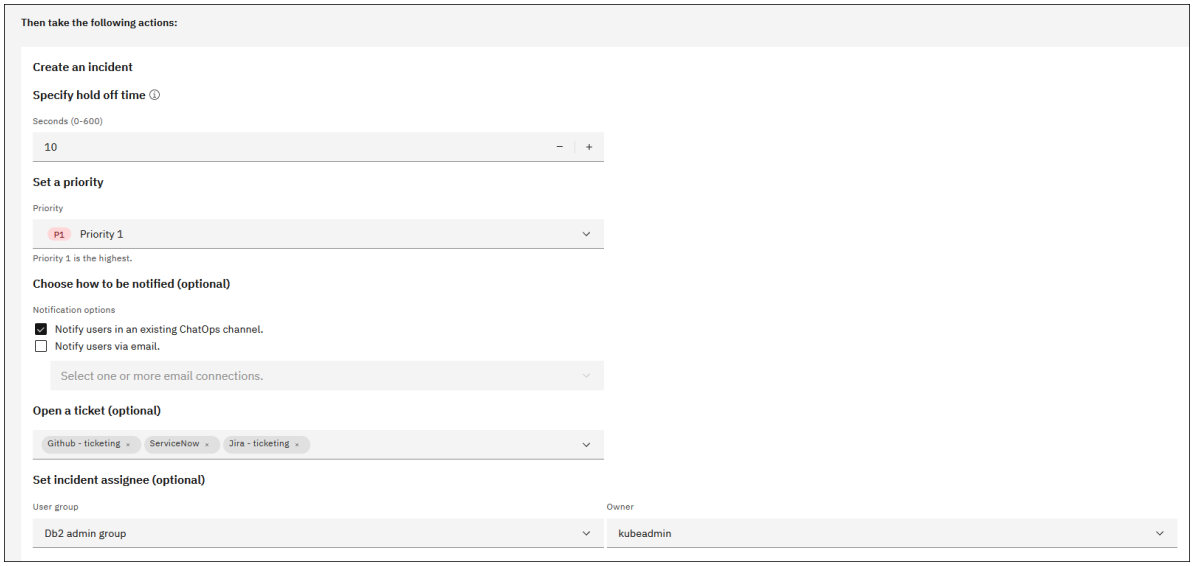
1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Promote alerts to an incident** tile.
5. Enter a name in **Policy name**, for example, "Assign high priority Db2 incidents to Db2 admins". You can also add an explanation of the policy in **Description** to help you and others understand the purpose of the policy, for example, "Automatically assign any priority 1 incidents from Db2 hosts to the Db2 admin group, and notify team leader".
6. Set the **Execution order** to 1.
7. Define how the policy is triggered to create an incident. Once triggered, it will look for alerts that match the conditions you specify in order to take action. The policy triggers are when an alert is created, updated, or both. Updated means the alert state changes. For example, if an alert changes from Severity 1 to Severity 5. In this example, select **An alert is created**.
8. Define the following conditions for alerts that will trigger the creation of an incident:
    1. Click **Add condition** and select **Alert property**.
    2. From the Property drop-down list, select **alert.severity**. You can type "sev" and the system will show in the property drop-down list all alert properties that contain the text "sev", which in this case is only alert.severity. From the Operator drop-down list, select **greater or equal**. From the Matches drop-down list, select **only**. In the Values field, select **5-Major**.
    3. Click **Add condition** and select **Business criticality**.
    4. The Property field contains the business criticality value by default. From the Operator drop-down list, select **greater or equal**. From the Matches drop-down list, select **only**. In the Values field, select **65-Tier 2**.

    Incident policy condition set
9. **Set hold off time**. Delaying the creation of an incident provides time for data about the issue to be ingested. This can allow the system to make a better decision about what incidents should be created. For example, for flapping alerts or alerts which might be resolved by an automated runbook. In these cases, the incident would only be opened if the alert remains open beyond the hold off time. Eliminating incidents from the UI which ultimately are not actionable.

    Enter the number of seconds that you want to wait before the incident is created (maximum 600 seconds). In this example, the hold off time is 10 seconds.
10. **Set priority**. Give a priority level to the incident that is created when the policy conditions are met. For this example, select **Priority 1**.
11. Select how you want to receive **Notifications** about the incident (optional):
    - Notify users in an existing ChatOps channel.
    - Notify users via email. To use this option, you must have an email notification integration set up.
12. Open a ticket (optional). You can multi-select ticketing options from the drop-down menu. The required connections for GitHub, ServiceNow, or Jira must be set up in Integrations.
13. Assign a user group and owner to route the incident to (optional). In this example, the user group is the Db2 admin group, and the owner the group's team leader.

    Example incident policy actions:



Incident policy actions

14. Click **Create policy**.

New and updated policies can take up to 2 minutes to take effect.

Invoke an IBM Tivoli Netcool/Impact policy

Enable Cloud Pak for AIOps to invoke an IBM Tivoli Netcool/Impact policy that will take actions on your alerts.

Note: If you have existing **Impact Policy Language (IPL) policies** (versus JavaScript policies) that you want to use in Cloud Pak for AIOps to return the event, you must modify the return statement from:

```
ReturnEvent(EventContainer);
```

To:

```
AIOPS_IPL_Utils.AIOPSReturnEvent(EventContainer);
```

For example:

```
@Summary=EventContainer.mynewsummary + " Enriched with this message";

AIOPS_IPL_Utils.AIOPSReturnEvent(EventContainer);
```

About this task

For example, you want Cloud Pak for AIOps to invoke an existing IBM Tivoli Netcool/Impact policy called Notify_Customer_Email. The policy accepts incoming event information from an event reader service and uses this information to look up affected internal customers in an external database. The policy then sends an email to each customer and reports the event data to a third-party help desk system that uses the web services API provided by that application.

For more information about setting up IBM Tivoli Netcool/Impact policies, see the IBM Tivoli Netcool/Impact documentation.

Important:

- To successfully enable your IBM Tivoli Netcool/Impact policy with Cloud Pak for AIOps, you must first configure an IBM Tivoli Netcool/Impact integration in **Define** > **Integrations**.
- By default, IBM Tivoli Netcool/Impact policies are invoked without input or transformation. Editing options are available in the policy template.

Example

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Invoke Netcool/Impact** tile.
5. Enter a name in the **Policy name**, for example, "Notify customer help desk". You can also add an explanation of the policy in the **Description** to help you and others understand the purpose of the policy, for example, "Look up customers affected by the event in the external database."
6. Set the **Execution order** to 50.

7. Define how the policy is triggered to invoke an IBM Tivoli Netcool/Impact policy. Once triggered, it looks for alerts or incidents that match the conditions you specify in order to take action. First, select the entity that this policy acts on. Choose either **Alert** or **Incident** properties. The default trigger entity is Alert.

   The policy triggers are when an alert or incident is created, updated, or both. Updated means the alert or incident state changes. For example, if an alert changes from Severity 1 to Severity 6, or if an incident escalates from Priority 5 to Priority 1 (this is an escalation as Priority 1 is the highest priority level).

   Note: Switching between alert and incident entities resets the input fields for triggers, condition sets, and actions.

   For this example, we select **After an alert is created**.

8. Define the following conditions for alerts that trigger your IBM Tivoli Netcool/Impact policy:

   1. Click **Add condition** and select **Alert property**.
   2. From the Property drop-down list, select **alert.severity**. You can type "sev" and the system will show in the property drop-down list all alert properties that contain the text "sev", which in this case is only alert.severity. From the Operator drop-down list, select **greater or equal**. From the Matches drop-down list, select **only**. In the Values field, select **5-Major**.

9. In **Invoke Netcool/Impact**, specify the IBM Tivoli Netcool/Impact integration instance to invoke the policy. If you don't see anything in the **Impact connection** dropdown list, see IBM Tivoli Netcool/Impact integrations.

10. Select your **Impact policy** Notify_Customer_Email from the drop-down list. You can start typing the policy name in the field that is provided and the matching IBM Tivoli Netcool/Impact policy names are suggested.

11. By default, your IBM Tivoli Netcool/Impact policy parameters are mapped to all alerts or all incidents. Select from the following IBM Tivoli Netcool/Impact policy parameter-mapping options:

    - Send the full alert or incident
    - No input
    - Customize

Send the full alert or incident

Send the entire alert or incident without alteration. Click **Create policy** to save the policy.

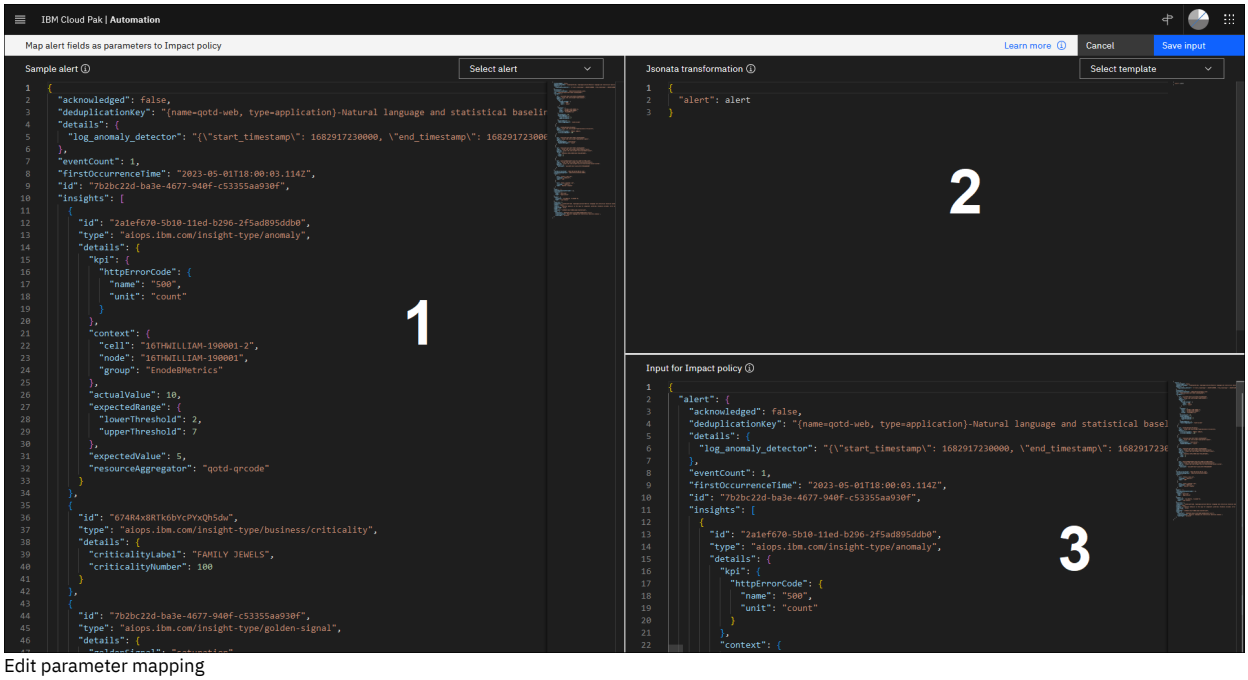
No input

Your IBM Tivoli Netcool/Impact policy requires no input. Click **Create policy** to save the policy.

Customize

Select this option if you want to customize alert or incident field parameters that are used as input for your IBM Tivoli Netcool/Impact policy. Click **Edit parameter mapping** to continue.

In this example, alert fields are being mapped as parameters to an IBM Tivoli Netcool/Impact policy. If incidents are selected as the policy trigger entity (see step 7 in the preceding example), then incident fields are mapped as parameters here. The editor is divided into three sections. You can click and drag the vertical and horizontal middle lines to resize the sections.



Edit parameter mapping

- Section 1: **Sample alert**

  Use the **Select alert** drop-down menu to load a sample alert, or go to the Alert Viewer to copy and paste a raw alert by opening **Alert details** > **Information** > **Raw** > **Copy to clipboard**. For more information, see Alert details.

  The sample alert provides variables available for use in your mapping, including the alert that the policy is acting on. You can modify the variables to test out different possible inputs to your IBM Tivoli Netcool/Impact policy, for example, by overriding the alert with one from your alert list.

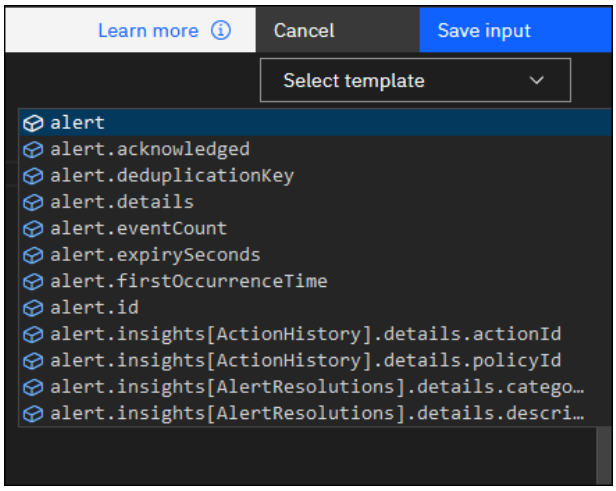
Note: The sample alert is only used to test your transformation and has no bearing on the execution of your policy.

- Section 2: **JSONata transformation**

  A JSONata expression translates between the variables available to the Cloud Pak for AIOps policy and the EventContainer object that is made available to the IBM Tivoli Netcool/Impact policy during execution. Use the JSONata transformation section to include your own arguments for an IBM Tivoli Netcool/Impact action by making a selection from the **Select template** drop-down menu. If you are invoking an IBM Tivoli Netcool/Impact policy that was originally written for an IBM Tivoli Netcool/Impact EventReader, you will likely want to translate the alert variable to a set of ObjectServer columns here.

  When you start typing in **JSONata transformation** (or press Ctrl + Space), a suggestion panel with alert properties is shown. Move the cursor into the panel to scroll and select from the list of properties. As you make changes in this section, the **Input for Impact policy** (section 3) shows a preview of your transformation.

  Note: Policies support individual details by name. However, there is no `alert.details.detailsname` displayed in the JSONata transformation list of properties. To pass such a value, select `alert.details` and then add `."details_name"`. The `alert.details` name must be specified in quotation marks (`" "`). That is `alert.details."details_name"`. The value of `alert.details."details_name"` will then be passed at run time.

  Parameter suggestion panel

  Warning: Any changes that you make to the source alert are reset if you reedit the parameter mapping.

- Section 3: **Input for Impact policy**

  This section shows the results of your transformation and what the content of the EventContainer variable in IBM Tivoli Netcool/Impact will contain, based on the current sample alert and transformation rules that are used in section 2.

12. Click **Save input** to save your changes and close the editor. Then, click **Create policy** to save the policy.

    New and updated policies can take up to 2 minutes to take effect.

Populate a dashboard or report

BETA The dashboard or report policy is available only for technology preview, and is not for production usage.

Send alerts or incidents from Cloud Pak for AIOps to populate a dashboard or report.

About this task

Adding an IBM Db2 integration enables your organization to store alert and incident data in a Db2 database. Using this template, you can create policies that send that alert or incident data from Cloud Pak for AIOps to populate a dashboard or report in IBM Cognos® Analytics.

In this example, you want to send alerts to populate a dashboard when a new alert is created or when an alert severity level changes. The policy should look for and match only alerts that have an alert summary.

Example

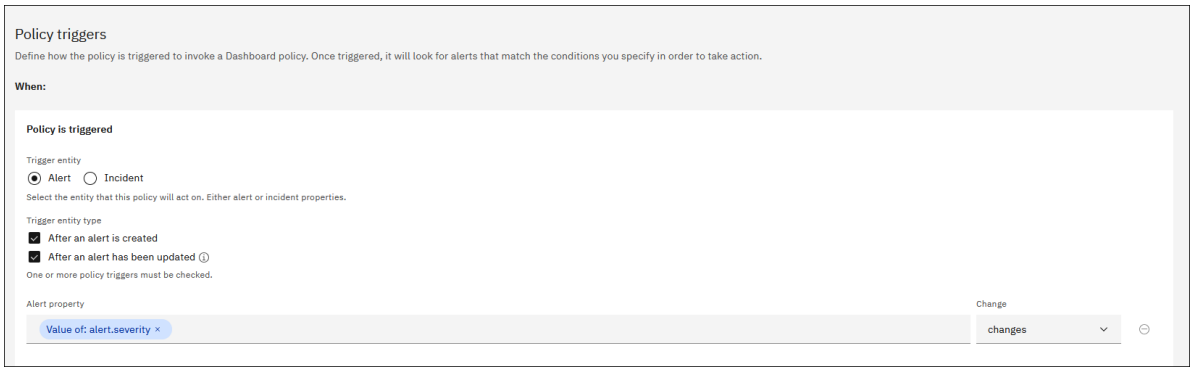
1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Populate a dashboard or report** tile.
5. Enter a name in the **Policy name**, for example, "Alerts Dashboard". You can also add an explanation of the policy in the **Description** to help you and others understand the purpose of the policy.

6. Set the **Execution order** to 50.
7. Define how the policy is triggered to invoke a dashboard policy. Once triggered, it looks for alerts or incidents that match the conditions you specify in order to take action. First, select the entity that this policy acts on. Choose either **Alert** or **Incident** properties. The default trigger entity is Alert.

   The policy triggers are when an alert or incident is created, updated, or both. Updated means the alert or incident state changes. For example, if an alert changes from Severity 1 to Severity 6, or if an incident escalates from Priority 5 to Priority 1 (this is an escalation as Priority 1 is the highest priority level).

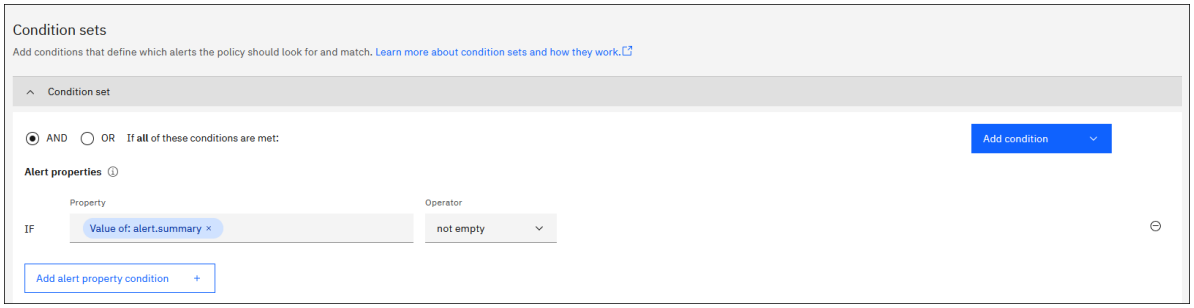
   Note: Switching between alert and incident entities resets the input fields for triggers, condition sets, and actions.

   For this example, we select **After an alert is created** and **After an alert has been updated**.
8. In the alert property drop-down list, select **alert.severity**. You can type "sev" and the system will show in the property drop-down list all alert properties that contain the text "sev", which in this case is only alert.severity.



Policy triggers

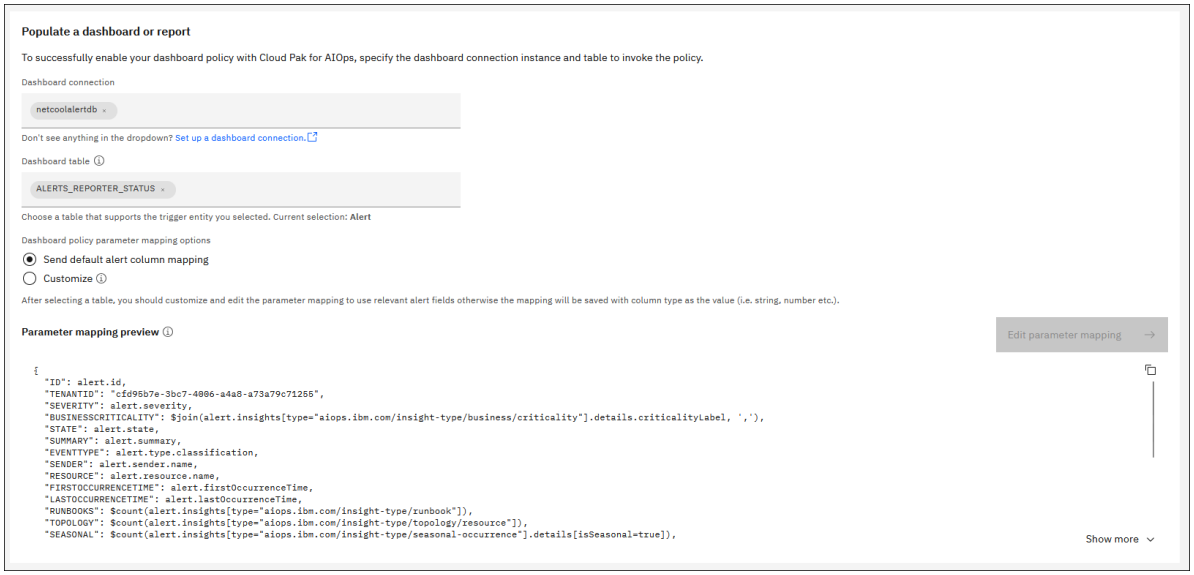
9. Define the following conditions for alerts that trigger your dashboard policy:
   1. Click **Add condition** and select **Alert property**.
   2. From the Property drop-down list, select **alert.summary**. From the Operator drop-down list, select **not empty**.



Policy conditions

10. In **Populate a dashboard or report**, specify the Dashboard connection instance and table to invoke the policy. If you don't see anything in the **Dashboard connection** dropdown list, see IBM Db2 integration.
11. Select your Dashboard table from the drop-down list. You can start typing the table name in the field that is provided and the matching table names are suggested. When selecting the default tables **`ALERTS_REPORTER_STATUS`** or **`INCIDENTS_REPORTER_STATUS`**, a default mapping is provided that matches table columns with relevant alert and incident fields. Ensure that you include the columns "ID" and "TENANTID".

    Note: The database schemas must be installed so that the relevant reporting tables are in place. For more information, see Data flow configuration.



Policy actions

12. Select from the following dashboard policy parameter-mapping options:

    - Send default alert column mapping
    - Customize

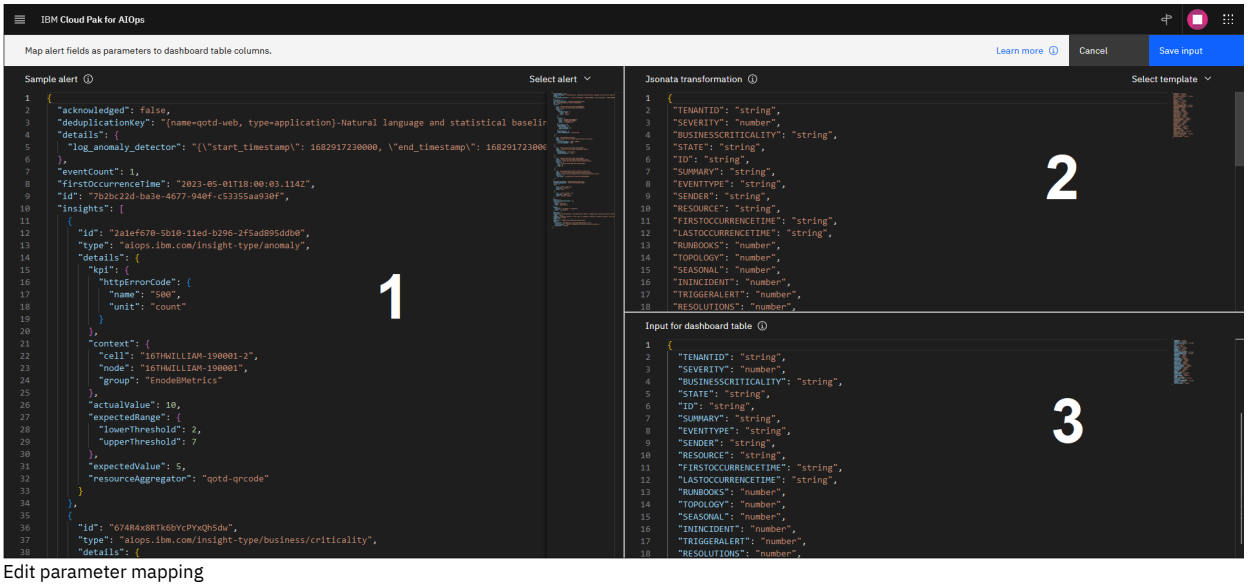
Send default alert column mapping

Send the default alert column mapping without alteration. Click **Create policy** to save the policy.

Customize

Select this option if you want to customize alert or incident field parameters that are used as input for your Dashboard table. Click **Edit parameter mapping** to continue.

In this example, alert fields are being mapped as parameters to a Dashboard table. If incidents are selected as the policy trigger entity (see step 7 in the preceding example), then incident fields are mapped as parameters here. The editor is divided into three sections. You can click and drag the vertical and horizontal middle lines to resize the sections.



Edit parameter mapping

- Section 1: **Sample alert**

  Use the **Select alert** drop-down menu to load a sample alert, or go to the Alert Viewer to copy and paste a raw alert by opening **Alert details** > **Information** > **Raw** > **Copy to clipboard**. For more information, see Alert details.

  The sample alert provides variables available for use in your mapping, including the alert that the policy is acting on. You can modify the variables to test out different possible inputs to your dashboard policy, for example, by overriding the alert with one from your alert list.

  Note: The sample alert is only used to test your transformation and has no bearing on the execution of your policy.

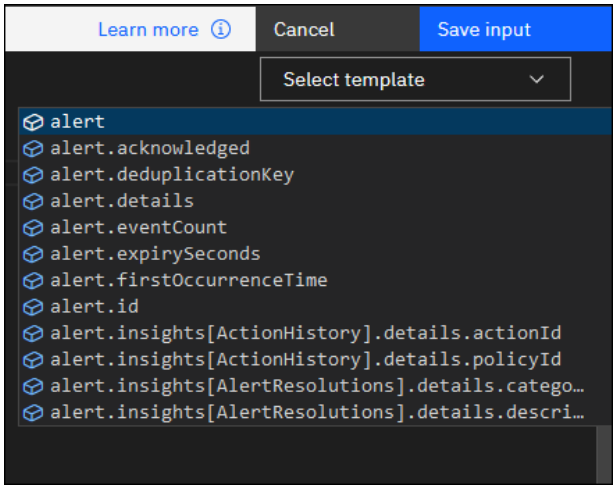
- Section 2: **JSONata transformation**

A JSONata expression translates between the variables available to the Cloud Pak for AIOps policy and the dashboard table columns from a given connection. To avoid max transfer sizes the number of transformation key/value pairs is limited to 100.

Instances of "string" and "number" indicate the type of data that a given column supports. Columns with type "number" only support integer values, any string sent to these columns will cause errors. Ensure that you replace instances of "string" and "number" in the transformation or remove the column if not needed, otherwise, those values are sent to the column.

The keys "ID" and "TENANTID" are required when using the default tables.

Use the JSONata transformation section to include your own arguments for dashboard table columns by making a selection from the **Select template** drop-down menu. Choose between **Show sample column mapping** or **Use columns from table**. When you start typing in the **JSONata transformation** (or press Ctrl + Space), a suggestion panel with alert properties is shown. Move the cursor into the panel to scroll and select from the list of properties. As you make changes in this section, the **Input for dashboard table** (section 3) shows a preview of your transformation.



Parameter suggestion panel

Warning: Any changes that you make to the source alert are reset if you reedit the parameter mapping.

- Section 3: **Input for dashboard table**

  This section shows the results of your transformation and what the content of the table columns contain based on the current sample alert and transformation rules that are used above. For columns that rely on date/time, such as "FIRSTOCCURRENCETIME", use the value from the selected entity (that is, `"FIRSTOCCURRENCETIME": alert.firstOccurrenceTime`).

13. Click **Save input** to save your changes and close the editor. Then, click **Create policy** to save the policy.

    New and updated policies can take up to 2 minutes to take effect.

Policy editor

You can create or customize policies by adding or editing conditions in the policy and the actions that are triggered. Policies are triggered upon alert creation only. This topic contains an example of editing a policy from the policy table. You can also edit a policy from the **Details** tab in the side panel.

Important: In incident policies, only the actions from the first matching condition set are executed. In runbook policies and scope-based policies, the actions from every matching condition set are executed. Suppression policies contain only one condition set.

Policy details and Execution order

The first two sections of the policy editor are **Policy details** and **Execution order**. Here you name (or rename) the policy and add a description. To set the execution order for the policy, use the slider or enter a value between 1-100. Policies with a lower number are processed first. Preset policies that handle system needs are executed before or after those with a value of 1-100. User-created suppression policies are always executed first. The order is set to 0 and cannot be changed.

To see how your current policies are set, click on the following number of policies listed for each range.

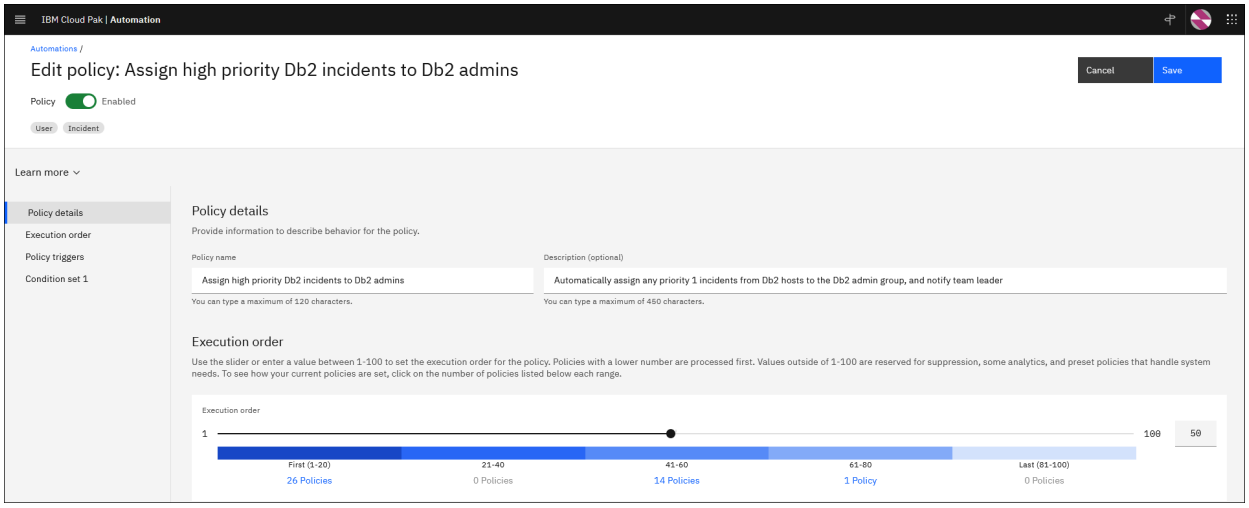


Figure. Policy details

Policy triggers

With the exception of suppression policies, all other policy types have a **triggers** section. These policies are triggered when an alert is created, updated, or both. Updated means the alert state changes. For example, if an alert changes from Severity 1 to Severity 5. If the policy is triggered by the alert create or update, then the conditions are consulted.

If you select **After an alert has been updated**, alert properties can be specified in the policy trigger. As with condition sets, when **`alert.details`** is selected you can specify a singular key within the alert's details to minimize the scope of the policy trigger. You must enter a string value that matches a key from an alert's details. If the Details name input is left empty, all of the details and the details' names are searched. You can also access additional properties of **`alert.sender`**, **`alert.resouce`**, and **`alert.type`** by typing your custom property (for example, **`alert.type.customProperty`**) and selecting the "**`Value of: alert.type.customProperty`**" option. The supported special characters "**-**" (dash) and "**_**" (underscore) must be used with square brackets (that is, [custom-name]). There is a character limit of 255.



Figure. Policy triggers

Note: With the Invoke an IBM Tivoli Netcool/Impact policy, you can select either alert or incident properties in the trigger section.

Condition sets

In a **Condition set** you configure the conditions that alerts must meet. For new policies, start by adding a condition set. At least one is required. Condition sets are created based on alert properties and Cloud Pak for AIOps alert insights. Click **Add condition** to select a new condition type from the drop-down menu.

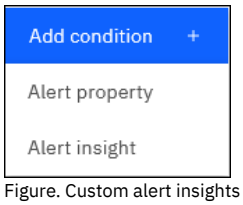


Figure. Custom alert insights

Choose from the following condition types:

- **Alert properties**: Alert attributes that are predefined for IBM Cloud Pak for AIOps and common to most alerts. For more information, see Alert details.
    - When **`alert.details`** is selected in the property field or the value field, a secondary input box is displayed underneath. The **Details name** field is an optional input where you can minimize the scope to a singular key within the alert's details. For example, if you enter **`specificKey`** in the Details name field, it is understood as **`alert.details.specificKey`**. If the Details name input is left empty, all of the details and the details' names are searched and operators of **`contains`**, **`does not contain`**, **`is empty`** and **`not empty`** are the most useful. The Details name field does not provide suggestions for possible keys. You must enter a string value that matches a key from an alert's details. Details name if provided cannot contain spaces, double quotes, backslash, or square brackets.

- You can also access additional properties of `alert.sender`, `alert.resouce`, and `alert.type` that are not in the default alert schema. Type your custom property (for example, `alert.type.customProperty`) and select the "`Value of: alert.type.customProperty`" option. The supported special characters "-" (dash) and "_" (underscore) must be used with square brackets. (that is, [custom-name]). There is a character limit of 255.

- **Alert insights**:
  - *Anomaly*: Log anomaly and metric anomaly attributes that are predefined for IBM Cloud Pak for AIOps. For more information, see Supported Anomaly insights.
  - *Business criticality*: Set in Resource management to define the importance of an application, resource group, or resource to the business. For more information, see Defining business criticality.
  - *Golden signal*: Select from predefined golden signal `labels` and `types`.
    - *Labels*: `traffic`, `latency`, `error`, `saturation`, `availability`, `information`.
    - *Types*: `effect`, `cause`, `none`.
  - *Seasonal*: Alerts that occur within a seasonal time window.

For more information, see Examples of policy conditions mapped to alert JSON.

Note: Schema descriptions for both Alert properties and Alert insights are displayed on the UI. Click anywhere in **Property** field and move the cursor over the properties to display the descriptions.

Schema descriptions

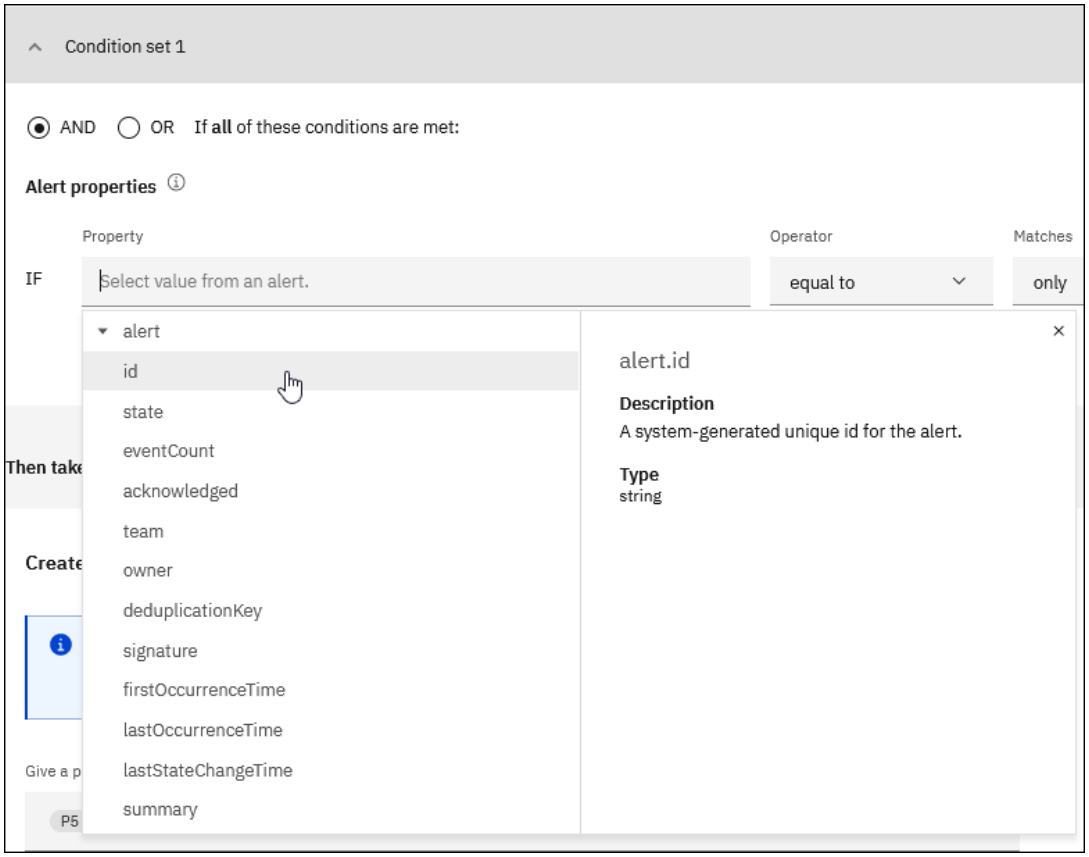


Figure. Schema descriptions

When adding conditions, you can join multiple condition types by using the AND and OR operators. The AND operator means that alerts are matched only if all of the individual conditions are true. The OR operator means that alerts are matched if any of the individual conditions are true.

In the fields provided, select the **Property**, **Operator**, **Matches**, and **Value** for the new condition. Setting a **Value** field to the value of another property allows you to compare the value of one property to one or more other properties. Click anywhere in the **Value** field and scroll the tree-view to select from the available options. You can also enter the last part of the fully qualified tree-view name to see and select the tree-view "value of" alert or insight field. To enter a string value, type the string (e.g. "Error") in the **Value** field and then select **String: Error** from the drop-down. If you enter a fully qualified tree-view value, you can save it as a string versus the "value of". To enter a numerical value, type the number (e.g. "10") in the **Value** field and then select **Number: 10** from the drop-down. Note, negative values are also supported for type Number fields.

If you need more than one condition set, click **Add condition set**. Alternatively, to copy and paste an existing condition set click **Copy condition set**. To rename a condition set, point the cursor over the condition set name and click Edit. You can use the sidebar to navigate a policy's condition sets and actions. To remove a condition set, click **Delete**. At least one condition set is required.

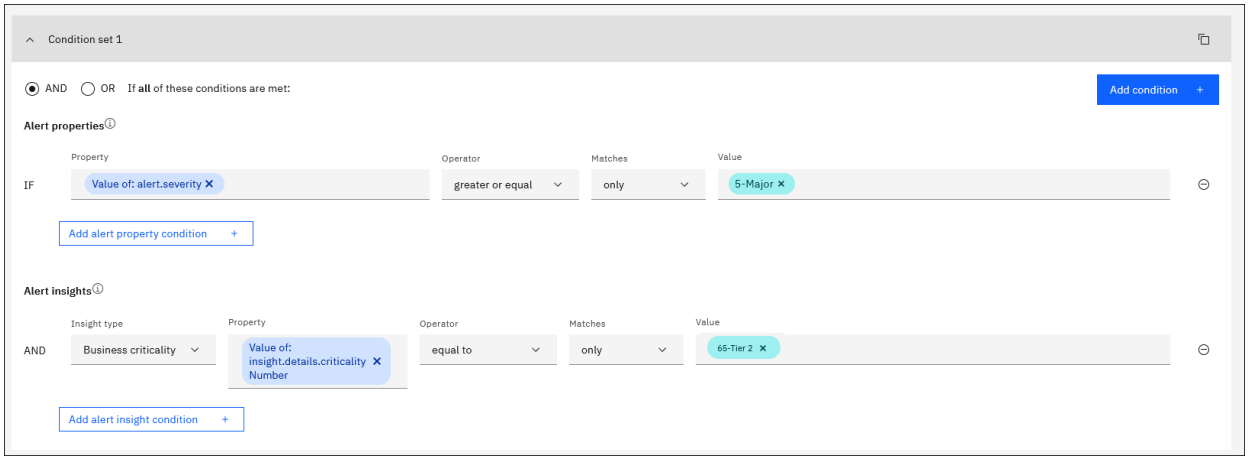


Figure. Condition set

Notes:

- Suppression policies contain only one condition set. You cannot add or copy condition sets in these policies.
- Some information icons in the Policy editor are activated by hover-over, others are activated by clicking. This depends on nearby controls and groups on the UI. Some pages can contain a mixture of hover-over and clickable information icons.
- Unlike **Alert details** properties, **Alert insights** conditions like **Business criticality** and **Anomaly** conditions cannot evaluate `true` where Business criticality or Anomaly insights contain no value. For these two property types, Cloud Pak for AIOps checks first for a value, and if it is not present, `false` is returned. If the value is present, the condition that is assigned by the user is evaluated, and `true` or `false` is returned as determined. Therefore, it is a best practice to avoid use of the negative **Operator** property "**not equal to**" when you use Business criticality and Anomaly insights properties.

Actions

Actions are where you specify what is triggered when the policy conditions are met:

- Suppression policies: the action is always to suppress the alerts described in the condition sets.
- Scope-based policies: the action that is taken is defined by alert properties and strings to create the ScopeID for a group. These properties define the scope for how the alerts are grouped.
- Runbook policies: choose one or more runbooks to assign when the policy conditions are met, and select how parameters are passed.
- Incident policies: the action is the creation of an incident with a specified priority level (example below). You can optionally delay the creation of an incident by setting a hold off time, select how you want to be notified about the incident, open a ticket (GitHub or ServiveNow), and assign a user group and owner.
- Invoke an IBM Tivoli Netcool/Impact policy: enable Cloud Pak for AIOps to invoke an IBM Tivoli Netcool/Impact policy that will take actions on your alerts.

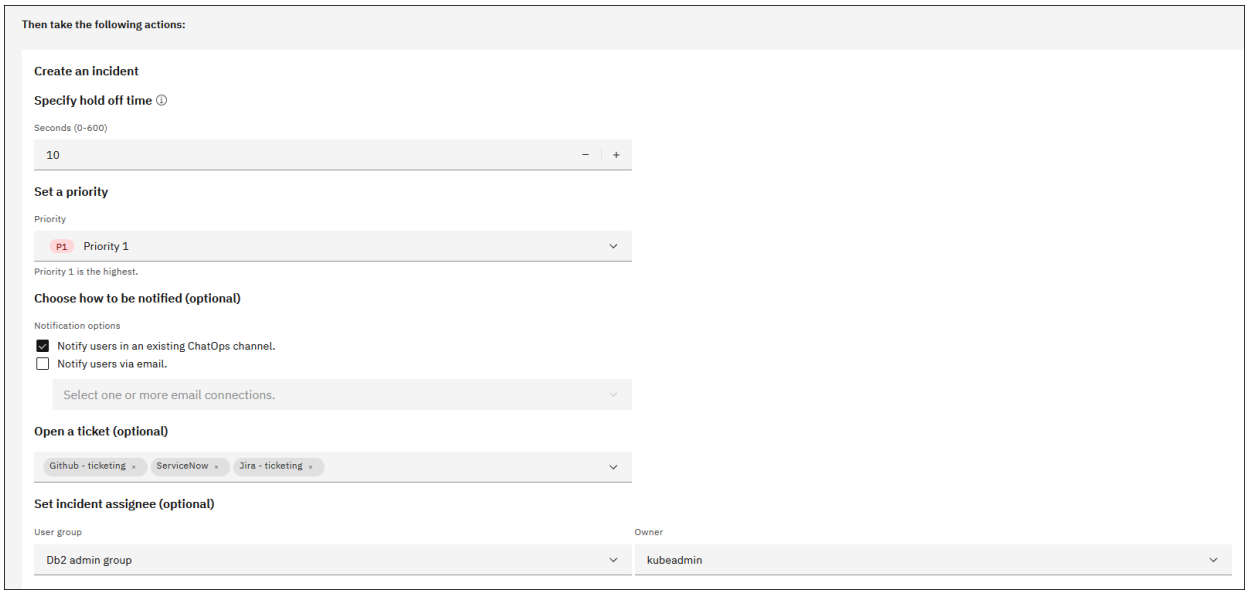


Figure. Actions

Example

In the following example, a condition is added to an existing policy so that the policy applies only to alerts that have a prefix of either "Error" or "Warning" in their Summary field.

1. Click the navigation icon in the toolbar to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. In the table row of the policy that you want to edit, click the menu overflow icon (the three dots at the end of the row) to and select **Edit policy**.
4. Click **Add condition** and select **Alert property**.
    1. From the Property drop-down list, select **alert.summary**. You can type "sum" and the system will show in the property drop-down list all alert properties that contain the text "sum", which in this case is only alert.summary.
    2. From the Operator drop-down list, select **Contains**.
    3. From the Matches drop-down list, select **any of**.
    4. In the Values field, type **Error** and then click **String:Error** in the pop-up. In the same field, type **Warning** and click **String:Warning**.

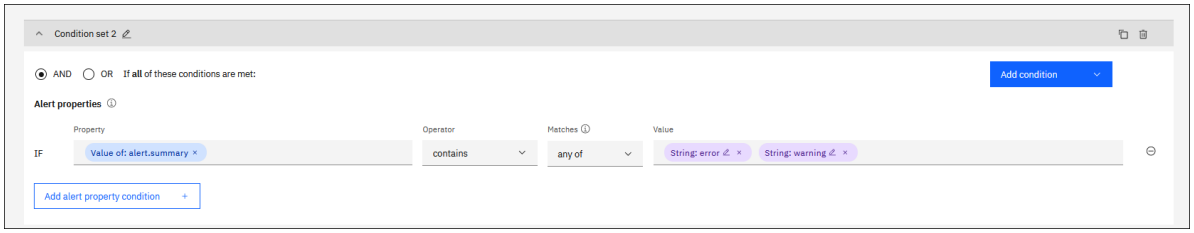


Figure. Policy example

5. Click **Save** to save your changes.

    If the save button is not active, one or more items might be missing in the conditions or runbook parameter sections. If clicking save fails to exit the edit session, again, check that no required details are missing.

Note: New and updated policies can take up to 2 minutes to take effect.

IBM Cloud Pak® for AIOps determines which of your alerts were caused by the same problem. If an active incident exists for a given problem, a new incident is not created.

For more information about creating policies, see the following topics:

- Suppress alerts
- Group alerts based on scope
- Assign a runbook to alerts
- Promote alerts to an incident

Supported Anomaly insights properties

In a policy condition set, you can suppress alerts and create reports from alerts by using the following types of **Anomaly insights** properties:

Table. Anomaly insights properties

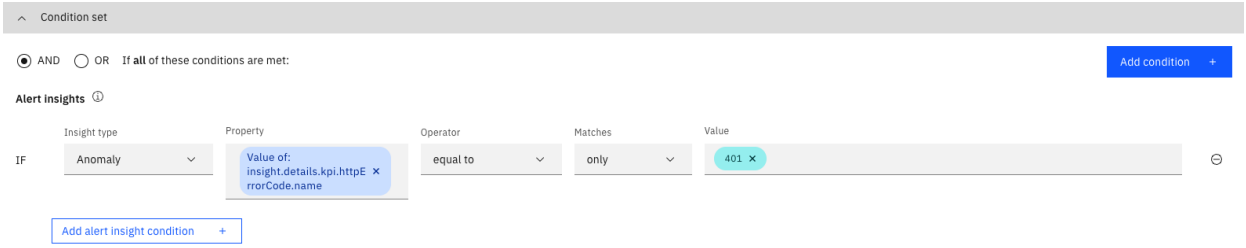
Property	Description	Type	Possible values
`httpErrorCode.name`	HTTP error code	`string`	400, 401, 403, 408, 429, 500, 502, 503, 504, 5xx
`httpErrorCode.unit`	Unit for the HTTP error code KPI	`string`	Count
`exception.name`	Name of the exception (for example, a token that contains the string `exception` found within a Log)	`string`	
`exception.unit`	Unit for the exception KPI	`string`	Count
`messageId.name`	An IBM WebSphere MQ message ID	`string`	
`messageId.unit`	Unit for the message ID KPI	`string`	Count
`metric.name`	Metric name	`string`	
`metric.unit`	Unit for the metric KPI (for example, bytes, mb, ms, s)	`string`	
`actualValue`	The observed value for a KPI.	`number`	
`expectedlValue`	The expected value for a KPI.	`number`	
`expectedRange.upperThreshold`	The expected upper bound of the count/frequency of the KPI in a given time aggregated window.	`number`	
`expectedRange.lowerThreshold`	The expected lower bound of the count/frequency of the KPI in a given time aggregated window.	`number`	

Property	Description	Type	Possible values
`resourceAggregator`	This field is optional. An entity in the underlying client application that can be used to logically aggregate all anomalies that are related to the same application resource. For example, in the context of Metric anomalies this would be a server that might be used to aggregate anomalies across all ports exposed by that server. For Log anomalies, this might be a Kubernetes ReplicaSet that aggregates all the pods it manages.	`string`	

Using properties for alert suppression

Scenario 1: Alert suppression based on HTTP error codes

You might expect the system to generate certain HTTP error types at regular intervals. However, these alerts do not require action and you do not want alerts and incidents generated based on these errors. You can enable alert suppression to avoid new alerts from being generated.

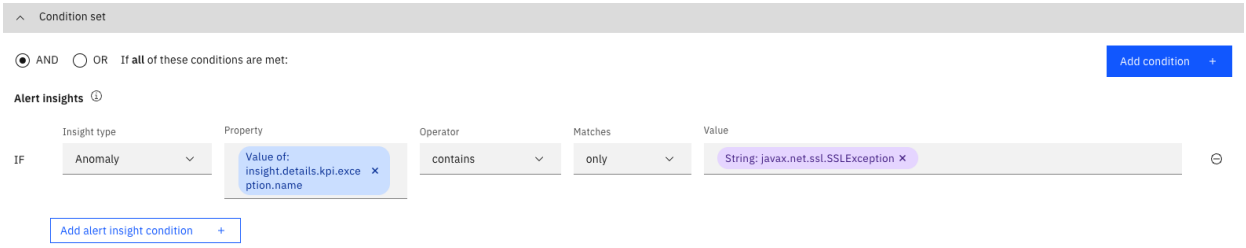


Example 1: Suppress alerts for any 401 error code.

Scenario 2: Alert suppression based on exceptions present in log messages

Some exceptions present in log messages might not be a priority or cause any outages for your environment. As a result, you might want to suppress any alerts that are generated by these exceptions captured in logs.

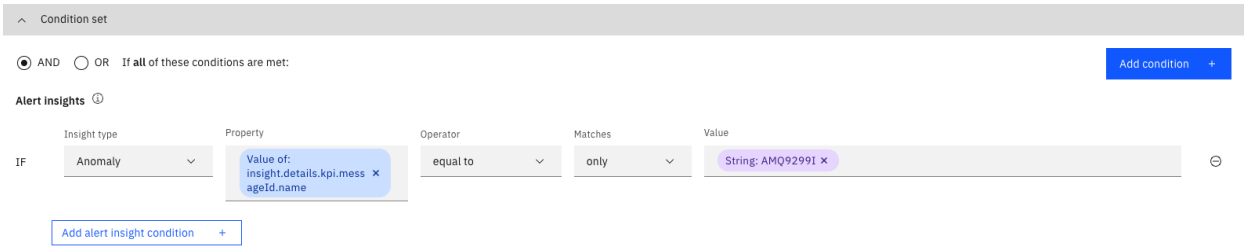
Log anomaly suppression policies can reference only computed or derived entity values from the RSA algorithm, such as the exception name. To suppress certain log anomaly alerts based on the exception name, go to the IBM Cloud Pak® for AIOps Suppress Alerts automation page. Then, from the Alert insight category suppression options, choose the **`insight.details.kpi.exception.name`** property. However, you cannot define a log anomaly suppression policy that references the string message that is found in the log record.



Example 2: Suppress all alerts generated by exceptions of the type javax.net.ssl.SSLException.

Scenario 3: Alert suppression based on domain-specific error code entities that are present in the logs

Domain-specific logs, such as IBM MQ or WebSphere, include error code entities that you can use to set policies to suppress alerts. For example, use the **`insights.details.kpi.messageId.name`** field in an Anomaly insights condition set to suppress alerts that match that entity.

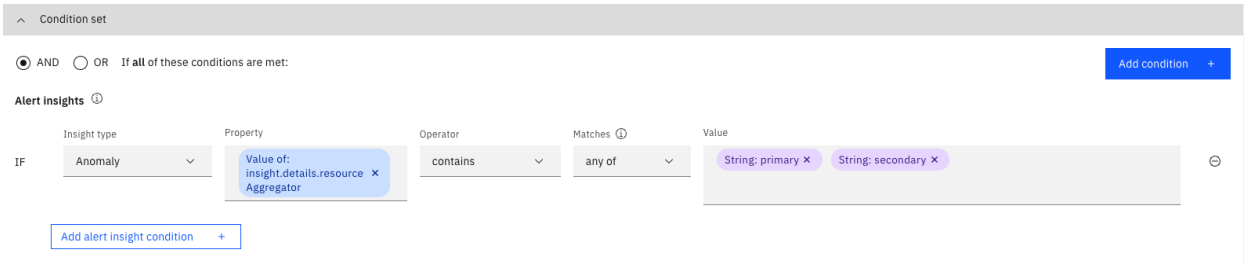


Example 3: Alert suppression based on an incident where the message name equals AMQ9299I.

Using properties for creating runbooks

Scenario 4: Create a runbook based on resource name

Use the **`resourceAggregator`** field in an Anomaly insights condition set to create a runbook based on resource name.

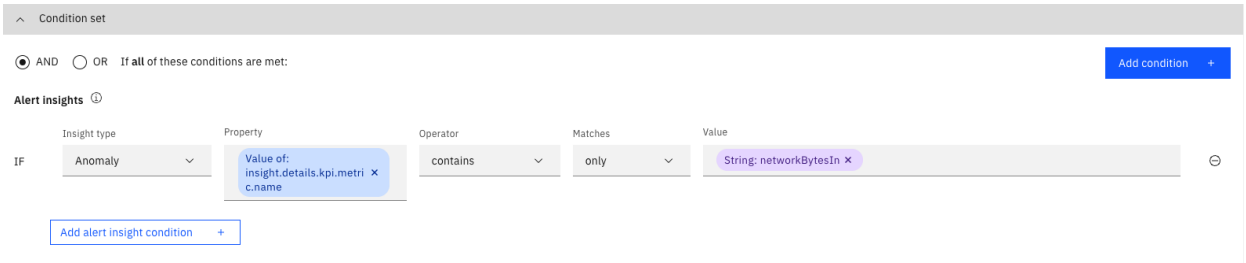


Example 4: Create a runbook for any resource that contains either primary or secondary.

Using properties to invoke IBM® Tivoli® Netcool®/Impact

Scenario 5: Invoke Netcool/Impact based on metric name

Use the **`metric.name`** field in an Anomaly insights condition set to invoke Netcool/Impact based on metric name.

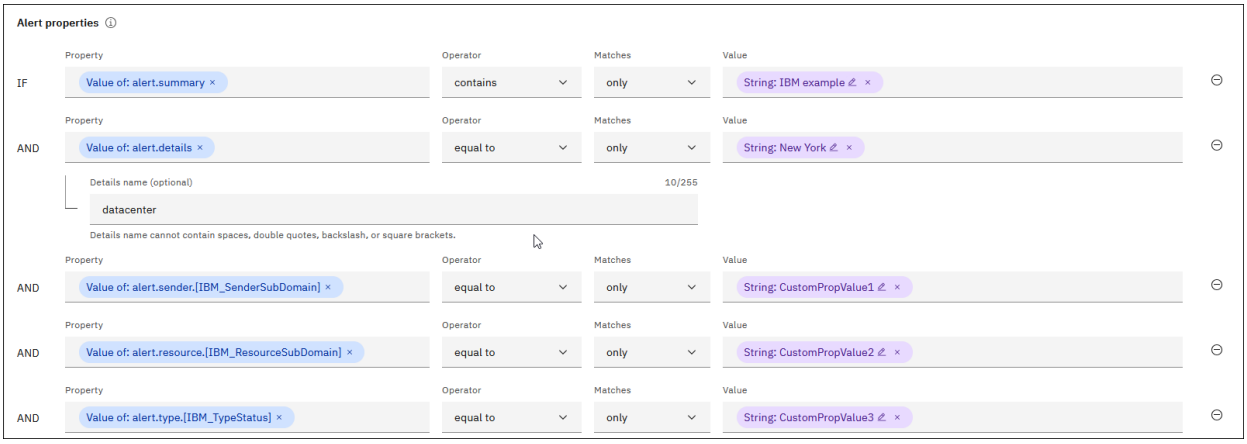


Example 5: Invoke Netcool/Impact where the metric name contains networkBytesIn.

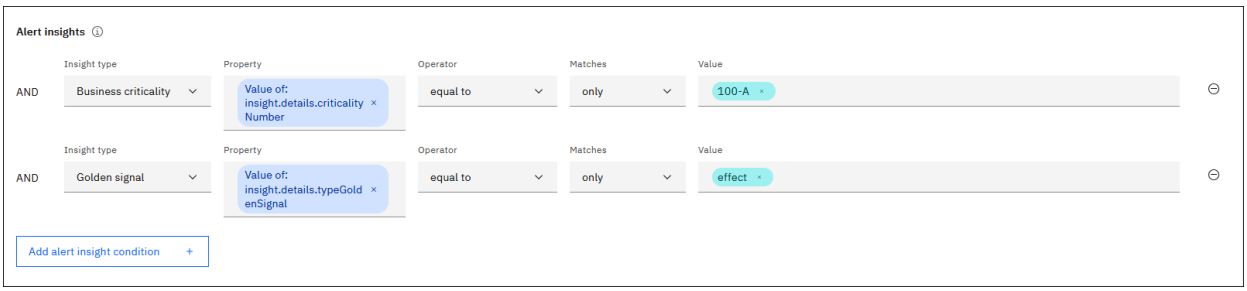
Examples of policy conditions mapped to alert JSON

This is an example of an IBM Cloud Pak® for AIOps policy condition set, which is mapped to corresponding alert JSON. The policy conditions contain both Alert properties and Alert insights. For the alert properties, 'Details name' and custom properties are specified as conditions that alerts must meet.

Policy condition set



Condition set - Alert properties



Condition set - Alert insights

Alert JSON

```
{   "acknowledged": false,   "deduplicationKey": "\{IBM_ResourceSubDomain=CustomPropValue2, IBM Example, type=IBM
Solution}

  "details":

{     "original_problem_id": "7973772",     "vendor": "ICP",     "environment": "Prod",     "region": "US",
"datacenter": "New York",     "location": "40.7685° N, 73.9822° W"    }

,
  "eventCount": 1,
  "expirySeconds": 0,
  "fieldaddedbyuser": "Custom fields can be inserted anywhere, however using custom props increases usability",
  "firstOccurrenceTime": "2024-10-09T18:37:47.999Z",
  "id": "6d3f5066-72fb-41f3-97c1-4df6edd77e01",
  "insights": [
    {
      "id": "674R4x8RTk6bYcPYxQh5dw",
      "type": "aiops.ibm.com/insight-type/business/criticality",
      "details":

{         "criticalityLabel": "Critical Services",         "criticalityNumber": 100       }

    },
    {
      "id": "ccb146a0-c1dc-4f58-a06b-dea3b135dcf7",
      "type": "aiops.ibm.com/insight-type/relationship/causal-union",
      "details":

{         "policyId": "38b6eeb0-804b-11ef-80fc-a972dd876928",         "subsumedUnions": []       }

    },
    {
      "id": "event-occurrence",
      "type": "aiops.ibm.com/insight-type/deduplication-details",
      "details":

{         "lastProcessedEventOccurrenceTime": "2024-10-09T18:37:47.999Z"       }

    },
    {
      "id": "fb9ba49c-4c0c-4f71-9c85-ee7060b76c29",
      "type": "aiops.ibm.com/insight-type/golden-signal",
      "details":

{         "goldenSignal": "error",         "typeGoldenSignal": "effect"       }

    },
    {
      "id": "6d3f5066-72fb-41f3-97c1-4df6edd77e01",
      "type": "aiops.ibm.com/insight-type/probable-cause",
      "source": "aiops.ibm.com/insight-source/probable-cause",
      "details":

{         "id": "27d8486a-dedc-4057-b6d7-862572746059",         "rank": 1,         "time": 0       }

    },
    {
      "id": "manny3 IBM Example#0fcd0e83-714e-4882-a54a-aed59d32170f",
      "type": "aiops.ibm.com/insight-type/relationship/causal",
      "source": "aiops.ibm.com/insight-source/relationship/causal/custom",
      "details":

{         "policyId": "38cac4d0-804b-11ef-80fc-a972dd876928"       }

    }
  ],
  "lastOccurrenceTime": "2024-10-09T18:37:47.999Z",
  "lastStateChangeTime": "2024-10-09T18:38:02.831Z",
  "links": [


{       "url": "https://www.ibm.com/us-en",       "linkType": "HOT LINK",       "name": "IBM"     }

  ],
  "owner": "-",
  "relatedContextualStoryIds": [],
  "relatedStoryIds": [
    "27d8486a-dedc-4057-b6d7-862572746059"
  ],
  "resource":

{     "name": "IBM Example",     "type": "IBM Solution",     "IBM_ResourceSubDomain": "CustomPropValue2"    }

,
  "sender":

{     "type": "Example",     "IBM_SenderSubDomain": "CustomPropValue1",     "name": "SentbyIBM"    }

,
  "severity": 6,
  "signature": "{IBM_ResourceSubDomain=CustomPropValue2, name=manny3 IBM Example, type=IBM Solution}IBM example
documentation problem alert",
```

```
  "state": "open",
  "summary": "IBM example documentation problem alert",
  "suppressed": false,
  "team": "-",
  "tenantId": "cfd95b7e-3bc7-4006-a4a8-a73a79c71255",
  "type":

{     "IBM_TypeStatus": "CustomPropValue3",     "classification": "IBM example documentation problem alert",
"eventType": "problem"    }

}
```

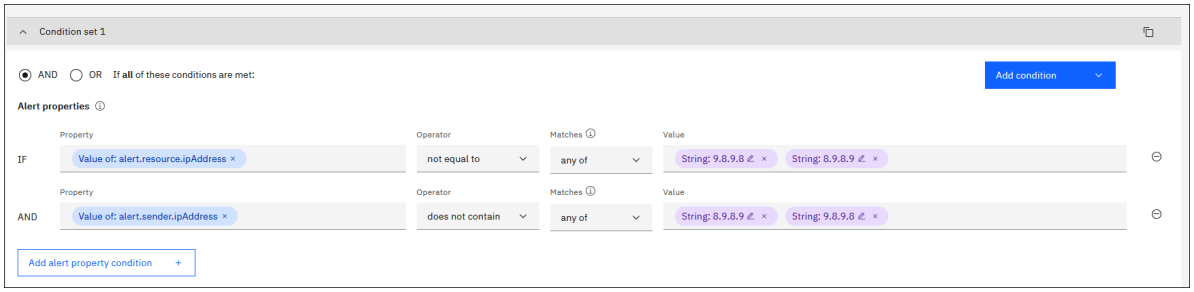
Best practices for condition sets

For certain scenarios, some best practices are recommended for the **Operator** and **Matches** fields in a policy condition set.

Scenario: use of "any of" in negative operator cases

- In this example, "not equal to" and "does not contain" a set of IP addresses evaluates true for "any of" if either value matches the alert property value. Think of "any of" for these negative operator conditions as reading "any one of". Where any value matches a true will be "or'd" with false, resulting in true. For this reason, it is recommended to use **"all of"** instead which then operates as a definitive exclusion list.

  For the following example, the policy evaluates true for an alert with `resource.ipAddress` of `9.8.9.8` or `8.9.8.9`.



Example: use of any of in negative operator cases

Scenario: looking for unassigned owners

- If you need a policy condition to search for unassigned owners or teams, the **is empty** operator does not work with the `alert.owner` or `alert.team` properties. Use the **equal to** operator instead, with **only** and enter a minus sign (-) value.



Example: search for unassigned owners

Refreshing the policies table

Refresh the policies table to view all the latest policies at the current point in time.

If any filters are applied to the table, click **Clear filters** first to view a complete list of policies.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click Refresh to refresh the table at anytime. A refreshed policy count is displayed above the table beside **Policies**.
4. You can also click start to refresh the table with with auto-refresh every 90 seconds. Click stop to turn off auto-refresh.

Note: Applying filters automatically refreshes the table.

Managing multiple policies

You can multi-select policies and apply actions to all of your selections at one time.

Note: When a multi-select of policies includes different policy types, not all actions can be applied. For example, preset policies cannot be deleted.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.

2. In the main navigation menu, click **Operate** > **Automations**.
3. Select multiple policies from the table by using the checkboxes at the beginning of the table rows.

   Tip: You can select all of the policies in the table by clicking the checkbox in the table header beside **Policy name**.
4. A banner above the table displays the number of policies that are selected, together with a menu of options. The following actions can be applied to multiple policies at the same time:

   - **Enable**
   - **Disable**
   - **Archive** (Seasonal and Temporal policies only)
   - **Delete**

Renaming policies

A global unique identifier string is used to identify analytics policies. You can rename policies to give them more meaningful names. Preset policy names cannot be customized.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Point the cursor over the **Policy name** that you want to change and click Edit.
4. Enter a new policy name in the field provided.
5. Click the check mark ✓ to save the policy name or click cancel ✕ to discard your changes.

Note: When editing is enabled for the policy, the name can also be updated in the policy editor.

Archiving policies

You can archive alert seasonality and temporal grouping policies that are not in use. Archived policies don't act on incoming alerts.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Filter the table to display **Temporal** and/or **Seasonal** policies, or search for a policy by entering a search term in the field provided.
4. In the table row of the policy that you want to archive, click the options menu (three vertical dots) at the end of a table row and select **Archive policy**.
5. To confirm the archive, click **Archive policy**.

Archived policies can be viewed and managed in Draft and archived policies.

Migrating Netcool correlations to Cloud Pak for AIOps

Many deployments of Cloud Pak for AIOps are done where an existing Netcool environment already exists. Part of the migration over to Cloud Pak for AIOps is the deactivation of event correlation in the Netcool layer, and its reimplementation in Cloud Pak for AIOps. This is necessary because how groupings are created is different between Netcool and Cloud Pak for AIOps and the representation and linkages are not compatible:

- Netcool uses either real or synthetic parent events whose children are linked via a key linkage field - often `@ParentIdentifier`.
- Cloud Pak for AIOps stores grouping information as metadata within each correlated event - parent events are only virtual and are rendered in the Alert Viewer.

The major advantage of Cloud Pak for AIOps over Netcool is that events can be a member of any number of groups, either of the same or different kinds of groups. Whereas Netcool requires a linkage between a parent event and a child event via a static field, Cloud Pak for AIOps stores this metadata within the alert payload. There is no programmatic limit on how much metadata can be stored in an Cloud Pak for AIOps alert, which results in a much more flexible grouping model.

There are three main grouping capabilities that are provided in both Netcool Operations Insight and Cloud Pak for AIOps. The following paragraphs outline how these groupings can be migrated from the Netcool layer to the Cloud Pak for AIOps layer:

- **Temporal**

Temporal groupings that are found in either the Netcool/Impact based Event Analytics or the newer Cloud Native Event Analytics (CNEA) engine are not transferable to Cloud Pak for AIOps. Cloud Pak for AIOps must ingest event data from Netcool (and other sources) and retrain on the data internally. Any temporal grouping in Netcool Operations Insight should be deactivated before enabling temporal grouping in Cloud Pak for AIOps.

- **Topology-based**

  Netcool Operations Insight includes the same topology capability as Cloud Pak for AIOps and also supports topology-based event correlation. It works the same in both environments and must be set up in Cloud Pak for AIOps to match the configuration in Netcool Operations Insight. Any topology-based grouping in Netcool Operations Insight should be deactivated before enabling topology-based grouping in Cloud Pak for AIOps.

- **Scope-based**

  Netcool Operations Insight includes a similar scope-based grouping capability as Cloud Pak for AIOps. By default, the Netcool Connector maps `@ScopeID` in Netcool Operations Insight to `resource.scopeId` in Cloud Pak for AIOps. The default Cloud Pak for AIOps policy called Default Netcool scope-based grouping - with a rolling window of 15 minutes will automatically detect any inbound events with the `resource.scopeId` attribute set and perform scope-based grouping on it. As the name suggests, this is done with a rolling 15-minute window. This policy provides a convenient like-for-like function and makes migration from Netcool to Cloud Pak for AIOps relatively easy in this respect. You can also disable this default policy and create your own, if you have more specific or complex requirements - for example, you require various time windows for different correlation scenarios.

  Note: Netcool/OMNIbus in the absence of Netcool Operations Insight only has scope-based event correlation, if enabled. For more information, see https://www.ibm.com/docs/en/netcoolomnibus/8.1?topic=netcoolomnibus-enabling-scope-based-event-grouping.

Custom correlations in Netcool

Many implementations of Netcool include custom correlations where one or more child events are linked to a parent event. While the grouping engine itself is eventually deactivated at the Netcool layer, in a combined Netcool and Cloud Pak for AIOps deployment, the correlation logic can remain in-place. With the grouping instead done at the Cloud Pak for AIOps layer. A method for achieving this is outlined here.

Procedure

1. Create a custom correlation field.

   Create a custom field to hold the key data value that will tie each event set together. Use the `nco_sql` utility to create the field:

   ```
   1> ALTER TABLE alerts.status ADD COLUMN CustomCorrelation1 VARCHAR(64);
   2> go
   ```

2. Modify the correlation logic.

   The next step is to modify the automation that is correlating the events together, whether it's a Netcool/OMNIbus ObjectServer trigger or a Netcool/Impact policy. The logic of the automation remains the same as before, the only change being how the events are tied together. In the new correlation design, the automation must compute a key data value that is set in the newly created custom correlation field - in this case `@CustomCorrelation1`. The value must be set to the same value across all members of the correlation set, and be sufficiently unique so that incorrect items are not inadvertently correlated together.

3. Map the custom correlation field.

   Map the custom correlation field to Cloud Pak for AIOps. This is done in Cloud Pak for AIOps by editing the mapping of your Netcool integration instance and adding in a details attribute. The following extract from a Netcool integration mapping shows the insertion of the details section into the mapping, immediately under the **expirySeconds** attribute:

   ```
   "eventCount": alert.@Tally,
   "signature": alert.@Identifier,
   "firstOccurrenceTime": alert.@FirstOccurrence,
   "lastOccurrenceTime": alert.@LastOccurrence,
   "severity": alert.@Severity < 0 ? 1 : alert.@Severity = 0 ? 2 : alert.@Severity < 6 ? alert.@Severity + 1 :
   alert.@Severity >= 6 ? 6,
   "state": alert.@Severity = 0 ? "clear" : "open",
   "acknowledged": alert.@Acknowledged = 1 ? true : false,
   "expirySeconds": alert.@ExpireTime = 0 ? undefined : alert.@ExpireTime,
   "details": {
       "customCorrelation1": alert.@CustomCorrelation1 = "" ? undefined : alert.@CustomCorrelation1
   },
   ```

   Click **Save** to save your new integration mapping. Note, this results in the Netcool Connector pod restarting. The new mapping will only take effect on events that are newly inserted into the Netcool/OMNIbus ObjectServer. You can add as many custom correlation fields (comma separated) as required to this details section.

4. Create a Cloud Pak for AIOps correlation policy.

   The final step is to create a scope-based grouping automation policy in Cloud Pak for AIOps to group the incoming event stream by your newly created attribute.

   1. Give your policy a meaningful name.
   2. Set the priority.
   3. For the policy trigger, select **Before an alert is created**.
   4. Add a condition that the policy fires when your custom attribute is "**not empty**".
   5. Under **Create a scope-based grouping**, enter your custom attribute.

6. Specify the required time window.
7. Specify whether the time window is rolling or fixed.

   Example scope-based policy

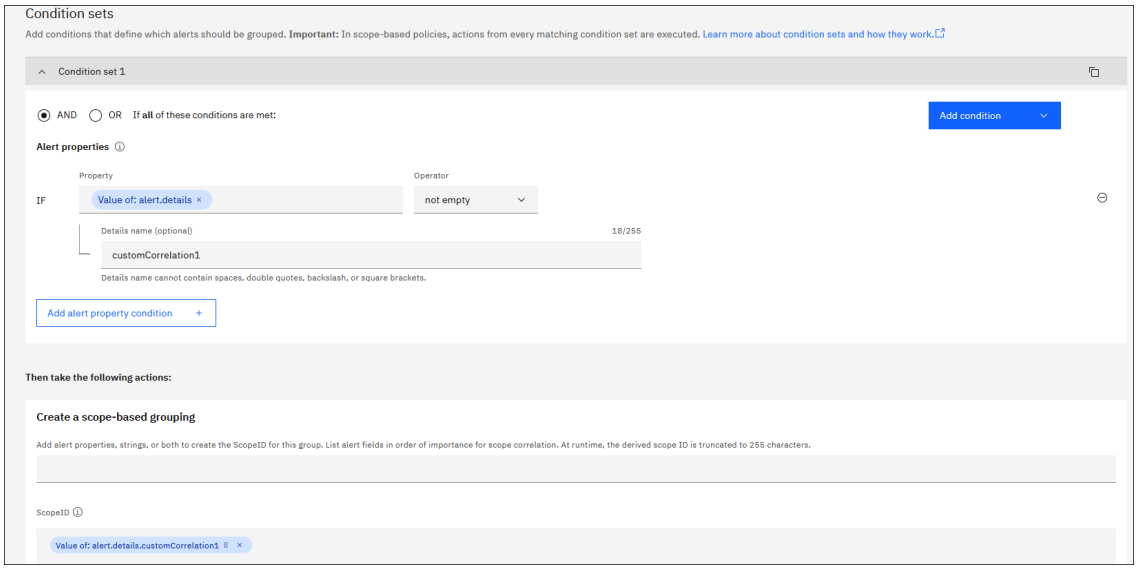


Figure. Scope-based policy

8. Click **Create Policy**.

You should now start to see your custom correlation working in Cloud Pak for AIOps. Remember, this policy will only work for new events, not existing events.

One of the major advantages in Cloud Pak for AIOps over Netcool is that you can set up as many scope-based grouping automations as required. Therefore, you can use this process to migrate all of your custom correlations in Netcool over to Cloud Pak for AIOps.

About runbooks

Use Runbook Automation to build and execute runbooks that can help IT staff to solve common operational problems. Runbook Automation can automate procedures that do not require human interaction, thereby increasing the efficiency of IT operations processes. Operators can spend more time innovating and are freed from performing time-consuming manual tasks.

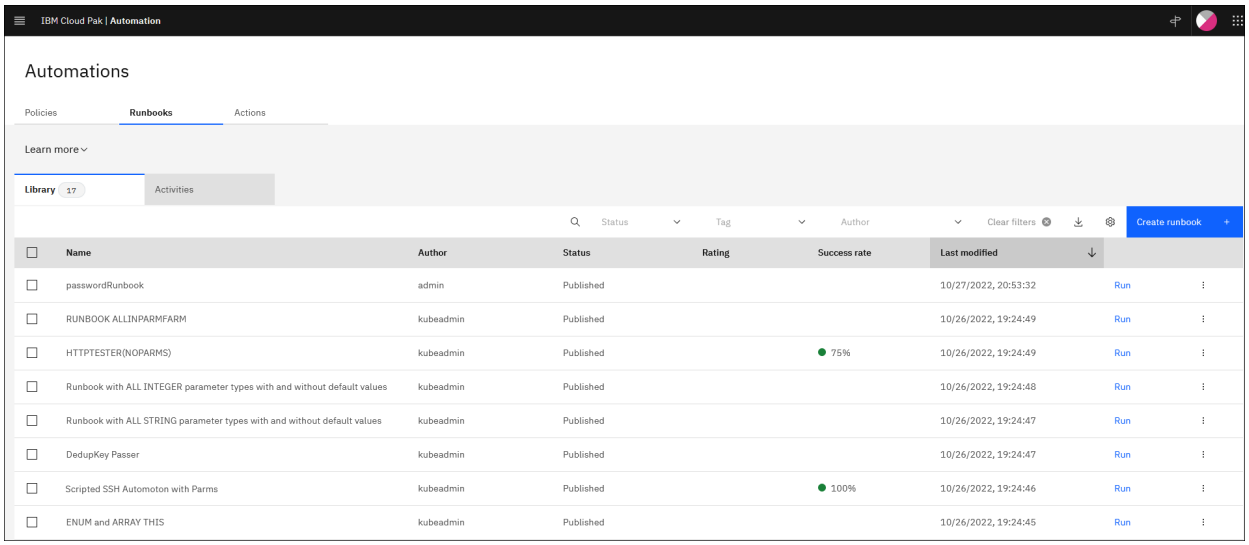


Figure. Runbooks Library UI

- What problems does Runbook Automation address?

  IT systems are growing. The number of events are increasing, and the pressure to move from finding problems to fixing them is increasing. Runbook Automation supports operational and expert teams in developing consistent and reliable procedures for daily operational tasks.

- How can I simplify daily operational tasks?

  Using Runbook Automation, you can record standard manual activities, so that they are run consistently across the organization. The next step is to replace manual steps with script-based tasks.

- What is a runbook?

  A runbook is a controlled set of automated and manual steps that support system and network operational processes. A runbook orchestrates all types of infrastructure elements, such as applications, network components, or servers.

Runbook Automation helps users to define, build, orchestrate, and manage runbooks. Metrics about runbooks are also provided. A knowledge base is built up over time through collaboration between Subject Matter Experts (SMEs) and operational staff.

Runbook types

Transfer your procedure based documentation to manual runbooks. Then add actions and continue with semi-automated runbooks or fully automated runbooks.

Runbook Automation supports three different types of runbooks:

- Manual runbooks

  Steps describe the exact procedure that an operator must follow. The operator uses standard tools, which can be accessed from their working environment.

- Semi-automated runbooks

  Each step describes exactly what an operator must do. Additionally, the operator can run an automated task on a target system.

- Fully automated runbooks

  The runbook runs automatically without operator interaction.

Actions

Actions are a task that run autonomously on a target system. For example, a shell script that identifies many files in a file system.

Runbook Automation supports the following technologies to connect to a backend system:

- Script

  Actions of this type contain a complete script. When a SSH Action runs, the content is sent to the target system and executed. Scripts can be of type Bash or PowerShell, depending on the target system.

- HTTP

  Actions of this type define the specific properties to make a secure HTTP call.

- Ansible

  Actions of this type reference existing Job Templates or Workflow Job Templates on an Ansible Automation Platform Controller server and run them with the defined variables.

For more information, see Actions.

Runbooks Library

View the available runbooks, find matching runbooks for the alerts that you need to resolve, and run the runbooks. You can also review the runbooks that you have used to date. The library also provides ready-to-go sample runbooks. Load the examples to explore what you can do with runbooks. Document your existing operational procedures using the runbook editor. Share the knowledge and experience that you documented in a runbook with your team members. The runbook editor provides editing capabilities as well as runbook elements that help to document operational procedures. Runbook elements are designed to avoid mistakes.

For more information about the Runbooks Library, see the following topics:

- Run your first runbook
- Load example runbooks
- Create a runbook
- Edit a runbook
- Configuring runbook creation
- Runbook versions
- Save and publish a runbook
- Filtering runbooks
- Assigning runbooks to groups
- Export and import runbooks
- Delete a runbook

Run your first runbook

If you are using Runbook Automation for the first time, it is recommended to load example runbooks and then edit and publish a runbook. For more information, see Load and reset example runbooks and Edit a runbook.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Runbooks**.

   The **Library** tab is displayed by default.
4. Select a runbook and click the menu overflow icon (the three dots at the end of the row) and select **Preview** to find out what a runbook consists of:

   - **Procedure**:

     The Procedure section displays the runbook steps. For more information about the runbook elements, see Edit a runbook.

   - **Information**:

     The information tab provides descriptive details of the runbook and contains the following fields:

     - Description: shows a description of the runbook.
     - Runbook ID: ID of the runbook which can be used to identify the runbook using the Runbook Automation HTTP REST Interface.
     - Created by: the user who created the runbook.
     - Last modified at: the date and time the runbook was last modified.
     - Last modified by: the user who last modified the runbook.
     - Type: indicates the runbook type, that is manual, semi-automated, or fully automated.
     - Tags: categorization tags associated with the runbook.
   - **Versions**:

     Shows all draft, published, and archived versions of the runbook. You can also see the user that created each version and the creation date.
5. Back on the main **Runbooks** page, select one of the runbooks and click **Run**.
6. The **Run runbook** page opens, which consists of a procedure description on the left side of the window and a **Details** and **Information** section on the right. **Details** contains the following collapsible sections:

   - **Parameters** – displays a list of the parameters used.
   - **Actions** – displays a list of the actions used.
7. In the **parameters** section you are requested to enter parameter values. Click the information icon to see a description and a reference to where this parameter is used within the runbook. Enter the parameter values, for example host1 for the parameter **`HOSTNAME`** and click **Start runbook**.

   Note:

   - In some cases, you might not know the parameter value when starting the Runbook. For example, the parameter could be determined by a runbook step during the runbook execution. In this scenario, as soon as the parameter value is known, go back to the parameter section and enter the parameter value. To apply the value click **Update parameters**. The **Update parameters** button is only displayed if the parameter value has been entered.
   - In any step of a runbook, you can select the parameter section of the runbook in the **Details** on the right of the window. You might need to first expand the section if it is collapsed. The parameter can then be changed to a new value. To apply the new value click **Update parameters** (the **Update parameters** button is only displayed if the parameter value has been changed). As soon as a parameter has been used by a Runbook step, it cannot be changed.
8. Click **Start runbook**.
9. Follow the instructions in the runbook and proceed to the next step by clicking **Next step**.
10. If the step contains a command, use the **Copy** button within a command to copy the command to your clipboard. Open a command-line window and paste the command.
11. If the step contains an action, click on the action and select **Run** to run the action. **Note**: If running the action is not appropriate in the particular situation, then you can also choose to proceed without selecting **Run**. If you select **Run**, the action output is displayed in a text box. If an action produces JSON output, the JSON document is automatically formatted for better readability. Select **More info** to see description, parameter values, and type information.

    If the type of action is "Client-side", then the label of the **Run** button is replaced with **Launch**. Clicking this button launches the URL from the Client-side action in a new browser tab. You can review the resulting URL in the action output area and launch it again by clicking the displayed URL hyperlink.
12. You can pause or cancel the runbook by clicking **Pause** or **Cancel**. Pause temporarily suspends the execution of the runbook. You can pick up later from where you left off. You can use **Pause** to handle long running actions, for example actions that take longer than 10 minutes. The actions that are in progress continue while you pause the runbook. If you cancel the runbook you are prompted to select a reason. This feedback is helpful to the author of the runbook.
13. After you completed all steps, click **Complete** at the end of the runbook. Provide a rating, a comment (both optional), and a final assessment of the runbook. You are taken back to the **Library** page.
14. The **Activities** tab provides the **Runbook activities** table. Here you can find all the runbooks that were executed with information such as Status, Start time, Version, Comments, and if the runbook is automated. Filters such as user, runbook status, runbook type and runbook name can be applied. For more information, see Activities.

Load and reset example runbooks

Load ready-to-go sample runbooks to explore what you can do with runbooks.

Warning: If you want to edit a sample runbook and adjust it to your requirements, or if you are considering publishing a modified sample runbook to use in production, it is recommended to first create a copy of the sample runbooks and then edit and publish the copy. Similarly, if the sample runbook uses an automated

step, then create a copy of the automation action and use that copy within your new runbook. This is to avoid losing your work (or even disrupting production) if somebody deletes or reloads the sample runbooks.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Integrations**.
3. Ensure that a connection of type "SSH" has been configured. If not, you can configure it now (with default values).
4. From the main navigation menu, click **Operate** > **Automations**.
5. Click **Runbooks**.

   The **Library** tab is displayed by default.

Load sample runbooks

Load all pre-defined examples. You can also edit the example runbooks and create new versions.

1. Click **Load sample runbooks** on the upper right of the window (beside the **Create runbook** button) and select **Load sample runbooks**.
2. The sample runbooks are shown in the **Library** page. The runbooks will show up for Operators if they are published. The samples are marked by the text "Example:".
3. Click **Preview** to view the runbook and see what it does.
4. Click **Run** to run your first runbook. For more information about how to run a runbook, see Run your first runbook.

Sample runbooks with special administrative use cases

Some of the following sample runbooks go beyond just illustrating the features of runbooks:

- **Example: Convert existing operational procedures into runbooks**: This runbook explains how to convert your existing operational procedures into runbooks. It covers file based procedures and web/wiki based procedures.
- **Example: Import runbooks from an existing runbook deployment**: Provides instructions on how to export runbooks from one deployment and import them to another deployment. It uses the RBA API for both operations.
- **Example: Reduce amount of data in runbook data stores** and its automated variant **Example: Automatically reduce amount of data in runbook data stores**: Offload runbook execution data into a secondary data store for auditing or long-term storage purposes. This will delete that data from the production data store to ensure less disc space usage and faster response times on the active system.
- **Example: Add new SSH key to target systems** and **Example: Remove old SSH key from target systems**: These sample runbooks assist you during deployment migration. As part of the migration, you must install the SSH key of the new deployment to all target systems that were previously reached from the old deployment, using that old deployment's SSH key. This is a prerequisite step when you plan to migrate your runbooks and script actions from the old deployment to the new deployment. For more details, see Installing new SSH key during migration.
- **Example: Using the alerts API (basic)** and **Example: Using the alerts API (advanced)**: These sample runbooks demonstrate how you can create actions that use the Issue Resolution API to get and modify alerts. You can reference these actions within new custom menu items in the Alert Viewer as described in Working with alerts and menu customization to extend the built-in capabilities for managing alerts.

  As the Issue Resolution API is one of the supported IBM Cloud Pak® for AIOps APIs for HTTP actions, authentication is handled automatically provided that they are being run manually. For more information about automatic authentication, see the API endpoint section in Creating HTTP actions.
- **Example: Expand Kafka broker storage (troubleshoot Cloud Pak for AIOps cluster)**: This runbook assists in analyzing the Kafka broker's health, and in particular in getting the current size of storage that is available to the Kafka broker. If needed it assists in expanding the Kafka broker related storage. See also the related troubleshooting documentation, Kafka broker pods are unhealthy because Kafka PVCs are full.

Reset sample runbooks

Follow these steps if you edited the examples and would like to reload the original version, or if you have upgraded to a new version and want to get the latest versions of the examples:

- Open the runbook **Library** page.
- Open the **Reload sample runbooks** menu.
- Select **Delete sample runbooks** and wait for the sample runbooks to be deleted.
- Open the Load sample runbooks menu.
- Select the **Load sample runbooks** menu item and wait for the sample runbooks to be loaded.

Convert a sample runbook to a regular runbook

To save a sample runbook as a regular runbook:

1. On the runbook that you want to convert, click the menu overflow icon (the three dots at the end of the row) and select **Copy runbook**.
2. Change the runbook name. You can optionally update the description.
3. Click **Save**.

After saving, the runbook will no longer be tagged as an example.

4. If the sample runbook references an action, then you need to create a "deep copy" manually:
   - Navigate to the **Actions** table and select the referenced sample action.
   - Copy the sample action and modify its name so you can identify the copy.
   - Edit the copy of the sample runbook, and replace the reference to the original sample action with a reference to your new copy of the sample action.

Delete sample runbooks

1. Click **Load sample runbooks** and select **Delete sample runbooks**.
2. Confirm that you want to delete the sample runbooks from the **Library** page.

Creating a runbook

Document your existing operational procedures using the runbook editor. Share the knowledge and experience that you documented in a runbook with your team members.

Runbooks are created for one of the following reasons:

- Operators document their daily tasks in documents. Runbooks allow you to document and store these procedures in an ordered manner. When a runbook is published the knowledge can be shared between all operators.
- As a special case, create reusable runbook assets to combine those sequences of steps that otherwise would have to be repeated over and over again, and add them to your runbook as nested runbooks.
- A new set of events has occurred and a couple of skilled operators work out a new procedure to solve the issue that caused these events.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Runbooks**.

   The **Library** tab is displayed by default.
4. Click **Create runbook**. The runbook editor opens. For more information about how to use the editor, see Edit a runbook.
5. Provide data to describe the runbook. Make sure that you use a **Name** for the runbook that describes what problem this runbook solves.

   Restriction: Avoid using commas in runbook names. Runbook names containing commas cannot be filtered in the **Library** or on the **Activities** page.
6. (Optional): Provide a **Description**. The description contains information about specific situations that this runbook solves. Operators use the description to decide if a runbook meets their requirements.
7. (Optional): Add **Tags**. These help to filter a large amount of runbooks, as well as group runbooks with the same content.
8. Start adding steps to the runbook. Click **Add manual step**, **Add nested runbook step**, or **Add automatic step**.
9. Click inside the editor and describe the first step of your procedure. Use **commands**, **parameters**, **actions**, or **GOTO** elements to distinguish between different types of elements within a runbook. Remember to always create a new step if you start to describe a new task that the operator can run.
10. When all steps have been documented, click **Save draft** > **Save draft & close**.
11. Test the runbook. Search for your runbook on the **Library** page. Click **Run** to test the execution of the new runbook. Modify the runbook if necessary.
12. Publish your runbook.
    1. Click **Edit** to open the runbook editor.
    2. Click **Publish**. The runbook is now available to all operators.

Note: Runbooks are limited to a size of 1 MB. This relates to runbook metadata and its content (including embedded images). Attempts to create or edit runbooks in a way that exceeds this limit will result in an error.

Editing a runbook

The runbook editor provides editing capabilities as well as runbook elements that help to document operational procedures. Runbook elements are designed to avoid mistakes.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.

3. Click **Runbooks**.

   The **Library** tab is displayed by default.

4. Click **Create runbook** to create a new runbook, or scroll through the list of runbooks and click the menu overflow icon and select **Edit** to open an existing runbook in the editor.

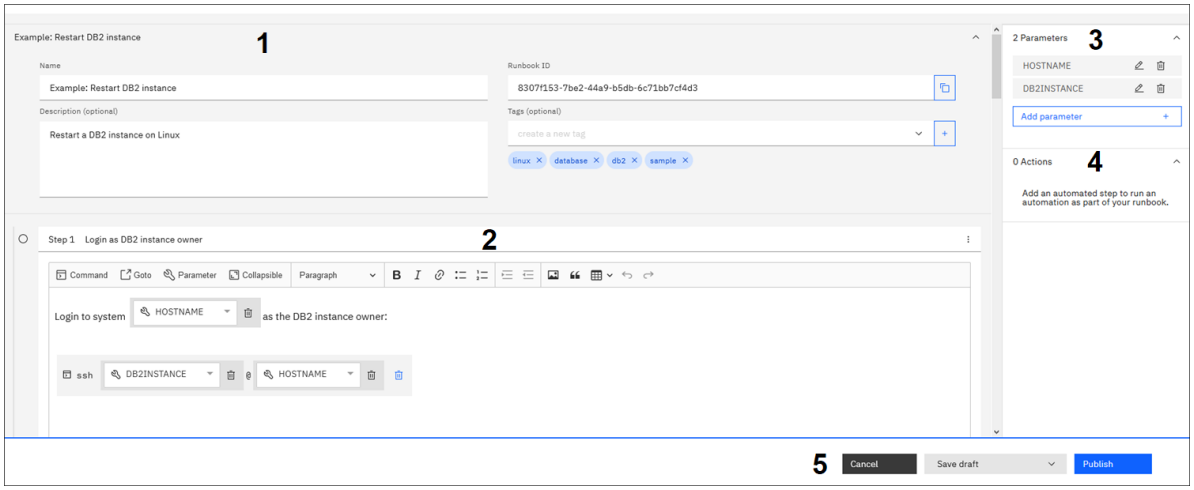


Figure. Runbook editor

Region 1 **Runbook details:** Enter descriptive data about the runbook.

- Name: the name identifies the runbook. Try to be as precise as possible and describe the problem that this runbook solves.
- Description: describe the problems that this runbook solves in more detail and sketch out the solution.
- Tags: provide tags to filter runbooks. You can select any existing tags from the drop-down list and click the plus icon "**+**" to add them. To create a new tag, type the text into the field, select the text string displayed, and click the plus icon "**+**".
- Runbook ID: this field is generated by Runbook Automation. The Runbook ID can be used to identify the runbook using the Runbook Automation HTTP REST Interface. Use the copy button to copy the ID to the clipboard.

Region 2 **Editor:** Provides text edit actions and runbook elements to describe operational tasks. Each step has its own section with an optional title. Manual steps cannot contain automation elements. Automated steps must be added for actions.

Region 3 **Parameters:** Create parameters that are used in this runbook.

Region 4 **Actions:** Add pre-defined actions to the runbook. Action steps only contain an action, no text.

Region 5 **Save and Publish** the runbook.

For more information about editing a runbook, see the following topics:

- Runbook elements and editing actions
- Adding actions
- Adding parameters
- Parameter types
- Adding a GOTO element

Runbook elements and editing actions

Use the editor to describe your daily tasks step by step. The editor provides editing actions and runbook specific elements to help you write the instructions.

Click **Add manual step** and use the following runbook elements and text editing actions to edit and format your operator instructions.

Icon	Name	Editing action
Command	Add command	Click the editor canvas. Select **Add command** and enter your command. Click **Enter** to quit the command and continue in the next row. Another way to enter a command is to mark the command and click **Add command**. As the command is still selected, you have to click on the canvas to deselect the command. Click **Enter** to quit the command and continue in the next row. **Keyboard shortcut:** Use the **tab key** to move the focus on the canvas. Press Ctrl + 4 and begin typing your command. **Remove command:** You can remove the command by placing the cursor in the command and clicking **Remove**.
Goto	Add GOTO	Click the editor canvas. Select **Add GOTO**. A goto element is added with the label **END**. Click the GOTO element and select the step that you want to jump to. You can select any subsequent step. **Keyboard shortcut:** Use the **tab key** to move the focus on the canvas. Press Ctrl + 5. The GOTO element is added. **Remove GOTO:** Select the GOTO element and click **Remove**. For more information about how to use the GOTO element, see Adding a GOTO element.

Icon	Name	Editing action
Parameter	Add Parameter	To add a parameter, go to the **Parameter** palette and drag the parameter that you require to the canvas. Or mark the text which should be a parameter name and click **Add parameter**. **Keyboard shortcut:** Use the **tab key** to move the focus on the canvas. Press Ctrl + 8. A parameter menu is added. Click the arrow and select the required parameter. **Remove parameter:** Select the parameter and click **Remove**. For more information about creating parameters, see Adding parameters.
Collapsible	Add collapsible details	To add collapsible details, click the icon and enter a title for the collapsible section. Then enter the information that you want to collapse and expand in a section. **Keyboard shortcut:** None. **Remove collapsible details:** Use the delete or backspace key to delete collapsible details.
Paragraph	Paragraph or Heading	Switch between paragraph style or select from the available heading styles.
B	Bold	Select a text and make it bold by selecting **b**.
I	Italic	Select a text and make it italic by selecting **i**.
	Insert Link	If you want to insert a link, click **Insert Link** and enter in the URL field the link. Do not forget to provide the link text and decide whether you want this link to be opened in another tab.
	Bulleted list	Select text that is spread over a couple of rows and click bulleted list. All lines are indented and bulleted.
	Numbered list	Select text that is spread over a couple of rows and click numbered list. All lines are indented and numbered.
	Increase indent	Select text that is spread over a couple of rows and click increase indent. The whole text section is moved to the right.
	Decrease indent	Select text that is spread over a couple of rows and click decrease indent. The whole text section is moved to the left.
	Insert image	If you want to add a screen capture for example, select **Insert Image**.
	Block quote	If you want to add a quotation that is set off from the main text, select **Block quote**.
	Insert table	If you want to insert a table, click **Insert Table**. Enter the number of rows and columns that you need and specify the column with and width unit.
	Undo	You can undo your last 20 editing actions.
	Redo	You can redo your last 20 editing actions.

Adding actions

Actions are used to run several steps automatically. Use the **Actions** page to create actions.

The job of an operator can involve repetitive tasks, such as, *"Logon to Db2"*, or running the command **`start server dagobert45`** to start a server. Actions are scripts or Ansible templates that document these steps. If you use actions within your runbook, then the operator does not have to manually run these steps. They can run all steps with one click.

For more information about how to create actions, see Creating an action.

The runbook editor displays any created actions in the **Actions** pane.

Manual steps cannot contain action elements. Automated steps must be added for actions.

It is common practice for actions to contain parameters. When adding actions to a runbook, you must decide how the values of the action parameters are filled by creating a parameter mapping. Select from the following options:

- **Use a runbook parameter:** a runbook parameter, which can be filled manually by an operator running the runbook, is used to fill the specific action parameter. When adding an action to a runbook, you can select existing runbook parameters or directly create a new runbook parameter.
- **Define a fixed value:** the action added to the step of a runbook will always be launched with a fixed value for the parameter. (If the same action is used in a different runbook – another fixed value might be defined there.)
- **Use the default from the action:** if the action defines a default value for this parameter, it is possible to use that value.
- **Use the output of a previous action:** the output of an action that was added to a previous step can be used as input for the action of the current step.
- **Use the logged in user:** the parameter value is filled with the username of the user, who is logged in to Runbook action at the time when the runbook is run.

  Note: If the username contains the @ symbol, for example because the username is an email address, the @ symbol and all characters that follow it are removed from the username.

Conditional branching is supported for runbooks that contain automated steps. You can configure which step to run next if running the current action ends with status ***successful***, and another step to run next if the current action ends with another status. Depending on the remediation procedures that are available for the current step, this allows you to automatically perform error handling such as the following:

- Analysis and remediation of the error
- Cleanup of leftovers from the error
- Falling back to the previous state
- Collecting diagnostic data

If a runbook runs fully automated and has conditional branching that is enabled for some steps, the resulting runbook activity reports as ***successful*** even if one or multiple steps have failed. Only if the last step fails, or if another step fails that has no conditional branching that is configured, will the overall runbook activity also be marked as not successful. For more information about the available exit states of actions, see How is the success rate of an action calculated?.

Conditional branches are primarily intended to take effect when running the runbook in fully automated mode. When running the same runbook manually, the conditional branching is also observed. The flow of execution proceeds to the appropriate target step based on the result of the automated action. If the automated action has not been run yet, the default is to proceed with the next step.

Procedure

1. In the New (or edit) runbook editor, click **Add automated step**.
2. Search for the action. Click the search icon and type in the search term that describes the action that you need.
3. Choose the action that you want and then click **Select this action**.
4. Select the parameter-mapping type. A window opens with action details and the parameter definitions. In the **Mapping** column, select how action parameters are to be filled with values. You can choose from the following options:
   - **New runbook parameter:** add a parameter. If default values are provided, update the default values as required. Enter a new parameter name and description.
   - **Fixed value:** an entry field is provided to enter the parameter value.
   - **(Existing) runbook parameter:** select the parameter from the list of existing parameters.
   - **Use default:** use the default value set by the action.
   - **Action output:** use the action output from a previous runbook step. The output value of the action will be used as the parameter value for the current action. Note, if the action in the previous runbook step is a script action or a HTTP action then you can filter the output of that action as described in Creating SSH actions or Creating HTTP actions, respectively. This option is only available if the runbook contains a previous step with an action.
   - **Use logged in user:** select this option to fill the parameter value with the username of the user who is logged in to Runbook action at the time when the runbook is run.

     Note: If the username contains the @ symbol, for example because the username is an email address, the @ symbol and all characters that follow it will be removed from the username.
5. Each automated step, except for the last step, provides the option to configure conditional branching. With conditional branching enabled, you can select which step to run as the next step depending on the result of the current action (***successful*** or ***not successful***), or if the runbook should stop (that is, go to the end).

Next steps

- **Edit parameter configuration:** if you want to edit the parameter configuration of your action, select the action and click **Change parameter mappings**. Change your settings and click **Save**.
- **Action with errors:** if the action is decorated with an error symbol, then the parameter settings are not correct. Click **Change parameter mappings** to correct the parameter settings.
- **Delete an action:** if you want to remove the action, click **Remove selection**.

Adding nested runbooks

Nested runbooks are used to collect sets of steps that are commonly used, so they can be reused by other runbooks.

About this task

Create reusable runbook assets to combine those steps that otherwise would have to be repeated over and over again in various runbooks. For example, if you have a 3-step login procedure, you can describe this procedure once in a runbook, and then reuse this runbook by adding it as a nested runbook step within all other runbooks that require the 3-step login procedure. When the 3-step login procedure needs to be changed to become a 5-step login procedure, you only need to change the nested runbook. Once the nested runbook has been published, all runbooks that refer to this nested runbook inherit the updated 5-step procedure.

Once the reusable runbook has been created and published, it is ready to be referenced by other runbooks as a nested runbook.

Runbooks that include nested runbooks are always considered as "manual" runbooks, even if all nested runbooks contain automated steps only, and the runbook itself does not contain manual steps.

Nested runbooks allow a maximum nesting depth of 10 runbooks, and the complete nesting structure cannot contain more than 100 nested runbooks. Attempts to create or change runbooks in ways that exceed these limits will fail.

The statistics for nested runbooks are unchanged when running the parent runbook. Statistics (such as success and failure, but also rating and feedback comment) are always recorded at the level of the runbook that was started, and not propagated to any nested runbooks.

It is common practice for nested runbooks to contain parameters. When adding a nested runbook to a runbook you must decide how the nested runbook's parameter values are filled by creating a parameter mapping. Select from the following options:

- **Use a runbook parameter**: a runbook parameter, which can be pre-filled automatically from an event or edited manually by an operator executing the runbook, is used to fill the parameter of the nested runbook. When adding a nested runbook to a runbook, you can select from existing runbook parameters or directly create a new runbook parameter within the parent runbook.

- **Define a fixed value**: the nested runbook added to the step of a runbook will always be launched with a fixed value for the parameter. (If the same nested runbook is used in a different runbook, another fixed value could be defined there.)
- **Use the default value from the nested runbook**: if the nested runbook defines a default value for this parameter, it is possible to use that value.
- **Use the output of a previous automation**: the output of an automation that was added to a previous step can be used as input for the nested runbook of the current step.
- **Use the logged in user**: the parameter value will be filled with the username of the user who is logged in at the time when the runbook is run.

Note: If the username contains the @ symbol, for example because the username is an email address, the @ symbol and all characters that follow it are removed from the username.

Procedure

1. In the New (or edit) runbook editor, click **Add nested runbook step**.
2. Search for the published runbook that you want to add. Click the search icon and type in the search term that describes the runbook that you require as a nested runbook.
3. Choose the runbook and click **Select this runbook**.
4. If the selected runbook contains parameters, you are now asked to define the parameter mappings. A window is displayed with the list of parameters that are defined in the nested runbook. In the **Mapping** column, select how each of the parameters of the nested runbook is to be filled with its corresponding value. You can choose from the mapping options that are described above.

Next steps

- **Edit parameter configuration**: if you want to edit the parameter mapping, click **Change parameter mappings** Change your settings and click **Apply**.
- **Nested runbook with errors**: an error message is displayed when you open the runbook editor again after the parameter list of the nested runbook has been changed, so the parameter mappings are no longer valid. This can happen when the parameter list of the nested runbook has been changed, and the changed nested runbook has been published again. Click **Change parameter mappings** to adjust the parameter mappings.
- **Delete nested runbook**: if you want to remove the nested runbook, click **Remove selection** .

Adding parameters

Parameters are used as general placeholders for values that are used on a frequent basis. Examples of parameters are **DATABASE_NAME**, **DIRECTORY**, **HOSTNAME**, **URL** and so on.

You can use runbook parameters in the following scenarios:

- Parameters can be used as variables that get substituted by a value in the text of a runbook step, for example:

  ```
  STEP 2
  In this step you determine the longest running process of system $SYSTEM
  ```

- As input to a command defined in a runbook, for example:

  ```
  STEP 5
  Now issue the command: cd $DIRECTORY
  ```

- Parameters of an automation can be filled with the value of a parameter defined for the runbook.
- Actions of type Script have a system parameter `target` which defines the target system where the script is run.
- Actions of type Script have a system parameter `user` that specifies the UNIX username which is used to run the script on the target system. This parameter can be mapped automatically to the user who is currently logged in to Runbook Automation. For more information, see Adding actions.

Operators enter values such as `username`, `password`, `hostname`, `database name` or `URL` on a frequent basis. If you have to deal with many different computer systems, you must remember numerous user names, passwords, host names or database names. In order to avoid typos or having to remember values, you can define parameters. For example, you can define a parameter `DATABASE_NAME` and either provide a default value or enter the value when you run the runbook. Default values are the best fit if you do not want operators to remember and enter the value.

Parameters are local to the runbook. They are not available for other runbooks.

Parameters are used within runbooks and for actions. Within runbooks, you can use parameters either standalone, for example `URL`, or within a command, for example `connect to HOSTNAME`.

Parameters must be filled with values when executing a runbook. This can be done in the following ways:

- When a runbook is launched by clicking the execute button from the Runbook Automation UI, an operator must first enter the parameter values. The operator is prompted with the name and description of all parameters defined for the runbook, as well as an optional default value.
- When a runbook is launched from an event, a Trigger defines which values of the event data are mapped to which parameter of the runbook. Values that are entered in this way cannot be changed.

Runbook parameter values can be changed during the execution of a runbook. In any step of a runbook, you can select the parameter section of the runbook in the sidebar. You might need to expand the section first, if it is collapsed. The parameter can then be changed to a new value. To apply the new value click **Update** (the Update button is only displayed if the parameter value has been changed). Steps of the runbook that are not yet executed will reflect the new value.

Runbook parameters can be used to define parameter values for actions within the same runbook. Parameter values for actions can be filled using the result of a previously executed automation.

In this way, the parameter Action output of a previous automation **`Find large file systems`** is an automatic parameter available for actions in the current step. It is possible to use the output of any automation that ran before in this runbook.

Procedure

1. Create a parameter. Click **Add Parameter** Parameter. A dialog box opens where you can enter the parameter name, the description, and the default value. If you select **Optional** you can start the runbook execution without defining a value, and set the parameter value during runbook execution. This setting is useful if the parameter value is not available at the beginning of the execution, but during execution.

   When you enter the runbook parameter name you can use alphanumeric characters, characters from national languages, and some special characters. But do not use the special characters ampersand (&), backslash (\), space ( ), greater than (>) or less than (<) when you define runbook parameter names.

   As a result, a parameter of the runbook is created and can be used as a variable in the text of a step, as a variable value of a command, or as input of an automation added to a runbook step. For more information about how to use parameters, see Runbook elements and editing actions.

2. Edit a parameter. Click **Edit** next to the parameter that you want to edit.

3. To edit a parameter used by an automation, click **Edit** for the automation you have added to a runbook step.

Parameter types

Runbook Automation supports multiple parameter types. You can select parameter type when you create or modify a runbook parameter in the runbook editor, or when you create or modify an action parameter in the edit action dialog.

Depending on the action type, certain parameter limitations might apply. Supported parameter types are string, integer, Boolean, and array.

String

When defining a parameter of type string, an operator running your runbook has a free text input element to enter a parameter value. You can specify the minimum or maximum number of characters that the operator can use. You can also mark a string parameter as a password so that any input that is entered by the operator is not displayed.

Any runbook or action parameter that is created before June 2021 is of type string.

Integer

When defining a parameter of type integer, an operator running your runbook has an integer input element to enter a parameter value.

You can specify the minimum or maximum value for the integer value.

Boolean

When defining a parameter of type Boolean, an operator running your runbook has a toggle element to enter a parameter value.

Array

When defining a parameter of type array, an operator running your runbook has an input element to define a various number of inputs. The array parameter must define what type of inputs are allowed. You can use an array of either strings, integers, or Booleans.

Enumerations

To restrict the input for a specific parameter to a discrete set of values, you can extend the parameters with an optional enumeration list. Enumeration support is available for parameters of the following types:

- string
- integer
- arrays with string or integer as their item type

You can define any number of allowed discrete values for your enumeration. The user will then be given the choice of selecting any one of them in the Runbook Automation UI, or multiples of those in the case of an array parameter. If a user tries to send a value which is not part of the enumeration, for example through an API call, the action is rejected with an error message.

Additionally, you can assign labels for any or all of the values of an enumeration for integer or string parameters. Labels are optional. If a label is present, the label is displayed in the Runbook Automation UI together with the value.

Example: If you are writing an application which accepts the days of the week as numeric values from 0-6, you can define the following enumeration list with labels:

Value	Label
0	Monday
1	Tuesday
2	Wednesday
3	Thursday
4	Friday
5	Saturday
6	Sunday

This way, the user can only choose a valid value (0-6), but is also shown the appropriate meaning of the value, as presented by the label.

Action parameter support

The following table contains details about the parameter type to use for each type of automation.

Action type	String	Integer	Boolean	Array
Script/Bash	String value is exported as a variable.	Integer value is exported as a variable.	Boolean is exported as variable with either value "true" or value "false".	Not supported
Script/Powershell	String value is available as a variable of type `System.String`.	Integer value is available as a variable of type `System.Int32`.	Boolean value is available as a variable of type `System.Boolean`.	Array variable is available as a variable of type `System.Array`.
HTTP	String value is substituted where ever the parameter name occurs in HTTP action API endpoint, credentials, or body.	Integer value is substituted where ever the parameter name occurs in HTTP action API endpoint, credentials, or body.	Not supported	Not supported
Ansible	Ansible parameters extra variables, limit, tags, skip tags, job type and all survey parameters of type text, password, or multiple choice are harvested as string parameters.	Ansible parameters verbosity, inventory, credentials, and all survey parameters of type integer are harvested as integer parameter types.	Ansible parameter diff mode is harvested as Boolean parameter type.	Survey parameters of type "multiselect" are harvested as an array parameter type.

Adding a GOTO element

The GOTO element is used to skip steps or jump to the end of the runbook.

See the following example scenarios where the GOTO element is used.

- **Example: Run optional steps as result of a check**:

  1. Log in to the server where you want to start the application server.
  2. Check the current CPU consumption. If it is less than 80% **GOTO Step 5**.
  3. Find the process consuming the most CPU.
  4. Determine the process.
  5. Start the application server.
  6. Check the log file for confirmation that the application server is started.

- **Example: The execution was successful and you can complete the runbook**:

  1. Restart the Application Server.
  2. If this was successful, then GOTO END.
  3. Check your log files.

Configuring runbook creation

Use these procedures to define the runbook creation process and control the publication of runbooks.

- Configuring runbook approval

  Define whether to use the approval workflow or the direct publish workflow to make runbooks available for execution by operators.

- Runbook approval process

  The Runbook approval process offers control over the publication of runbooks. This is beneficial if a mandatory review process is necessary, if prematurely released runbooks could be harmful, or when auditable information on the publication is required.

Configuring runbook approval

Define whether to use the approval workflow or the direct publish workflow to make runbooks available for execution by operators.

1. Navigate to the **Runbooks Library**.
2. Click **Settings** next to the **Create runbook** button.
3. Select **Approve runbooks before publish** if you want runbooks to be approved before they are published.
4. If you want to enforce that an approval must be refreshed at a regular interval, select **Set expiration date after xxx days** and choose a time interval. If this option is enabled, the runbook approval will expire after the specified number of days and can no longer be executed until the runbook is reviewed, submitted for approval, and approved again.

   The deactivated approval process is described in Create a runbook.

   The activated approval process is described in Runbook approval process.

   Note: If the approval process is deactivated, all pending approvals are deleted.

Runbook approval process

The runbook approval process offers control over the publication of runbooks. This is beneficial if a mandatory review process is necessary, if prematurely released runbooks could be harmful, or when auditable information on the publication is required.

Figure 1. Runbook approval process

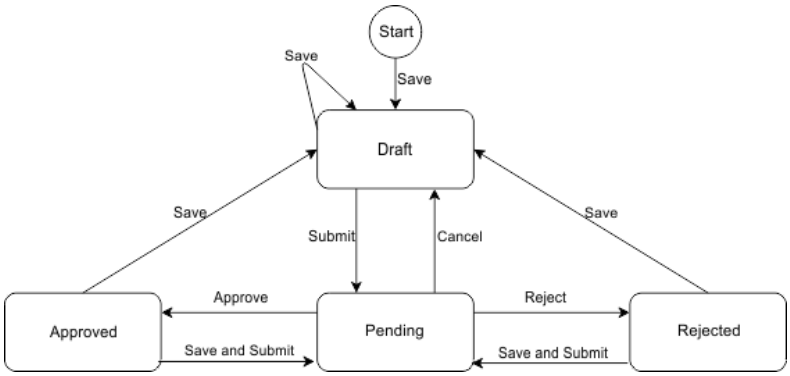


Figure. Runbook approval process

Figure 1 illustrates the states and transitions of the runbook approval process. If the approval process is enabled, it is not possible to directly publish a runbook. Instead an option to submit the runbook is displayed. The following options are then available:

List all runbooks that must be approved

1. Open the **Library** page.
2. From the **Status** filter drop-down menu, select **Approval pending**.
3. All runbooks for which approval is pending are shown.

List all runbooks that are approved

1. Open the **Library** page.
2. From the **Status** filter drop-down menu, select **Approved**.
3. All approved runbooks are shown.

Note: You can also filter for a status of **Approval expired** and **Approval rejected**.

Submit a runbook draft for approval

1. Create or edit a runbook and click **Submit for approval**.
2. Enter the name of the approver in the field provided and click **Send**.

The runbook is now in a pending state. It can no longer be edited, except for changes to its state.

Cancel a pending runbook approval

1. Locate the runbook on the **Library** page.
2. Click the menu overflow icon (the three dots at the end of the row) and select **Remove approval**.

The runbook is now in normal draft mode and can be edited. The assignee can no longer approve or reject the runbook.

Approve a runbook

Only the assigned person from the submission can approve a runbook.

1. Locate the runbook on the **Library** page.
2. Click the menu overflow icon and select **Approve**.

The runbook now has a status of approved, which is the default for execution. New drafts can be created. Information pertaining to the approval assignee and timestamp is stored within the runbook.

Reject a runbook

Only the assigned person from the submission can reject a runbook.

1. Locate the runbook on the **Library** page.
2. Click the menu overflow icon and select **Reject**.

The runbook now has a status of rejected and can be edited again. Information pertaining to the approval assignee and timestamp is stored within the runbook until a new draft version is created and the old information is deleted.

Runbook versions

You can use versioning to create incremental improvements to runbooks and monitor the success of changes. Subject Matter Experts can continuously improve the quality of runbooks by using the comments provided and the available run metrics. As a result, different versions of a runbook are created.

All available runbooks are listed on the **Library** page. Click the menu overflow icon (the three dots at the end of the row) and then click **Preview** > **Versions** to display all the versions of a runbook.

- **Draft**:

  The Draft version is a work-in-progress runbook. It is marked with a draft flag. A draft runbook is not yet published and does not run in production. It is not visible to any operator. You can run and test the current draft.

- **Latest published**:

  The latest published version is the version that is running and used by all operators. This is the version that the operations analyst can preview and work with.

- **Archived**:

  Archived versions are previously used versions. They are not in production and cannot be used by any operations analyst. The metrics of the previous versions help the Subject Matter Expert to monitor continuous improvements of the runbook. You can run archived version to learn why this runbook did not run successfully.

Save and publish a runbook

Creating a runbook involves documenting complex steps. These steps must be reviewed and tested before you can publish the runbook for use in production.

After a runbook is published and used in production, you might improve it based on the feedback that you receive from comments, success rate, and ratings. The runbook editor provides the following actions.

- **Cancel:** closes the runbook editor without saving any changes.
- **Save as:** use this action to save a copy of the runbook using a different runbook name.
- **Save draft:** saves the updates that are added. The editor remains open. **Save draft** is useful if you are editing a runbook and you want to take a break and continue later. Or you just want to save your changes as you go.
- **Save draft & close:** saves your changes and closes the editor. You use this option if you need to stop and work on something else.
- **Publish:** this button is only available when the approval process has been deactivated. After a runbook has been published it can be used by all operators in a production environment.
- **Submit for Approval:** this button is only available when the approval process is enabled.

If the approval process has been activated, see Runbook approval process for more information about approval before publication.

Filtering runbooks

Filter by status, groups or tags to easily locate runbooks.

1. Open the **Runbooks Library** page.
2. To filter for runbook status, select one or more items in the **Status** filter. The filter matches any selected status (or condition).

   Note: Filtering for status **Draft** displays runbooks that have never been published. Draft versions of already published runbooks are not displayed.

   - To filter for tags, select one or more items in the **Tag** filter. All tags must match to apply the filter (and condition).
   - To filter for authors, select one or more authors in the **Authors** filter. The filter matches any selected Authors.
   - If a combination of several filters are used, all filters must match to apply the filter (and condition).
   - Additionally, you can filter by runbook name using the **Search** field.

     The search field allows to search using regular expressions. If you want to search for special characters that are used by regular expressions, for example plus (+), asterisk (*), question mark (?), brackets (() [] {}), or backslash (\), you must escape those characters. For example, if you want to search for (T), you must enter \(T\) as the search string.

     Note: The filter drop-down list displays up to 50 available filtering options. You can start typing the name of the filter option that you are looking for. As you type, the list displays only items that contain the typed text.

Assigning runbooks to groups

Each runbook can be assigned to one or more groups, or can remain unassigned.

Runbooks can only be assigned to user groups if the respective feature is by an administrator. If the feature is enabled and if you only want to check what groups a runbook is assigned to, complete steps 1 - 3 from the following instructions for *Using the feature* section but do not make any changes.

Assigning runbooks to groups allows you to organize runbooks in a more efficient manner. There are two main reasons for assigning runbooks to groups:

- Groups determine the visibility of runbooks for operations analysts.
- It is easier to find relevant runbooks in the **Library** page when Filtering runbooks by groups.

Part 1: Enabling the feature

1. Navigate to the Runbook **Library** page.
2. Click the Configure settings icon next to the New runbook button.
3. Select the **Allow users to assign runbooks to groups** checkbox.
4. Save the changed setting.

Part 2: Using the feature

1. Go to **Library**.
2. Use the check boxes to select one or more runbooks.
3. An action bar is displayed. The bar shows available actions and how many runbooks are selected.
4. On the action bar, click **Grant permission**.
5. Use check boxes to choose which groups the selected runbooks should belong to. You can also select all or clear all groups by using the checkbox in the header row.

   Note: Some checkboxes might appear selected, but dimmed. This is an intermediate state that occurs when some (but not all) of the selected runbooks belong to the group.
6. Click **Assign** to confirm the selection or **Close** to discard any changes.

Export and import runbooks

Export and import runbooks by sending requests to the HTTP API. You can use existing tools to send requests, for example cURL, Postman for Chrome, or RESTClient for Firefox.

Before you can send requests, you need to create an API key for authentication. See API Keys and navigate to section *Get authentication by using Zen API Key* for instructions on how to create and encode an API key (**ZenApiKey**).

In addition, determine the host name part of your IBM Cloud Pak for AIOps user interface URL. The base URL for sending API requests has the format **https://<waiops-hostname>**, for example, **https://cpd-cp4waiops.apps.mycluster.mycompany.com**.

Runbooks are identified by an ID that gets generated when you create the runbook. For example, this ID is used when you create a runbook policy to link the runbook to alerts that match a specific filter pattern. When you plan to export the runbook and import it again at a later time (for example, to restore a backup version of the runbook after the runbook has been removed), you must use the "**keepId**" parameter to include the runbook ID during the export. This is so the integrity of the runbook policy is preserved when the runbook is imported later on.

Depending on your certificate configuration, you might need to add the "**-k**" or the "**--insecure**" option for the sample curl commands listed below.

Export Runbooks

The following two calls create output, which can be imported again.

Use the following API to export all runbooks (including runbook IDs):

```
GET api/v1/rba/runbooks?exportFormat=keepId
```

When enabling the option exportFormat via the query parameter, the usual output of the command is replaced by an array of JSON objects. These JSON objects can be used to create the runbooks by using the import API endpoint. The following call is an example:

```
curl --header "Authorization: ZenApiKey <ZenApiKey>" https://cpd-cp4waiops.apps.mycluster.mycompany.com/aiops/api/story-manager/rba/v1/runbooks?exportFormat=keepId
```

Replace **<ZenApiKey>** with the encoded Platform UI (Zen) API key token that you created before.

Use the following API to export a single runbook:

```
GET api/v1/rba/runbooks/<RunbookId>?exportFormat=keepId
```

Replace **<RunbookId>** with the ID of the runbook. You can retrieve this ID, for example, from the **Edit Runbook** window. The following call is an example:

```
curl --header "Authorization: ZenApiKey <ZenApiKey>" https://cpd-cp4waiops.apps.mycluster.mycompany.com/aiops/api/story-manager/rba/v1/runbooks/*<RunbookId\>*?exportFormat=keepId > exportedRunbook.json
```

Replace **<ZenApiKey>** with the encoded Zen API key token that you created before. Replace **<RunbookId>** with the ID of the runbook.

Import Runbooks

Use the following API to import runbooks:

```
POST api/v1/rba/runbooks/import
```

The following call is an example:

```
curl -H "Expect:" -H "Content-Type: application/json" --header "Authorization: ZenApiKey <ZenApiKey>" -d@./exportedRunbooks.json https://cpd-cp4waiops.apps.mycluster.mycompany.com/aiops/api/story-manager/rba/v1/runbooks/import
```

Replace **<ZenApiKey>** with the encoded Zen API key token that you created before. Note: **./exportedRunbooks.json** is the output file that is created from the previous example.

Note: If you previously exported a lot of runbooks (or some very large runbooks), you might face the 10 MB upload limit of the Runbook API when trying to import them again. The runbook file must be split into multiple JSON files, and each of these chunks must be imported separately. The following examples assume that you have the **jq** tool installed.

- **Example**: Split by maximum file size

  If your runbooks do not reference any automation actions, or if your automation actions are not shared between multiple runbooks, you can use the following command to split the file into chunks of a maximum of 10 MB each, named **exportedRunbooks.json.aa**, **exportedRunbooks.json.ab**, etc.:

  ```
  cat exportedRunbooks.json | jq .[] -c | split -C 10485760 --filter='jq . -cs > $FILE' - exportedRunbooks.json.
  ```

- **Example**: Split by automation actions

  If **exportedRunbooks.json** is larger than 10 MB and if it contains multiple runbooks that reference the same automation action, then try to group all these runbooks into a single chunk file. This avoids creating multiple copies of the same automation action during the import. You can use the following command to split the original file into multiple JSON chunk files, named **exportedRunbooks.json.aa**, **exportedRunbooks.json.ab**, etc. Each of these files contains all the runbooks that reference the same set of automation actions:

  ```
  cat exportedRunbooks.json | jq 'group_by(.automations)'[] -c | split -l 1 - exportedRunbooks.json.
  ```

General Notes on export and import

- Exporting runbooks returns the data of referenced automations inline. These automations are created new when the runbook is imported.
- Import actions do not overwrite existing runbooks, automations, or both.
- Exporting multiple runbooks at once will consolidate automations that are referenced by more than one runbook and only create the minimum necessary number of new automations.

- Imported runbooks are in draft state and do not retain any kind of version, execution, or approval history. The user performing the import action is visible as the runbook creator.
- Exporting runbooks returns the data of referenced runbooks depending on the chosen export format mode.
- In normal mode, nested runbooks are exported in the "flat" structure, where all runbook references are resolved and runbooks are exported as if they did not contain references to other runbooks.
- In keepId export mode, nested runbooks are exported while retaining their referencing structure. Referenced runbooks are included in the data as additional runbooks. Runbooks that are referenced by their latest version keep their ID. Runbooks that are referenced by a fixed version create an additional entry with a placeholder ID, which is replaced by a new random ID upon importing the data set.
- Importing a data set which includes nested runbooks fails to create any runbooks referencing other runbooks if the "publish" flag is not set. This occurs because referencing other runbooks is only allowed on published runbooks, not on drafts.
- When exporting multiple runbooks, ones with invalid nested runbook references and invalid automation action references will not be exported. The export will include all other runbooks, and all errors are logged in the runbook service.

For more information about creating an API key, see API Keys.

For more information about the API, see Automation (runbook) API.

Related links

Installing new SSH key during migration

The SSH key that is associated with an SSH connection cannot be moved or copied to another deployment.

Delete a runbook

Users assigned the Service Administrator, Automation Administrator, or Administrator role can delete a runbook.

1. Open the **Runbooks Library** page.
2. Use the check boxes to select one or more runbooks.
3. An action bar is displayed showing the available actions and the number of runbooks selected.
4. On the action bar, click **Delete** . If you delete a runbook, all versions and the activity history of the runbook are deleted as well.

Activities

See recently executed runbooks, including runbooks in progress. Up to 1000 records matching the current criteria are displayed on the **Activities** UI. More records can be retrieved through the runbook API. If you need to see the records on the UI, use the filters on the page to refine the search criteria so that less records match.

On the **Activities** page, you can find all runbooks that have been started. You can filter your list by the following criteria:

- User

  Click the **User** filter and select one or more user names. All runbook activities for the selected user names are shown.

- Runbook name

  Click the **Runbook** filter and select one or more runbook names. All runbook activities for the selected runbook names are shown.

- Runbook type

  Click the filter **Type** and select **Automatic Manual**. Fully automated runbooks are runbooks in which each step contains an automation. An operator does not need to interact with the runbook. Automated runbooks can be mapped with events by creating a trigger. To see the automated runbooks that are started by a trigger, click **Automatic**.

- Runbook execution status

  Click the **Status** filter and choose a status such as Success, Failed, Cancelled, In progress, or Completed. For example, you can find all runbook activities that are paused by selecting **In progress**. When you are ready to return to the task you can resume the runbook. Select the **Canceled** status if you want to see all canceled runbooks and the reason why they have been canceled.

You can take the following actions on the **Activities** page:

- **Resume** activities that are paused or in progress.

  This lets you continue with the activity's current step, and it displays the activity's history up to that point including which steps have been skipped as well as all relevant action results.

- **View** past runbook activities' details.

  This displays the entire history of the activity. In the case of a fully automated activity, the details also include action activity errors when available. Only activities that are in progress or paused can be changed, so this view of the activity is read only.

Monitor runbook history

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Runbooks**.
4. Select a runbook.
5. Click the menu overflow icon (the three dots at the end of the row) and select **View history**.

   The activity history table and runbook details such as description, type, rating, success rate, and activity statistics are displayed.
6. Use the filter boxes on the page to display only specific runbooks, for example:
   1. To see runbook activities for a specified time frame, select the dates in the **Start** and **End** fields.
   2. To see runbook activities that are still in progress, select **In progress** from the Status filter. If you want to resume a runbook execution, click **Resume**.

Actions

In runbooks, actions are the collection of several manual steps into a single automated entity. An action improves runbook efficiency by automatically performing procedures and operations.

The Service Delivery Engineer can provide an action to replace frequently used steps with a single click. Depending on the action type, defining an integration might be a prerequisite. For more information about how to set up a integration, see Configuring Integrations.

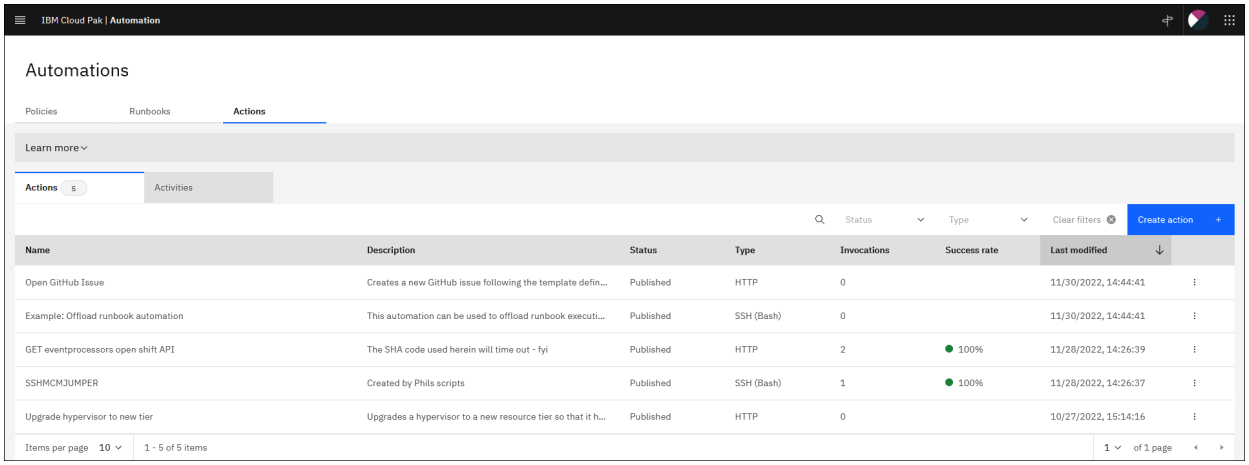


Figure. Actions UI

For more information about runbook actions, see the following topics:

- Managing actions
- Creating an action
- Ansible actions
- Predefined actions
- Action versions
- Save and publish an action
- Testing an action
- Running an action
- Activities

Managing actions

As a Service Delivery Engineer, you are interested in how the actions run and how to keep them running smoothly. Use the **Actions** page to test actions, view statistics, and administer the available actions.

The following types of runbook contain actions:

- Semi-automated runbooks: runbooks that contain one or more actions. These runbooks consist of manual and automated steps.
- Fully automated runbooks: runbooks that contain an action in each step. If fully automated runbooks are started by a trigger, you can find their status on the **Activities** page. Select the filter **Type** and select **Triggered automatically**.

On the **Actions** page, you can find all available actions with the following information displayed in the table:

- **Name**:

  Name of the action.

- **Description**:

  Description of the action.

- **Type**:

  Actions are of the type SSH, HTTP, Client-side, Predefined, or Ansible, depending on the connection type used.

- **Invocations**:

  The number of times the action has been used in runbooks.

- **Rating**:

  Average rating, based on the ratings provided by operators and administrators who ran the action and provided a rating.

- **Success rate**:

  How reliably did this action run so far? The success rate indicates how well an action ran.

- **Last modified**:

  A timestamp of when the action was last modified.

- **Actions**:

  As a Subject Matter Expert you can **Test**, **Edit**, **Copy**, and **Delete** an action unless the action is of type **Predefined** or **Ansible**.

  Select an action in the table to display the **Preview** side panel for that action. You can Edit, Copy, Test, and Delete actions directly from the side panel. The **Details** section is open by default and displays the **Description**, **Action ID**, **Prerequisites**, **Success rate**, and **Last modified by**.

  The **Parameters** tab contains the name, description, and default values of the parameters contained in the action.

  The **Content** tab of the side panel shows the script content in the case of script actions, or the Method and API endpoint in the case of HTTP actions.

  Click the **X** in the top right corner to close the side panel.

How is the success rate of an action calculated?

An action can have the following exit states:

- **failed**:

  The action failed to complete, either because it could not even be started, for example because of an invalid hostname, unsuccessful authentication, or incorrect parameters (in the case of an Ansible action), or it failed during execution because of an internal issue such as a zone failure.

- **unsuccessful**:

  The action could be started but has a return code indicating an error, for example for ssh actions the exit code is not 0, for http actions the http return code is >= 400.

- **successful**:

  The action could be started and has a return code indicating success, for example for ssh actions the exit code is 0, for http actions the http return code is <400.

The success rate indicates the quality of an action. For example, for ssh actions it indicates the quality of the corresponding scripts. For Ansible actions, the success rate indicates the quality of the corresponding Ansible playbooks.

It does not indicate if the configuration is correct. For example, if an ssh action cannot be executed because of an invalid hostname or authentication failure, you will see this in the success rate of the runbook, but not in the success rate of the action.

Therefore, the success rate is calculated using the number of unsuccessful and successful action activities. It does not include the number of failed action activities.

The total number of action invocations that is listed in the **Actions** table includes all action activities, including the failed ones.

Creating an action

Create an action by defining the action type and configuring the parameters and fields.

Actions are routines that run a task automatically. There are different types of actions depending on how tasks are run. Runbook Automation supports actions of type SSH (script), HTTP, Client-side, Predefined, and Ansible. For actions of type script and Ansible, an integration must be configured. Ansible automations are harvested directly from Ansible, so they appear in the automations table as soon as the Ansible connection is configured. For more information about how to set up a connection, see Configuring Integrations.

- **Creating SSH actions** - You can create an action of type script by using the **SSH provider**. The Operating System of the endpoint system can be either UNIX® (including AIX® and Linux®) or Windows™. For UNIX®, the SSH provider executes a script using the default shell. Alternatively, a shebang (#!) can be used to execute the script in any scripting language the target system can interpret. The SSH provider does not require an additional agent as it uses a direct remote execution connection through SSH.
- **Creating HTTP actions** - The HTTP action provider allows you to send HTTP requests to a specified web service.
- **Creating Client-side actions** - Client-side actions provide a mechanism to launch out from the Cloud Pak for AIOps user interface in the user's browser to any third-party web site.

Creating SSH actions

You can create an action of type script by using the SSH provider. The Operating System of the endpoint system can be either UNIX (including AIX® and Linux®) or Windows. For UNIX, the SSH provider executes a script using the default shell. Alternatively, a shebang (#!) can be used to execute the script in any scripting language the target system can interpret. The SSH provider does not require an additional agent as it uses a direct remote execution connection through SSH.

For Windows, the SSH provider can execute a Powershell script.

Note:

- To access the system through SSH you must set up an SSH server on the Windows system, for example OpenSSH for Windows Server 2019 or Windows 10. The Windows System must have Powershell installed as the default shell of the SSH server.
- If you are using a jumpserver, the jumpserver must be a Linux system. You cannot use a Windows system as a jumpserver.

Complete the following steps to enable script actions:

1. Create an SSH integration. For more information about how to set up an integration, see Configuring Integrations.
2. Create an action of type SSH. Click **Actions** > **Create action**. Complete the following fields:
   - **Type**:

     Select SSH. If you want to run a script as an action, you must configure an SSH Action Provider on the **Integrations** page.
   - **Name**:

     Provide a name that describes what this action does. For example, the name can be *Find large file systems*.
   - **Prerequisites**:

     If this action requires prerequisites, add this information. For example, `Db2 10.5 or higher`.
   - **Description**:

     Provide any helpful additional information so that the user can immediately understand which problem this action solves.
   - **Tags**:

     Provide tags to filter actions. You can select any existing tags from the drop-down list and click the plus icon "+" to add them. To create a new tag, type the text into the field, select the text string displayed, and click the plus icon "+".
   - **Shell**:

     Select **Bash** for scripts that run on UNIX systems using `bash` as the ssh shell. Within the script you can specify other script interpreters that are installed on the target system, for example `python`. See the item "Script" below for an example.

     - Notes about how Runbook Automation invokes Bash scripts:
       - It generates a temporary file with the script content.
       - It exports parameters as environment variables.
       - It runs the temporary script file.
       - It deletes the temporary script file.
       - It returns the script return code of the temporary script execution.

     Select **Powershell** for scripts that run on Windows systems using `powershell` as the ssh shell (using `cmd` as the ssh shell is not supported).

     - Notes about how Runbook Automation invokes Powershell scripts:
       - It assigns parameter values to variables with the selected type.
       - It executes the commands from the script.
   - **Script**:

     You can either select **Import** to select the script from your file system, or you can directly enter the script into the editor. If you selected **Bash** the script can only be run on UNIX systems using the default shell or the interpreter, which is specified after the shebang `#!`. If you selected **Powershell** the script can only be run on Windows systems using Powershell.

     Notes:

     - Imported scripts must be in UTF-8 format if they contain non-ascii characters (for example, Chinese).
     - If the "Bash" shell type is selected, then any Windows line endings from the imported script are automatically converted to UNIX line endings.

     Examples:

     - Create an SSH action with the following Bash script that uses the `python` interpreter:

```
#!/usr/bin/env python3
import os
import platform
print("Your message \'" + os.environ.get("message") + "\' is presented by python version " +
platform.python_version())
```

Create a parameter **message** and assign the default value *Hello python* to the parameter. Run the script on a target system where **python3** is installed. The output looks similar to the following output:

```
Your message 'Hello python' is presented by python version 3.9.13
```

- Create an SSH action with the following Bash script that uses the IBM WebSphere Application Server CLI to start, stop, or get the server status of the server:

  ```
  $profilePath/bin/$serverAction.sh $serverName
  ```

  Create the parameters:

  - **profilePath** (the path name of the WAS profile, set the type to a single string, example default value: **/opt/IBM/JazzSM/profile**)
  - **serverAction** (the action to take on the WAS server, set the type to an enumeration of strings (**serverStatus**, **startServer**, **stopServer**), example default value: **serverStatus**)
  - **serverName** (the name of the WAS server, set the type to a single string, example default value: **server1**).

  Run the script on a target system where IBM WebSphere Application Server is installed and has administrative credentials configured in **$profilePath/properties/soap.client.props**. Alternatively, add parameters for the administrative user name and password and modify the CLI command accordingly.

- **Action output**:

  - **Filter action output** By default, all action output gets collected and printed to the action output field. If you are only interested in a subset of the action output (for example, because you want to create a chain of two actions within a runbook, where some particular information from the first action is used as input to the second action, then you can enable the **Filter action output** toggle and specify a pattern that controls which part of the action output gets returned.
  - **Choose query syntax** Depending on the type of output that gets created you can choose between three types of pattern-matching engines and their respective syntax:
    - **XPath**: best suited for parsing XML data
    - **JSONPath**: best suited for parsing JSON data
    - **Regular expression**: best suited for parsing unstructured data
  - **Choose query syntax at runtime** Select this checkbox if you want to allow the query syntax to be chosen when testing the automation, and also within the parent runbook. This checkbox is typically disabled. If this checkbox is enabled, the system property **filter-syntax** is added to the action.
  - **Enter query pattern** Enter a pattern that is supported by the selected query syntax. For example, if the JSON output object is expected to contain an attribute called "version" and you selected the "JSONPath" query syntax, then you can use the query pattern "$.version" to pick the value of the version attribute out of the overall JSON output.
  - **Enter query pattern at runtime** Select this checkbox if you want to allow the query pattern to be chosen when testing the automation, and also within the parent runbook. If this checkbox is enabled, the system property **filter-query** is added to the action.
  - **Test output filter** Click this link (or the **Test** button in the actions menu) when you are ready to test the action. This saves the action and directly opens the **Test action** page for the current action. For many developers, working with query patterns is an iterative process. Therefore it is recommended to enable **Enter query pattern at runtime** at least while you are developing the action, and switch to **Test action** to fine-tune the query pattern. The development of the query pattern is also supported within the **Test action** page itself, where you can toggle between **Show filtered output** and **Show unfiltered output** after you have run the action. You can also toggle between **Edit action** and **Test action** until you are satisfied with the result.

- **Parameters:**

  Add input parameters to run the action script. Those input parameters are available to the script as environment variables. For example, a parameter **filesystem** is used in a Linux script as **$filesystem**. The **$** sign is automatically added to distinguish user-defined parameters from system parameters such as **target** and **user**. The parameter is exported without the dollar sign. **Note:** input parameter names must consist of alphanumeric characters and the underscore character only. System parameters are not available for use within your own script code. If you want to use some of that data within your script code, you must create related user-defined parameters. The following system parameters exist:

  - **target**: Mandatory parameter. Applies to actions of type Script. The **target** parameter is created to define the target machine where the script is running. By default, it is assumed that the SSH server on the target machine is listening on port 22. If a different SSH port is used, then set the value for the **target** parameter to **hostname:port**, for example, **mytarget.mycompany.com:2022**.

  - **user**: Mandatory parameter. Applies to actions of type SSH. The **user** parameter defines the UNIX or Windows username that is used to run the script on the target machine.

  - **filter-syntax, filter-query**: These system parameters are added if you enable the related checkboxes for action output parsing.

    Note: If the value of the user parameter contains the @ symbol, for example an email address, the @ symbol and all characters that follow it are ignored.

3. Add the action to the runbook, see Adding actions.
4. Run the runbook, see Run your first runbook.

For more information about SSH actions, see the following topics:

- **Single and Multi Target Actions**
- **Output format of Multi Target Actions**
- **Comparison of Single and Multi Target Actions**

Single and multi target actions

You can use single target or multi target actions to execute script actions.

Single Target

The parameter with the name **target** has special meaning to script actions as it defines the system on which the action will run. The system can be defined by a short hostname, a FQDN, or an IP address. When executed a script action will connect to the system identified by the **target** and execute the script there.

The execution result of the script will be reflected in the status of the action. Possible action states are **executing**, **successful**, **unsuccessful**, **failed**, and **unknown**. In the case of a fully automated runbook execution, this status will also be used to decide whether the runbook will be canceled or continued.

Examples:

- An action which receives **prod-server1**, as the content of the target variable will execute this action on the system **prod-server1**. The script exited with return code of zero and the state of the action instance is set to **successful**.
- An action which receives **192.168.55.56**, as the content of the target variable will execute this action on the system with the matching IPv4 adress. The script exited with a non-zero return code and the state of the action instance is set to **unsuccessful**.
- An action which ran in step 1 of a fully automated runbook failed to reach the endpoint (for example because the server was not available) and the action could not be executed at all. The status of the action instance is **failed**. This status is reflected in the status of the execution. The execution will stop after the first step and report a failure.

Multi Target Actions

The other mode of operation of an action is the **Multi Target Action** (MTA). In order to execute an action as an MTA, specify the target string in the following format: **[$target1, $target2, ..., $targetN]** or as a comma-separated list without square brackets: **$target1, $target2, ..., $targetN**.

The following rules apply to MTAs:

- If you use the format with brackets, then the target string must begin with a left square bracket (**[**) and must end with a right square bracket (**]**).
- Define a list of systems separated by comma (**,**).
- Duplicated entries are detected and ignored. This means specifying the same target multiple times will have no effect.
- An empty list in the format **[ ]** is also allowed.

Multi Target Actions allow you to execute the same action on any number of targets in parallel, removing the necessity to execute the same action with different targets sequentially. If the target is specified in such a way, how the action is executed and how the results are treated will be different. The following criteria apply to MTAs:

1. For every entry in the comma-separated list, the script will be executed on the specified target. **Note:** you cannot specify the same target more than once.
2. All actions and information are summarized in one log. This log follows a specific format, see Output format of Multi Target Actions.
3. The status of the execution will be **executing** as long as at least one execution is still ongoing and **successful** once all script processes are finished. The result will be **successful** even if some or even all actions reported **unsuccessful**, **failed**, or **unknown**.
4. Specifying an empty array is allowed. No actual execution occurs, but an execution record is still created.
5. The record will always report **successful** as its state. So if the runbook is executed fully automated, the runbook instance will proceed to the next step. This allows the use of multi-target automations as **Conditional Automations**: use the returned lists of previous multi-target automations in subsequent multi-target automations to only run the subsequent automation where the script in the previous automation was successful. Or where it was unsuccessful.

Examples:

- An action receives **[prod-server1]** as the target. It will execute the script on the system **prod-server1**. The script exited with a non-zero return code. The status of the execution is **successful**. The output of the action instance contains information that this script execution was **unsuccesful** on **prod-server1**.
- An action receives **[prod-server1**, **prod-server2**, **prod-server3]** as the target. It will execute the script on three systems in parallel. All scripts exit with zero as the return code. The output and status of each of the three script activities is written to the output. The status of the overall execution is **successful**.
- An action receives **[]** as the target. This triggers the conditional action workflow No execution will take place. The execution record will have an almost empty output and the status will be **successful**.

Output format of multi target actions

Despite being executed on any number of endpoints, the results of multi target actions (MTA) are consolidated into one activity record. This record combines the information from all activities.

The output is formatted as follows:

- If the script produces output the output is displayed after two prefixes. The first item is the name of the target. The second item is either **out** or **err** depending on whether the output was written to **`stdout`** or **`stderr`**.

- The log will group the output by target. So the complete output of the first target will be displayed before the output of the second target is displayed and so on. This means the output is not ordered chronologically, although the output by target is.
- At the end of the output a summary is always present. The summary consists of three additional lines. They follow the pattern **$status : [$listOfTargetsWithThisState]**. Where **$status** is one of the following: **unsuccessful**, **successful**, or **failed**.
- If the target array is an empty array, the output will only contain the summary.

Example

An automation has been executed with the target **[prod-server1, prod-server2]**. The execution on **prod-server1** was successful, while the execution on **prod-server2** was unsuccessful. The output will look as follows:

```
prod-server1 out: Some output from the script to stdout happening at 08:34
prod-server1 out: More output from the script to stdout happening at 08:36
prod-server1 out: Script prints more to stdout
prod-server1 status: successful

prod-server2 out: Some output from the script happening at 08:35
prod-server2 out: More output from the script happening at 08:37
prod-server2 err: Script prints output concerning a failure to err
prod-server2 status: unsuccessful

unsuccessful: [prod-server2]
successful: [prod-server1]
failed: []
```

Comparison of single and multi target actions

The following table outlines the difference between the operation modes:

Table. Operation mode differences for target actions

Aspect	Single Target	Multi Target Action
Number of parallel activities	One	Number of entries in target array
Final execution status	Equal to status of automation	Always successful
Failure during fully automated runbook (FARB) execution	Aborts the FARB	Proceeds the FARB
Specifying an empty target	Failure to execute automation	Enters special case for empty array

Creating HTTP actions

The HTTP action provider allows to send HTTP requests to a specified web service. It is highly recommended to send these requests using the HTTPS protocol, in order to have the communication encrypted and signed. If required in your environment, and if compliant with your company's security guidelines, you can also send requests using the unencrypted HTTP protocol (for example, in some test scenarios).

To create an action of type HTTP, click **Actions** > **Create action** and complete the following fields:

- **Type**:

  Select **`HTTP`**.

- **Name**:

  Provide a name that describes what this action does. For example, **`IBM Watson Translate`**.

- **Description**:

  Provide any helpful additional information so that the user can immediately understand which problem this action solves.

- **Prerequisites**:

  If this action requires prerequisites, add this information. For example, **`Watson service credentials are required`**.

- **Tags**:

  Provide tags to filter actions. You can select any existing tags from the drop-down list and click the plus icon "+" to add them. To create a new tag, type the text into the field, select the text string displayed, and click the plus icon "+".

- **API endpoint**:

  Specify the API endpoint of the web service to which the HTTP request will be sent to. It is highly recommended to use the HTTPS protocol, so the API endpoint URL should start with "https://". Only if absolutely needed and if in compliance with your company's security guidelines, you can also use the HTTP protocol, so the API endpoint URL may start with "http://". Alternatively, specify a parameter that contains a full valid web service URL at the time when the HTTP action is started. For example: **`$ticketingSystemApiURL`**.

  You can choose between a custom API endpoint or one of the supported Cloud Pak for AIOps APIs.

Note that these supported Cloud Pak APIs automatically handle authentication. Provided that the action is run manually (for example, inside of a manual runbook or via a contextual menu item in the Alert Viewer) by passing through the executing user's information.

This automatic authentication is not supported in fully automated runbooks. To create an action that calls a supported Cloud Pak API and that can be used in a fully automated runbook, you must explicitly specify an Authorization header as documented for the respective API.

If you want to access any other Cloud Pak for AIOps API you should do this via the internal hostname **`https://ibm-nginx-svc:443/aiops/api`** instead of using the external hostname. In this case, you must specify an Authorization header as documented for the respective API.

For example, to retrieve some data from the Metric API, you can use **`https://ibm-nginx-svc:443/aiops/api/app/metric-api/v1/metrics`** with an Authorization header that contains your API (Zen) key. Alternatively, you can use a parameter for the entire URL that contains a full valid web service URL at the time when the HTTP action is started.

- **Method**:

  Choose the HTTP method. For example **`POST`**.

- **Username**:

  If basic authentication is required to use the web service, specify the API user name for basic authentication.

- **Password**:

  If basic authentication is required to use the web service, specify the API password for basic authentication.

- **Accept**:

  Specify the accept request header. The accept header is used to specify the media types which are acceptable for the response of the request. For example **`text/html`**.

- **Accept-Language**:

  Specify the accept-language request header which is used to restrict the set of languages that are preferred for the response of the request. For example **`en-US`**.

- **Additional header**:

  Optionally, specify any additional request headers that are needed for the request. For example **`accept-charset: utf-8`**.

- **Ignore certificate errors**:

  Select this check box to ignore certificate errors. Use this option only for test purposes. In production environments, ensure that the correct certificates are installed on the target web service.

- **Action output**:

  - **Filter action output** By default, all action output gets collected and printed to the action output field. If you are only interested in a subset of the action output (for example, because you want to create a chain of two actions within a runbook, where some particular information from the first action is used as input to the second action, then you can enable the **Filter action output** toggle and specify a pattern that controls which part of the action output gets returned.
  - **Choose query syntax** Depending on the type of output that gets created you can choose between three types of pattern-matching engines and their respective syntax:
    - **XPath**: best suited for parsing XML data
    - **JSONPath**: best suited for parsing JSON data
    - **Regular expression**: best suited for parsing unstructured data
  - **Choose query syntax at runtime** Select this checkbox if you want to allow the query syntax to be chosen when testing the automation, and also within the parent runbook. This checkbox is typically disabled. If this checkbox is enabled, the system property **`filter-syntax`** is added to the action.
  - **Enter query pattern** Enter a pattern that is supported by the selected query syntax. For example, if the JSON output object is expected to contain an attribute called "version" and you selected the "JSONPath" query syntax, then you can use the query pattern "$.version" to pick the value of the version attribute out of the overall JSON output.
  - **Enter query pattern at runtime** Select this checkbox if you want to allow the query pattern to be chosen when testing the automation, and also within the parent runbook. If this checkbox is enabled, the system property **`filter-query`** is added to the action.
  - **Test output filter** Click this link (or the **Test** button in the actions menu) when you are ready to test the action. This saves the action and directly opens the **Test action** page for the current action. For many developers, working with query patterns is an iterative process. Therefore it is recommended to enable **Enter query pattern at runtime** at least while you are developing the action, and switch to **Test action** to fine-tune the query pattern. The development of the query pattern is also supported within the **Test action** page itself, where you can toggle between **Show filtered output** and **Show unfiltered output** after you have run the action. You can also toggle between **Edit action** and **Test action** until you are satisfied with the result.

- **Parameters**:

  Add input parameters to run the action. Those input parameters can be referred in the entry fields. For example, a parameter **`text`** can be referred as **`$text`**.

Creating Client-side actions

Client-side actions provide a mechanism to launch out from the Cloud Pak for AIOps user interface in the user's browser to any third-party website.

The **Client-side** action provider allows you to build a URL from a core URL string and some optional parameters. It is highly recommended to use the HTTPS protocol, in order to have the communication encrypted and signed. If required in your environment, and if compliant with your company's security guidelines, you can also use the unencrypted HTTP protocol (for example, in some test scenarios). To run a client-side action, (for example, as an automated step within a runbook) click the **Launch** button. When running a client-side action, the parameters are replaced with the actual values, and the resulting URL is launched in a new browser tab. You can review the URL in the action output area and launch it again by clicking the displayed URL hyperlink.

Client-side actions are composed in the runbook back-end service but they are run in the browser context. Therefore, all runbooks that contain automated steps with client-side actions are considered as manual runbooks. Runbooks that are intended to be run fully automated must not contain client-side actions.

If the third-party website requires authentication, then it will typically ask to provide login credentials before opening the requested page. It is not recommended to use parameters within the client-side action to contain sensitive information (such as user passwords), as they can be seen in clear text in the resulting URL. As a result, password-style string parameters are not supported for client-side actions.

1. To create a client-side action, click **Actions** > **Create action** and complete the following fields:.

- **Type**:

  Select `Client-side`.

- **Name**:

  Provide a name that describes what this action does. For example, `IBM homepage search`.

- **Description**:

  Provide any helpful additional information so that the user can immediately understand which problem this action solves. For example, `Search the IBM home page for the given subject`.

- **Tags**:

  Provide tags to filter actions. You can select any existing tags from the drop-down list and click the plus icon "+" to add them. To create a new tag, type the text into the field, select the text string displayed, and click the plus icon "+".

- **Action client URL**:

  Configure the URL for your action. It is highly recommended to use the HTTPS protocol, so the API endpoint URL should start with "https://". Only if absolutely needed and if in compliance with your company's security guidelines, you can also use the HTTP protocol, so the API endpoint URL may start with "http://". You can use parameters. Parameter names must be prefixed with the `$` character. For example, `https://www.ibm.com/search?q=$query`. Alternatively, specify a parameter that contains a full valid URL of the third-party website at the time when the client-side action is started. For example: `$openTicketWebPageURL`.

- **Action parameters**:

  Add input parameters to run the action. Those input parameters can be referred in the URL. For example, a parameter `query` can be referred as `$query`. The parameter query can be defined as follows:

  - Name: `query`
  - Description: `The subject to be searched on the IBM home page`
  - Type: `A single string value`
  - Minimum length: `1`
  - Maximum length: `not selected`
  - Password: `not selected`
  - Set a default value: `selected`
  - Default value: `runbook`

2. Add the action to the runbook, see Adding actions.
3. Run the runbook, see Run a runbook.

Deleting an action

Users assigned the Service Administrator, Automation Administrator, or Administrator role can delete an action.

1. Open the **Actions** page.
2. Use the check boxes to select one or more actions.
3. An action bar is displayed showing the available batch actions and the number of actions selected.
4. On the action bar, click **Delete**.
5. A dialog will open, showing you any potential warnings or reasons why some actions cannot be deleted. Select the ones that you want to delete and confirm. If you delete an action, all versions and the activity history of the action are deleted as well.

Ansible actions

The Ansible action provider harvests Ansible type actions from the connected Ansible Automation Platform Controller server. These actions are created automatically and reference an existing asset from an integrated Ansible installation.

Ansible Actions can reference the following:

- Job Templates
- Workflow Job Templates

Unlike SSH Actions, Ansible Actions do not define new automation content inside IBM® Runbook Automation. Instead they point to an existing automation, present in the Ansible Automation Platform Controller server, which is a job or job workflow template.

When an Ansible Action runs, it calls the specific "launch" operation of that template on the connected Ansible Automation Platform Controller server. If the template requires prompting for specific variables, these variables are sent by Runbook Automation. Runbook Automation allows these variables to be defined as Action parameters.

The integration with Ansible has been verified with Ansible Automation Platform 2.4 and Automation Controller 4.5.2. Note that the set of parameters that can be actively used in the related automation actions is limited to what is documented at Job Template parameters and surveys.

Warning: Runbook integration with Ansible Automation Platform 2.5 is not supported by IBM Cloud Pak for AIOps 4.8.0.

See the topics that follow this section for more information about Ansible Actions.

- Harvesting Ansible actions
- Ansible Action interactions
- Job Template parameters and surveys

Harvesting Ansible actions

Complete the following steps to allow Runbook Automation harvest all Job Templates and Workflow Job Templates and create Ansible actions from those templates.

1. Set up an integration to Ansible.
2. Click **Actions**.
3. Check the Actions for the harvested Ansible Actions.

Ansible action interactions

The Ansible Action Provider automatically converts job templates into actions. These actions are read only in Runbook Automation, but they pick up any change to the original template made in Ansible. To change any aspect of the automation, you can edit the original template.

If you change the Ansible connection, for example to point to another Ansible Automation Platform Controller server with different templates, the previously harvested templates are no longer visible from the Actions page. Runbooks containing those templates can no longer run the actions. Instead, the templates from the new Ansible Automation Platform Controller server are visible.

If you change the Ansible connection back to a previous configuration, the actions that are harvested from this connection reappear, together with their statistics. All runbooks that contain those templates are again able to run those actions.

Note that harvested templates are cached in Runbook Automation, so changes made to the original templates might take up to five minutes to be reflected in the corresponding actions.

All runbook instances continue to be fully visible, independent of the state of the Ansible connection.

Job Template parameters and surveys

In Ansible a Job Template or Workflow Job Template can define a number of parameters and properties. They can be defined directly in Ansible's template itself, or can be set to *prompt on launch*. Every parameter that's set to be prompted must be supplied during template launch by Runbook Automation. In the case of an Ansible action executed through Runbook Automation, this translates to action parameters.

Job Templates in Ansible can accept input for the following parameters:

- diff_mode (type: boolean)
- extra_vars (type: string in JSON format)
- inventory (type: integer)
- credentials (type: integer; enumeration with labels)
- limit (type: string)
- verbosity (type: integer; range 0-5)
- tags (type: string)
- skip_tags (type: string)
- scm_branch (type: string)
- execution_environment (type: integer; enumeration with labels)
- labels (type: integer array; enumeration with labels)

- forks (type: integer)
- job_slice_count (type: integer)
- timeout (type: integer)
- instance_groups (type: integer array; enumeration with labels)

Note: Runbook Automation supports only single values for the "credential" parameter. Launching job templates that require more than one credential value to be specified is not supported.

Workflow Job Templates in Ansible can accept input for the following parameters:

- extra_vars (type: string in JSON format)
- inventory (type: integer)
- labels (type: integer array; enumeration with labels)
- tags (type: string)
- skip_tags (type: string)

Additionally, you can define any number of additional parameters to be prompted from the user through the **`survey`**. For every parameter defined in the survey an additional Action parameter is created. These parameter names are prefixed with **`extraVariables`**. and have a schema similar to the one defined for the survey parameter. For example, if you define a survey parameter with the name "age" and define it as an integer with minimum value **`0`** and maximum value of **`99`**, then the Action will have a parameter called "extraVariables.age" of type "integer", with minimum value of **`0`** and maximum value of **`99`**.

Note: Float and textarea parameters are not yet supported and will not show up as independent parameters. If you have surveys using these types of survey parameters, enable the *Prompt on launch* flag for the *Extra Variables* selection in your template. That way users can specify the values using the **`extraVariables`**. parameter.

The Ansible Action Provider will harvest the existing Job Templates and Workflow Job Templates and create matching Actions out of them. Those Actions are automatically created and kept synchronized with Ansible. They cannot be changed inside Runbook Automation. To change these Actions, directly edit the Job Template or Workflow Job Template inside Ansible. Runbook Automation will pick up the change and update the Action referencing the asset. Note that harvested templates are cached in Runbook Automation, so changes made to the original templates might take up to five minutes to be reflected in the corresponding Actions.

If an Ansible Action changes, make sure to update any parameter mappings made towards that Action in your runbooks.

If a parameter is set to *Prompt on launch* in Ansible, but also defines a value, Runbook Automation will import the default value defined in Ansible into the automation parameter.

During execution of Ansible Actions the Runbook Automation UI ensures that you can only pass values that conform to the parameter schemas. If you are using the API, make sure to pass the correct values by comparing your input to the parameter schemas.

Define default values in Ansible for parameters you want to prompt for. For more details on parameters, see the Ansible Documentation for Job Templates or Workflow Job Templates.

Predefined actions

Predefined actions perform some standard changes on alerts, like acknowledging the alert, assigning the alert, or modifying the severity of the alert.

Predefined actions are typically invoked from the right-click menu on entries in the Alert Viewer. You can also use predefined actions in the context of your own runbooks, for example to clear the related alert once all steps of the runbook have been performed successfully. Predefined actions may be used in fully-automated runbooks.

If you run an action (either from the actions table or as an automated step within a runbook) and no error occurs, then the resulting success message is not localized. Instead, an English message is displayed, so you can parse the output in a subsequent automated step of your runbook and proceed as appropriate. See the following table for the expected success message for each predefined action.

If you run an action and an error occurs, then whenever possible the output area displays a localized error message, for example: **`ARBA1102E Alert with ID '1234' does not exist`**. In some cases we receive just a generic message and wrap it, for example: **`ARBA1103E Predefined action failed: 400: Bad Request`**. In those cases check if the values of the other parameters (like **`team`** or **`owner`** for action **`Assign owner to alert`**) are typed correctly.

Predefined actions are immutable actions that exist after product installation. Each predefined action indicates that the action was last modified by user **System**. The modification date of a predefined action indicates when its source code was created or modified and is independent of the time when the product was installed.

You cannot create additional predefined actions. Similarly, you cannot copy, edit, test, or delete predefined actions.

The following table lists the existing predefined actions.

Predefined actions

Predefined action	Parameters	Expected output
Acknowledge alert	**`alert`**: ID of the alert to be acknowledged	**`Alert acknowledged`**
De-Acknowledge alert	**`alert`**: ID of the alert to be de-acknowledged	**`Alert deacknowledged`**
Assign owner to alert	**`alert`**: ID of the alert to be assigned; **`team`**: Name of the user group to be assigned to the alert (the user who performs this action must be a member of that particular user group; use the value **`-`** (hyphen) to set the **`team`** to "unassigned"); **`owner`**: User ID of the owner to be assigned to the alert (use the value **`-`** (hyphen) to set the **`owner`** to "unassigned")	**`Alert assigned`**

Predefined action	Parameters	Expected output
Update severity of alert	`alert`: ID of the alert to be changed; `severity`: New severity (choose from enumeration)	`Alert severity updated`
Suppress alert	`alert`: ID of the alert to be suppressed	`Alert suppressed`
De-Suppress alert	`alert`: ID of the alert to be de-suppressed	`Alert desuppressed`
Clear alert	`alert`: ID of the alert to be cleared	`Alert cleared`

Action versions

You can use versioning to create incremental improvements to actions and monitor the success of changes.

- **Draft**:

  The Draft version is a work-in-progress action. It is marked with a draft flag. A draft action is not yet published and does not run in production. The draft is not visible to any operator. You can run and test the current draft.

- **Latest published**:

  The latest published version is the version that is running and used by all operators. This is the version that will be used in runbooks where the action is referenced.

- **Archived**:

  Archived versions are previously used versions. They are not in production and cannot be used by any operations analyst. You can still run archived actions in "Test action".

Save and publish an action

Creating an action involves defining routines that run a task automatically.

After an action is published and used in production, you might improve it based on its success rate. The following actions are available:

- **Cancel:** closes the editor without saving any changes.
- **Save draft:** saves the updates that are added. The editor remains open. **Save draft** is useful if you are editing an action and you want to take a break and continue later. Or you want to save your changes as you go.
- **Save draft & close:** saves your changes and closes the editor. You use this option if you need to work on a different item.
- **Publish:** publishes a new version of the action and makes it available for use in runbooks and for operators.

Testing an action

As a Service Delivery Engineer, you can test an action before you use it within a runbook. This allows you to quickly develop and test actions.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Actions**.
4. Select an action in the table to display the **Preview** side panel for that action.
5. Click **Test** to start the test procedure for the action. Alternatively, click the options menu of the action and click the **Test** option.

   Note: You can select a different version of the automation if you don't want to test the latest version.
6. Follow the instructions on the **Test action** page.
7. Verify the result of the action.
8. You can test the action multiple times.
9. Parameter values can be modified between test runs of the action. To copy test parameter values to the action editor, click **Edit action**. In the dialog that's displayed, select the test parameter values that you want to bring to the action editor and click **Edit action**. Any changes made in the editor must still be saved.
10. When you are finished testing the action, click **Close**.

Running an action

As an Operator or as a Service Delivery Engineer, you can run the latest published action directly, without wrapping the action into a runbook first. If the action has never been published before, Service Delivery Engineers can also run the draft (although they would typically use **Test action** instead, as the test page is more tightly integrated with **Edit action** and generally allows you to test the latest draft).

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Actions**.
4. Select an action in the table to display the **Preview** side panel for that action.
5. Click **Run action** at the bottom of the side panel to start the run procedure for the action. Alternatively, click the options menu of the action and click the **Run** option.
6. Follow the instructions on the **Run action** page.
7. Verify the result of the action.
8. You can run the action multiple times. Parameter values can be modified between runs of the action.
9. When you are finished running the action, click **Complete**.
10. In the **Provide feedback** section, you can provide a rating and a comment. This information is intended to be evaluated by administrators, so they can easily identify those actions that need to be improved, and understand what exactly needs to be improved.
11. When you are finished with providing feedback, click **Close**.

Activities

In the **Activities** tab on the main page, you can find all actions that have been run, including actions in progress.

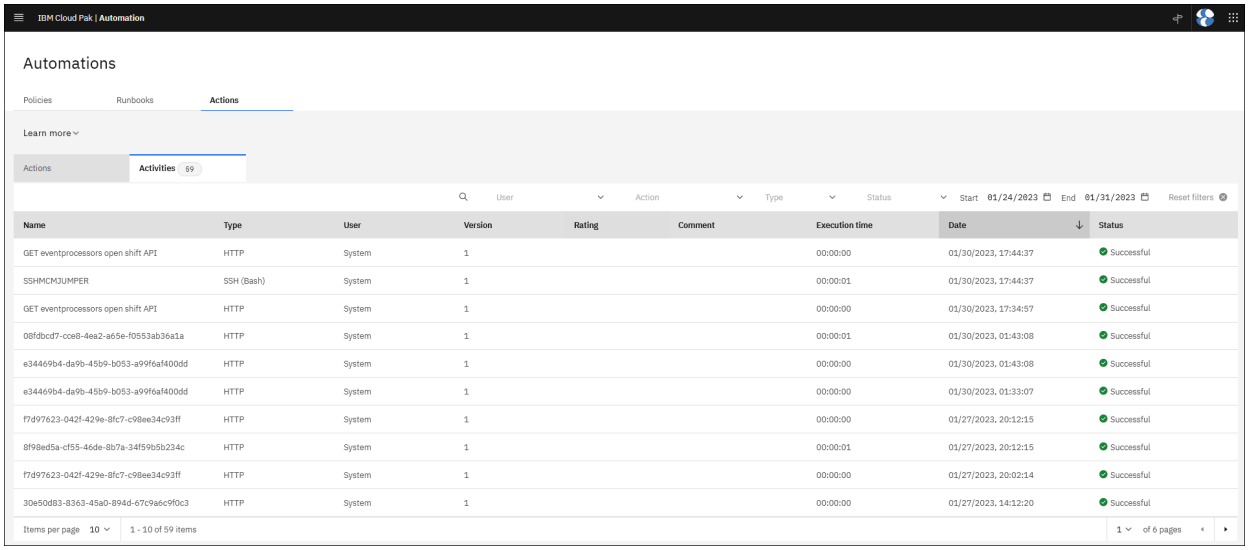


Figure. Activities UI

You can filter your list by the following criteria:

- **User**:

  User that ran the action.

- **Action name**:

  The action that was run.

- **Type**:

  Ansible, HTTP, SSH (Bash), SSH (PowerShell), Client-side, Predefined

- **Status**:

  Current status of the action activity. Valid statuses are "Executing", "Failed", "Successful", and "Unsuccessful".

The columns **Rating** and **Comment** display the detailed feedback information that has been provided by the operators who have run the action. Service Delivery Engineers can evaluate this feedback to identify those actions that should be improved (for example, as a combination of a high number of invocations and a low

average rating, as displayed in the **Actions** table, and the detailed feedback comments on the **Activities** page). Ideally, the feedback comments provided by the operators describe the pain points in sufficient detail, so the Service Delivery Engineer can improve the action efficiently.

Visualizing dashboards and reports

IBM Cloud Pak for AIOps includes support for analyzing and visualizing data through an integration with IBM Cognos Analytics. IBM Cognos Analytics integrates reporting, modeling, analysis, exploration, dashboards, and stories, so you can understand your organization's data, and make effective business decisions.

IBM Cognos Analytics is a proven, self-service, dependable analytics solution that provides scalability and analytics governance.

This integration provides you with tools to create custom IBM Cognos Analytics dashboards and reports.

With IBM Cognos Analytics dashboards, you can build visualizations of your IBM Cloud Pak for AIOps data, such as alert and topology data, to monitor your data at a glance. IBM Cognos Analytics includes a wide range of customizable visualization options to help you create these interactive dashboards. You can then use the exploration, analysis, and reporting capabilities within IBM Cognos Analytics to discover patterns and relationships and gain further insights into your IBM Cloud Pak for AIOps data.

With IBM Cognos Analytics reports, you can use the provided web-based report authoring tool to build sophisticated, multiple-page, multiple-query reports whenever you require, such as for creating alert resolution rates or impacted resource reports.

When you set up this integration, you can open IBM Cognos Analytics from the Cloud Pak for AIOps console.

Note: This functionality is not enabled for deployments of IBM Cloud Pak for AIOps on Linux®.

Before you begin

- IBM Cognos Analytics requires a database to act as a 'Content store' that contains IBM Cognos Analytics configuration and settings. You can use the same IBM Db2 instance that you use to sync your IBM Cloud Pak for AIOps data to, but in a separate database. For more information about using IBM Db2 as a content store in IBM Cognos Analytics, see the IBM Cognos Analytics documentation.

  The complete list of components that must be installed and configured for the integration between IBM Cloud Pak for AIOps and IBM Cognos Analytics is the following:

  1. Install IBM Db2.
  2. Install IBM Cognos Analytics (using IBM Db2 as a content store).
  3. Link IBM Cognos Analytics and IBM Cloud Pak for AIOps.
  4. Create the data flow.
  5. Create the data module.
  6. Create a dashboard.
- Entitlement to IBM Cloud Pak for AIOps provides limited entitlement to using IBM Cognos Analytics and IBM Db2 for IBM Cloud Pak for AIOps data sources only. Review the IBM Cloud Pak for AIOps license terms.

Installing and configuring the integration

You can configure an existing IBM Cognos Analytics environment with IBM Cloud Pak for AIOps. For more information, see Integrating IBM Cognos Analytics with IBM Cloud Pak for AIOps.

For information about new IBM Cognos Analytics installations, see the IBM Cognos Analytics documentation.

You must configure IBM Cloud Pak for AIOps to synchronize data with IBM Cognos Analytics through an IBM Db2 integration. For more information and instructions, see IBM Db2 integration.

Getting started with IBM Cognos Analytics

When you have the integration configured you can access and use IBM Cognos Analytics.

As a starting point, review the IBM Cognos Analytics documentation Getting started - User interface. This page and section of documentation provide basic information about navigating the IBM Cognos Analytics user interface, and creating and managing the application content.

The full set of documentation for using IBM Cognos Analytics is available in the IBM Cognos Analytics documentation. This documentation describes all features of IBM Cognos Analytics, including features that are not supported in IBM Cognos Analytics for IBM Cloud Pak for AIOps.

Opening IBM Cognos Analytics

You can open the IBM Cognos Analytics UI from the Cloud Pak for AIOps console main navigation menu by clicking **Visualize** > **Cognos**. The IBM Cognos Analytics UI opens in a new browser tab.

The IBM Cognos Analytics UI includes an AI **Assistant** (left pane of the UI), and a **Details** pane (right side of the UI), to help interpret the chart that you are viewing. The **Assistant** supports natural language exploration, while the **Details** pane provides system-generated plain-language information to augment the chart. The **Explore** feature reduces the guesswork that is associated with data exploration by finding the strongest relationships in your data. The relationships are displayed in the relationship diagram, and suggested starting points are offered for more data exploration.

Important: To access and use IBM Cognos Analytics to create and use dashboards and other visualization tools, you must have a user role with the following permissions:

- **Manage dashboards/reports**
- **View dashboards**

Supported features

The following features are supported in IBM Cognos Analytics for use with IBM Cloud Pak for AIOps data:

- Dashboards

  Note: Embedding visualization code is not supported.

- Stories
- Explorations
- Data modules
- Reporting

Working with dashboards

You can create a dashboard within IBM Cognos Analytics to build visualizations of your IBM Cloud Pak for AIOps data, such as alert and topology data. IBM Cognos Analytics dashboards help you to monitor events or activities at a glance by providing key insights and analysis about your data on one or more pages or screens. For more information, see Creating a dashboard. For more information about editing an existing dashboard, see Editing a dashboard.

For detailed instructions and information about working with dashboards, review the IBM Cognos Analytics documentation Dashboards. This page lists various topics to explore to learn more about using and understanding dashboards within IBM Cognos Analytics. For tutorial about the basics for creating a dashboard, see Cognos Analytics dashboards: a tutorial

For more information about building the visualizations on your dashboard, see the IBM Cognos Analytics documentation Visualizations. As part of building the visualizations, you can use IBM Cloud Pak for AIOps widgets within IBM Cognos Analytics.

You can also create custom visualizations to meet your specific business needs. For more information, see Custom Visualizations Developer. For a step-by-step guide for how to set up your development environment, create custom visualizations, and deploy and authorize the visualization, see Step-by-step guide for custom visualizations.

IBM Cloud Pak for AIOps widgets

IBM Cloud Pak for AIOps provides a default setup for creating 4 custom widgets. When you have configured the data flow to include IBM Cloud Pak for AIOps data in IBM Cognos Analytics, you can create these widgets. For more information, see:

- Alert list custom widget
- Topology viewer custom widget
- Incident list custom widget
- Button custom widget

These widgets can be combined with other IBM Cognos Analytics widgets in your IBM Cognos Analytics dashboards to create custom views for analyzing your data to meet your operational requirements and use cases.

Working with reports

You can create reports to meet your various needs, such as alert resolution rates or impacted resource reports. When you create a report, you are actually creating a report specification. The report specification defines the queries and prompts that are used to retrieve data and the layouts and styles used to present the data. For simplicity, the report specification is named by the same name as the report. For more information about creating and using reports, the IBM Cognos Analytics documentation Reporting. This page lists various topics to get started with creating reports based on your data within IBM Cognos Analytics.

Integrating IBM Cognos Analytics with IBM Cloud Pak for AIOps

BETA The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

Run the **`setupCognos.sh`** command to configure OpenID Connect (OIDC) authentication and single sign-on (SSO) between IBM Cloud Pak for AIOps and IBM Cognos Analytics.

Before you begin

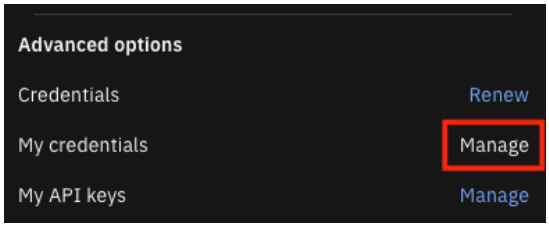
Important: You must have a deployment of IBM Cognos Analytics set up in order to integrate with IBM Cloud Pak for AIOps. For information about IBM Cognos Analytics installations, see the IBM Cognos Analytics documentation.

- Clone the repository containing the required scripts from github.com: aiops-cognos-analytics. For example:

  ```
  git clone git@github.com:IBM/cp4waiops-samples.git
  ```

- If required, install jq where you plan to run this script. To install jq, see Download jq.
- Obtain a IBM Cognos Analytics API key from an administrator.

- The IBM Cognos Analytics administrator can get the key from any existing namespace under **Profile and Settings** > **My API keys**.
- The IBM Cognos Analytics administrator must first **Renew** credentials if **My credentials - Manage** is disabled.



- The IBM Cognos Analytics user should have the administration capabilities to **Manage Namespaces** and **Users, Groups, and Roles**.

- From the command-line, authenticate with the IBM Cloud Pak for AIOps cluster as an admin user.
- Add the IBM Cloud Pak for AIOps cluster CA to IBM Cognos Analytics as documented here. If you're using the default Red Hat OpenShift ingress CA, find it by running the following command:

  ```
  kubectl get -n openshift-ingress-operator secrets router-ca -o jsonpath={.data."tls\.crt"} | base64 --decode
  ```

- Restart the IBM Cognos Analytics server.

Procedure

Run the following command:

```
./setupCognos.sh -u cognos_url [-k cognos_api_key] [-n cognos_namespace] [-g cognos_gateway] [-c aiops_client] [-r]
```

Where:

`-u cognos_url` (required) IBM Cognos Analytics server in the form of **`hostname:port`**.

`-k cognos_api_key` (optional) API key of the IBM Cognos Analytics user under which the setup will run. If no API key is specified, anonymous user is used.

`-n cognos_namespace` (optional) name of the IBM Cognos Analytics namespace that will be created for IBM Cloud Pak for AIOps users. If not specified, the IBM Cloud Pak for AIOps cluster name is used.

`-g cognos_gateway` (options) path of the IBM Cognos Analytics gateway. If not specified, **`/bi`** is used.

`-c aiops_client` (optional) name of the OIDC client that will be created in IBM Cloud Pak for AIOps. If not specified, the IBM Cloud Pak for AIOps cluster name is used.

`-r` (optional) remove the IBM Cognos Analytics namespace and OIDC client.

Configuring permissions to see IBM Cognos Analytics navigation entry in IBM Cloud Pak for AIOps

For any roles that require visibility of the IBM Cognos Analytics navigation entries, you can add the required permission through the **Access control** page. For more information about managing permissions, see Managing user access control and Roles and permissions.

The following permissions are required for such users:

- **Manage dashboards/reports**
- **View dashboards**

Notes

- SSO with Red Hat OpenShift is not supported.
- If the IBM Cloud Pak for AIOps cluster uses self-signed certificates, you must add a browser security exception before using these widgets in IBM Cognos Analytics. Refer to your browser documentation for instructions.
- IBM Cognos Analytics users cannot modify a namespace that they're using.
- For SSO to be effective, anonymous access must be disabled after successfully running this script.

Data flow configuration

BETA The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

After you install IBM Cognos® Analytics, you can configure Cloud Pak for AIOps to synchronize data with IBM Cognos® Analytics through an IBM Db2 integration. You can then create a policy to send alerts or incidents from Cloud Pak for AIOps to populate a dashboard or report.

The following components are required for alert data to flow from IBM Cloud Pak for AIOps to IBM Cognos Analytics.



Figure. Alert data flow from Cloud Pak for AIOps to IBM Cognos Analytics

Before you begin

You must have IBM Cloud Pak for AIOps 4.8.0, IBM Db2, and IBM Cognos Analytics installed before proceeding.

Complete the following steps to push Cloud Pak for AIOps alerts to IBM Db2 for use with IBM Cognos Analytics dashboards.

Procedure

1. Implement the database schemas so that the relevant reporting tables are in place. The schemas can be installed from: github.com: aiops-cognos-analytics/schemas.
2. Create a new IBM Db2 integration so that you can store IBM Cloud Pak for AIOps data in your IBM Db2 database. For more information, see IBM Db2 integration.
3. Connect IBM Cognos Analytics to your IBM Db2 database.
    1. In IBM Cognos Analytics, navigate to **Menu** > **+ New** > **Data server connections**.
    2. In **Select sources**, click **Data servers and schemas**, and select your database, followed by the schema.
    3. Click **New +** and select **Data server**.
    4. Provide a name, optional description, and select the relevant IBM Db2 connection type.
    5. Provide the connection details. In the **Authentication** section, you can create a **Signon** in which you provide the username and password for IBM Db2. Select **Use signon** in the drop-down list, click **Add signon** and then provide the details.
    6. Test the connection before saving.
    7. Select the newly created data server.
    8. Click the Action menu for the respective schema row in the table (the three dots at the end of the row).
    9. Select **Assets**.
    10. Click the Action menu for the asset row and select **Load metadata**. This loads the tables for the database.
    11. Click **Done** when complete.

    For more information, see Using data servers as data module sources.
4. Switch to the Cloud Pak for AIOps Policy UI and create policies to send alerts or incidents from IBM Cloud Pak for AIOps to populate a dashboard or report in IBM Cognos Analytics. For more information, see Populate a dashboard or report.
5. Ensure that you configure the policy to trigger when new alerts are added, or existing alerts are updated, so that the live alerts and the alerts in IBM® Db2® are in sync.

Creating a Data Module in IBM Cognos Analytics

BETA The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

When you have connected to IBM Db2, you can start using the database. The next step is to create a Data Module.

Procedure

1. In IBM Cognos Analytics, navigate to **Menu** > **+ New** > **Data Module**.
2. From **Select sources**, click **Data servers and schemas**, and select your database, followed by the schema.
3. Click **OK**.
4. Now, you can choose either to **Select tables** or **Discover related tables**. Discover related tables is a tool that is provided by IBM Cognos Analytics that uses NLP and AI to help suggest relationships between tables. For more information, see the IBM Cognos Analytics documentation. For this example, choose **Select tables** then **Next**.
5. Select all the tables and click **OK**.
6. You now have a Data Module containing the entire schema. If you view the **Relationships** tab, you can see that IBM Cognos Analytics has detected some relationships due to the structure of the schema.
7. If more data sources are needed for your dashboard requirements, you can include them into the data module and relationships that are formed between the Alerts/Incidents schema and others.
8. Save your Data Module after completing any changes.

Creating a dashboard

The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

BETA

You can create a dashboard within IBM Cognos Analytics to build visualizations of your IBM Cloud Pak for AIOps data, such as alert and topology data. IBM Cognos Analytics dashboards help you to monitor events or activities at a glance by providing key insights and analysis about your data on one or more pages or screens.

Building a dashboard involves dragging data sources, visualizations, and widgets to the main canvas. You can also customize the appearance of the visualizations to make the data more visually appealing.

With a dashboard created, you can then use the analysis and reporting capabilities within IBM Cognos Analytics to discover patterns and relationships and gain further insights into your IBM Cloud Pak for AIOps data.

Before you begin

To create a dashboard within IBM Cognos Analytics, you must have a role with the following permission:

- **Manage dashboards/reports**

You can download example dashboards for Cloud Pak for AIOps using IBM Cognos Analytics from github.com: aiops-cognos-analytics/examples.

Procedure

1. From the main navigation menu, click **New** .
2. Click **Dashboard**.
3. Select a template to use and click **Create**.

   Predefined templates contain grid lines for easy arrangement and alignment of visualizations and other elements. For example, you can create an infographic to convey information with pictures on one page that you scroll through.

   The dashboard displays a blank canvas that is ready for you to visualize your data. If this is your first time using this dashboard, the dashboard also processes your data sources. The **Selected sources** pane displays the progress of the data sources as a percentage until the processing is complete. Some functions in the dashboard are only available after the processing is complete.
4. Ensure that the data sources that you want to use are displayed in the **Selected sources** pane.

   If no data source is displayed, click **Select a source** to find and chose your data source. If you want to add another data source, click the **Add a source** Add icon icon and select your data source.
5. Work with charts on the dashboard. You can add visualizations to the dashboards in one of the following ways:
   - Have Cognos Dashboards choose the visualization for you. Cognos Dashboards provides a number of recommended visualizations based on the data that you are working with.

     To have Cognos Dashboards choose the visualizations for you, drag a measure or column on to the canvas. You can change to another visualization from either Recommended or from More.
   - Choose the visualizations yourself. You can customize your dashboard by choosing your own visualizations:
     - Click the **Visualizations** icon to display the available charts for your dashboard.
     - Click the chart to insert it automatically to the canvas. You can also drag the chart and drop it to the canvas. After you insert the chart into the canvas, the dashboard automatically selects the **Sources** icon for you to select the data for your chart.
     - To move the chart on the canvas, click and drag the corresponding **Move widget** icon
     - From the Sources section, select your data source, then browse to a data column.
     - Drag the data columns and drop them to an available section of the chart or graph. The chart uses the entire set of data from the data column.
     - To filter the data column values, click the **Filters** Filters icon icon in the toolbar, then drag the data column and drop it to the **All tabs** icon or the **This tab** icon, then specify the filter settings. If you have multiple tabs, the filter settings that you specify in All tabs apply to all your tabs.
6. Customize the appearance of the dashboard.
   1. Click the Widgets icon to display the list of available widgets and icons.
   2. Click a widget insert it automatically to the canvas. You can also drag the widget and drop it to the canvas.

      For more information about IBM Cloud Pak for AIOps widgets, see IBM Cloud Pak for AIOps widgets within IBM Cognos Analytics.
   3. To move the widget on the canvas, click and drag the corresponding **Move widget** icon.
7. Click **Save** to save your changes and create the dashboard.

   Note: Even if you do not manually save your dashboard, IBM Cognos Analytics automatically saves any changes that you made every 5 seconds.

Sharing the dashboard

You can share your dashboard with other users by copying the dashboard's URL. From the toolbar, click the **Share** icon. Chose your preferred method for sharing the dashboard, and then share the link, embedded code, or PDF. The user that you share the dashboard with can only edit the dashboard if they have the required permissions.

Editing the dashboard

If you return to the dashboard, the dashboard is in read-only mode if the **Edit** icon is gray . If you want to edit the dashboard again, enable the edit mode by clicking the **Edit** icon until it is green with a checkmark .

IBM Cloud Pak for AIOps widgets within IBM Cognos Analytics

BETA The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

As part of the integration with IBM Cognos Analytics, IBM Cloud Pak for AIOps provides a default setup for creating 4 custom widgets.

When you have configured the data flow to include IBM Cloud Pak for AIOps data in IBM Cognos Analytics, you can create these widgets. For more information, see:

- Alert list custom widget
- Topology viewer custom widget
- Incident list custom widget
- Button custom widget

These widgets can be combined with other IBM Cognos Analytics widgets in your IBM Cognos Analytics dashboards to create custom views for analyzing your data to meet your operational requirements and use cases.

Alert list custom widget

BETA The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

The Cloud Pak for AIOps Alert list custom widget can use data selection events from IBM Cognos Analytics visualizations. For example, a bar chart visualization represents severity data. If a high severity bar is selected in the visualization, the Alert list custom widget corresponds and shows high severity alerts only. The assumption is that the related data source column and Alerts property names are matched. Configure the column mapping if that is not the case.

Procedure

1. Download the IBM Cloud Pak for AIOps widgets and other artifacts that are needed from: github.com: aiops-cognos-analytics/widgets.
2. Create a zipped file out of the entire **widgets** folder.

   On macOS: right-click the widgets folder and select **Compress "widgets"**.

   On Windows: right-click the widgets folder and select **Send to** > **Compressed (zipped) folder**.
3. Log in to IBM Cognos Analytics. The base URL is `https://<CLUSTER_NAME>/cognosanalytics/aiops/bi/?perspective=home`.
4. From the main navigation menu, click **Manage** > **Customization** > **Extension**.

   Click the Upload extension icon and navigate to your compressed file.

   The Cloud Pak for AIOps Alert list custom widget is now available to use in new dashboards.
5. From the main navigation menu, click **New** > **Dashboard** > **Create**.
6. Click Custom widgets.
7. Select the **AIOps Alert list** and drag it onto the canvas.
8. Click **Properties** to configure the Cloud Pak for AIOps Alert list in the properties side panel:
   - **Filter**: Sets the default Filter.
   - **View**: Sets the default View.
   - **Show toolbar**: Enables or disables the Alert list toolbar.
   - **Show details**: Enables or disables the Alert list details side panel.
   - **Column mapping**: Allows mapping of IBM Cognos Analytics data source columns to Alert properties. The mapping is defined through JSON format, for example:

     ```
     {"SEVERITY": "severity"}
     ```

The value on the left represents the IBM Cognos Analytics data source column. The value on the right represents the Alert properties.

Note: Identifier strings for data source columns in IBM Cognos Analytics cannot consist entirely of a reserved keyword. Therefore, some identifiers are appended with an underscore ("_") which must be added to the column name for mapping to work. For example, the mapping of the STATE column would be as shown:

```
{"STATE_":"state"}
```

Related links

- Creating a dashboard
- Editing a dashboard

Topology viewer custom widget

BETA The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

Procedure

Skip steps 1 - 4 if you have already completed them for the Alert list custom widget.

1. Download the IBM Cloud Pak for AIOps widgets and other artifacts that are needed from: github.com: aiops-cognos-analytics/widgets.
2. Create a zipped file out of the entire **widgets** folder.

   On macOS: right-click the widgets folder and select **Compress "widgets"**.

   On Windows: right-click the widgets folder and select **Send to** > **Compressed (zipped) folder**.
3. Log in to IBM Cognos Analytics. The base URL is `https://<CLUSTER_NAME>/cognosanalytics/aiops/bi/?perspective=home`.
4. From the main navigation menu, click **Manage** > **Customization** > **Extension**.

   Click the Upload extension icon and navigate to your compressed file.

   The Cloud Pak for AIOps Topology viewer custom widget is now available to use in new dashboards.
5. From the main navigation menu, click **New** > **Dashboard** > **Create**.
6. Click Custom widgets.
7. Select **AIOps Topology viewer** and drag it onto the canvas.
8. If you have the alert list viewer open, double-clicking an alert with topology generates the topology map for that alert.
9. Click **Properties** to configure the Cloud Pak for AIOps Topology viewer in the properties side panel:
   - **Hops**: Sets the number of hops in the topology map.
   - **Show toolbar**: Enables or disables the topology viewer toolbar.
   - **Show search**: Enables or disables the topology viewer search bar.
   - **ResourceId/StatusId**:
     - If you have a custom column name that stores the resource/status ID for the topology viewer, enter the name in one of these boxes.
     - The topology viewer looks for this column name in the IBM Cognos Analytics data to retrieve the resource ID.
     - Otherwise, topology viewer will only work with the Alert list viewer.

Related links

- Creating a dashboard
- Editing a dashboard

Incident list custom widget

BETA The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

The Cloud Pak for AIOps Incident list custom widget can use data selection events from IBM Cognos Analytics visualizations. For example, a bar chart visualization represents priority data. If a high priority bar is selected in the visualization, the Incident list custom widget corresponds and shows high priority incidents only (if the related data source column and Incident property names are matched in the column mapping).

You can use the Incident list widget in conjunction with an Alert list widget. To see the alerts related to a specific incident, select the incident in the widget and the alerts in the Alert list widget are filtered down to the alerts for that incident.

Procedure

Skip steps 1 - 4 if you have already completed them for another widget.

1. Download the IBM Cloud Pak for AIOps widgets and other artifacts that are needed from: github.com: aiops-cognos-analytics/widgets.
2. Create a zipped file out of the entire **widgets** folder.

   On macOS: right-click the widgets folder and select **Compress "widgets"**.

   On Windows: right-click the widgets folder and select **Send to** > **Compressed (zipped) folder**.
3. Log in to IBM Cognos Analytics. The base URL is `https://<CLUSTER_NAME>/cognosanalytics/aiops/bi/?perspective=home`.
4. From the main navigation menu, click **Manage** > **Customization** > **Extension**.

   Click the Upload extension icon and navigate to your compressed file.

   The Cloud Pak for AIOps Incident list custom widget is now available to use in new dashboards.
5. From the main navigation menu, click **New** > **Dashboard** > **Create**.
6. Click Custom widgets.
7. Select the **AIOps Incident list** and drag it onto the canvas.
8. Click **Properties** to configure the Cloud Pak for AIOps Incident list in the properties side panel:
   - **Filter**: Sets the default Filter.
   - **View**: Sets the default View.
   - **Show toolbar**: Enables or disables the Incident list toolbar.
   - **Show details**: Enables or disables the Incident list details side panel.
   - **Column mapping**: Allows mapping of IBM Cognos Analytics data source columns to Incident properties. The mapping is defined through JSON format, for example:

     ```
     {"Priority": "priority"}
     ```

     The value on the left represents the IBM Cognos Analytics data source column. The value on the right represents the Incident properties.

Related links

- Creating a dashboard
- Editing a dashboard

Button custom widget

BETA The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

The Cloud Pak for AIOps Button custom widget allows you to create a button with an associated alert filter. Customize the button properties to your requirements and set the desired target. The button can update an alert list in your existing dashboard, or launch an alert list in a new browser tab.

Procedure

Skip steps 1 - 4 if you have already completed them for another widget.

1. Download the IBM Cloud Pak for AIOps widgets and other artifacts that are needed from: github.com: aiops-cognos-analytics/widgets.
2. Create a zipped file out of the entire **widgets** folder.

   On macOS: right-click the widgets folder and select **Compress "widgets"**.

   On Windows: right-click the widgets folder and select **Send to** > **Compressed (zipped) folder**.
3. Log in to IBM Cognos Analytics. The base URL is `https://<CLUSTER_NAME>/cognosanalytics/aiops/bi/?perspective=home`.
4. From the main navigation menu, click **Manage** > **Customization** > **Extension**.

   Click the Upload extension icon and navigate to your compressed file.

   The Cloud Pak for AIOps Button custom widget is now available to use in new dashboards.
5. From the main navigation menu, click **New** > **Dashboard** > **Create**.
6. Click Custom widgets.
7. Select the **AIOps Button** and drag it onto the canvas.

8. Click **Properties** to configure the Cloud Pak for AIOps Button in the properties side panel:
    - **Filter**: Select from the list of filters to associate a filter to the button.
    - **Shape**: Choose between Rectangular or Round for the button shape.
    - **Button label**: Select from Filter name, Alert count, or Both for the button label.
    - **Target**: The action to be taken when the button is clicked. It can update the alert list in your existing dashboard, or launch an alert list in a new browser tab.
    - **Refresh interval**: Auto-refresh interval in seconds.

Related links

- Creating a dashboard
- Editing a dashboard

Example configurations

BETA The installation and integration of IBM Cognos® Analytics is available only for technology preview, and is not for production usage.

You can quickly appreciate the value of your IBM Cognos Analytics integration with some example configurations that are ready for immediate use.

Before you begin

This requires IBM Cognos Analytics version 11.2 or higher.

Download the Example dashboards zip file.

Procedure

Download installation video demo

1. **Find the location of IBM Cognos Analytics extensions**.

   For Cognos running within an Red Hat OpenShift cluster, locate the pod that contains **/data/deployment/**. Search for **ca-** to find Cognos pods.

   ```
   kubectl get pods | grep ca-

   ca1721852474940188-artifacts-bc9d68dc4-gzk6p 1/1 Running 0 21d
   ca1721852474940188-ca-cpd-cm-78676c5d8f-gctsj 1/1 Running 0 21d
   ca1721852474940188-ca-cpd-reporting-94bd774b8-d2thf 2/2 Running 0 21d
   ca1721852474940188-ca-cpd-rest-bd5667547-m7dxq 2/2 Running 0 21d
   ca1721852474940188-ca-cpd-smarts-7bbdd6ff97-vnwvl 2/2 Running 0 21d
   ibm-ca-operator-controller-manager-7b7d67667b-tks6b 1/1 Running 6 (4h36m ago) 21d
   ```

   Check whether the pod contains **/data/deployment/**.

   ```
   kubectl exec <pod_name> -- find / -type d -name 'deployment' 2>/dev/null

   /data/deployment
   /deployment
   ```

   For stand-alone IBM Cognos Analytics, this file is on the Cognos server, typically under **/opt/ibm/cognos/analytics/deployment**.

2. **Install the example .zip file.**

   For IBM Cognos Analytics running within a Red Hat OpenShift cluster:

   ```
   kubectl cp "./<example>.zip" "<pod_name>:/data/deployment/<example>.zip"
   ```

   For a stand-alone IBM Cognos Analytics server:

   ```
   cp ./<example>.zip /opt/ibm/cognos/analytics/deployment/<example>.zip
   ```

3. Open the Administration console in the IBM Cognos Analytics UI.
4. Go to **Configuration** > **Content Administration** and then select the new import button.
5. Select the **Cloud_Pak_for_AIOps_examples** package and click **Next**.
6. You are be prompted for a passcode. Enter **examples**, and click **OK**.
7. Optionally, update the name and description and click **Next**.
8. Select the checkbox next to the **Cloud_Pak_for_AIOps_examples** folder, and click **Next**.
9. Click **Next** through the remaining steps, updating defaults as required.
10. Click **Finish** > **Run** > **OK**. The examples package is displayed in the content list.

11. The incident dashboard example requires a custom funnel extension. Download and install this extension from the **Manage** > **Customization** > **Custom Visuals** menu option on the IBM Cognos Analytics home page.
12. The examples folder is displayed under **Content** > **Team content** from the IBM Cognos Analytics home page.
    - If you don't immediately see the examples, you might need to wait a minute and then reload the **Team content** page.
13. Example dashboards use mock data by default. Complete the following steps to use your own live data with the examples:
    1. Create an IBM Db2 integration.
    2. Relink the example Db2 data module to use your database server connection from the previous step.
    3. Relink the dashboard to use the Db2 data module.

Troubleshooting

1. Color of severity is off: Install the Custom color extension, which provides a richer color palette.
2. The funnel graph is blank: Install the Custom funnel extension.
3. Missing Data in the Db2 module: Relink the provided Db2 data source in the **Data** subfolder to your own IBM Db2 server connection. Or try loading the page in a private browser window.
4. Warning "One or more data caches could not be loaded." This message can be dismissed. It can occur after relinking data sources to inform you that the data from the new source is not found in the local cache.

Related links

Exporting just the dashboard without a data source

Exporting reports or deployment packages

Cloud Pak administration

After you install IBM Cloud Pak® for AIOps, you can access several options for administering and customizing the IBM Cloud Pak and IBM Cloud Pak® foundational services.

- UI console
- Managing user access control
- Secret rotation
- Administering data storage
- Restricting the number of alerts returned by the datalayer
- Updating usage data collection preferences
- Clearing out topology data without reinstalling
- Setting the idle session timeout
- Configuring probable cause

For platform specific administration topics, see also:

- Administering IBM Cloud Pak for AIOps on OpenShift
- Administering IBM Cloud Pak for AIOps on Linux

Administering the UI console

Learn how to access the IBM Cloud Pak for AIOps console, and how to use some of the console components.

- Supported browsers
- Accessing the console
- Customizing the platform

Supported browsers

You can access the UI console from Mozilla Firefox, Google Chrome, Microsoft™ Edge, and Safari.

Table. Supported browsers

Platform	Supported browsers
Microsoft Windows	• Microsoft Edge - latest version • Mozilla Firefox - latest version for Windows • Google Chrome - latest version for Windows
Linux®	• Mozilla Firefox - latest version for Linux • Google Chrome - latest version for Linux

Platform	Supported browsers
macOS	• Mozilla Firefox - latest version for Mac • Google Chrome - latest version for Mac • macOS Safari - latest version

Accessing your cluster by using the console

You can access your console from a supported web browser.

Procedure

1. In a browser, go to the URL **<https://<cluster_address>**. See the following definitions:

   <cluster_address> is the IBM Cloud Pak console route. You can get the IBM Cloud Pak console route by running the following command:

   ```
   oc get route -n <your-foundational-services-namespace> cp-console -o jsonpath='{.spec.host}' && echo
   ```

   Following is a sample output:

   ```
   cp-console-cp4aiops.apps.mycluster.mydomain.com
   ```

   Based on the example output, your console URL would be **https://cp-console-cp4aiops.apps.mycluster.mydomain.com** .

   The username and password are defined in the **config.yaml** file.

2. Access your cluster from a web browser. For a list of supported browsers, see Supported browsers.

3. Log in to the console. You might have different authentication types available for you to select from to authenticate your user credentials.

   The default username to access the console is **cpadmin**.

   You can get the password for the **cpadmin** username by running the following command:

   ```
   oc -n <your-foundational-services-namespace> get secret platform-auth-idp-credentials -o
   jsonpath='{.data.admin_password}' | base64 -d && echo
   ```

   Following is a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the example output, you would use **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA** as the password.

   You can change the default password at any time. For more information, see Changing the administrator password.

   Note: Any user with access to your IBM Cloud Pak foundational services project (namespace) can retrieve this password since it is stored in a secret in that namespace. To minimize password exposure, allow limited users to access your IBM Cloud Pak foundational services namespace.

Customizing the platform

As an administrator, you can customize the IBM Cloud Pak for AIOps console to meet the needs of your enterprise.

Complete the appropriate types of customization for your environment:

- Enabling and disabling tours

  As an administrator, you can customize the Cloud Pak for AIOps console to enable or disable the Tours feature for all users on the platform. For instance, if you do not need the tours to be available to users, you can disable the feature. If you, or another administrator, previously disabled the feature and you want the tours available for your users, you can enable the feature.

- Customizing the Platform UI (Zen) route

  You can set or change parameters to control the Platform UI (Zen) route. This change can be done before or after installation to change the route.

- Customizing the branding of the Cloud Pak for AIOps console

  As an administrator, you can customize the Cloud Pak for AIOps console with your company's branding. You can customize the name that is displayed in the Cloud Pak for AIOps console and use a custom logo on your home page.

- Customizing the home page

  As an administrator, you can customize the content that users can see on the home page. You can specify which cards are enabled and which links are displayed in the **Support** section of the home page.

Enabling and disabling guided tours

As an administrator, you can customize the IBM Cloud Pak for AIOps console to enable or disable the *Tours* feature for all users on the platform.

If you do not need the tours to be available to users, you can disable the feature. If you, or another administrator, previously disabled the feature and you want the tours available for your users, you can enable the feature.

Tours are included in the Cloud Pak for AIOps console to help users learn about the IBM Cloud Pak for AIOps platform and features. Some tours introduce high-level concepts, while others offer step-by-step guidance. More tours can be accessed by a user at any time from the Cloud Pak for AIOps console page that is assoicated with the feature to get a better understanding of how to get started with different tasks and components. Users can access available tours by clicking the **Tours** icon in the Cloud Pak for AIOps console toolbar, and clicking their preferred tour.

First-time users are also presented with a guided tour upon login to the Cloud Pak for AIOps console. This tour introduces them to the various home page components, shows how to access tools and capabilities, and where to complete key tasks across the platform.

Before you begin

Required permissions: To customize the home page, you must have one of the following permissions:

- *Administer platform*

Procedure

1. Log in to the IBM Cloud Pak for AIOps console.
2. From the navigation menu, select **Administration** > **Customizations**.
3. Click **Tours**.
4. Change the setting of the **Enabled**/**Disabled** toggle to either enable or disable the tours feature and to hide the **Tours** icon in the Cloud Pak for AIOps console toolbar.

   Your selection is saved and applied automatically.

Disabling auto complete on the login page

In Common Web UI version 4.x.x, there is a new option to disable auto complete on the login page.

Note: The disable option is not supported on Firefox since it does not honor the autocomplete off setting in HTML.

Procedure

To disable auto complete on the login page, complete the following steps:

1. Edit the configmap named **`common-web-ui-config`** in the common services namespace by running:

   ```
   oc -n <common services ns> edit cm common-web-ui-config
   ```

2. In the **`data`** section, add the following option:

   ```
   disable-autocomplete: "true"
   ```

3. Save the changes and auto complete will be disabled when you access the login page.

Customizing the Platform UI (Zen) Route

Overview

The ZenService CR spec contains a few parameters that control the Zen route. This document describes what those parameters are and how to set them. At any time before or after installation, you can modify the ZenService CR to trigger a Zen operator reconcile, so that the route can be changed even after installation.

Reencrypt

When the Zen route is set to reencrypt, a TLS connection that is established from a client terminates at the Red Hat® OpenShift® Container Platform router. A second TLS connection is established from the router to the ibm-nginx pods. It is this second hop that requires packets to be “reencrypted”, which is where this routing type gets its name.

The following parameters show what must be set in the ZenService custom resource (CR) to configure a reencrypt route. After changing the values in the ZenService CR, in order for them to take effect, a full Zen operator reconcile will take place. To know whether the reconcile is complete, check the **`Status`** → **`Progress`** field in the ZenService CR object. The **`Status`** → **`Progress`** value will show as 100% after reconcile is complete.

```
zenDefaultIngressReencrypt: true
zenCustomRoute:
  route_host: <hostname> # OPTIONAL
```

```
  route_reencrypt: true
  route_secret: anything-but-external-tls-secret
```

The ZenService CR parameters relevant to a reencrypt connection are defined as follows:

- **zenDefaultIngressReencrypt**: If present, this value must be set to `true`.
- **route_host**: This value must reflect the external domain that a user uses to interact with the cluster. The domain specified here must match the domain that is presented by the Certificate in the `tls.crt` value of `route_secret`.
- **route_reencrypt**: Set to `true` for reencrypt.
- **route_secret** : This new value of `route_secret`, should indicate the name of a Kubernetes Secret that will hold the `tls.crt`, `tls.key`, and `ca.crt` values necessary for customizing the certificate that is provided by a reencrypt route. `route_secret` is ONLY valid for a reencrypt route type.

Important: Key names for `route_secret` must use `tls.crt`, `tls.key`, and `ca.crt`. If any other names are used (for example, `cert.crt`, `cert.key`), the route does not apply the values. If any of the three values of `tls.crt`, `tls.key` or `ca.crt` are missing, the ibm-zen-operator ignores the secret, use the default ingress-controller certificate, and issue a WARNING.

Example `create secret` command:

```
oc create secret generic cpd-tls-secret --n <aiops_project>
--from-file=ca.crt=./<ca-certificate-file-name>
--from-file=tls.crt=./<tls-certificate-file-name>
--from-file=tls.key=./<tls-key-file-name>
```

Rotating the Reencrypt Certificate

To rotate the certificate that is used in a reencrypt route, the easiest way to force a reconcile of the Zen operator to accept the new Certificate is to create a new `route_secret` and update that in the ZenService CR. The updated ZenService CR triggers a Zen operator reconcile, which updates the route.

If the same secret is reused after the data values have been updated, there would be no noticeable change to the ZenService CR spec, which means that the Zen operator will not know to reconcile. If the same secret is being used, the Zen operator can still be triggered to reconcile by making changes to the ZenService CR for reconciliation.

Example YAML:

1. Create a reecrypt route with a custom hostname

```
spec:
zenCustomRoute:
  route_host: cloudpak.mycompany.com
  route_reencrypt: true
  route_secret: cpd-tls-secret
```

2. Create a reencrypt route with a custom hostname but accept the default Ingress Controller certificate

Create a reencrypt route with a custom hostname, but accepting the default Ingress Controller certificate (that is, no custom route Certificate provided). This will only work if the Ingress Controller certificate has already been setup to protect the “cloudpak.mycompany.com” domain.

```
spec:
zenCustomRoute:
  route_host: cloudpak.mycompany.com
  route_reencrypt: true
```

Updating default routes to custom hostname

If needed, you can change the default hostname for IBM Cloud Pak® for AIOps and Managed services.

Use the following steps to use a custom hostname for both Cloud Pak for AIOps and Managed services.

1. Follow the DNS update steps in Customizing the Platform UI (Zen) route.
2. Log in to Red Hat OpenShift Container Platform online console.
3. Click **ConfigMap** and click on **router-config** ConfigMap. If the value of the **ZEN_Host** shows as default value, then update this field to use a custom hostname.
4. Follow the DNS update steps for `cp-console` hostname in Updating cp-console hostname and TLS secret by using a configmap.
5. Click **ConfigMap** and click on **oauth-client-map** ConfigMap. Verify the values for **CLUSTER_CA_DOMAIN** and **MASTER_IP** fields and update the fields with the correct domain.
6. Run the following three commands to restart the platform pods:

   ```
   oc delete pod -l app=platform-auth-service
   ```

   ```
   oc delete pod -l app=platform-identity-management
   ```

   ```
   oc delete pod -l app=platform-identity-provider
   ```

   Verify that the pods starting with the name **platform** are up and running.

7. Run the following command to restart Zen base pods:

   ```
   oc delete pod -l app=0020-zen-base
   ```

   Verify that the pod names starting with **zen** are up and running.

8. Run the following command and verify that the progress under Status shows as 100%.

   ```
   oc get zenservice iaf-zen-cpdservice -o yaml
   ```

9. Run the command to restart Managed services pods.

   ```
   oc delete pod -l release=cam
   ```

   Verify that the pod names starting with **cam** are up and running.

10. When all the pods are up and running, navigate to **Library** and **Instances** pages in Infrastructure Automation UI console.

11. If any issue persists, try the following steps:

    1. Check the `cam-tenant-api` pod logs.
    2. Search the logs for the term **apikey** in configboot. A valid key is displayed. If the key is empty restart the **cam-tenant-api** pod.

12. Go to **Service catalog** page. If the page shows an error, restart the following two pods:

    ```
    oc delete pod -l app=cam-service-library-ui-api
    ```

    ```
    oc delete pod -l app=cam-service-library-ui
    ```

13. If the issue still persists, contact IBM Support.

(Optional) Refresh ChatOps pods

If the default cpd route is updated, then ChatOps does not work until the new URL is passed into the deployments. Run the following 3 commands to trigger a rolling restart of the pods:

```
oc set env deployment/$(oc get deploy -l app.kubernetes.io/component=chatops-orchestrator -o
jsonpath='{.items[*].metadata.name }') ZEN_URL=<new_url>
oc set env deployment/$(oc get deploy -l app.kubernetes.io/component=chatops-slack-integrator -o
jsonpath='{.items[*].metadata.name }') ZEN_URL=<new_url>
oc set env deployment/$(oc get deploy -l app.kubernetes.io/component=chatops-teams-integrator -o
jsonpath='{.items[*].metadata.name }') ZEN_URL=<new_url>
```

Where `<new_url>` is the updated URL for the default cpd route.

Customizing the console branding

As an administrator, you can customize the console with your company's branding. You can customize the name that is displayed in the console, and use a custom logo, browser icon, home page and login image.

Before you begin

Required permissions

To customize the branding of the console, you must have one of the following permissions:

- **Administer platform**

About this task

Product name

If you change the product name in the console, the custom name is used throughout the console. For example, the custom name is used on the home page and on the banner of every page in the console.

Logo

If you upload a logo, the logo is displayed in the welcome section of the home page.

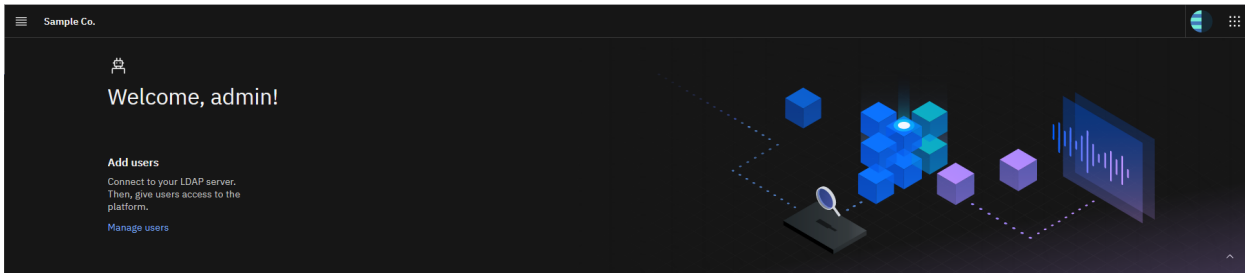


Figure 1. Console with a custom name and logo

Browser icon

If you upload a browser icon, the icon is displayed on the console tab in the browser.

Home page image

If you upload a home page image, the image is displayed on the console home page.

Login image

If you upload a login image, the image is displayed upon login.

Procedure

1. Log in to the console.
2. From the navigation menu, select **Administration** > **Customizations**.
3. Click **Branding**.
4. To use a custom name, select the **Custom name** option under **Product name** and enter the name that you want to use in the console.
   - The name must be 40 characters or less.
   - The product name cannot include the following characters: `<`, `>`, `/`, `\`, `'`, `(`, `)`, or `-`.
5. To use a custom logo, select **Use your own logo** and drop a JPG or PNG file or browse for a file to upload as the home page logo.
   - You can supply a JPG or PNG file.
   - The image is re-sized to fit in a 512 px × 512 px space.
   - The image must be 500 KB or less.
6. To use a custom browser icon, select **Use your own icon** and drop a JPG or PNG file or browse for a file to upload as the browser icon.
   - You can supply a JPG or PNG file.
   - The image is re-sized to fit in a 512 px × 512 px space.
   - The image must be 500 KB or less.
7. To use a custom home page image, select **Use your own image** and drop a JPG or PNG file or browse for a file to upload as the home page image.
   - You can supply a JPG or PNG file.
   - The image must be less than 500 KB for best results, though the limit for an image is 20 MB.
8. To use a custom login image, select **Use your own image** and drop a JPG or PNG file or browse for a file to upload as the login image.
   - You can supply a JPG or PNG file.
   - The image must be less than 500 KB for best results, though the limit for an image is 20 MB.
9. Click **Apply**.

If you want to remove your customizations, click **Restore defaults**.

Customizing the home page

As an administrator, you can customize the content that users can see on the home page. You can specify which cards are enabled and which links are displayed in the **Support** section of the home page.

Before you begin

Required permissions

To customize the home page, you must have one of the following permissions:

- **Administer platform**

About this task

Cards

Cards give users easy access to recently used items and an overview of important changes and alerts.

The list of available cards is determined by the services that are installed on the platform. You can also add custom cards to the home page with the custom cards API. You can prevent certain cards from being displayed on the home page by disabling them.

Each user might see only a subset of the cards based on their permissions and the services that they have access to, so consider how disabling a card will affect users with different roles.

Support links

Support links give users easy access to content that they can use to learn about the platform or to get help.

The default support links are accessible to all users on the platform. However, you can add custom links that are available to specific users based on their permissions.

When you save a change to the home page configuration, the change is immediately applied to the home page. However, if a user is currently logged in to the console, the user must refresh the home page to see the changes.

Procedure

1. Log in to the console.
2. From the navigation menu, select **Administration** > **Customizations**.
3. Click **Home page**.
4. To customize the cards that are displayed on the home page:
    1. Open the **Cards** tab.
    2. Select a card from the list of cards and decide whether you want to display the card on the home page. If you want to suppress the card, toggle the card to **Disabled** and click **Save**.

    Restriction: You cannot add custom cards from the console. You must use the custom cards API.
5. To customize the list of links that are displayed in the **Support** section of the home page:
    1. Open the **Support** tab.
    2. Select a link from the list of links and decide whether you want to display the link on the home page. If you want to suppress the link, toggle the link to **Disabled** and click **Save**.
6. To reorder the list of links that are displayed on in the **Support** section of the home page:
    1. Click the **Reorder** icon .
    2. Drag the links to reorder them.
    3. Click **Save**.
7. To add a link to the list of links that are displayed on in the **Support** section of the home page:
    1. Open the **Support** tab.
    2. Click the **Add custom link** icon .
    3. Specify the label for the link.
    4. Specify the icon to display next to the link.

        You must specify an icon from the Carbon Icon Library.

        By default, when you copy an icon code from the Carbon Design Library, the icon code specifies the 32 px version of the icon. For example, if you copy the code for the Home icon, the following snippet is copied to your clipboard: `<Home32 />`. However, the console uses the 16 px version of the icon. When you paste the snippet into the console:

        1. Remove the angle brackets (`< >`) and the forward slash (`/`).
        2. Change 32 to 16
    5. Specify the URL that you want users to access.
    6. Specify who can see this link:
        - If you select **All users**, all platform users can see the link.
        - If you select **Users with any of the selected permissions**, only users with one or more of the selected permissions can see the link.
    7. Click **Save**.

Supported languages

IBM Cloud Pak Administration panel and IBM Cloud Pak for AIOps console both support multiple languages.

The following table lists the languages that are supported with IBM Cloud Pak Administration panel and IBM Cloud Pak for AIOps console.

Table. Supported languages

Language	Language code
English	"en"
French	"fr"
German	"de"
Spanish	"es"
Italian	"it"
Japanese	"ja"

Language	Language code
Brazilian Portuguese	"pt"
Simplified Chinese	"zh-Hans"
Traditional Chinese	"zh-Hant"

Managing user access control

As an administrator for Role Based Access Control (RBAC) in IBM Cloud Pak for AIOps, you are responsible for determining and implementing the best approach for authenticating and managing users.

You use the access control features within IBM Cloud Pak for AIOps to add users and user groups, create and assign roles, and create user group profiles to fine-tune access control. For instance, you can use profiles to control the visibility of management artifact properties and content that is visible to users. Profiles and associated filters and policies can be used to control the visibility of alerts, incidents, applications, resource groups and resources.

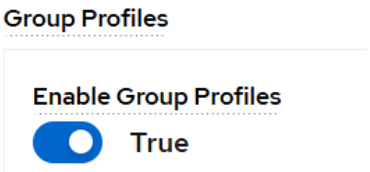
To add to the platform, enable their access to features, and fine-tune their access, you need to complete the following tasks:

1. Configure identity provider connections so that your users can be added to IBM Cloud Pak for AIOps.
2. Add your users to the platform.
3. (Optional) create, and include your users within, user groups.
4. Create and assign roles and permissions to your users and user groups.
5. (Optional) Define any redaction policies or restriction filters for accessing features or data for specific user groups.
6. (Optional) Create profiles to apply redaction policies, restriction filters, and other conditions to your user groups.

Prequisites

As an administrator for Role Based Access Control (RBAC) in IBM Cloud Pak for AIOps, you are responsible for determining and implementing the best approach for authenticating and managing users. To work with profiles, and their associated restriction filters and redaction policies, you must possess an administrative user account with the **Manage profiles** role. For more information, see Roles and permissions.

Note: In the installation resource you are using you need to set the *Enable Group Profiles* radio button to *True*.



Identity provider connections

User records are stored in an internal repository database. You are recommended to use an enterprise-grade password management solution through identity provider connection, such as an LDAP server for password management. To connect to an identity provider, use the Identity Management (IM) service. You can open the IM service from the **Access control** page, by clicking **Configure identity provider**. For more information about connecting to an LDAP provider with the IM service, see Identity providers.

If you configure an identity provider connection to an LDAP server, ensure that you grant administrator privileges to a user in your LDAP server.

Important:

- You cannot create a user from the **Access control** page of your cluster Cloud Pak for AIOps console. You can use the **Access control** page to add only an existing LDAP user to your cluster.
- You must use the configured identity provider (LDAP) to manage users that are added from that provider system. For instance, to change or reset the password for a user, you must complete that change in your configured provider system. You cannot change or reset a user password with the Cloud Pak for AIOps console.

  To see your configured identity providers, navigate to **Administration** > **Identity providers** in the Cloud Pak for AIOps console.
- Identity Management (IM)

Adding users and user groups

An administrator can manage the permissions that users and groups have on the platform. However, users might need additional permissions.

A user can have multiple roles. The roles can be assigned directly to a user or can be assigned to the user through a user group. If a user has multiple roles, the user has all of the permissions from all of the roles that are assigned to them.

Tip: You can see all of the roles (and permissions) that a user has from the user's profile page, which you can access from the **Access control** page, on the **Users** tab.

If you update a user's role or their group membership and the user is logged in, the user must log out and log back in for the changes to take effect. If the user does not log out, their session will be refreshed after the session times out.

Before you add users to the platform, consider the following questions:

- Do you want to use an LDAP server to manage users' passwords?
- Do you want to use an LDAP server to manage access to the platform?
- Do you want to use user groups to manage users with similar access requirements?
- Do you want to be able to add all of the users in an LDAP group to a user group?
- Do the default roles meet my business requirements?
- Adding users
- Managing user groups

Role-based access control

An administrator can manage the permissions that users and groups have on the platform. However, users might need additional permissions. Cloud Pak for AIOps provides enhancements to meet customers' RBAC/ABAC needs around data-access control for many data types such as incidents, alerts, resources, resource groups and applications.

A user can have multiple roles. The roles can be assigned directly to a user or can be assigned to the user through a user group. If a user has multiple roles, the user has all of the permissions from all of the roles that are assigned to them.

- Roles and permissions

Tip: You can view all of the roles (and permissions) that a user has from the user's profile page, which you can access from the **Access control** page, on the **Users** tab.

Note: If a user role or user group is updated while an associated user is logged in, the user must log out and log back in for any changes to take effect. If they do not log out, their session is refreshed when a session timeout occurs.

Fine-tuning role-based access control with profiles

As an administrator, you can define redaction policies and restriction filters for hiding specific data for alerts, incidents, applications, resource groups, and resources in a unified way based on available properties, including custom ones. The policies and filters can be assigned to a profile that can then be applied to one or more user groups. This level of access control is an addition to the use of roles and permissions for accessing features.

By using these profiles, policies, and filters, you can determine the specific data that your users can and cannot view. With this finer level of access control you can better manage data privacy to meet your business needs.

For instance, if IBM Cloud Pak for AIOps is monitoring a network infrastructure that includes multiple devices, each device, and any related alerts, can contain properties for identifying the managing team (user group). This property can be a simple enrichment such as a field that denotes the managing group. The property can also be based on the naming characteristics of the device or alert payload. You can use profiles with redaction policies and restriction filters to ensure that users from one location (A) are not able to view resources in another location (B), such as VM resources. You can also configure the policies and filters so the same users (A) cannot view any alerts from those VM resources. This can ensure that users in one location do not have full visibility of alert and topology data coming from another location (B).

This fine-tuning configuration of access control is managed from the **Profiles** UI page. This page includes three key features across individual tabs:

- Profiles
- Restriction filters
- Redaction policies

User preferences and authentication

When your users are added to the platform and able to access the UI and features they can complete the following user account management tasks to manage their user settings and obtain authentication keys for accessing the IBM Cloud Pak for AIOps API:

- Setting your preferences
- Generating API keys for authentication

Identity Management (IM)

This topic covers the Identity Management (IM) features and concepts that are available with IBM Cloud Pak for AIOps. These Identity Management are provided by IBM Cloud Pak foundational services and are available for you to leverage when you install IBM Cloud Pak for AIOps

At a high level, your IBM Cloud Pak for AIOps provides authentication support through the OpenID Connect (OIDC) specification. IBM Cloud Pak for AIOps also supports Security Assertion Markup Language (SAML)-based federated authentication is also supported.

The authentication and authorization details are explored in the following topics:

- IM for your product platform users

For troubleshooting, see Troubleshooting Identity Management (IM)

For APIs, see Identity Management (IM) APIs

Multiple LDAP domains

There is a growing need for IBM Cloud Pak® for AIOps users to be able to authenticate across multiple LDAPs. Sometimes large organizations might have an LDAP domain controller for different global regions or subsidiaries.

Users can have a mix of directory types such as AD, Tivoli, OpenLDAP etc.

Users can configure multiple directories in the LDAP configuration in IBM Cloud Pak for AIOps. IBM Cloud Pak for AIOps uses WebSphere Liberty Server OpenID Connect as an authentication service which does administration and authentication against the appropriate directory.

Note: Currently, IM doesn't support LDAP failover.

Multiple LDAP registration

As a cluster administrator, you can configure multiple LDAP domains by adding multiple directory entries to the LDAP configuration in **`server.xml`**.

Open LDAP server.xml AD Tivoli Cluster Persist LDAP config Configure each LDAP WebSphere Liberty

In a multiple domain configured environment, a new user administration on IBM Cloud Pak for AIOps platform enforces a selection of appropriate domains and the user is added to the Team.

The user profile and the domain name is maintained by IBM Cloud Pak for AIOps and is further used for user management. The ability to chose domain before selecting users, such as for a team, allows administrator to isolate teams with a specific domain.

Note: User credentials are passed by IBM Cloud Pak for AIOps to WebSphere Liberty OIDC server which resolves the user domain and authenticates the user with a matching domain.

Authentication types

You can configure multiple authentication types for single sign-on (SSO).

You can then access your IBM Cloud Pak® console by using the configured authentication type.

The following authentication types are supported:

- IBM provided credentials (admin only). The default authentication type that you configure during IBM Cloud Pak® installation. This configuration is usually the Administrator username and password. The foundational services installer generates a password for the default username `admin`, which is a cluster administrator role.
- Enterprise LDAP
- Enterprise SAML
- OpenShift authentication
- OpenID Connect username

Note: Multiple factor authentication (MFA) uses multiple authentication technologies to verify a user's identity. foundational services IAM supports MFA using the third-party identity provider MFA capability. If IAM is integrated with the third-party identity provider through SAML or OIDC, Cloud Pak can support MFA to authenticate users when the identity provider enables MFA.

After you install your IBM Cloud Pak®, when you access the console, you can see the login options that are available. You can see the login options only for the authentication types that are configured in your cluster. If you configure only the default authentication type (IBM provided credentials), you do not see any login option. You must then access your IBM Cloud Pak® console by using the default authentication type.

The console login cookie saves the authentication type that you select for 24 hours. When you access the login page within 24 hours, you see the login page for the same authentication type. You can choose another authentication type by clicking **Change your authentication type**.

Note: If you remove the Enterprise LDAP or disable Enterprise SAML configuration from your cluster, first clear the browser cache before you access the console login page.

Setting the preferred login options

When you configure multiple authentication types in your cluster, all the configured types are displayed on the console login page as authentication options.

If you want the console login page to display any one or a selected set of the configured login options, you can set the preferred login options.

To set the preferred login options before IM service installation, see setting the preferred login options in Configuring foundational services by editing the CommonService custom resource.

To set the preferred login options after IM service installation, complete these steps:

1. Log in to your infrastructure node by using the oc login command.
2. Edit the `platform-auth-idp` configmap.

   ```
   oc edit cm platform-auth-idp -n <your-foundational-services-namespace>
   ```

   In the data section, you see the `PREFERRED_LOGIN` data definition, which has no values set by default.
3. Add the login options that you want to display on the console page.

   For example, `PREFERRED_LOGIN: "SAML","LDAP","OIDC"`. Use the following parameter values:

   - For default authentication, use `DEFAULT`
   - For enterprise LDAP, use `LDAP`
   - For enterprise SAML, use `SAML`
   - For OpenShift authentication, use `ROKS`
   - For OpenID Connect, use `OIDC`

   Note: If you set the `PREFERRED_LOGIN` to SAML or OIDC with one provider only, the console redirects you to the login page of the configured provider.
4. Restart the `platform-identity-management` pod by deleting it.
   1. Get the platform-identity-management pod name.

   ```
   oc get pods -n <your-foundational-services-namespace> | grep platform-identity-management
   ```

   Following is a sample output:

   ```
   platform-identity-management-785df784f5-qcx4z                     1/1     Running   0          39d
   ```

   2. Delete the platform-identity-management pod.

   ```
   oc delete pod <platform-identity-management-pod-name> -n <your-foundational-services-namespace>
   ```

   5. Restart the `common-web-ui` pod by deleting it.
   6. Get the common-web-ui pod name.

   ```
   oc get pods -n <your-foundational-services-namespace> | grep common-web-ui
   ```

   2. Delete the `common-web-ui` pod.

   ```
   oc delete pod <common-web-ui-pod-name> -n <your-foundational-services-namespace>
   ```

   After the pods restart, your preferred login options are visible on the console login page.

Using foundational services IM for authentication

You can log in to the Red Hat® OpenShift® Container Platform console by using your foundational services or the IBM Cloud Pak access credentials.

Delegating authentication to OpenShift

Authenticate with Red Hat® OpenShift® Container Platform.

If you are installing IBM Cloud Pak for AIOps on an environment that uses the built-in OpenShift OAuth server, you can use the OpenShift OAuth server for authentication.

Note: In foundational services version 4.x.x, OpenShift authentication is enabled by default. You can disable it by using Identity Provider V3 API. To disable the OpenShift authentication, see Disabling OpenShift authentication by using IdP V3 API.

Note: If you are upgrading your cluster, the authentication configuration is not changed after the upgrade. For example, if you did not configure OpenShift authentication in the previous release, the OpenShift authentication is not enabled after you upgrade.

Updating OpenShift authentication before you install the IM Operator

Complete the following steps to update the OpenShift authentication parameters. You must complete these steps after you install the IBM Cloud Pak foundational services operator, but before you install the foundational services in your cluster:

1. From your Red Hat® OpenShift® Container Platform console, click **Operators** > **Installed Operators**.
2. Search for **IBM Cloud Pak foundational services** and click the **IBM Cloud Pak foundational services** operator name to open the Details tab for the operator.
3. Select the **Common Service** tab. You see an instance of the CommonService that is installed in your cluster.
4. Edit the `common-service` instance.
5. In the YAML file, scroll down to `spec.services.ibm-im-operator.spec.authentication.config` section.

   Note: Create the `spec.services.ibm-im-operator.spec.authentication.config` section if it does not exist.

   The following example shows the `spec.services.ibm-im-operator.spec.authentication.config` section in the YAML file:

   ```
   apiVersion: operator.ibm.com/v3
   kind: CommonService
   metadata:
     name: common-service
     namespace: <your-foundational-services-namespace>
   spec:
   size: ...
     ...
     services:
      - name: ibm-im-operator
        spec:
          authentication:
           config:
            roksEnabled: false
            roksURL: <endpoint url>
            roksUserPrefix: 'IAM#'
   ```

6. Update the values of the following parameters:

   - `roksEnabled:` Set to `false:` to disable authentication with Red Hat® OpenShift® Container Platform. The default value is `true`.
   - `roksURL:` The public service endpoint URL of your public cloud cluster. Use the following command to get the endpoint URL.
   - `roksUserPrefix:` Prefix to be used with the username. When you access your cluster console or CLI, you use the prefix along with the username to authenticate with OpenShift. The default value is `""`. If you are using IM with Red Hat® OpenShift® on IBM Cloud® service in the IBM Cloud, you must set the prefix to `IAM#`.

   Note: You need `jq` to run the following command. To install `jq`, see Download jq.

   ```
   curl -sk https://<server-URL>/.well-known/oauth-authorization-server | jq -r '.issuer'
   ```

   For example:

   ```
   curl -sk https://api.par-test.os.fyre.ibm.com:6443/.well-known/oauth-authorization-server | jq -r '.issuer'
   ```

   Following is a sample output:

   ```
    https://oauth-openshift.apps.par-test.os.fyre.ibm.com
   ```

   The following shows a configuration of the `ibm-im-operator` with the OpenShift authentication configuration in the **CommonService** YAML file:

   ```
   - name: ibm-im-operator
       spec:
         authentication:
           config:
             roksEnabled: true
             roksURL: 'https://oauth-openshift.apps.par-test.os.fyre.ibm.com'
             roksUserPrefix: 'IAM#'
   ```

7. Save the YAML file.

When you access IBM Cloud Pak for AIOps console, you can see the **OpenShift authentication** as a login option.

Changing the default cloud pak administrator

The IBM Cloud Pak foundational services installation creates a default **admin** user, who is a administrator. If you are delegating authentication to OpenShift, you can customize the default openshift username by using the **bootstrapUserId** parameter. This user will have the cloud pak administrator access.

To add the **bootstrapUserId** parameter before IM service installation, see Assigning the cloud pak administrator privileges to an OpenShift user.

To add the **bootstrapUserId** parameter after IM service installation, complete these steps:

1. Log in to your infrastructure node by using the **oc login** command.
2. Edit the **platform-auth-idp** configmap.

   ```
   oc edit cm platform-auth-idp -n <your-foundational-services-namespace>
   ```

   In the **data** section, the **"BOOTSTRAP_USERID"** is set to **kubeadmin** by default. Change **kubeadmin** to the OpenShift user who you want to make the cloud pak administrator.
3. Run the following CURL command to update default admin username:

   ```
   curl -X PUT 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idmgmt/identity/api/v1/users/defaultAdmin' \
       --header 'Content-Type: application/json' \
       --header 'Accept: application/json' \
       --header "Authorization: Bearer $ACCESS_TOKEN" \
       -d "{\"username\": \"${new_admin_username}\"}" \
       --insecure
   ```
4. Restart the **platform-identity-provider** and **platform-auth-service** pods by deleting them.
   1. Get the **platform-identity-provider** pod name and **platform-auth-service** pod name .

      ```
      oc get pods -n <your-foundational-services-namespace> | grep platform-identity-provider

      oc get pods -n <your-foundational-services-namespace> | grep platform-auth-service
      ```
   2. Delete the **platform-identity-provider** pod and **platform-auth-service** pod.

      ```
      oc delete pod <platform-identity-management-pod-name> -n <your-foundational-services-namespace>

      oc delete pod <platform-auth-service-pod-name> -n <your-foundational-services-namespace>
      ```

After the pods restart, the OpenShift user can access with administrator privileges.

OIDC registration with the client custom resource

Create a custom resource for OpenID Connect (OIDC) registration.

The controller for OIDC registration generates secret objects by using a client ID and secret key. The secret objects can be used for OIDC challenges. The identity management (IM) service applies role-based access control (RBAC) to the OIDC registration custom resource and binds the ability to allow operator or higher roles to create, view, and list these objects.

- The custom resource is added to the appropriate roles so that users with that role can create OIDC Client resources.
- The OIDC Client resource contains the OIDC registration payload and the name of the secret that is required to inject the resulting OIDC Client ID and secret.
- The OIDC registration controller watches for the OIDC Client resource to be created and then processes it.
- The processing status of the OIDC Client resource is stored in the status portion of the resource, and the secret is created with the client ID and secret information.

Complete these steps to create the **Client** custom resource. After you create the custom resource, you can integrate it with your application to enable OIDC-based authentication.

1. Define a custom resource named **Client** and save it as a YAML file. For example, **sample-oidc-client.yaml**. To ensure that the namespace is authorized by the namespace scope operator, see Authorizing foundational services to perform operations on workloads in a namespace.

Following is a sample custom resource:

```
apiVersion: oidc.security.ibm.com/v1
kind: Client
metadata:
  name: eventstreams-client
  namespace: default
spec:
  secret: eventstreams-oidcclient-secret
  oidcLibertyClient:
    post_logout_redirect_uris:
    - <console URL>:443
    redirect_uris:
    - <console URL>:443/api/auth/callback
    trusted_uri_prefixes:
    - <console URL>:443
```

The **spec** section includes information about initiating the OIDC Client registration process.

- **secret** is the name for the secret that is generated during registration.
- **oidcLibertyClient** contains the post logout redirect urls (**post_logout_redirect_uris**), trusted url prefixes (**trusted_uri_prefixes**), and redirect urls (**redirect_uris**).

Following is a sample custom resource:

```
apiVersion: oidc.security.ibm.com/v1
kind: Client
metadata:
  name: eventstreams-client
  namespace: default
spec:
  secret: eventstreams-oidcclient-secret
  oidcLibertyClient:
    post_logout_redirect_uris:
    - https://myapp.example.com:443
    redirect_uris:
    - https://myapp.example.com:443/api/auth/callback
    trusted_uri_prefixes:
    - https://myapp.example.com:443
```

2. Create the custom resource.

   ```
   oc create -f <crd-yaml-file-name>.yaml
   ```

   Following is an example command:

   ```
   oc create -f sample-oidc-client.yaml
   ```

   The command creates the resource, registers the client, and generates the secret (which is **eventstreams-oidcclient-secret** in the example), which includes the client ID and client secret.

3. Check whether the registration is successful.

   ```
   oc get Client <client-resource-name> -n <client-resource-namespace>
   ```

   Following is an example command:

   ```
   oc get Client eventstreams-client -n default
   ```

   If the resource is successfully registered, you see a message similar to the following message:

   ```
   status:
     conditions:
       lastTransactionTime: "2019-12-12T23:17:42Z"
       message: OIDC client registration successful
       reason: CreateClientSuccessful
       status: "True"
       type: "Ready"
   ```

APIs

You can use the following APIs to create and manage OIDC registration custom resource.

Before you use these APIs, see Preparing to run component or management API commands.

The **<cluster_address>** is the IBM Cloud Pak console route. You can get the IBM Cloud Pak console route by running the following command:

```
oc -n <your-foundational-services-namespace> get route cp-console -o yaml
```

The IBM Cloud Pak console route is in the **spec.host** section.

Create OIDC registration

```
curl -X "POST" --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization:
bearer $ACCESS_TOKEN" --data @registration.json  "https://<cluster_address>:443/idprovider/v1/auth/registration" --
insecure
```

Update existing OIDC registration

```
curl -X "PUT" --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization:
bearer $ACCESS_TOKEN" -d --data @registration.json
"https://<cluster_address>:443/idprovider/v1/auth/registration/$CLIENT_ID" --insecure
```

GET existing OIDC registration information

```
curl -X "GET" --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization:
bearer $ACCESS_TOKEN"  "https://<cluster_address>:443/idprovider/v1/auth/registration/$CLIENT_ID" --insecure
```

DELETE existing oidc registration

```
curl -X "DELETE" --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization:
bearer $ACCESS_TOKEN" "https://<cluster_address>:443/idprovider/v1/auth/registration/$CLIENT_ID" --insecure
```

Auditing IM service

Identity Management (IM) uses HTTP ingest endpoint to send the IM related audit records to the Audit logging service of the foundational services. IM sends the audit logs to **zen-audit-svc** when you enable auditing service for IM.

By default, the audit logs are disabled in IM. You can enable the auditing service in IM with the following methods:

- Enabling auditing service with **oc** command
- Enabling auditing service with console

Prerequisites

- Ensure that the audit logging service (**zen-audit**) is running in your cluster.

```
$ oc get pods |grep zen-audit
```

```
zen-audit-789bfdc9d4-2hkmn                                        1/1     Running     0          3h25m
```

- If **zen-audit** is not installed in your cluster, IM sends the audit logs to the docker logs in the container.

  The following is the sample **platform-auth-idp** configmap:

```
apiVersion: v1
data:
 AUDIT_ENABLED_IDMGMT: "true"
 AUDIT_ENABLED_IDPROVIDER: "true"
 BASE_AUTH_URL: /v1
 BASE_OIDC_URL: https://127.0.0.1:8443/idauth/oidc/endpoint/OP
 CLUSTER_NAME: mycluster
 HTTP_ONLY: "true"
 MASTER_HOST: mycluster.icp
 NODE_ENV: production
 OAUTH2DB_DB_PORT: "3306"
 OIDC_ISSUER_URL: https://mycluster.icp:8443/oidc/endpoint/OP
 SESSION_TIMEOUT: "43200"
  .
  .
  .
```

Enabling auditing service with oc command

To enable the auditing in the IM with the **oc** command, complete the following steps:

1. Edit the **platform-auth-idp** configmap in the **foundational-services** namespace:

   ```
   oc -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

2. Set the following attribute values to **true**:

   - AUDIT_ENABLED_IDPROVIDER: 'true'
   - AUDIT_ENABLED_IDMGMT: 'true'

3. Save the configmap.

4. Delete the the **platform-auth-service**, **platform-identity-provider**, and **platform-identity-management** pods.

   ```
   oc -n <your-foundational-services-namespace> delete pods <pod_name>
   ```

5. To check whether the audit logs are generated, go to the **zen-audit** pod logs. Run the following command to see the **zen-audit** pod name. Replace the **<common-services-namespace>** variable with the namespace in which foundational services is installed.

   ```
   oc get pods -n <common-services-namespace> | grep zen-audit
   ```

6. To see the pod logs, enter the following command. Replace the **<zen-audit-pod-name>** variable with the **zen-audit** pod name.

   ```
   oc logs -f <zen-audit-pod-name>
   ```

After you complete these steps, the audit logs are forwarded to the appropriate security information and event management [SIEM] tool that is configured within the **foundational-services** namespace.

Enabling auditing service with console

1. Log in to the Red Hat® OpenShift® Container Platform console.

2. From the navigation menu, click **Workloads** > **Config Maps.**

3. Search for **platform-auth-idp**.

4. Click **...** > **Edit Config Map**.

5. Set the following attribute values to **true**:

   - AUDIT_ENABLED_IDPROVIDER: 'true'
   - AUDIT_ENABLED_IDMGMT: 'true'

Note: A **resource is managed by example-authentication and any modifications may be overwritten** warning message is displayed. You can ignore this message.

6. Click **Save**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate `platform-auth-service` pods.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the `platform-auth-service` pods with the latest ConfigMap values.
11. Click `platform-auth-service`.
12. Wait for some time. Then, check the status of the `platform-auth-service` pods in the **Pods** pane. The status of the pods must show as `1/1` under the **Ready** field name.
13. To check whether the audit logs are generated, go to the `zen-audit` pod logs.
14. Repeat steps 8 through 13 for the `platform-identity-provider` and `platform-identity-management` deployments.

To forward audit records to a SIEM solution such as Splunk, Mezmo, or QRadar, see Exporting Cloud Pak for Data audit records to a security information and event management solution.

IM for IBM Cloud Pak for AIOps platform users

Identity Management (IM) for platform users includes authentication that includes OIDC and SAML.

Authentication

IBM Cloud Pak® for AIOps uses WebSphere Liberty OpenID Connect (OIDC) 1.0 for authentication. It calls the standard OIDC endpoints `/authorize` and `/token` to initiate an OAuth dance. OpenID in Liberty can be configured with Lightweight Directory Access Protocol (LDAP), after which an LDAP user can authenticate to IBM Cloud Pak for AIOps by using the same OpenID endpoints. For single sign-on (SSO) based authentication, OIDC is configured with Security Assertion Markup Language (SAML) to interact with your enterprise identity source.

Authentication protocols supported

IBM Cloud Pak for AIOps supports the following two authentication protocols:

1. OIDC-based authentication
2. SAML-based federated authentication

OIDC and SAML are both used for SSO with IBM Cloud Pak for AIOps but for different purposes.

IBM Cloud Pak for AIOps is an OIDC identity provider that provides authentication and authorization services to IBM Cloud Pak for AIOps console and APIs. It works along with one or more LDAP providers to authenticate the user ID and password with the LDAP service and to provide an access token for subsequent requests to IBM Cloud Pak for AIOps services. IBM Cloud Pak for AIOps is an identity provider through LDAP.

IBM Cloud Pak for AIOps can be configured as a SAML service provider, which allows federated authentication with an external SAML 2.0 identity provider. When you configure SSO, IBM Cloud Pak for AIOps redirects your console browser to the third-party login page, and OIDC issues you a bearer token.

The OIDC-based authentication service is the default authentication service in IBM Cloud Pak for AIOps. If required, you can configure a SAML server to provide federated authentication.

OIDC-based authentication

You must configure and connect an LDAP directory with your product cluster, and provide cluster administrator, Cloud Pak administrator, or administrator access level. For more information, see Configuring LDAP connection. You must set up the LDAP connection before you create a team and add users to the team. Only LDAP users who are assigned to a team can log in to the console.

Configuring single sign-on with the SAML using IdP APIs

Metadata files are used for communication between IBM Cloud Pak for AIOps and your enterprise SAML server.

Note: If you see the error `404` during SAML configuration, check with the administrator whether SAML is enabled for the Cloud Pak at the Identity Provider.

Before you configure SSO, you must configure a fully qualified domain name (FQDN) for accessing your cluster.

Note: If you are configuring SSO by using SAML, you must manually register the IdP by using Identity provider APIs in the following scenarios:

- If you do not have the IdP registration.
- If you are installing foundational services for the first time.
- Irrespective of SAML dependency on LDAP.

To verify whether you have an IdP registration, see Get IdP registration by query.

To configure SSO, complete the following sequence of steps:

1. Configure SAML and import metadata that is sent by your enterprise SAML server by using IdP V3 APIs.
2. Export the metadata of IBM Cloud Pak for AIOps to your enterprise SAML server. After you complete this task, a metadata file is downloaded. For more information, see SAML metadata export by using samlmetadata API.
3. Verify whether SAML was successfully configured.
4. Configure the SAML connection with one of the following methods:
   - **SAML with SCIM dependency:** You can directly connect with the SAML IdP if it is SCIM-enabled. SCIM is supported to connect with the registered IdP and import the users by using the IdP APIs. If you are manually registering SAML with SCIM-enabled IdP, see SAML with SCIM dependency registration. Based on your requirement, you can update the values of the schema elements, **`name`**, **`description`**, **`idp_type`**, **`scim_base_path`**,**`token_attribute_mappings`**, **`scim_attribute_mappings`**, and **`config`**. To understand the use of schema elements, see Different schema elements.
   - **SAML without any dependency:** If you are using SAML without LDAP dependency or SCIM-enabled IdP, you can register the IdP by using SAML registration without any dependency. Based on your requirement, you can update the values of schema elements, **`name`**, **`description`**, and **`token_attribute_mappings`**. To understand the use of schema elements, see Different schema elements.
   - **SAML with LDAP dependency:** Optional: Connect with an LDAP server and import users who might use the SSO request. For more information, see Configuring LDAP connection. See the following notes:
     - You can also connect IBM Cloud Pak for AIOps with the same LDAP server that your enterprise SAML server uses for authentication.
     - If you are connecting LDAP by using SAML, see SAML with LDAP dependency registration. Based on your requirement, you can update the values of the schema elements, **`name`**, **`description`**, **`scim_base_path`**, **`token_attribute_mappings`**, and **`saml_ldap`**. To understand the use of schema elements, see Different schema elements.

Configuring single sign-on with the SAML using IBM Cloud Pak for AIOps console

Configure single sign-on (SSO) by using Security Assertion Markup Language (SAML) from IBM Cloud Pak for AIOps console.

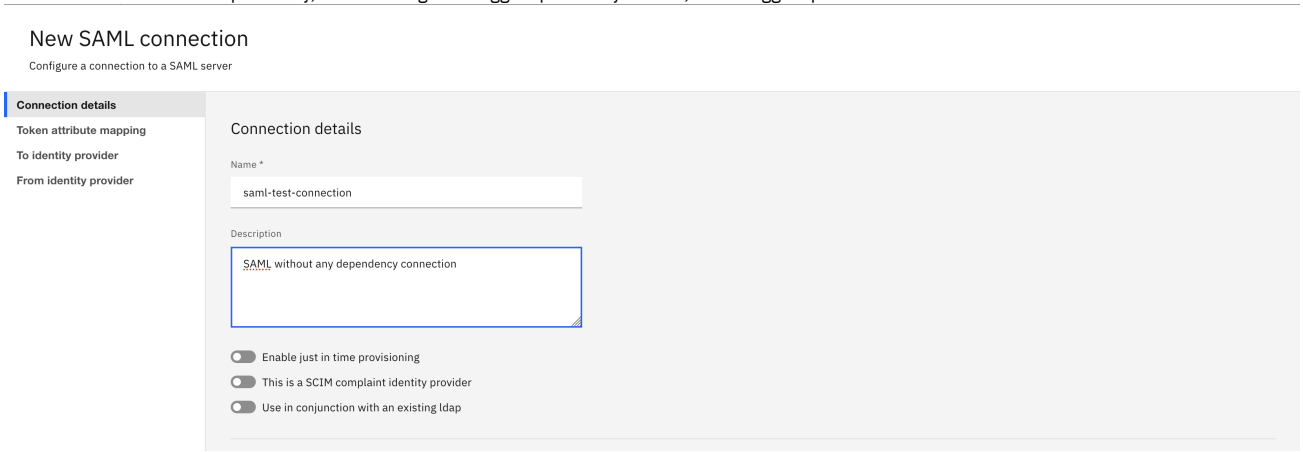
Note: You can configure only one SAML connection.

You can configure the SAML connection by using any of the following methods:

- SAML with SCIM dependency
- SAML without any dependency
- SAML with LDAP dependency

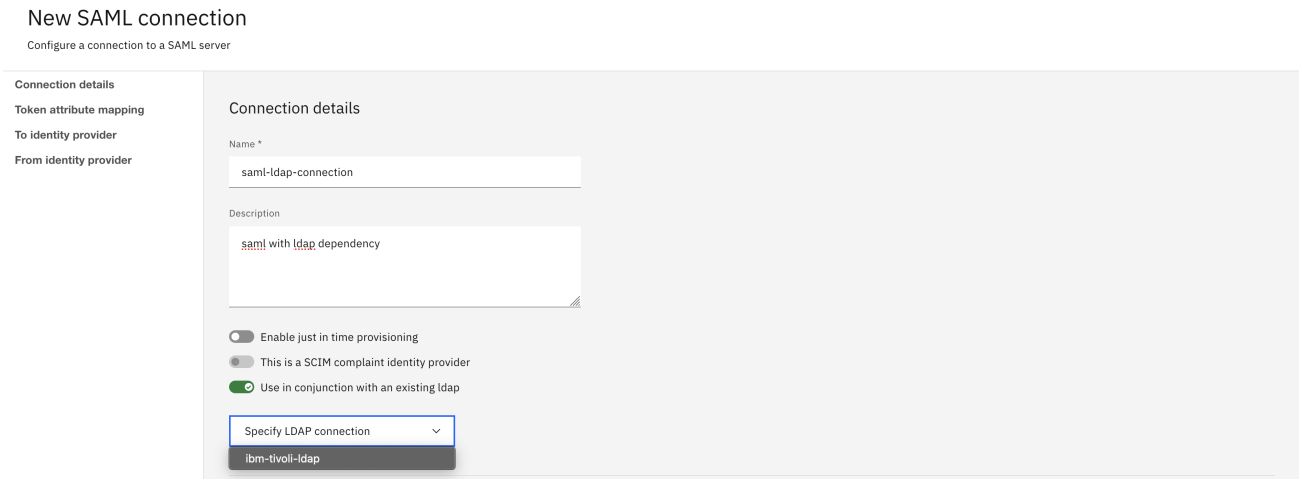
Setting up the SAML connection

1. Log on to the console as an administrator.
2. From the navigation menu, click **Administer** > **Identity providers**.
3. Click **Create Connection**. Select **SAML 2.0** as the protocol type and click **Next**. The **New SAML Connection** page is displayed.
4. In the **New SAML Connection** page, select the **Connection details** tab. It contains the following options that you are required to fill:
   - **`Name`**: A name for the SAML connection.
   - **`Description`**: Description of the SAML connection.
5. Choose the type of SAML that you want to configure.
   - For SAML without dependency, do not change the toggle options. By default, these toggle options are disabled.

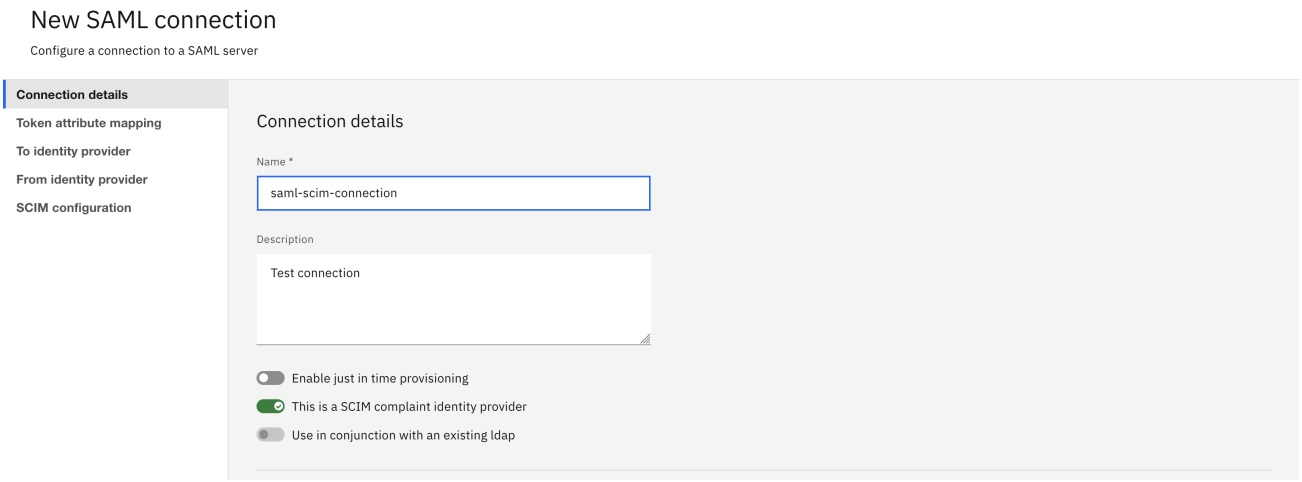


   - For SAML with LDAP dependency, enable **Use in conjunction with an existing ldap** toggle option. From the **Specify LDAP Connection** drop-down list, choose the existing LDAP to integrate.

Note: You must configure the LDAP connection before you configure SAML with LDAP dependency. **Important:** When you enable **Use in conjunction with an existing ldap**, you won't see the login option for LDAP authentication on the console login page.



- For SAML with SCIM dependency, enable **This is a SCIM compliant identity provider** toggle option.



6. Select the **Token attribute mapping** tab. It contains the following options that you are required to fill:

   - `Subject`
   - `Given name`
   - `Family name`
   - `Groups`
   - `Email`

   See the following notes:

   - By default, the **Token attribute mapping** values are pre-defined if you do not specify the mapping. It is recommended to modify the default values according to the SAML claim. The following default values are displayed if you do not specify the mapping values:
   - `Subject` : uid
   - `Given name` : firstName
   - `Family name` : lastName
   - `Groups` : blueGroups
   - `Email` : emailAddress
   - The `uniqueSecurityName` option is not supported from the product console. If the `uniqueSecurityName` is required in **Token attribute mapping**, see Different schema elements for IdP V3.

7. Select **To identity provider** tab. Click **Download metadata** link to download the SAML 2.0 metadata. Once you download the metadata, you can upload that metadata to your identity provider to generate identity provider metadata.

8. Select **From identity provider** tab. Upload the identity provider metadata in `*.xml` file that is supplied by your identity provider.

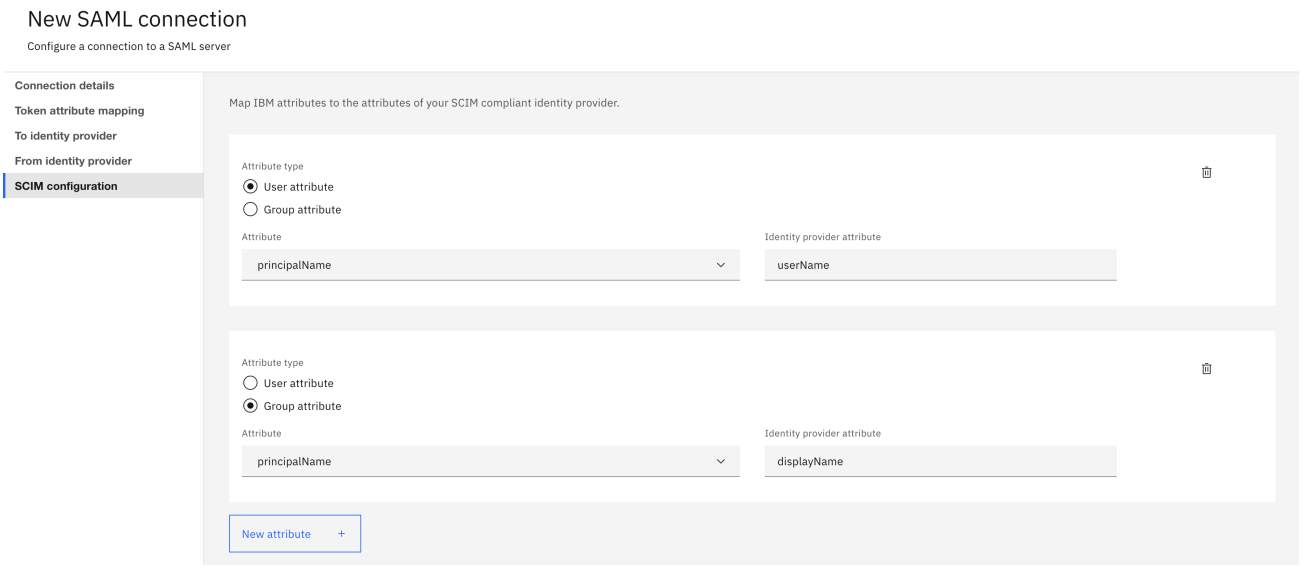
   Note: The maximum accepted size of the metadata file is `2 MB`. If the size of the metadata file is more than 2 MB, the `Payload Too Large` error is displayed in the console.

9. The **SCIM configuration** tab is enabled when you choose to configure SAML with SCIM dependency. It contains the following fields:

- **`Scim base path`**: It is required to uniquely identify the service instance.
- **`Grant type`**: It identifies the token generation flow.
- **`Token URL`**: It is the token endpoint of the IdP. It is used to request the token.
- **`Client ID`**: It is used for the client credentials flow.
- **`Client Secret`**: It is used for the client credentials flow.



Also, you need to configure the SCIM attribute mapping for the **`user`** and **`group`** resources. By default, these fields are pre-defined as the **`principalName`** in both **`user`** and **`group`** attributes. It is recommended to modify the values based on the SCIM IdP configuration.



10. Click **Create**.

Configuring single sign-on using OpenID Connect (OIDC)

Using OIDC, you can configure single sign-on (SSO) between IBM Cloud Pak® for AIOps and supported providers. You can configure OIDC with just-in-time (JIT) provision only.

OIDC is an authentication protocol layer that is built on the OAuth 2.0 protocol. It allows the third-party applications to:

- Verify the identity of the users.
- Get the basic profile information about the users.

The following steps represent the flow that happens during SSO configuration by using OIDC:

1. A user attempts to access a service in IBM Cloud Pak for AIOps through a web browser.
2. IBM Cloud Pak for AIOps verifies whether an authentication token is present.
3. If no authentication token is present, IBM Cloud Pak for AIOps redirects the request for authentication to the third-party authorization server of the user.
4. The authorization server presents a login page to the user.
5. If the user logs in successfully, the authorization server redirects the user, along with the OIDC response, to IBM Cloud Pak for AIOps.

6. IBM Cloud Pak for AIOps generates an authentication token and grants access to the service that the user requested.

Note: You can use PKCE with OIDC for protection from authorization code interception attacks for OAuth 2.0 public clients. For more information, see PKCE support for OpenID Connect clients. The IM service enables PKCE by default. If your OIDC provider configures PKCE, you can use it for authentication in the IM login flow. If the OIDC provider does not configure PKCE, it does not impact the IM login flow.

Configuring SSO in IBM Cloud Pak for AIOps by using OIDC

See the following notes:

- You can configure SSO with OIDC support by using the IdP V3 APIs. For more information, see IdP V3 APIs.
- Currently, OIDC is the only supported protocol with the IdP V3 version.
- To verify whether you have OIDC support or not, see Getting the list of registered OIDC clients by query.

Before you begin

Log in to any of the following platforms by using the provided IdP links and register yourself in the application. During registration, use application URL as cp-console URL and redirect URL as **`https://<cp-console-url>/ibm/api/social-login/redirect/<name of the oidc>`**. Once you are registered, you can get the unique client ID, client secret and **`discovery_url`** endpoint. Currently, only these platforms are verified to register the OIDC by using the IdP V3 API.

Verified platforms to register OIDC provider using IdP V3

Platform	IdP link
IBM Security Verify (ISV)	https://www.ibm.com/docs/en/erqa?topic=using-security-verify-as-oidc-provider **Note:** Whenever you perform the attributes mapping through the OIDC provider in the ISV application, enable the **Send all known user attributes in the ID token** option. And, ISV groups must not contain spaces.
Google Cloud Platform	https://cloud.google.com/identity-platform/docs/web/google
Microsoft	https://cloud.google.com/identity-platform/docs/web/microsoft
Okta	https://help.okta.com/en-us/Content/Topics/Apps/Apps_App_Integration_Wizard_OIDC.htm

Procedure

To configure SSO, complete the following sequence of steps:

1. Register the OIDC clients. For more information, see Registering the OIDC clients.
2. Get the list of registered OIDC clients. For more information, see Getting the list of registered OIDC clients.

   If you need to delete the registration of the client, see Deleting the registration of the client.
3. Open the Cloud Pak home page to verify whether the OIDC is successfully configured.

   Note: After the OIDC is successfully configured, you can log in to the cp-console. However, to log in to the Cloud Pak Platform and to get the OIDC successfully configured, first you need to onboard the OIDC users into the Cloud Pak Platform. Then, the OIDC users are able to log in to the Cloud Pak Platform.
4. To authenticate, click the corresponding OIDC link in the Cloud Pak home page.

Granting an admin role to Cloud Pak Platform

Before you begin, onboard OIDC users into the Cloud Pak Platform.

Important: If you configure multiple authentication types (SAML, LDAP, and OIDC) in your cluster, and onboard same users to all the authentication types, the users might not be able to log in the first time by using OIDC. They might be able to log in the second time. To overcome the limitation, onboard users by following the instructions in Integrating IBM Cloud Pak SAML with Azure Active Directory.

Now, complete the following steps to grant an Admin role for Cloud Pak Platform. The procedure is based on the APIs.

1. Configure environmental variables:
   1. Obtain an IM access token for the default admin user. For more information, see Preparing to run API commands.
   2. Set the HOST environment variable. To get the host value, run the following command:

   ```
   oc get route -n <your-foundational-services-namespace>
   ```

   Set the HOST to the cp-console route. For example, **`export HOST=<cp-console>`**.

   3. Set CPHOST to the cpd route with corresponding **`ibm-nginx-svc`** service name. For example, **`CPHOST=cpd-cp4i.apps.bunny.cp.abc.ibm.com`**.
2. Get the IBM Cloud Pak access token for the admin user.

   ```
   curl -ki -X GET 'https://'$CPHOST'/v1/preauth/validateAuth' --header 'username: cpadmin' --header "iam-token: $TOKEN"
   ```

   Note: The default **`username`** for the new cluster installation is **`cpadmin`**. If you upgrade the existing cluster, the default username is **`admin`**.
3. Create a user group with the role that you want to assign to the user.

   ```
   curl -k -s  -X POST 'https://'$CPHOST'/usermgmt/v2/groups' \
   --header 'Content-Type: application/json'   \
   ```

```
--header 'Accept: application/json'   \
--header "Authorization: Bearer $CPTOKEN" \
-d
'{"name": "isv group",
"description": "Group default ISV group",
"role_identifiers": ["zen_user_role"]
}'
```

Note: To create an admin group, in "role_identifiers", specify the “zen_administrator_role”.

Verify whether the group is listed, and get the group ID:

```
curl -k -s  -X GET 'https://'$CPHOST'/usermgmt/v2/groups' \
--header 'Content-Type: application/json'   \
--header 'Accept: application/json'   \
--header "Authorization: Bearer $CPTOKEN"
```

4. Add the `isv` group into the user groups:

```
curl -k -s  -X POST 'https://'$CPHOST'/usermgmt/v2/groups/<group_ID>/members' \
--header 'Content-Type: application/json'   \
--header 'Accept: application/json'   \
--header "Authorization: Bearer $CPTOKEN" \
-d
'{
"ldap_groups": ["admin"]
}'
```

In the command, the **group_ID** is the `isv` group ID that you got in the previous step.

The configuration is complete.

5. Verify whether the user who is a member of the `isv` group can log in to Cloud Pak Platform with proper permissions. To verify, open the Cloud Pak Platform in your browser by using the URL of $CPHOST.

Configuring LDAP connection

Configure an LDAP (Lightweight Directory Access Protocol) connection for your product cluster.

You must connect an LDAP directory with your product cluster. You can then add users from your LDAP directory into your cluster.

IM supports the `LDAPEntityType` configuration for LDAP User and Group entities as default to define the `ObjectClass` or `ObjectCategory` attributes automatically in the Liberty XML file. The `ObjectClass` or `ObjectCategory` attributes of the User and Group are retrieved from the `User` and `Filter` fields that are defined in the LDAP IDP V3 idsource API. For existing LDAP connection, you need to restart the `platform-auth-service` pods to enable the `LDAPEntityType` configuration. For more information, see Enabling `LDAPEntityType` configuration for existing LDAP User and Group entities.

The following LDAP types are supported:

- IBM Tivoli Directory Server
- IBM Lotus Domino
- IBM SecureWay Directory Server
- Novell eDirectory
- Sun Java™ System Directory Server
- Netscape Directory Server
- Microsoft Active Directory
- Custom

Note: You can configure an account lockout policy when you set up your LDAP server. An account lockout policy provides more security by restricting access to the account if multiple login attempts fail.

Important: If Platform UI (`ibm-platformui-operator`) service is installed, user management capabilities are available for managing LDAP user and user groups roles, and permissions. You can add an LDAP or OpenShift user to your cluster. When you add a user, you must specify the OpenShift or LDAP login ID of the user in the `Username` field.

Required user type or access level: Cluster administrator or Cloud Pak administrator

Enabling `LDAPEntityType` configuration for existing LDAP User and Group entities

Restart the `platform-auth-service` pods to enable the `LDAPEntityType` configuration for the existing LDAP connection.

1. Get the `platform-auth-service` pod name.

```
oc get pods -n <your-foundational-services-namespace> | grep platform-auth-service
```

2. Delete the `platform-auth-service` pod.

```
oc delete pod <platform-auth-service-pod-name> -n <your-foundational-services-namespace>
```

The procedure is not applicable for new LDAP connection.

Connecting to your LDAP directory

Follow these steps to set up your LDAP connection.

Pre-requisite: Install the cluster. Once the cluster is installed, the admin username, password, and the URL are displayed.

1. Log on to the console by using the URL and enter the username and password.

   Note: The console can be Platform UI or Common UI. For more information, see Accessing your cluster by using the console.

2. From the navigation menu, click **Identity and access** > **Identity providers**.

3. Click **Create Connection**. The "LDAP connection" page is displayed.

4. Enter the following details to set up your LDAP connection.

   Note: You can configure multiple LDAP connection instances to the same LDAP server. However, ensure that you meet the following requirements:

   - Both the **`Base DN`** and connection **`Name`** must be unique.
   - The username from the LDAP attribute that you specified in the **`User ID map`** must be unique across your LDAP instances. This requirement is because WebSphere Liberty uses federated registries for the multiple LDAPs by default. For more information, see Configuring LDAP user registries in Liberty.
   - IBM Cloud Pak foundational services provides a default **`admin`** user ID. If your LDAP registries have **`admin`** user IDs, you must rename the default **`admin`** user ID that foundational services provides. For more information about how to change the default name, see Changing the default admin username or Changing the cluster administrator username.

LDAP connection

Enter connection information.

- **Connection name:** A unique name for the LDAP connection. Format: 3 - 30 alphanumeric characters; Special characters that are allowed: **-** **_**
- **Server type:** A list of directory server types to which you can connect. Select one from the list.
- **Case insensitive user search:** Set the value to **`true`** or **`false`** to enable or disable case sensistivity for the LDAP usernames.

LDAP authentication

Enter authentication information.

- **Base DN**: The distinguished name of the search base. Example: dc=abc,dc=com. Format: 1 - 255 alphanumeric characters; Special characters that are allowed:
  - `=`
  - `.`
  - `,`
  - `-`
- **Bind DN**: The user who is allowed to search the base DN. Example: cn=admin,dc=abc,dc=com. This parameter is optional. If specified, the **`Bind DN password`** parameter must also be specified. If no user is specified in the **`Bind DN`** parameter, the LDAP connection is established without authentication. Format: 0 - 255 alphanumeric characters; white space is allowed; Special characters that are allowed:
  - `=`
  - `.`
  - `,`
  - `-`
  - `:`
  - `@`
  - `(`
  - `)`
  - `_`
  - `\`
- **Bind DN password**: The password of the user specified in the **`Bind DN`** parameter. This parameter is required if a user is specified in **`Bind DN`**. A maximum of 255 characters are allowed. Only ASCII characters are allowed; non-ASCII characters are not supported.

  Note: The configuration of **Base DN** and **Bind DN** values must be set as case-sensitive and must be a full distinguished name (DN) path. The DN path, including spaces, commas, and other characters, must be the same as configured in the LDAP server. See the following example:

  ```
  Base DN : DC=mycompany,DC=com
  Bind DN : CN=Administrator,CN=Users,DC=mycompany,DC=com
  ```

  For **`Base DN`**, the following values are invalid:

  - dc=mycompany,dc=com because **`DC`** is lowercase alphabet.
  - DC=mycompany, DC=com because there is a space between the parameters.

  For **`Bind DN`**, the following values are invalid:

  - cn=Administrator,cn=Users,dc=mycompany,dc=com because **`CN`** and **`DC`** are lowercase alphabets.
  - CN=Administrator,DC=mycompany,DC=com because **`CN=Users`** parameter is missing.
  - CN=Administrator,CN=Users, DC=mycompany,DC=com because there is a space between the parameters.
  - CN=administrator,CN=Users,DC=mycompany,DC=com because the **`administrator`** parameter value starts with a lowercase alphabet.

  Note: Microsoft Active Directory server does a strict check of **`Base DN`** and **`Bind DN`** values while it establishes a connection.

You can click **Test connection** to verify whether the LDAP connection details are valid.

LDAP server

Enter the server URL for your connection.

- **URL**: The LDAP directory domain name or IP address, and the LDAP port number. The domain name must begin with `ldap(s)://`. Example URL: `ldap(s)://corpldap.abc.com:389` or `ldap://10.10.10.1:389`.

For LDAP over SSL (LDAPS), you must use the domain name, and the URL must begin with `ldaps://`. Example URL: `ldaps://corpldap.abc.com:636`.

Note: If you are unable to connect to your LDAPS server by using the host name, add the IP address and host name of the LDAPS server in your local DNS. The LDAPS server host name must be resolvable from your master node.

LDAP filters

Note:

- If your LDAP server is compliant with LDAP version 3, then the filters are not sensitive to alphabet case or to white spaces.
- From foundational services version 3.19 and later, `Group filter` and `Group member ID map` filters support one or more than one value.

Enter information about the search filters. For default LDAP filters by LDAP type, see Default LDAP filters by LDAP type.

- **Group filter**: The filter clause for searching groups. From foundational services version 3.23 and later, IM supports all the special characters but following are the validated and tested special characters: white space, `= ; . , & % () {} <> | !`
- **Group ID map**: The filter to map a group name to an LDAP entry. Format: 1 - 255 alphanumeric characters; Special characters that are allowed: white space, `* : = ; . , & % () {}`
- **Group member ID map**: The filter to map a user to a group. Format: 1 - 255 alphanumeric characters; Special characters that are allowed: white space, `* : = ; . , & % () {}`
- **User filter**: The filter clause for searching users. From foundational services version 3.23 and later, IM supports all the special characters but following are the validated and tested special characters: white space, `= ; . , & % () {} <> | !`
- **User ID map**: The filter to map a username to an LDAP entry. Format: 1 - 255 alphanumeric characters; Special characters that are allowed: white space, `* : = ; . , & % () {}`

5. Click **Create**.

Your cluster is now connected with your LDAP directory.

Note: To block other hosts, see Adding egress firewall to prevent DNS loop to unknown hosts

Note: If you are using an LDAPS connection, the SSL (Secure Sockets Layer) certificates that are required for your LDAPS connection are automatically configured when you connect with your directory.

If your LDAPS connection is not successful, you can try setting up the connection manually. For more information, see Configuring LDAP over SSL.

Next, you can add your LDAP users and user groups to your cluster.

Note: The Cluster Administrator or Cloud Pak administrator must provide the Administrator with view access to the LDAP directory. Only then, the Administrator can use the console to add users or groups to teams.

Default LDAP filters by LDAP type

Note: There is a limitation on liberty, that is, the LDAP `userFilter` and `groupFilter` fields must contain an attribute with `=%v` value. For example, `(cn=%v)` for Cloud Pak and Cloud Pak for data login flow to work.

The following table represents the name of the filters, their data types and their default values. The `groupFilter` and `groupMemberIdMap` can have one or more than one default value from foundational services version 3.19 and later.

For example:

- `groupFilter` = (&(cn=%v)((objectclass=groupOfNames)(objectclass=groupOfUniqueNames)))
- `groupMemberIdMap` = groupOfNames:member;groupOfUniqueNames:uniqueMember

Table 1. List of default IBM Tivoli Directory Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=groupOfUniqueNames))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfUniqueNames:uniquemember
userFilter	String	(&(emailAddress=%v)(objectclass=person))
userIdMap	String	*:emailAddress

Table 2. List of default Custom Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=groupOfUniqueNames))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfUniqueNames:uniquemember
userFilter	String	(&(uid=%v)(objectclass=ePerson))
userIdMap	String	*:uid

Table 3. List of default Microsoft Active Directory LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectcategory=group))
groupIdMap	String	*:cn
groupMemberIdMap	String	memberOf:member
userFilter	String	(&(sAMAccountName=%v)(objectclass=person))
userIdMap	String	user:sAMAccountName

Table 4. List of default IBM Lotus Domino LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=dominoGroup))
groupIdMap	String	*:cn
groupMemberIdMap	String	dominoGroup:member
userFilter	String	(&(uid=%v)(objectclass=Person))
userIdMap	String	person:uid

Table 5. List of default IBM SecureWay Directory Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)((objectclass=groupOfNames)(objectclass=groupOfUniqueNames)))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfNames:member;groupOfUniqueNames:uniqueMember
userFilter	String	(&(uid=%v)(objectclass=ePerson))
userIdMap	String	*:uid

Table 6. List of default Sun Java System Directory Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=ldapsubentry))
groupIdMap	String	*:cn
groupMemberIdMap	String	nsRole:nsRole
userFilter	String	(&(uid=%v)(objectclass=inetOrgPerson))
userIdMap	String	inetOrgPerson:uid

Table 7. List of default Netscape Directory Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)((objectclass=groupOfNames)(objectclass=groupOfUniqueNames)))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfNames:member;groupOfUniqueNames:uniqueMember
userFilter	String	(&(uid=%v)(objectclass=inetOrgPerson))
userIdMap	String	inetOrgPerson:uid

Table 8. List of default Novell eDirectory LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=groupOfNames))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfNames:member
userFilter	String	(&(cn=%v)(objectclass=Person))
userIdMap	String	person:cn

Adding egress firewall to prevent DNS loop to unknown hosts

Run the following command to retrieve the egress network policy:

```
oc get egressnetworkpolicy ldap-firewall
```

Sample output:

```
NAME             AGE
ldap-firewall    16m
```

The following sample YAML spec defines the LDAP firewall egress network policy. To block other hosts, update the spec to specify the hosts that you want to deny access. Then, save and apply the updated policy.

```
apiVersion: network.openshift.io/v1
kind: EgressNetworkPolicy
metadata:
  name: ldap-firewall
spec:
  egress:
  - type: Allow
    to:
      dnsName: bluepages.ibm.com
  - type: Allow
    to:
      cidrSelector:  9.30.192.0/24
  - type: Deny
    to:
      cidrSelector: 0.0.0.0/0
  podSelector:
```

```
    matchLabels:
      app: platform-auth-service
  policyType:
  - Egress
```

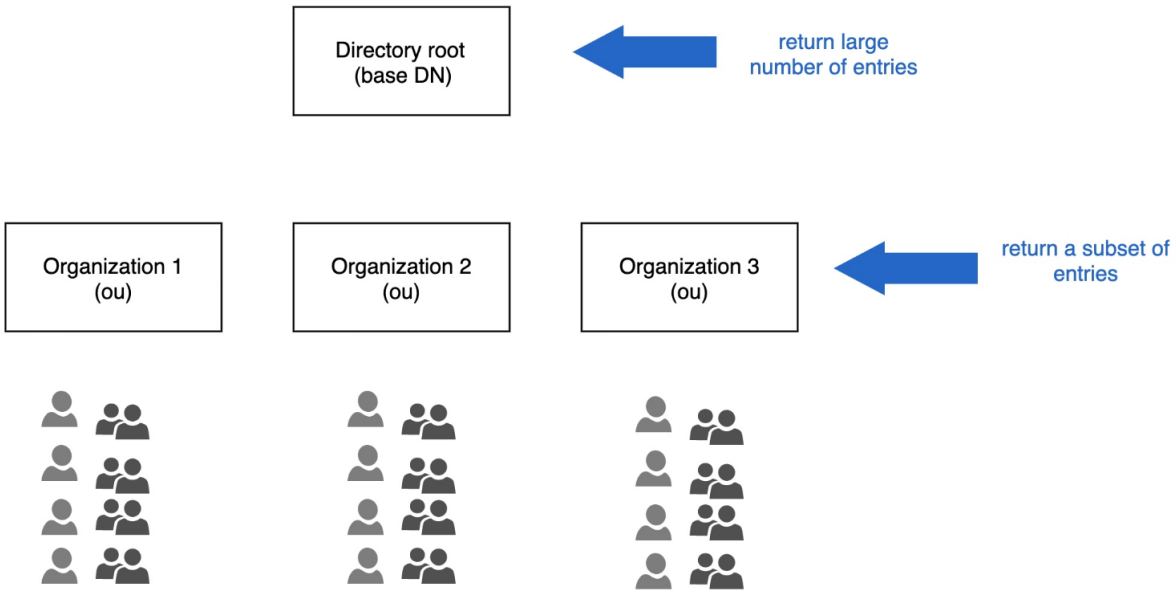
Custom search base support for LDAP group and user entity in SCIM group and user APIs

You can use custom search base for LDAP group and user entities in SCIM group APIs. For now, the custom search base is supported only through APIs, and not from your product UI.

The structure of the LDAP server is organized as a tree hierarchy. The leaves of the tree represent an individual user account or a group with a common name (cn) and other attributes. The classification of the LDAP tree is as follows:

- The root directory structure of the LDAP server is the Base DN, which is the distinguished name of the search base. It is further classified into subclasses such as country (c), domain component (dc), and organization name (o). For example, dc=abc,dc=com.
- The subtree of the LDAP server is Organizational Unit (ou) that you can use for a group search in a specific search base.

The custom search is useful in a scenario when you have multiple organizations in your LDAP server. By using the custom search base, you can isolate each organization in separate namespaces, and search the LDAP group and user under the organizations.



See the following notes:

- The custom search base specifies the subtree of the LDAP server for the search call for an entity type. The subtree overrides the base DN in search operations. For example, if the base DN is **o=ibm,c=us** and the search base is **ou=iUsers,o=ibm,c=us**, then all search calls are made under the **ou=iUsers,o=ibm,c=us** subtree. Multiple search bases can be configured for the same entity type.
- By using a custom search base for LDAP user entity in SCIM user APIs, you can configure search bases per LDAP connection. The corresponding search bases are used over base DN for the user search.

Enabling a custom search base

The **LDAP_GROUP_SEARCHBASE_LIST** and **LDAP_USER_SEARCHBASE_LIST** parameters define the list of custom search bases for LDAP groups and users. The data type must be a list. And, it is an optional parameter. To enable a custom search base, add the parameters with the list of search bases. See the following examples:

```
"LDAP_GROUP_SEARCHBASE_LIST":
["OU=searchbase1,DC=ibmtest,DC=com","OU=searchbase2,DC=ibmtest,DC=com","OU=searchbase3,DC=ibmtest,DC=com"]
```

```
"LDAP_USER_SEARCHBASE_LIST":
["OU=searchbase1,DC=ibmtest,DC=com","OU=searchbase2,DC=ibmtest,DC=com","OU=searchbase3,DC=ibmtest,DC=com"]
```

▶ The following API shows how to enable a custom search base:

To use the API, you need the host details and the base64-encoded password for the **"LDAP_BINDPASSWORD"** parameter. For more information, see Registering an LDAP connection by using IdP V3 API.

The following curl command is an example of adding a custom search base for an LDAP group entity while you create an LDAP connection:

```
curl -k -X POST 'https://<cluster_address>/idprovider/v3/auth/idsource/'
--header "Authorization: bearer $ACCESS_TOKEN" \
--header 'Content-Type: application/json'
-d
'{
 "LDAP_ID": "msad",
 "LDAP_URL": "ldap://corp.abc.com:389",
 "LDAP_BASEDN": "DC=ibmtest,DC=com",
 "LDAP_BINDDN": "CN=Administrator,cn=Users,DC=ibmtest,DC=com",
 "LDAP_BINDPASSWORD": "password in base64 encoded",
 "LDAP_TYPE": "Microsoft Active Directory",
 "LDAP_USERFILTER": "(&(sAMAccountName=%v)(objectclass=person))",
 "LDAP_GROUPFILTER": "(&(cn=%v)(objectcategory=group))",
 "LDAP_USERIDMAP": "user:sAMAccountName",
 "LDAP_GROUPIDMAP":"*:cn",
 "LDAP_GROUPMEMBERIDMAP": "memberOf:member"
 "LDAP_GROUP_SEARCHBASE_LIST":
["OU=searchbase1,DC=ibmtest,DC=com","OU=searchbase2,DC=ibmtest,DC=com","OU=searchbase3,DC=ibmtest,DC=com"]
}'
```

The following curl command is an example of adding a custom search base for an LDAP user entity while you create an LDAP connection:

```
curl -k -X POST 'https://<cluster_address>/idprovider/v3/auth/idsource/'
--header "Authorization: bearer $ACCESS_TOKEN" \
--header 'Content-Type: application/json'
-d
'{
 "LDAP_ID": "msad",
 "LDAP_URL": "ldap://corp.abc.com:389",
 "LDAP_BASEDN": "DC=ibmtest,DC=com",
 "LDAP_BINDDN": "CN=Administrator,cn=Users,DC=ibmtest,DC=com",
 "LDAP_BINDPASSWORD": "password in base64 encoded",
 "LDAP_TYPE": "Microsoft Active Directory",
 "LDAP_USERFILTER": "(&(sAMAccountName=%v)(objectclass=person))",
 "LDAP_GROUPFILTER": "(&(cn=%v)(objectcategory=group))",
 "LDAP_USERIDMAP": "user:sAMAccountName",
 "LDAP_GROUPIDMAP":"*:cn",
 "LDAP_GROUPMEMBERIDMAP": "memberOf:member"
 "LDAP_USER_SEARCHBASE_LIST":
["OU=searchbase1,DC=ibmtest,DC=com","OU=searchbase2,DC=ibmtest,DC=com","OU=searchbase3,DC=ibmtest,DC=com"]
}'
```

You can also enable the custom search base for LDAP group or LDAP user while you update the existing LDAP directory. For more information, see Updating an LDAP connection by using IdP V3 API.

Using realName as a query filter

You can use the **realmName** parameter in SCIM APIs to filter user and group queries.

When you use the **/identity/v3/idsource** API, the **realmName** parameter shows the IdP connection name. See the following example:

```
"urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User": {
        "realmName": "openLDAP"
}
```

You can filter user and group queries based on the **realmName**.

Get the RealmName for IdP

Use the following API to get the realmName.

See these API parameters:

- The protocol values can be **ldap** for LDAP servers, **oauth** for OpenShift, and **saml** for SAML and OIDC.
- If you have multiple LDAP servers, replace **[0]** with the LDAP server sequence number. Use 0 for the first LDAP server, 1 for the second LDAP server, and so on.

```
curl -k -X "Authorization: Bearer $ACCESS_TOKEN" "https://<cp-console>/idprovider/v3/auth/idsource?protocol=ldap" | jq -r
.idp | jq -r '.[0].idp_config.name'
```

The following output is an example:

```
OutPut: openLDAP
```

Use the realmName to query users and groups

Use the realmName that you get from the **https://<cp-console>/idprovider/v3/auth/idsource?protocol=<protocol-value>** API to run a filter on user and group queries.

- For SCIM users query, the filter name must be equal to **urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User:realmName** or **realmName**. See the following example:

```
https://<cp-console>/idmgmt/identity/api/v1/scim/users/?
filter=urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User:realmName eq "openLDAP"
```

Or

```
https://<cp-console>/idmgmt/identity/api/v1/scim/users/?filter=realmName  eq "openLDAP"
```

- For SCIM groups query, the filter name must be equal to **`urn:ietf:params:scim:schemas:extension:ibmcp:2.0:Group:realmName`** or **`realmName`**. See the following example:

```
https://<cp-console>/idmgmt/identity/api/v1/scim/groups/?
filter=urn:ietf:params:scim:schemas:extension:ibmcp:2.0:Group:realmName eq "openLDAP"
```

Or

```
https://<cp-console>/idmgmt/identity/api/v1/scim/groups/?filter=realmName  eq "openLDAP"
```

Note: The only supported operator for realmName is **`equal`** or **`eq`**. You can use any other filter with the **`realmName`** by using **`and`** or **`or`** operator.

See the following example:

```
https://<cp-console>/idmgmt/identity/api/v1/scim/users/?
filter=urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User:realmName eq "openLDAP" and displayName eq "icpuser1"
https://<cp-console>/idmgmt/identity/api/v1/scim/users/?
filter=urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User:realmName eq "openLDAP" or displayName eq "icpuser1"
```

Limitations

The use of a realmName filter has the following limitations:

- The SCIM query returns all users for an identity provider. You must set the pagination to query identity providers with many users.
- For OpenShift, the groups are not returned when you run a query by using SCIM.
- You cannot run a query on OIDC users by using SCIM.
- For SAML with LDAP dependency, the SCIM query returns the LDAP connection name as realmName.

Enabling LDAP Nested Search

Nested search enables the Cloud Pak users to onboard the users or groups that belong to Nested hierarchy in the LDAP tree.

You can enable the nested search for the following:

- Microsoft Active Directory (MSAD) server
- Tivoli Directory Server (TDS) and Security Directory Server (SDS)

IMPORTANT: Nested Search can be an expensive operation. If the Nested search is enabled, it can affect the response time.

Nested search for MSAD

You can enable Nested search for Microsoft Active Directory by using the console in the following scenarios:

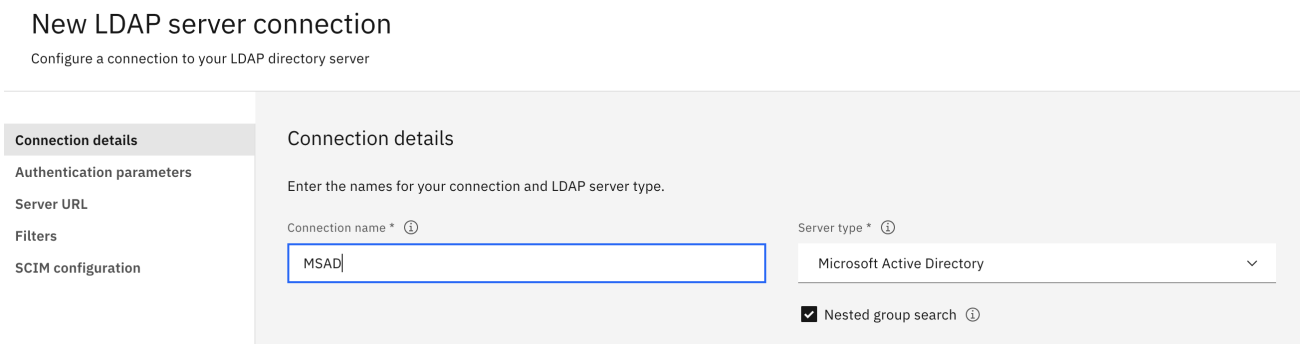
- Creating the MSAD connection
- Editing connection for existing MSAD

Enabling Nested search for MSAD while creating the connection

1. Log on to the console by using the URL and enter the username and password.

   Note: The console can be Platform UI or Common UI. For more information, see Accessing your cluster by using the console.

2. From the navigation menu, click **Identity and access** > **Identity providers**.
3. Click **Create Connection**. The LDAP connection page is displayed.
4. Select the Server type as **`Microsoft Active Directory`**. The **Nested group search** option is enabled.
5. Select the **Nested group search** to enable the nested search for the Microsoft Active Directory.



Enabling Nested search for existing MSAD connection

1. Log on to the console by using the URL and enter the username and password.

   Note: The console can be Platform UI or Common UI. For more information, see Accessing your cluster by using the console.

2. From the navigation menu, click **Identity and access** > **Identity providers**.

3. In the **Identity providers** page, the name of the MSAD connection is displayed. Select the MSAD connection name to edit and click **Edit connection**. The **Edit your LDAP connection** page is displayed.

4. Select the **Nested group search** option to enable the nested search for an existing Microsoft Active Directory server.

5. Click **Save**.

Nested search support for Tivoli Directory Server (TDS) and Security Directory Server (SDS)

For now, TDS and SDS nested search can only be configured using APIs, not from the product UI. You can enable nested search for Tivoli Directory Server (TDS) and Security Directory Server (SDS) by two ways:

- **TDS or SDS recommended way:** From IBM Cloud Pak foundational services version 3.19.9, 3.22, and later, you can enable nested search for TDS/SDS by creating nested group data as per TDS or SDS recommendations.
- **Recursive method:** From IBM Cloud Pak foundational services version 3.19.9 and 3.23, IM supports recursive method to enable nested search for TDS/SDS if LDAP nested group data set up is not created as per TDS or SDS recommendations.

IMPORTANT: It is mandatory to use the TDS or SDS recommended way method to enable nested search for TDS/SDS when you have created the LDAP nested group data set up as per TDS or SDS recommendations. In this case, if you use the recursive method, nested group functionality might not work properly.

IMPORTANT: It is mandatory to use the recursive method to enable nested search for TDS/SDS when you have not created the LDAP nested group data set up as per TDS or SDS recommendations. In this case, if you use the TDS or SDS recommended way method, nested group functionality might not work properly.

Tivoli Directory Server (TDS) and Security Directory Server (SDS) recommended way to create Nested group

IMPORTANT: The IBM Cloud Pak foundational services IM will be retrieving the Nested Group members only if data has been created with the TDS or SDS recommended format.

In IBM TDS/SDS LDAP server, a nested group is created by adding the **`ibm-nestedGroup`** object class in the group schema. And, the nested child groups must be added using **`ibm-memberGroup`** attributes.

The **`ibm-nestedGroup`** and **`ibm-memberGroup`** attributes must be present in the nested group schema as following:

```
dn: cn=TeamStaff,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com
cn: TeamStaff
objectClass: groupOfUniqueNames
objectClass: top
objectclass: ibm-nestedGroup
ibm-memberGroup: cn=TeamDoctors,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com
ibm-memberGroup: cn=Coaches,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com
```

IM uses operational attribute that is provided by TDS/SDS LDAP to query aggregate group membership as following. Ensure that LDAP data is created as per the TDS or SDS recommended format.

- **`ibm-allMembers`**: For a given group entry, the **`ibm-allMembers`** operational attribute returns all nested members.
- **`ibm-allGroups`**: For a given user entry, the **`ibm-allGroups`** returns all nested groups.

Note:

- If your data for nested groups is set up as per the TDS recommended way, IBM Cloud Pak foundational services IM will be able to use the **`ibm-allMembers`** attribute, and the **`ibm-allMembers`** attribute will return all nested members for a group, and **`ibm-allGroups`** attribute will return all nested groups for a user.
- If your data for nested group is not set up as per the TDS recommended way, Nested search feature will not work as expected, means, when IBM Cloud Pak foundational services IM uses the **`ibm-allMembers`** attribute, the **`ibm-allMembers`** attribute will only return all first-level members for a group, and **`ibm-allGroups`** attribute will return only the parent groups for a user.

Enabling Nested search feature in TDS/SDS recommended way by using SCIM API

The following parameters are used in directory management API to enable nested search in TDS/SDS recommended way:

- The **LDAP_NESTEDSEARCH** parameter in the directory management API is used to support the nested search feature in SCIM APIs. Set the **LDAP_NESTEDSEARCH** parameter to **true** to enable nested search option for TDS/SDS.
- From IBM Cloud Pak foundational services version 3.19.9 and 3.23,**LDAP_TDS_NESTEDCONFIG** parameter is introduced in directory management API. The value of this parameter is used to determine that which method (TDS/SDS recommended way or recursive method) is used to enable nested search for TDS/SDS.
    - If this parameter is set as **true**, the TDS/SDS recommended method will be used for nested search. However, make sure that to use TDS/SDS recommended method, you must create the LDAP nested group data set up as per TDS or SDS recommendations.
    - If this parameter is set as **false**, the recursive method logic will be used for nested search. Note that if you have not created the LDAP nested group data set up as per TDS or SDS recommendations, you need to use recursive method to enable nested search for TDS/SDS.
    - If the parameter is not provided, its default value will be considered as **false**, that means by default, recursive method is used to enable nested search.

Note: To use the directory management APIs, you need to get the host details. To get the host details, see Getting host details. And, use the base64-encoded password for the **"LDAP_BINDPASSWORD"** parameter. To encode the password, use the following command:

```
echo -n <password> | base64

Following is an example output:
```cmd {: pre}

UGFzc3cwcmQ=

#### Enabling nested search in TDS/SDS recommended way while creating LDAP connection by using the Directory management API

The following API shows how to enable the nested search option in TDS/SDS recommended way while creating an LDAP connection by using the Directory management API.

<dl>
<dt><b>API version</b></dt>
<dd>1.0.0</dd>
<dt><b>API URI components</b></dt>
<dd>
<dl>
<dt><b>Scheme</b></dt>
<dd><pre>HTTPS</pre></dd>
<dt><b>Host IP</b></dt>
<dd><pre>Cluster Master Host</pre></dd>
<dt><b>Port number</b></dt>
<dd><pre>Cluster Master API Port</pre></dd>
<dt><b>Path</b></dt>
<dd><pre>idmgmt/identity/api/v1/directory/</pre></dd>
<dt><b>Command</b></dt>
<dd><pre>POST</pre></dd>
</dl>
</dd>
<dt><b>Command output format</b></dt>
<dd><pre>application/json</pre></dd>
</dl>

**Note:** For more information about the LDAP parameters, see [Configuring LDAP authentication](configure_ldap.html).

The curl command resembles the following code:

```cmd {: codeblock}
curl -k -X POST \
-H "Authorization: Bearer ${ACCESS_TOKEN}" \
-H "content-type: application/json" \
"https://${CP_CONSOLE_URL}/idmgmt/identity/api/v1/directory/" \
-d
'{
  "TYPE": "LDAP",
  "LDAP_ID":"SDS",
  "LDAP_REALM":"SDSRealm",
  "LDAP_HOST":"ip",
  "LDAP_PORT":"389",
  "LDAP_IGNORECASE":"true",
  "LDAP_BASEDN":"dc=ibm,dc=com",
  "LDAP_BINDDN":"cn=admin",
  "LDAP_BINDPASSWORD":"password in base64 encoded",
  "LDAP_TYPE":"IBM SecureWay Directory Server",
  "LDAP_USERFILTER":"(&(uid=%v)(objectclass=person))",
  "LDAP_GROUPFILTER":"(&(cn=%v)(objectclass=groupOfUniqueNames))",
  "LDAP_USERIDMAP":"*:uid",
  "LDAP_GROUPIDMAP":"*:cn",
  "LDAP_GROUPMEMBERIDMAP":"groupOfUniqueNames:uniquemember",
  "LDAP_NESTEDSEARCH": "true"
  "LDAP_TDS_NESTEDCONFIG": "true"
}'
```

Enabling nested search in TDS/SDS recommended way while updating existing LDAP connection by using the Directory management API

The following API shows how to enable nested search in TDS/SDS recommended way while updating existing LDAP connection.

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster Master Host

Port number

Cluster Master API Port

Path

idmgmt/identity/api/v1/directory/ldap/{id}

Command

PUT

Command output format

application/json

Note: Use **id** to update the LDAP directory. To get the id, see Getting information about all LDAP connections.

The curl command resembles the following code:

```
curl -k -X PUT \
-H "Authorization: Bearer ${ACCESS_TOKEN}" \
-H "content-type: application/json" \
"https://${CP_CONSOLE_URL}/idmgmt/identity/api/v1/directory/ldap/<id>" \

-d
 '{
    "TYPE": "LDAP",
    "LDAP_ID":"SDS",
    "LDAP_REALM":"SDSRealm",
    "LDAP_HOST":"ip",
    "LDAP_PORT":"389",
    "LDAP_IGNORECASE":"true",
    "LDAP_BASEDN":"dc=ibm,dc=com",
    "LDAP_BINDDN":"cn=admin",
    "LDAP_BINDPASSWORD":"password in base64 encoded",
    "LDAP_TYPE":"IBM SecureWay Directory Server",
    "LDAP_USERFILTER":"(&(uid=%v)(objectclass=person))",
    "LDAP_GROUPFILTER":"(&(cn=%v)(objectclass=groupOfUniqueNames))",
    "LDAP_USERIDMAP":"*:uid",
    "LDAP_GROUPIDMAP":"*:cn",
    "LDAP_GROUPMEMBERIDMAP":"groupOfUniqueNames:uniquemember",
    "LDAP_NESTEDSEARCH": "true",
    "LDAP_TDS_NESTEDCONFIG": "true"
  }'
```

Recursive approach to enable nested search for TDS/SDS

You might have LDAP directories for years and your data set up for nested search might not be as per Tivoli Directory Server (TDS) and Security Directory Server (SDS) recommendations. In such case, from IBM Cloud Pak foundational services version 3.19.9, 3.23, and later, IM supports recursive approach to enable nested search. It is mandatory to use the recursive method for nested search when you have not created the LDAP nested group data set up as per TDS or SDS recommendations. Otherwise, nested group functionality might not work properly.

Enabling recursive method to support nested search for TDS/SDS by using SCIM API

The following parameters are used in directory management API to enable recursive method/logic to support nested search for TDS/SDS:

- **LDAP_NESTEDSEARCH** : The **LDAP_NESTEDSEARCH** parameter in the Directory management API is used to support the nested search feature in SCIM APIs. Set the **LDAP_NESTEDSEARCH** parameter to **true** to enable recursive logic for nested search in TDS/SDS.
- **LDAP_TDS_NESTEDCONFIG**: From IBM Cloud Pak foundational services version 3.19.9 and 3.23,**LDAP_TDS_NESTEDCONFIG** parameter is introduced in directory management API. The value of this parameter is used to determine that which method (TDS/SDS recommended way or recursive method) is used to enable nested search for TDS/SDS.
  - If this parameter is set as **true**, the TDS/SDS recommended method will be used for nested search. However, make sure that to use TDS/SDS recommended method, you must create the LDAP nested group data set up as per TDS or SDS recommendations.
  - If this parameter is set as **false**, the recursive method logic will be used for nested search. Note that if you have not created the LDAP nested group data set up as per TDS or SDS recommendations, you need to use recursive method to enable nested search for TDS/SDS.

- If the parameter is not provided, its default value will be considered as **`false`**,that means by default, recursive method is used to enable nested search.

Note:

- For SCIM group API, IM returns only first level members (users and groups) for any groups.
- For SCIM users API, IM returns all nested groups of any users.
- To use the directory management APIs, you need to get the host details. To get the host details, see Getting host details. And, use the base64-encoded password for the **`"LDAP_BINDPASSWORD"`** parameter. To encode the password, use the following command:

  ```
  echo -n <password> | base64
  ```

  Following is an example output:

  ```
  UGFzc3cwcmQ=
  ```

Enabling recursive logic to support nested search for TDS/SDS while creating an LDAP connection

The following API shows how to enable recursive logic for nested search while creating an LDAP connection by using the Directory management API.

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster Master Host

Port number

Cluster Master API Port

Path

idmgmt/identity/api/v1/directory/

Command

POST

Command output format

application/json

Note: For more information about the LDAP parameters, see Configuring LDAP authentication.

Curl command resembles the following code:

```
curl -k -X POST \
-H "Authorization: Bearer ${ACCESS_TOKEN}" \
-H "content-type: application/json" \
"https://${CP_CONSOLE_URL}/idmgmt/identity/api/v1/directory/" \
-d
 '{
    "TYPE": "LDAP",
    "LDAP_ID":"SDS",
    "LDAP_REALM":"SDSRealm",
    "LDAP_HOST":"ip",
    "LDAP_PORT":"389",
    "LDAP_IGNORECASE":"true",
    "LDAP_BASEDN":"dc=ibm,dc=com",
    "LDAP_BINDDN":"cn=admin",
    "LDAP_BINDPASSWORD":"password in base64 encoded",
    "LDAP_TYPE":"IBM SecureWay Directory Server",
    "LDAP_USERFILTER":"(&(uid=%v)(objectclass=person))",
    "LDAP_GROUPFILTER":"(&(cn=%v)(objectclass=groupOfUniqueNames))",
    "LDAP_USERIDMAP":"*:uid",
    "LDAP_GROUPIDMAP":"*:cn",
    "LDAP_GROUPMEMBERIDMAP":"groupOfUniqueNames:uniquemember",
    "LDAP_NESTEDSEARCH": "true",
    "LDAP_TDS_NESTEDCONFIG": "false"
  }'
```

Enabling recursive logic to support nested search for TDS/SDS while updating the existing LDAP directory

By using Directory management API, you can also enable the Nested search using recursive method while updating the existing LDAP directory.

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster Master Host

Port number

Cluster Master API Port

Path

idmgmt/identity/api/v1/directory/ldap/{id}

Command

PUT

Command output format

application/json

Note: Use `id` to update the LDAP directory. To get the id, see Getting information about all LDAP connections.

Curl command resembles the following code:

```
curl -k -X PUT \
-H "Authorization: Bearer ${ACCESS_TOKEN}" \
-H "content-type: application/json" \
"https://${CP_CONSOLE_URL}/idmgmt/identity/api/v1/directory/ldap/<id>" \

-d
 '{
    "TYPE": "LDAP",
    "LDAP_ID":"SDS",
    "LDAP_REALM":"SDSRealm",
    "LDAP_HOST":"ip",
    "LDAP_PORT":"389",
    "LDAP_IGNORECASE":"true",
    "LDAP_BASEDN":"dc=ibm,dc=com",
    "LDAP_BINDDN":"cn=admin",
    "LDAP_BINDPASSWORD":"password in base64 encoded",
    "LDAP_TYPE":"IBM SecureWay Directory Server",
    "LDAP_USERFILTER":"(&(uid=%v)(objectclass=person))",
    "LDAP_GROUPFILTER":"(&(cn=%v)(objectclass=groupOfUniqueNames))",
    "LDAP_USERIDMAP":"*:uid",
    "LDAP_GROUPIDMAP":"*:cn",
    "LDAP_GROUPMEMBERIDMAP":"groupOfUniqueNames:uniquemember",
    "LDAP_NESTEDSEARCH": "true",
    "LDAP_TDS_NESTEDCONFIG": "false"
  }'
```

Changing LDAP cache settings

Changing the Lightweight Directory Access Protocol (LDAP) cache settings that are used for authentication in IBM Cloud Pak® for AIOps.

The LDAP cache setting parameters are listed in the following tables. For more information, see LDAP User Registry (ldapRegistry).

Note: Specify a positive integer followed by a unit of time, which can be hours (h), minutes (m), seconds (s), or milliseconds (ms). For example, specify 500 milliseconds as 500ms. You can include multiple values in a single entry. For example, 1s500ms is equivalent to 1.5 seconds. This note is applicable only to the `LDAP_ATTR_CACHE_TIMEOUT` and `LDAP_SEARCH_CACHE_TIMEOUT` parameters.

Table 1. LDAP attribute cache properties

Parameter	Description	Default value
`LDAP_ATTR_CACHE_ENABLED`	Enable or disable LDAP attribute cache.	true
`LDAP_ATTR_CACHE_SIZE`	Number of entities that can be stored in the cache.	2000
`LDAP_ATTR_CACHE_SIZELIMIT`	Maximum number of attributes per LDAP entity that are cached.	2000
`LDAP_ATTR_CACHE_TIMEOUT`	Maximum time that the contents of the LDAP attribute cache are available. When the specified time elapses, the LDAP attribute cache is cleared.	1200s

Table 2. LDAP search results cache properties

Parameter	Description	Default value
`LDAP_SEARCH_CACHE_ENABLED`	Enable or disable LDAP search results cache.	true
`LDAP_SEARCH_CACHE_SIZE`	Number of search results that are stored in the cache.	2000
`LDAP_SEARCH_CACHE_SIZELIMIT`	Maximum number of results that can be cached for a single LDAP search.	2000
`LDAP_SEARCH_CACHE_TIMEOUT`	Maximum time that the contents of the search results cache are available. When the specified time elapses, the search results cache is cleared.	1200s

Changing the parameter values by using kubectl

To change the parameter values, complete the following steps:

1. Install the Kubernetes CLI (kubectl) tool.
2. Edit the **platform-auth-idp** ConfigMap.

   ```
   kubectl -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

3. Change the following attribute values as required:

   - **LDAP_ATTR_CACHE_ENABLED**
   - **LDAP_ATTR_CACHE_SIZE**
   - **LDAP_ATTR_CACHE_SIZELIMIT**
   - **LDAP_ATTR_CACHE_TIMEOUT**
   - **LDAP_SEARCH_CACHE_ENABLED**
   - **LDAP_SEARCH_CACHE_SIZE**
   - **LDAP_SEARCH_CACHE_SIZELIMIT**
   - **LDAP_SEARCH_CACHE_TIMEOUT**

4. Save and close the ConfigMap.
5. Restart the **platform-auth-service** pods

   ```
   kubectl -n <your-foundational-services-namespace> delete pod -l k8s-app=platform-auth-service
   ```

6. Wait for some time. Then, check the status of the **platform-auth-service** pods. The status must show as **4/4 Running** for all the pods.

   ```
   kubectl -n <your-foundational-services-namespace> get pods | grep platform-auth-service
   ```

Changing the parameter values by using the console

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for **platform-auth-idp**.
4. Click **...** > **Edit Config Map**.
5. Change the following attribute values as required:
   - **LDAP_ATTR_CACHE_ENABLED**
   - **LDAP_ATTR_CACHE_SIZE**
   - **LDAP_ATTR_CACHE_SIZELIMIT**
   - **LDAP_ATTR_CACHE_TIMEOUT**
   - **LDAP_SEARCH_CACHE_ENABLED**
   - **LDAP_SEARCH_CACHE_SIZE**
   - **LDAP_SEARCH_CACHE_SIZELIMIT**
   - **LDAP_SEARCH_CACHE_TIMEOUT**
6. Click **Save**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate **platform-auth-service**.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the **platform-auth-service** pods with the latest ConfigMap values.
11. Click **platform-auth-service**.
12. Wait for some time. Then, check the status of the **platform-auth-service** pods in the **Pods** pane. The status of all the pods must show as **4/4** under the **Ready** field name.

Changing LDAP search settings

Changing the Lightweight Directory Access Protocol (LDAP) search settings in IBM Cloud Pak® for AIOps.

Configure attributes consideration for LDAP search

By default, IBM Cloud Pak for AIOps searches for users and groups on both the common name (CN) and identifier (ID) attributes as configured in the LDAP connection settings. Examples of ID attributes are **uid**, **sAMAccountName**, and **emailAddress**.

This behavior can be changed by setting the following configuration parameters:

The following two attributes are used for the search:

- **LDAP_SEARCH_CN_ATTR_ONLY** - Search by CN only. Default value is **false**.
- **LDAP_SEARCH_ID_ATTR_ONLY**- Search by ID only. Default value is **false**.

To use only the CN attribute for searching users or groups, set **LDAP_SEARCH_CN_ATTR_ONLY** value to **true**. To use only the ID attribute for searching users or groups, set **LDAP_SEARCH_ID_ATTR_ONLY** value to **true**.

Following are the steps to change the values of the LDAP search config parameters:

Changing the attribute values by using kubectl

To change the values, complete the following steps:

1. Install the `kubectl` CLI.
2. Edit the `platform-auth-idp` ConfigMap.

   ```
   kubectl -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

   Set the following attribute values as required to `true` or `false`.

   - `LDAP_SEARCH_CN_ATTR_ONLY`
   - `LDAP_SEARCH_ID_ATTR_ONLY`

   Note: Set only one value to `true`.
3. Save and close the ConfigMap.
4. Restart the `platform-auth-service` pods

   ```
   kubectl -n <your-foundational-services-namespace> delete pod -l k8s-app=platform-auth-service
   ```
5. Wait for some time. Then, check the status of the `platform-auth-service` pods. The status must show as `4/4 Running` for all the pods.

   ```
   kubectl -n <your-foundational-services-namespace> get pods | grep platform-auth-service
   ```

Changing the attribute values by using the console

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for `platform-auth-idp`.
4. Click **...** > **Edit Config Map**.
5. Change either of the following attribute values to `true`, as required:

   - `LDAP_SEARCH_CN_ATTR_ONLY`
   - `LDAP_SEARCH_ID_ATTR_ONLY`

   Note: Set only one value to `true`.
6. Click **Save**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate `platform-auth-service`.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the `platform-auth-service` pods with the latest ConfigMap values.
11. Click `platform-auth-service`.
12. Wait for some time. Then, check the status of the `platform-auth-service` pods in the **Pods** pane. The status of all the pods must show as `4/4` under the **Ready** field name.

Changing the LDAP search size limit and time limit values

Following are the LDAP search limit configuration parameters:

- `LDAP_SEARCH_SIZE_LIMIT`: Default value is "50". Range of values is 50 - 100.
- `LDAP_SEARCH_TIME_LIMIT`: Default value is "5". Range of values is 5 - 50.

Changing the values by using kubectl

To change the values, complete the following steps:

1. Set up the `kubectl` CLI. For more information, see Install tools.
2. Edit the `platform-auth-idp` ConfigMap.

   ```
   kubectl -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

   Change the following values as required:

   - `LDAP_SEARCH_SIZE_LIMIT`
   - `LDAP_SEARCH_TIME_LIMIT`
3. Save and close the ConfigMap.
4. Restart the `platform-auth-service` pods

   ```
   kubectl -n <your-foundational-services-namespace> delete pod -l k8s-app=platform-auth-service
   ```
5. Wait for some time. Then, check the status of the `platform-auth-service` pods. The status must show as `4/4 Running` for all the pods.

   ```
   kubectl -n <your-foundational-services-namespace> get pods | grep platform-auth-service
   ```

Changing the values by using the console

1. Log in to the console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for `platform-auth-idp`.
4. Click **...** > **Edit Config Map**.
5. Change the following attribute values, as required:
   - `LDAP_SEARCH_SIZE_LIMIT`
   - `LDAP_SEARCH_TIME_LIMIT`
6. Click **Submit**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate `platform-auth-service`.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Submit** without making any change. This step is to reload the `platform-auth-service` pods with the latest ConfigMap values.
11. Click `platform-auth-service`.
12. Wait for some time. Then, check the status of the `platform-auth-service` pods in the **Pods** pane. The status of all the pods must show as `4/4` under the **Ready** field name.

Change LDAP search wild characters usage

Following is the LDAP search wild characters configuration parameter:

- `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS`: Default value is "false".

In IBM Cloud Pak for AIOps, wild characters such as asterisk are used in the search string to match the substring value of the LDAP server attribute.

Set `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS` value to `true` to exclude wild characters (*) in the search string and to match the exact value of the LDAP server attribute.

Following is the default search string:

```
(|(&(cn=*<searchstring>*)(objectclass=person))(&(uid=*<searchstring>*)(objectclass=person)))
```

Following is the search string when the `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS` is set to `true`:

```
(|(&(cn=<searchstring>)(objectclass=person))(&(uid=<searchstring>)(objectclass=person)))
```

To change the wild characters usage in the search string, complete the following steps:

Changing the wild characters configuration parameter by using kubectl

To change the values, complete the following steps:

1. Install the Kubernetes CLI (kubectl) tool.
2. Edit the `platform-auth-idp` ConfigMap.

   ```
   kubectl -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

   Change the following value as required:

   - `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS`
3. Save and close the ConfigMap.
4. Restart the `platform-auth-service` pods

   ```
   kubectl -n <your-foundational-services-namespace> delete pod -l k8s-app=platform-auth-service
   ```
5. Wait for some time. Then, check the status of the `platform-auth-service` pods. The status must show as `4/4 Running` for all the pods.

   ```
   kubectl -n <your-foundational-services-namespace> get pods | grep platform-auth-service
   ```

Changing the wild characters configuration parameter by using the console

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Configuration** > **Config Maps.**
3. Search for `platform-auth-idp`.
4. Click **...** > **Edit Config Map**.
5. Change the following value, as required:
   - `LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS`
6. Click **Save**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate `platform-auth-service`.
9. Click **...** > **Edit**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the `platform-auth-service` pods with the latest ConfigMap values.
11. Click `platform-auth-service`.
12. Wait for some time. Then, check the status of the `platform-auth-service` pods in the *Pods* pane. The status of all the pods must show as `4/4` under the **Ready** field name.

Changing SCIM LDAP search limit settings

You can change the SCIM LDAP search limits settings in IBM Cloud Pak® for AIOps.

Following are the SCIM LDAP search limit description and configuration parameters:

- **`SCIM_LDAP_SEARCH_SIZE_LIMIT`**: It limits the number of entries that can be returned in the SCIM user/group API call. The default value is 4500. There is no limitation for the value, you can set any value based on the number of entries that is expected to return from the LDAP. When the LDAP returns the entries more than its configured value, it displays **`Size Limit Exceeded`** error.

  Note: Consider the LDAP maximum page size value while tuning the **`SCIM_LDAP_SEARCH_SIZE_LIMIT`** parameter. For now, IM SCIM does not fully support the LDAP pagination.

- **`SCIM_LDAP_SEARCH_TIME_LIMIT`**: It limits the maximum wait time for the SCIM users/groups search API call. The default value is 10 seconds and this value is configurable. If the wait time is greater than 10 seconds or the configured value, it displays **`Time Limit Exceeded`** error. It is a client-side restriction. The LDAP server can also provide a limit on the search time.

  Note: If the configured value of the **`SCIM_LDAP_SEARCH_TIME_LIMIT`** is greater than 30 seconds, increase the OpenShift Platform Container route ingress level timeout.

  It can be adjusted by using the following annotations:

  ```
  oc annotate route cp-console -n <your-foundational-services-namespace> --overwrite
  haproxy.router.openshift.io/timeout=60s
  ```

Changing the values by using the console

1. Log in to the OpenShift Container console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for **`platform-auth-idp`**.
4. Click **...** > **Edit Config Map**.
5. Change the following attribute values, as required:
   - **`SCIM_LDAP_SEARCH_SIZE_LIMIT`**
   - **`SCIM_LDAP_SEARCH_TIME_LIMIT`**
6. Click **Submit**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate **`platform-auth-service`**.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Submit** without making any change. This step is to reload the **`platform-auth-service`** pods with the latest ConfigMap values.
11. Click **`platform-auth-service`**.
12. Wait for some time. Then, check the status of the **`platform-auth-service`** pods in the **Pods** pane. The status of all the pods must show as **`4/4`** under the **Ready** field name.

Adding custom SCIM attributes

You can add custom SCIM user and group attributes.

The new attributes must be defined in the extended SCIM user schema (urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User). They must follow the SCIM specification of attribute properties, including name, type, multiValued, description, required, caseExact, mutability, returned, and uniqueness. For more information, see Resource Schema Representation.

Following table provides the payload of the attributes.

Table 1. Custom attribute payload

Subattribute field name	Mandatory or not	Default value	Allowed value
schemaId	Yes	NA	urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User or urn:ietf:params:scim:schemasibmcp:2.0:User; urn:ietf:params:scim:schemas:extension:ibmcp:2.0:Group or urn:ietf:params:scim:schemasibmcp:2.0:Group
screenName	Yes	NA	Any alphanumeric characters between 2 and 30 characters in length; may contain space
name	Yes	NA	Any alphanumeric characters between 2 and 30 characters in length
description	No	NA	NA
isSensitive	No	false	false
type	Yes	string	string, complex
multiValued	No	false	true, false
required	No	false	true, false
caseExact	No	false	false
mutability	No	readWrite	readWrite
returned	No	default	default
uniqueness	No	none	none

Important: The SchemaIds urn:ietf:params:scim:schemasibmcp:2.0:User and urn:ietf:params:scim:schemasibmcp:2.0:Group might be deprecated in a future release. Hence, use the SchemaId urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User or urn:ietf:params:scim:schemas:extension:ibmcp:2.0:Group when you create any custom SCIM attributes.

See these notes:

- For each attribute, you must provide the mandatory subattribute as described in Table 1. Otherwise, the POST API operation returns an error.
- **schemaId** is a mandatory field and its value must be **urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User** or **urn:ietf:params:scim:schemasibmcp:2.0:User** for user, and **urn:ietf:params:scim:schemas:extension:ibmcp:2.0:Group** or **urn:ietf:params:scim:schemasibmcp:2.0:Group** for group.
- For each attribute, it is ideal to add the nonmandatory subattribute fields with values in the payload. Otherwise, the default values are added as described in the Table 1.
- Attribute key and name must have the same value.
- IM supports only default values for these subattributes: isSensitive, caseExact, mutability, returned, and uniqueness. If you provide custom values, then IM changes the values of these subattributes to their default values and then creates the custom SCIM attribute.

Add custom SCIM attributes

Use the **POST** operation with the **idmgmt/identity/api/v2/scim/attributes** API to add one or more custom SCIM attributes in the users and groups section.

Before you use the API, create these environment variables to use with the APIs.

- Console route. For more information about how to get the console route, see Getting the console URL.

  ```
  export CP_CONSOLE_URL=$CP_CONSOLE_URL_ROUTE
  ```

- Access token. For more information, see Preparing to run component or management API commands.

  1. Run the following command to get all authentication tokens. Then, copy the access token from the output.

     ```
     curl -k -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -d
     "grant_type=password&username=admin&password=admin&scope=openid"
     https://$CP_CONSOLE_URL/idprovider/v1/auth/identitytoken
     ```

  2. Create an environment variable for the access token.

     ```
     export ACCESS_TOKEN= <Access-token>
     ```

Add custom SCIM attribute for users

Run the following curl command to add custom attribute only for users:

```
curl -k -X POST \
  -H "Authorization: Bearer $ACCESS_TOKEN" \
  -H "content-type: application/json" \
  "https://$CP_CONSOLE_URL/idmgmt/identity/api/v2/scim/attributes" \
  -d '{
      "user": {
        "userAttribute1": {
            "schemaId": "urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User",
            "screenName": "userAttribute1",
            "isSensitive": false,
            "name" : "userAttribute1",
            "type" : "string",
            "multiValued" : false,
            "description" : "Custom User Attribute 1",
            "required" : true,
            "caseExact" : false,
            "mutability" : "readWrite",
            "returned" : "default",
            "uniqueness" : "none"
           }
      }
   }'
```

Add custom SCIM attribute for groups

Run the following curl command to add custom attribute only for groups:

```
curl -k -X POST \
  -H "Authorization: Bearer $ACCESS_TOKEN" \
  -H "content-type: application/json" \
  "https://$CP_CONSOLE_URL/idmgmt/identity/api/v2/scim/attributes" \
  -d '{
        "group": {
            "groupAttribute1": {
              "schemaId": "urn:ietf:params:scim:schemas:extension:ibmcp:2.0:Group",
              "screenName": "groupAttribute1",
              "isSensitive": false,
              "name" : "groupAttribute1",
              "type" : "string",
              "multiValued" : false,
              "description" : "Custom Group Attribute 1",
              "required" : true,
              "caseExact" : false,
              "mutability" : "readWrite",
              "returned" : "default",
              "uniqueness" : "none"
            }
        }
      }'
```

Add custom SCIM attribute for users and groups

Run the following curl command to add custom attribute for both users and groups:

```
curl -k -X POST \
  -H "Authorization: Bearer $ACCESS_TOKEN" \
  -H "content-type: application/json" \
  "https://$CP_CONSOLE_URL/idmgmt/identity/api/v2/scim/attributes" \
  -d '{
      "user": {
        "userAttribute1": {
          "schemaId": "urn:ietf:params:scim:schemas:extension:ibmcp:2.0:User",
          "screenName": "userAttribute1",
          "isSensitive": false,
          "name" : "userAttribute1",
          "type" : "string",
          "multiValued" : false,
          "description" : "Custom User Attribute 1",
          "required" : true,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
           }
        },
        "group": {
            "groupAttribute1": {
              "schemaId": "urn:ietf:params:scim:schemas:extension:ibmcp:2.0:Group",
              "screenName": "groupAttribute1",
              "isSensitive": false,
              "name" : "groupAttribute1",
              "type" : "string",
              "multiValued" : false,
              "description" : "Custom Group Attribute 1",
              "required" : true,
              "caseExact" : false,
              "mutability" : "readWrite",
              "returned" : "default",
              "uniqueness" : "none"
            }
        }
      }'
```

Updating a custom SCIM attribute

Use the **PUT** operation with the **idmgmt/identity/api/v2/scim/attributes/<resourceType>/<attributekey>** API to modify a custom SCIM attribute in the users and groups section.

Update user attribute

To modify a user attribute, use the **idmgmt/identity/api/v2/scim/attributes/users/<attributekey>** API.

- To get the user **attributekey** value, use the following command:

  ```
  curl -sk -X GET --header "Authorization: Bearer $ACCESS_TOKEN" --header "Content-Type: application/json"
  "https://$CP_CONSOLE_URL:443/idmgmt/identity/api/v2/scim/attributes/users"
  ```

- To modify the user attribute, use the following curl command:

  ```
  curl -k -X PUT \
    -H "Authorization: Bearer $ACCESS_TOKEN" \
    -H "content-type: application/json" \
    "https://$CP_CONSOLE_URL/idmgmt/identity/api/v2/scim/attributes/user/<attributekey>" \
    -d '{
            "schemaId": "urn:ietf:params:scim:schemasibmcp:2.0:User",
            "screenName": "userAttribute1",
            "isSensitive": false,
            "name" : "userAttribute1",
            "type" : "string",
            "multiValued" : false,
            "description" : "Custom User Attribute 1",
            "required" : true,
            "caseExact" : false,
            "mutability" : "readWrite",
            "returned" : "default",
            "uniqueness" : "none"

        }
  ```

Update group attribute

To modify a group attribute, use the **idmgmt/identity/api/v2/scim/attributes/group/<attributekey>** API.

- To get the group **attributekey** value, use the following command:

  ```
  curl -sk -X GET --header "Authorization: Bearer $ACCESS_TOKEN" --header "Content-Type: application/json"
  "https://$CP_CONSOLE_URL:443/idmgmt/identity/api/v2/scim/attributes/group"
  ```

- To modify the group attribute, use the following curl command:

```
curl -k -X PUT \
  -H "Authorization: Bearer $ACCESS_TOKEN" \
  -H "content-type: application/json" \
  "https://$CP_CONSOLE_URL/idmgmt/identity/api/v2/scim/attributes/group/<attributekey>" \
  -d '{
          "schemaId": "urn:ietf:params:scim:schemasibmcp:2.0:Group",
          "screenName": "groupAttribute1",
          "isSensitive": false,
          "name" : "groupAttribute1",
          "type" : "string",
          "multiValued" : false,
          "description" : "Custom Group Attribute 1",
          "required" : true,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"

      }'
```

Deleting a custom SCIM attribute

Use the **DELETE** operation with the **idmgmt/identity/api/v2/scim/attributes/<resourceType>/<attributekey>** API to delete a custom SCIM attribute in the users and groups section.

Delete a user attribute

Use the **idmgmt/identity/api/v2/scim/attributes/users/<attributekey>** API.

Use the following curl command:

```
curl -k -X DELETE \
  -H "Authorization: Bearer $ACCESS_TOKEN" \
  -H "content-type: application/json" \
  "https://$CP_CONSOLE_URL/idmgmt/identity/api/v2/scim/attributes/user/<attributekey>"
```

Delete a group attribute

Use the **idmgmt/identity/api/v2/scim/attributes/group/<attributekey>** API.

Use the following curl command:

```
curl -k -X DELETE \
  -H "Authorization: Bearer $ACCESS_TOKEN" \
  -H "content-type: application/json" \
  "https://$CP_CONSOLE_URL/idmgmt/identity/api/v2/scim/attributes/group/<attributekey>"
```

Updating SCIM LDAP attributes mapping

You can update the SCIM LDAP attributes mapping.

To support the SCIM APIs for a configured LDAP connection in IBM Cloud Pak foundational services, you can update **SCIM_LDAP_ATTRIBUTES_MAPPING** data in the following ways:

- By using SCIM configuration for LDAP connection. For more information, see SCIM configuration by using IBM Cloud Pak for AIOps UI.
- By using the **attributemapping** API (available for IBM Cloud Pak foundational services version 3.11.0 or later)

Note: Use of the configmap to update the SCIM LDAP attributes mapping is not supported and removed from the foundational services version 4.8.

Updating with `attributemapping` API

To use the **attributemapping** API, you must add an authorization header to your request. You need to add an access token to the authorization header. To obtain the access token, see Preparing to run component or management API commands.

Procedure

Note: The **members** and **objectClass** attributes do not consider the custom SCIM attribute values. Use LDAP filter values for these attributes to map the data.

1. Check the existing attribute mappings by using the following API call:

   ```
   curl -sk -X GET --header "Authorization: Bearer $ACCESS_TOKEN" --header "Content-Type: application/json"
   "https://$BCS_URL:443/idmgmt/identity/api/v1/scim/attributemappings"
   ```

   The response resembles the following code:

   ```
   [{"idp_id":"default","idp_type":"ldap","user":
   {"id":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn","familyName":"sn","fullName":"
   cn","externalId":"dn","emails":"mail","created":"createTimestamp","lastModified":"modifyTimestamp","phoneNumbers":
   [{"value":"mobile","type":"mobile"},
   {"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"},"group":
   {"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created":"createTimestamp","lastMo
   dified":"modifyTimestamp","objectClass":"groupOfUniqueNames","members":"uniqueMember"}},
   ```

2. Create the attribute mappings with your mapping data by using the following API call:

   Note: If you already have the attribute mappings, do not create another attribute mapping. You can skip this step.

   ```
   export DATA='{"idp_id":"test","idp_type":"ldap","user":
   {"id_test":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn","familyName":"sn","fullNa
   me":"cn","externalId":"dn","phoneNumbers":[{"value":"mobile","type":"mobile"},
   {"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"},"group":
   {"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created":"createTimestamp","lastMo
   dified":"modifyTimestamp","objectClass":"groupOfUniqueNames","members":"uniqueMember"}}'
   ```

   The sample curl command resembles the following code:

   ```
   curl -sk -X POST --header "Authorization: Bearer $ACCESS_TOKEN" --header "Content-Type: application/json" -d $DATA
   "https://$BCS_URL:443/idmgmt/identity/api/v1/scim/attributemappings"
   ```

   The response resembles the following code:

   ```
   {"idp_id":"test","idp_type":"ldap","user":
   {"id_test":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn","familyName":"sn","fullNa
   me":"cn","externalId":"dn","phoneNumbers":[{"value":"mobile","type":"mobile"},
   {"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"},"group":
   {"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created":"createTimestamp","lastMo
   dified":"modifyTimestamp","objectClass":"groupOfUniqueNames","members":"uniqueMember"}}
   ```

3. Update the existing attribute mappings data with your mapping data by using the following API call:

   ```
   export DATA='{"idp_id":"$IDP_ID","idp_type":"ldap","user":
   {"id_test":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn","familyName":"sn","fullNa
   me":"cn","externalId":"dn","phoneNumbers":[{"value":"mobile","type":"mobile"},
   {"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"}},"group":
   {"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created":"createTimestamp","lastMo
   dified":"modifyTimestamp","objectClass":"groupOfUniqueNames","members":"uniqueMember"}'
   ```

   The sample curl command resembles the following code:

   ```
   curl -sk -X PUT --header "Authorization: Bearer $ACCESS_TOKEN" --header "Content-Type: application/json" -d $DATA
   "https://$BCS_URL:443/idmgmt/identity/api/v1/scim/attributemappings/$IDP_ID"
   ```

   The response resembles the following code:

   ```
   {"idp_id":"test","idp_type":"ldap","user":
   {"id_test":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn","familyName":"sn","fullNa
   me":"cn","externalId":"dn","phoneNumbers":[{"value":"mobile","type":"mobile"},
   {"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"},"group":
   {"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created":"createTimestamp","lastMo
   dified":"modifyTimestamp","objectClass":"groupOfUniqueNames","members":"uniqueMember"}}
   ```

4. Delete the existing attribute mappings by using the following API call:

   ```
   curl -sk -X DELETE --header "Authorization: Bearer $ACCESS_TOKEN" \
   --header "Content-Type: application/json" \
   "https://$BCS_URL:443/idmgmt/identity/api/v1/scim/attributemappings/$IDP_ID"
   ```

   The response resembles the following code:

   ```
   {"count":1}
   ```

Note: By default, the LDAP connection configuration considers the following mapping values for SCIM APIs.

Table 1. LDAP connection attributes

LDAP connection attributes	SCIM object	SCIM attributes
LDAP_USERIDMAP	user	userName
LDAP_USERFILTER	user	objectClass
LDAP_GROUPIDMAP	group	name
LDAP_GROUPFILTER	group	objectClass
LDAP_GROUPMEMBERIDMAP	group	members

Note: If you have multiple expressions in the LDAP connection attributes configuration, the first configuration value is considered.

LDAP attributes

For information about LDAP attributes, you can use the **ldapsearch** tool or any LDAP browser such as **Apache Directory Studio**.

Installing the ldapsearch tool

On Ubuntu, run the following command:

```
sudo apt-get install ldap-utils
```

On Red Hat® Enterprise Linux®, run the following command:

```
sudo yum install openldap-clients
```

The following examples show the ldapsearch tool command structure and an example output.

User

Command:

```
ldapsearch -x -H <LDAP_URL> -b <LDAP_BASEDN> -D <LDAP_BINDDN> -W -s sub "(sAMAccountName=pavann)" "*" "+"
```

Output:

```
# extended LDIF
#
# LDAPv3
# base <DC=ibmtest,DC=com> with scope subtree
# filter: (sAMAccountName=pava)
# requesting: * +
#

# Tom Northwood, Users, ibmtest.com
dn: CN=Tom Northwood,CN=Users,DC=ibmtest,DC=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Tom Northwood
givenName: Tom Northwood
distinguishedName: CN=Tom Northwood,CN=Users,DC=ibmtest,DC=com
instanceType: 4
whenCreated: 20180802104118.0Z
whenChanged: 20210305115346.0Z
displayName: Tom Northwood
uSNCreated: 13356
memberOf: CN=group9,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group8,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group7,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group6,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group5,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group4,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group3,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group2,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group1,CN=Users,DC=ibmtest,DC=com
memberOf: CN=security,CN=Users,DC=ibmtest,DC=com
uSNChanged: 145644
name: Tom Northwood
objectGUID:: pVKE4qv5MEyqxjQ3nUvsWA==
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 132285264754362226
lastLogoff: 0
lastLogon: 132285264944206361
pwdLastSet: 131776804917936927
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAAr/FodBO+7uVBDhlaXAQAAA==
accountExpires: 9223372036854775807
logonCount: 0
sAMAccountName: pavann
sAMAccountType: 805306368
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=ibmtest,DC=com
dSCorePropagationData: 20180802104118.0Z
dSCorePropagationData: 16010101000000.0Z
lastLogonTimestamp: 132594188266767176

# search reference
ref: ldap://ForestDnsZones.ibmtest.com/DC=ForestDnsZones,DC=ibmtest,DC=com

# search reference
ref: ldap://DomainDnsZones.ibmtest.com/DC=DomainDnsZones,DC=ibmtest,DC=com

# search reference
ref: ldap://ibmtest.com/CN=Configuration,DC=ibmtest,DC=com

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
```

Group

Command:

```
ldapsearch -x -H <LDAP_URL> -b <LDAP_BASEDN> -D <LDAP_BINDDN> -W -s sub "(cn=security)" "*" "+"
```

Output:

```
# extended LDIF
#
# LDAPv3
# base <DC=ibmtest,DC=com> with scope subtree
# filter: (cn=security)
# requesting: * +
```

```
#

# security, Users, ibmtest.com
dn: CN=security,CN=Users,DC=ibmtest,DC=com
objectClass: top
objectClass: group
cn: security
member: CN=Neil Wilson,CN=Users,DC=ibmtest,DC=com
member: CN=John Winston,CN=Users,DC=ibmtest,DC=com
member: CN=Jeff Rodrigue,CN=Users,DC=ibmtest,DC=com
member: CN=Kane Shatner,CN=Users,DC=ibmtest,DC=com
member: CN=Anna Reynolds,CN=Users,DC=ibmtest,DC=com
member: CN=Gessie Lemson,CN=Users,DC=ibmtest,DC=com
member: CN=Lauri Williamson,CN=Users,DC=ibmtest,DC=com
member: CN=Tom Northwood,CN=Users,DC=ibmtest,DC=com
distinguishedName: CN=security,CN=Users,DC=ibmtest,DC=com
instanceType: 4
whenCreated: 20180802103445.0Z
whenChanged: 20180802105558.0Z
uSNCreated: 13330
memberOf: CN=isl,CN=Users,DC=ibmtest,DC=com
uSNChanged: 13455
name: security
objectGUID:: UPoQ3uNfVk+fHn1W1b5KUg==
objectSid:: AQUAAAAAAAUVAAAAr/FodBO+7uVBDhlaWAQAAA==
sAMAccountName: security
sAMAccountType: 268435456
groupType: -2147483646
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=ibmtest,DC=com
dSCorePropagationData: 16010101000000.0Z

# search reference
ref: ldap://ForestDnsZones.ibmtest.com/DC=ForestDnsZones,DC=ibmtest,DC=com

# search reference
ref: ldap://DomainDnsZones.ibmtest.com/DC=DomainDnsZones,DC=ibmtest,DC=com

# search reference
ref: ldap://ibmtest.com/CN=Configuration,DC=ibmtest,DC=com

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
```

IBM Cloud attributes

To search IBM Cloud users in SCIM, custom attributes can be defined as shown in the following code example. By default, these attributes are not defined.

```
"IBMCloud" = {
    "user": {
        "id": "userId",
        "userName": "userId",
        "principalName": "userId",
        "displayName": "firstName",
        "emails": "email",
        "familyName": "lastName",
        "givenName": "firstName",
        "externalId": "userId"
    }
};
```

Enhancing SCIM group and user API performance

Enhanced SCIM group API performance

By default, foundational services supports the LDAP query filter size limit to up to 5000. You can run a query on a bulk of group members by using a single LDAP search. If you have a more number of members in a group, it is advised to increase the LDAP query filter size limit to at least 5000 for the configured LDAP to get the benefit from the enhanced performance of the group API calls.

Follow the tuning guidelines for the CPU and memory for the following observations, if you have a large number of members in a group:

Note: The values that are provided in the following observations and tuning guidelines are the estimated values. These values might also depend on the other environmental factors such as overall load, cluster health, network.

- A Group of approximately 21k members with concurrent access to 10 users, requires around 1500m of CPU and 1500Mi of memory, and 1 replica for the small profile.
- If the number of concurrent accesses increases, you need to increase the number of replicas as well. For example, the same 21k members for concurrent access to 20 users, require around 3 replicas with the 1500m of CPU and 1500Mi of memory for the small profile.

- If you have more members in the group, use the large profile instead of the small profile. For example, the large profile for a group of 87K members with the concurrent access to 10 users, needs around 3 replicas with 2550m of CPU and 2850Mi of memory.

Enhanced SCIM user API performance

IM has the following configuration parameters for SCIM users search API:

- **`SCIM_AUTH_CACHE_MAX_SIZE`**: It indicates the maximum number of tokens that are stored in the cache. The default value is 1000. If you have multiple concurrent calls, it is recommended to tune this value.
- **`SCIM_AUTH_CACHE_TTL_VALUE`**: It indicates the TTL value in seconds of each token that is stored in the cache. The default value is 60 seconds. If you have a long-running process that simultaneously calls the SCIM API for more than 60 seconds, it is recommended to tune this value.

These SCIM configured parameters improve the overall response time to avoid the repeated authorization calls.

Changing the configured SCIM attributes by using the console

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for **`platform-auth-idp`**.
4. Click **...** > **Edit Config Map**.
5. Change the following attribute values to **`true`**, as required:
   - **`SCIM_AUTH_CACHE_MAX_SIZE`**
   - **`SCIM_AUTH_CACHE_TTL_VALUE`**
6. Click **Save**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate **`platform-auth-service`**.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the **`platform-auth-service`** pods with the latest ConfigMap values.
11. Click **`platform-auth-service`**.
12. Wait for some time. Then, check the status of the **`platform-auth-service`** pods in the **Pods** pane. The status of all the pods must show as **`4/4`** under the **Ready** field name.

Changing the configured SCIM attributes by using CLI

1. Log in to your infrastructure node by using the **`oc login`** command.
2. Edit the **`platform-auth-idp`** configmap.

   ```
   oc -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

3. Change the following SCIM configured attributes parameter to a higher value:
   - **`SCIM_AUTH_CACHE_MAX_SIZE`**
   - **`SCIM_AUTH_CACHE_TTL_VALUE`**
4. Save the changes.
5. Restart the **`platform-auth-service`** pods by deleting the pods.
   1. Get the **`platform-auth-service`** pod names.

      ```
      oc -n <your-foundational-services-namespace> get pods | grep platform-auth-service
      ```

   2. Delete the **`platform-auth-service`** pod.

      ```
      oc -n <your-foundational-services-namespace> delete pods <pod-name>
      ```

Wait for some time and then check the status of the **`platform-auth-service`** pods. The status of all **`platform-auth-service`** pods must show as **`Running`**.

```
oc -n <your-foundational-services-namespace> get pods | grep platform-auth-service
```

Custom Group filter support in SCIM User API

You can use the LDAP Group filter value when you query for a group of SCIM users.

For example, User1 is part of four groups - **`icptest1`**, **`icptest2`**, **`isvtest1`** and **`isvtest2`** and you set this value **`"(&(cn=%v)(cn=*icptest*)(objectclass=person))"`** in the LDAP Group filter while you create the LDAP connection. When you query User1 through the SCIM API, the User1 has only **`icptest1`** and **`icptest2`** groups in the **group** section of the SCIM response.

SCIM pagination support for LDAP server

From the foundational services version 3.19.9, 3.21, and later, the server-side pagination is supported for the LDAP. The pagination feature in the LDAP server allows you to access the large results sets when there is a server-side administrative limit to the number of entries returned from a query.

The supported LDAP types in your product cluster enforce the page size limits upon the results that are returned from the LDAP in the SCIM user/group API call. The supported LDAP types in your product cluster with their corresponding default page size limit:

LDAP types with their corrresponding default page size limit

LDAP type	Default page size
IBM Tivoli Directory Server	20000
IBM Lotus Domino	20000
IBM SecureWay Directory Server	20000
Novell eDirectory	500
Sun Java™ System Directory Server	2000
Netscape Directory Server	5000
Microsoft Active Directory	1000
Custom	4500

For example, in Microsoft Active Directory, the default page size limit is 1000. It means, the default number of entries that are returned in a single request is, 1000. If you send the request to retrieve more than 1000 entries, the LDAP server returns the **LDAP_SIZELIMIT_EXCEEDED** error. To avoid the **LDAP_SIZELIMIT_EXCEEDED** error in the foundational services version 3.20 and earlier, the entry request in your query should be less than or equal to 1000.

Similarly, for other LDAPs server, if the number of entries in your request exceeds the default page size limit of the LDAP server, the server displays the **LDAP_SIZELIMIT_EXCEEDED** error. To overcome this page size limit of LDAP server, the SCIM pagination option is introduced with the foundational services version 3.19.9, 3.21, and later. The pagination option for LDAP server is applicable for the SCIM APIs. This option enables you to retrieve the result of your query even if your LDAP entry request is beyond the default page size limit of the LDAP server.

Enabling pagination option for LDAP server

By default, the SCIM pagination option for the LDAP server is disabled. You can enable the pagination option by using the Identity Provider V3 API. For now, you can enable pagination option through the APIs only.

The Identity Provider V3 API contains following parameters to support the server-side pagination option for LDAP:

- **LDAP_PAGINGSEARCH**: If the LDAP_PAGINGSEARCH parameter is set as "false", it indicates that the pagination option is disabled. Set this parameter to "true" to enable the pagination option.
- **LDAP_PAGING_SIZE**: It is a numeric field that defines the page size of the LDAP type. You can set the number as per your requirement. It is an optional field. If you do not provide the value to the **LDAP_PAGING_SIZE** parameter, the default page size of the LDAP server will be considered as a request in the API query. For example, when the LDAP type is Microsoft Active Directory (MSAD) and you do not provide the **LDAP_PAGING_SIZE** value in the API, then 1000 (default page size) value will be considered as a request in the API. Also, the **LDAP_PAGING_SIZE** value should not exceed the default server side page limit or the maximum configured value for the respective LDAP type.

See the following notes:

- The IM supports LDAP user and group search through SCIM APIs and LDAP APIs. However, the pagination option for the LDAP server is applicable for the SCIM APIs only.
- You can configure the server-side pagination option for each LDAP type.
- **LDAP_PAGINGSEARCH** and **LDAP_PAGING_SIZE** parameters are optional in the Identity Provider V3 API. Include these parameters in the API as per your requirement.

Procedure

The following example to enable the pagination and set the page size while creating an LDAP connection through Identity Provider V3 API:

Before you begin: Ensure that you are informed about the other LDAP configuration parameters that are used in the Identity Provider V3 APIs. For more information, see LDAP authentication, LDAP filters and Default LDAP filters by LDAP type.

Prerequisites: The LDAP server must have **bind dn** and **bind password**, otherwise you might get **InsufficientAccessRightsError** while performing the SCIM APIs query. You must use the admin **bind dn** and **bind password** while configuring the LDAP connection. If you use the non-admin **bind dn** and **bind password**, enable the non-admin page search option in the underlying LDAP server. For example, in IBM Tivoli Directory Server, the conf file attribute, **ibm-slapdPagedResAllowNonAdmin**(boolean) should be enabled. This file attribute decides whether the non-admin binds are allowed to perform page search.

API version

1.0.0

API URI components

Scheme

```
HTTPS
```

Host IP

```
Cluster Master Host
```

Port number

Cluster Master API Port

Path

idprovider/v3/auth/idsource/

Command

POST

Command output format

application/json

The sample curl command to create an LDAP connection:

```
curl -k -X POST 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $ACCESS_TOKEN"
--data-raw
'{
  "name": "openldap",
  "description": "",
  "protocol": "ldap",
  "type": "Custom",
  "idp_config": {
        "ldap_id": "openldap",
        "ldap_realm": "REALM",
        "ldap_url": "ldap://9.30.253.13:389",
        "ldap_host": "9.30.253.13",
        "ldap_port": "389",
        "ldap_protocol": "ldap",
        "ldap_basedn": "dc=ibm,dc=com",
        "ldap_binddn": "cn=admin,dc=ibm,dc=com",
        "ldap_bindpassword": "UGFzc3cwcmQ=",
        "ldap_type": "Custom",
        "ldap_ignorecase": "false",
        "ldap_userfilter": "(&(uid=%v)(objectclass=person))",
        "ldap_useridmap": "*:uid",
        "ldap_groupfilter": "(&(cn=%v)(objectclass=groupOfUniqueNames))",
        "ldap_groupidmap": "*:cn",
        "ldap_groupmemberidmap": "groupOfUniqueNames:uniqueMember",
        "ldap_nestedsearch": "false",
        "ldap_pagingsearch": "false"
        }
}'
```

To use the Identity Provider APIs for LDAP connection, you need to get the host details, and the base64-encoded password for the **"LDAP_BINDPASSWORD"** parameter. For more information, see Creating an LDAP connection.

By using Identity Provider APIs, you can also enable pagination option and set the page size while updating the existing LDAP directory. For more information, see Update an LDAP directory.

Changing LDAP contextPool settings

Change the contextPool settings to suit your Lightweight Directory Access Protocol (LDAP) server load.

The LDAP contextPool parameters are listed in the following tables. For more information, see LDAP User Registry (ldapRegistry).

Note: Specify a positive integer followed by a unit of time, which can be hours (h), minutes (m), seconds (s), or milliseconds (ms). For example, specify 500 milliseconds as 500ms. You can include multiple values in a single entry. For example, 1s500ms is equivalent to 1.5 seconds. This note is applicable only to the **LDAP_CTX_POOL_TIMEOUT** and **LDAP_CTX_POOL_WAITTIME** parameters. Also, the **LDAP_CTX_POOL_TIMEOUT** must be set lesser than the idle timeout on a firewall (or backend server) for it to be effective.

Table 1. LDAP contextPool parameter properties

Parameter	Description	Default value
LDAP_CTX_POOL_INITSIZE	An integer value that determines the initial size of the context pool. Set this value based on the load on the repository.	10
LDAP_CTX_POOL_MAXSIZE	An integer value that defines the maximum context pool size. Set this value based on the maximum load on the repository.	50
LDAP_CTX_POOL_PREFERREDSIZE	Maximum number of attributes per LDAP entity that are cached.	10
LDAP_CTX_POOL_TIMEOUT	The duration after which the context pool times out.	30s
LDAP_CTX_POOL_WAITTIME	The time interval that the request waits until the context pool checks again if an idle context instance is available in the pool when the number of context instances reaches the maximum pool size.	60s

Changing the parameter values by using kubectl

To change the parameter values, complete the following steps:

1. Install the Kubernetes CLI (kubectl) tool.
2. Edit the `platform-auth-idp` ConfigMap.

   ```
   kubectl -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

3. Change the following attribute values as required:
   - `LDAP_CTX_POOL_INITSIZE`
   - `LDAP_CTX_POOL_MAXSIZE`
   - `LDAP_CTX_POOL_PREFERREDSIZE`
   - `LDAP_CTX_POOL_TIMEOUT`
   - `LDAP_CTX_POOL_WAITTIME`
4. Save and close the ConfigMap.
5. Restart the `platform-auth-service` pods

   ```
   kubectl -n <your-foundational-services-namespace> delete pod -l k8s-app=platform-auth-service
   ```

6. Wait for some time. Then, check the status of the `platform-auth-service` pods. The status must show as `4/4 Running` for all the pods.

   ```
   kubectl -n ibm-common-services get pods | grep platform-auth-service
   ```

Changing the parameter values by using the console

Complete the following steps:

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for `platform-auth-idp`.
4. Click **...** > **Edit Config Map**.
5. Change the following attribute values as required:
   - `LDAP_CTX_POOL_INITSIZE`
   - `LDAP_CTX_POOL_MAXSIZE`
   - `LDAP_CTX_POOL_PREFERREDSIZE`
   - `LDAP_CTX_POOL_TIMEOUT`
   - `LDAP_CTX_POOL_WAITTIME`
6. Click **Save**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate `platform-auth-service`.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the `platform-auth-service` pods with the latest ConfigMap values.
11. Click `platform-auth-service`.
12. Wait for some time. Then, check the status of the `platform-auth-service` pods in the **Pods** pane. The status of all the pods must show as `4/4` under the **Ready** field name.

Cluster configurations

Learn to configure your cluster.

- Allowing limited users to log in by using only LDAP
- Updating cp-console hostname and TLS secret
- Changing the cluster administrator access credentials
- Adding a logout redirect URL
- Configuring access and identity token validity
- Changing the security role mappings refresh time interval that is used during authorization
- Changing LDAP search cache variable values
- Using a proxy server for outbound traffic
- Changing LDAP contextPool settings

Allowing limited users to log in by using only LDAP

You might have scenarios where you want to provide a strict subset of accounts the ability to authenticate by using an LDAP server that is configured in your Identity Management (IM) instance.

For example, when you configure a SAML connection to use an existing LDAP connection as its identity provider (IdP), you cannot use that LDAP for IM authentication. This behavior is the default and is implemented to enforce multi-factor authentication across all account logins. You might still need to allow a subset of nonpersonal accounts or functional IDs to authenticate directly with the LDAP server despite this broader implementation. Or, perhaps you need to block some or all use of LDAP for authentication with IM, but do not want to change the user and group filters that are already set on the configurations in your LDAP connections.

In such situations, you can use the ID source update API to set an allowlist of users on one or more LDAP configurations in your cluster. You can do so by adding an `ldap_allowlist` in the `idp_config` of the LDAP configuration. The field is optional.

ldap_allowlist field settings

The following table describes the parameter values that are in the **ldap_allowlist** field.

Table 1. ldap_allowlist field settings

Value type	Syntax	Description
An array with one or more string values	`"ldap_allowlist": [ "user1", "user2", "user3" ]`	Allows only these listed accounts to use LDAP authentication through the identity management (IM) service
Empty array	`"ldap_allowlist": []`	Deny all - blocks all attempts to use LDAP authentication through IM
No field present	NA	Default behavior - allows all attempts to use LDAP authentication through IM

Behavior on the IBM Cloud Pak console

If the **PREFERRED_LOGIN** field in the **platform-auth-idp** configmap has a nonempty value, the field must contain "LDAP", or else users cannot log in with LDAP. This requirement applies even in cases where an allowlist is defined for an LDAP connection.

When PREFERRED_LOGIN is set to a value that allows LDAP authentication, the enterprise LDAP login option is available to you on your IBM Cloud Pak console. Setting a nonempty allowlist on an LDAP connection that is being used as an IdP for a SAML connection adds the enterprise LDAP login option to the console if it was not already present. Then, the console no longer automatically redirects to the login page of your SAML provider: users from the allowlist can authenticate by using LDAP instead.

Behavior of ldap_allowlist with multiple LDAPs

If multiple LDAP servers are configured with IM, you can define an **ldap_allowlist** field on one or more of those configurations. When a user tries to log in to IM by using their LDAP credentials, IM checks whether allowlists are set on any of the configured LDAP servers. IM assumes that user or account names between the configured LDAP servers are unique. So, if any allowlists are set, IM checks whether the provided username is present in any of those allowlists. If present, IM allows the authentication to continue.

You must be careful when you add allowlists, especially when you use multiple LDAPs in IM. For example, if you add an allowlist on one LDAP and no allowlist on a second LDAP, then only the usernames from the allowlist of the first LDAP can authenticate with IM by using LDAP. Users from the second LDAP cannot use the LDAP authentication option because the second LDAP doesn't have an allowlist.

Also, if a username from the allowlist of the first LDAP exists in other LDAPs, even if the username is of a different user or account identity (for example, another person), IM allows the username to log in by using the other LDAP.

Adding ldap_allowlist field to the LDAP configuration

Use the **/idprovider/v3/auth/idsource** APIs to update the LDAP server with the **ldap_allowlist** field. Before you begin, make sure you have the following ready:

- An account with sufficient permissions to interact with the **idsource** APIs, **curl** commands, or any other means of making **HTTP**/**HTTPS** requests.
- **jq** installed, if you want to make it easier to read responses and construct requests.

Define an **ldap_allowlist** by following these steps. The steps work from a terminal that uses a POSIX-compliant shell. For example, **bash**.

1. Get the LDAP server information
2. Add the **ldap_allowlist** field
3. Update the LDAP configuration

Get the LDAP server information

Get the ID of the LDAP server.

1. Obtain an access token from IM. See Preparing to run component or management API commands.
2. Define the following environment variables:
   - access token

     ```
     export access_token=<access_token_value>
     ```

   - LDAP server name of the users that you want to add in the **ldap_allowlist** field

     ```
     export LDAP_NAME=<ldap_name>
     ```

   - **cp-console** domain name

     ```
     export CP_CONSOLE_DOMAIN=<domain_name>
     ```

3. Get the LDAP server details. If you are using **jq**, see Using jq with the commands.

```
curl \
  --location \
  --request GET \
  --insecure \
  --silent \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header "Authorization: Bearer ${access_token}" \
  --get \
```

```
  -d "name=${LDAP_NAME}" \
  -d 'protocol=ldap' \
  "https://${CP_CONSOLE_DOMAIN}/idprovider/v3/auth/idsource"
```

The response looks something like the following response. Note the **uid** value from the response. The **uid** value is used in the API call to update the LDAP connection configuration.

```
{
  "idp": [
    {
      "name": "example_ldap",
      "description": "An example LDAP",
      "protocol": "ldap",
      "type": "IBM Tivoli Directory Server",
      "idp_config": {
        "ldap_id": "example_ldap",
        "ldap_realm": "REALM",
        "ldap_url": "ldap://ldap.example.com:389",
        "ldap_host": "ldap.example.com",
        "ldap_port": "389",
        "ldap_protocol": "ldap",
        "ldap_basedn": "o=example.com",
        "ldap_binddn": "",
        "ldap_bindpassword": "",
        "ldap_type": "IBM Tivoli Directory Server",
        "ldap_ignorecase": "false",
        "ldap_userfilter": "(&(uid=%v)(objectclass=ePerson))",
        "ldap_useridmap": "*:uid",
        "ldap_groupfilter": "(&(cn=%v)(objectclass=groupOfUniqueNames))",
        "ldap_groupidmap": "*:cn",
        "ldap_groupmemberidmap": "groupOfUniqueNames:uniqueMember",
        "ldap_nestedsearch": "false",
        "ldap_pagingsearch": "false"
      },
      "uid": "917a0ac9-9cf0-49aa-9313-53aa5b9e87ec"
    }
  ]
}
```

4. Define environment variables for the UID and JSON response.

   - UID

     ```
     export uid=<ldap_uid_from_previous_command>
     ```

   - JSON response

     ```
     export json_response=<response_from_previous_command>
     ```

5. Write the JSON response to a file so that you can edit the response and submit an update request.

   ```
   echo "${json_response}" > ldap.json
   ```

Add the ldap_allowlist field

Add the **ldap_allowlist** field in the **ldap.json** file that you created in the earlier section.

You must add the **ldap_allowlist** field to the **idp_config** field and set it to an array value.

Important: Remove the **uid** field from the **ldap.json** file. **PUT** requests do not accept payloads if the **uid** is present. If you used **jq** filter in the previous section, the command automatically removes the **uid** field.

After you add the **ldap_allowlist** field and remove the **uid**, the **ldap.json** file content must resemble the following payload. The **"ldap_allowlist"** array must have your list of account names:

```
{
  "name": "example_ldap",
  "description": "An example LDAP",
  "protocol": "ldap",
  "type": "IBM Tivoli Directory Server",
  "idp_config": {
    "ldap_id": "example_ldap",
    "ldap_realm": "REALM",
    "ldap_url": "ldap://ldap.example.com:389",
    "ldap_host": "ldap.example.com",
    "ldap_port": "389",
    "ldap_protocol": "ldap",
    "ldap_basedn": "o=example.com",
    "ldap_binddn": "",
    "ldap_bindpassword": "",
    "ldap_type": "IBM Tivoli Directory Server",
    "ldap_ignorecase": "false",
    "ldap_userfilter": "(&(uid=%v)(objectclass=ePerson))",
    "ldap_useridmap": "*:uid",
    "ldap_groupfilter": "(&(cn=%v)(objectclass=groupOfUniqueNames))",
    "ldap_groupidmap": "*:cn",
    "ldap_groupmemberidmap": "groupOfUniqueNames:uniqueMember",
    "ldap_nestedsearch": "false",
    "ldap_pagingsearch": "false",
    "ldap_allowlist": [ "ldap_user1", "ldap_user2", "ldap_user_etc"]
```

```
  }
}
```

Update the LDAP configuration

Update the LDAP connection configuration by using the PUT method on the idsource API.

```
curl \
  --location \
  --request PUT \
  --insecure \
  --silent \
  --header "Authorization: Bearer ${access_token}" \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  -d @ldap.json \
  "https://${CP_CONSOLE_DOMAIN}/idprovider/v3/auth/idsource/${uid}"
```

Using jq with the commands

If you are using `jq`, you can pipe the JSON response into it when you run the curl commands.

1. Get the LDAP server details. The following command is the same as the curl command run in step 4, but stored as an array.

   ```
   curl_command=(
     curl
     --location
     --request GET
     --insecure
     --silent
     --header 'Content-Type: application/json'
     --header 'Accept: application/json'
     --header "Authorization: Bearer ${access_token}"
     --get
     -d "name=${LDAP_NAME}"
     -d 'protocol=ldap'
     "https://${CP_CONSOLE_DOMAIN}/idprovider/v3/auth/idsource"
   )
   ```

2. Read `uid` and LDAP idsource JSON into variables in your shell.

   Note: The following example command assumes that the query parameters are specific enough to narrow the results down to a single LDAP entry.

   ```
   IFS='|' read -r uid json_response <<< "$("${curl_command[@]}" | jq -r -c -M '.idp[] | [.uid, (. | del(.uid))] | "\(.[0])|\(.[1])"')"
   ```

3. Confirm that the variables were set properly.

   ```
   echo "UID: ${uid}"
   echo "JSON response: ${json_response}"
   ```

4. Write the JSON response to a file to edit and submit for update request.

   ```
   echo "${json_response}" > ldap.json
   ```

Next, create the `ldap_allowlist` field and update the LDAP configuration.

Updating cp-console hostname and TLS secret

You can update the cp-console hostname with the custom hostname and certificates using one of the following methods:

- Updating cp-console hostname and TLS using the script
- Updating cp-console hostname using the `ibmcloud-cluster-info` configmap (no TLS certificate update required)

Note: These instructions are for changing only the cp-console route hostname and certificates. This procedure is not applicable for the cpd route, zenservice CR, or Platform UI. Run the following command to verify whether the cp-console exists in the cluster:

```
oc get route | grep cp-console
```

If cp-console exists in the cluster, the following routes are displayed in the response:

```
cp-console              cp-console-ibm-common-services.apps.fidelitytest.cp.fyre.ibm.com   /
common-web-ui                3000                    reencrypt/Redirect     None
id-mgmt                 cp-console-ibm-common-services.apps.fidelitytest.cp.fyre.ibm.com   /idmgmt/
platform-identity-management   4500                    reencrypt/Redirect     None
platform-auth           cp-console-ibm-common-services.apps.fidelitytest.cp.fyre.ibm.com   /v1/auth/
platform-identity-provider     4300                    reencrypt/Redirect     None
platform-id-provider    cp-console-ibm-common-services.apps.fidelitytest.cp.fyre.ibm.com   /idprovider/
platform-identity-provider     4300                    reencrypt/Redirect     None
platform-login          cp-console-ibm-common-services.apps.fidelitytest.cp.fyre.ibm.com   /login
platform-identity-provider     4300                    reencrypt/Redirect     None
platform-oidc           cp-console-ibm-common-services.apps.fidelitytest.cp.fyre.ibm.com   /oidc
platform-auth-service          9443                    reencrypt/Redirect     None
saml-ui-callback        cp-console-ibm-common-services.apps.fidelitytest.cp.fyre.ibm.com   /ibm/saml20/defaultSP/acs
platform-auth-service          9443                    reencrypt/Redirect     None
social-login-callback   cp-console-ibm-common-services.apps.fidelitytest.cp.fyre.ibm.com   /ibm/api/social-login
platform-auth-service          9443                    reencrypt/Redirect     None
```

This procedure is applicable only for the cluster with the cp-console route.

You can refer to the following sections to restore SAML and revert the updates:

- Restore SAML
- Reverting the updates

Updating cp-console hostname using the script

You can update the **cp-console** hostname and certificates using the **cs-onprem-tenant-config.sh** script. Ensure that you complete the prerequisites before you run the script.

Prerequisites

Keep your new hostname and certificates ready.

- Create the custom hostname. If you modify the default Red Hat® OpenShift® Container Platform domain, ensure that the new hostname can resolve to the Red Hat® OpenShift® Container Platform router from inside and outside the Red Hat® OpenShift® Container Platform cluster.
- Optional: If you are updating the TLS secret, provide new certificates in an unencrypted PEM format. Place the certificates in the **installer_scripts** directory. For more information, see Download and extract the installer scripts for your CASE bundle.

  Note: This step is applicable only if you need to update the certificates. If the certificates change or expire, you must provide new certificates and run the script again to create a secret by using the new certificates.

  - **ca.crt**: Contains the intermediate CA signer certificates and the CA root signer certificate. The file must start with the intermediate CA certificates in order that to lead the root CA.
  - **tls.crt**: Contains only the route server certificate.

    Note: The route server certificate passes the TLS verification for the route hostname.
  - **tls.key**: Contains the private key of the route server certificate.

  See the following example of PEM encoding that uses header and footer lines for each certificate and private key.

```
-----BEGIN CERTIFICATE-----
(encoded set of characters)
-----END CERTIFICATE-----

-----BEGIN PRIVATE KEY-----
(encoded set of characters)
-----END PRIVATE KEY-----
```

Procedure

Complete the following steps to change the cp-console hostname and certificates of the **cp-console** route.

1. Create a **cs-onprem-tenant-config.yaml** file with the following configmap definition. In the **data** section, provide your custom hostname. If you are providing new certificates to update the TLS secret, do not change the name of the **custom_host_certificate_secret** secret in the configmap.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cs-onprem-tenant-config
  namespace: <your-foundational-services-namespace>
labels:
  cs_onprem_tenant_config: "true"
data:
  ##comment out or remove this setting if not changing the hostname
  custom_hostname: <your-custom-hostname>
  ##comment out or remove this setting if not changing the certificates
  custom_host_certificate_secret: custom-tls-secret
```

   The following is the sample custom hostname:

```
cp-testcustomhn.apps.o1-849902.cp.fyre.ibm.com
```

   Note: You need to update the **custom_hostname:** parameter with the URL that includes the domain name and new custom hostname without **https://** protocol.

2. Create the **cs-onprem-tenant-config** configmap.

```
oc apply -f cs-onprem-tenant-config.yaml
```

3. If your cluster topology has a separate namespace for all operators, identify that namespace and set it as the default namespace. You need to update the **$csNamespace** and **$map_to_common_service_namespace** in the script.

   1. Get the namespace where you deployed the foundational services.

```
oc get csv -A | grep common-service-operator
```

      The following is the sample output:

```
cp4itest                                           ibm-common-service-operator.v4.11.0            IBM Cloud Pak
foundational services    4.11.0            ibm-common-service-operator.v4.10.0            Succeeded
```

`cp4itest` is the namespace where foundational services is deployed and `v4.11.0` is the CASE version

2. Set the project with the default foundational services namespace.

   ```
   oc project <your-foundational-services-namespace>
   ```

   The following is the sample command:

   ```
   oc project cp4itest
   ```

4. Download and extract the installer scripts for your CASE bundle. To find your CASE version, see IBM: Product CASE to Application Version and search for the installed foundational services version to find the related CASE version.

   ```
   export CASE_VERSION=<your foundational service operator version>
   ```

   ```
   wget https://github.com/IBM/cloud-pak/raw/master/repo/case/ibm-cp-common-services/${CASE_VERSION}/ibm-cp-common-services-${CASE_VERSION}.tgz
   ```

   ```
   tar -xvzf ibm-cp-common-services-$CASE_VERSION.tgz
   ```

   ```
   cd ibm-cp-common-services/inventory/ibmCommonServiceOperatorSetup/installer_scripts
   ```

   Replace `CASE_VERSION` with the foundational services version. To get the operator version, run the following command:

   ```
   oc get csv | grep "<your-foundational-services-namespace>"
   ```

5. Set the following environmental variables in the `cs-onprem-tenant-config.sh` script. To update the script, you can use the text editor or run one of the following commands in a system editor:

   ```
   vi cs-onprem-tenant-config.sh
   ```

   ```
   sudo nano cs-onprem-tenant-config.sh
   ```

   - `$csNamespace` - Set the operator namespace of the foundational services. The operator namespace is the control namespace where foundational services is deployed.
   - `$map_to_common_service_namespace` - Set the operand namespace of the foundational services where `platform-auth-service-xxx` pod is running.

   Note: If the operators and operands are in the same namespace, set `$map_to_common_service_namespace` and `$csNamespace` with `<your-foundational-services-namespace>`.

6. If you are providing new certificates for TLS secret, place the `tls.key`, `tls.crt`, and `ca.crt` files in the `installer_scripts` directory. Run the script to create the TLS secret with the required certificate files.

   To change the hostname and use the selfsigned certificate, see Prerequisites to prepare your `cert` and `tls` files.

7. Run the script from the operator namespace.

   ```
   ./cs-onprem-tenant-config.sh
   ```

   If you see the following error, it means that you did not provide the new certificate files for TLS secret in the `installer_scripts` directory. To resolve the issue, place the certificate files and run the script again.

   ```
   tls.key is not present in current directory, pls keep tls.key, tls.crt and ca.crt files in current directory
   ```

   The script completes the following tasks:

   - Uses namespaces from the `common_service_maps` configmap that is in the `kube-public` namespace. The common_service_maps` configmap has all the namespaces in a tenant.
   - Creates a TLS secret if you provided new certificate files.
   - If your cluster topology has a separate namespace for all operators, the script runs the `iam-custom-hostname` job in that namespace.
   - Update the custom hostname in all applicable identity management (IM) components.

8. Restart the Zen operator pod.

   ```
   oc get pod -n <your-foundational-services-namespace> | grep ibm-zen-operator
   ```

   ```
   oc delete pod -n <your-foundational-services-namespace> <ibm-zen-operator-pod-name>
   ```

9. Restart the `usermgmt` pods.

   1. Get the `usermgmt` pod names.

      ```
      oc get pods |grep usermgmt
      ```

   2. Delete the `usermgmt` pods.

      ```
      oc delete pod <usermgmt-pod-name>
      ```

10. Wait until the Zen operator and `usermgmt` pods restart. Ensure that the pods are in the ready and running status.

```
oc get pods -n <your-foundational-services-namespace> | grep ibm-zen-operator
oc get pods -n <your-foundational-services-namespace>|grep usermgmt
```

Updating cp-console hostname using the `ibmcloud-cluster-info` configmap (no TLS certificate update required)

You can update the **cp-console** hostname with the wildcard certificate using the **ibmcloud-cluster-info** configmap. You cannot change the **cp-console** hostname that is not configured with the wildcard certificate and **cp-console** domain name.

The following is the sample wildcard certificate:


An example of wildcard certificate

Note: You cannot update the TLS using the **ibmcloud-cluster-info** configmap. To update the TLS, see Updating cp-console hostname using the script.

Complete the following steps to update the custom hostname with the **ibmcloud-cluster-info** configmap:

1. Update the **cluster_address** and **cluster_endpoint** parameters in the **ibmcloud-cluster-info** configmap.

   ```
   oc edit cm ibmcloud-cluster-info -n cpfs-data
   ```

   The following is the sample **ibmcloud-cluster-info** configmap:

   ```
   apiVersion: v1
   data:
     cluster_address: <hostname>-cpfs-data.apps.ocp415-qb.cp.fyre.ibm.com.
     cluster_endpoint: https://<hostname>-cpfs-data.apps.ocp415-qb.cp.fyre.ibm.com
     cluster_kube_apiserver_host: api.ocp415-qb.cp.fyre.ibm.com
     cluster_kube_apiserver_port: "6443"
     cluster_name: mycluster
     cluster_router_http_port: "80"
     cluster_router_https_port: "443"
     im_idmgmt_endpoint: https://platform-identity-management.cpfs-data.svc:4500
     im_idprovider_endpoint: https://platform-identity-provider.cpfs-data.svc:4300
     proxy_address: cp-proxy-cpfs-data.apps.ocp415-qb.cp.fyre.ibm.com
   kind: ConfigMap
   ```

   Replace **<hostname>** with the custom hostname.

2. Wait for the IM pods to restart.

   ```
   platform-auth-service-xxx
   platform-identity-management-xxx
   platform-identity-provider-xxx
   ```

3. Restart Zen pods with the following label:

   ```
   name=ibm-zen-operator
   component=usermgmt
   ```

   1. Get the **usermgmt** pod names.

      ```
      oc get pods |grep usermgmt
      ```

   2. Delete the **usermgmt** pods.

      ```
      oc delete pod <usermgmt-pod-name>
      ```

   You can login to **cp-console** and **cpd** with the updated hostname after the restart of the Zen pods.

Result

After you run the script, the **cp-console** route is updated with the new hostname. You can also see the new certificates, if applicable, in your browser when you access the **cp-console** URL.

Restore SAML

After you change the certificate, the Security Assertion Markup Language (SAML) connection breaks. You must update the SAML metadata with the new certificate information to restore the connection. For more information, see Identity and access management migration.

Reverting the updates

If you need to revert to the previous hostname and certificate, repeat the steps in Procedure by using the previous hostname.

To revert to the default name for **cp-console**, complete the following steps:

1. Update the **cs-onprem-tenant-config** ConfigMap to set the default name in the **custom_hostname** parameter.

   ```
   oc -n <your-foundational-services-namespace> edit configmap cs-onprem-tenant-config
   ```

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: cs-onprem-tenant-config
     namespace: $csNamespace
   labels:
     cs_onprem_tenant_config: "true"
   data:
   ```

```
  ##comment out or remove this setting if not changing the hostname
  custom_hostname: <Default hostname>
  ##comment out or remove this setting if not changing the certificates
  custom_host_certificate_secret: custom-tls-secret
```

2. Run the script again.

```
./cs-onprem-tenant-config.sh
```

3. To restore the default certificates, delete the **cp-console** route.

```
oc get route -n <your-foundational-services-namespace>

oc delete route cp-console  -n <your-foundational-services-namespace>
```

Changing the Cloud Pak administrator access credentials

You can update the administrator username and password.

Note: **cpadmin** is the default admin user name.

Before you begin

- Install the OpenShift client.
- Ensure that you must have **administrator** access to update the administrator username and password.

Supported languages

You can change the default administrator username **cpadmin** to a custom username. The username can be in the following languages:

Table 1. Supported languages

Language	Language code
Brazilian Portuguese	pt_br
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Chinese Simplified	zh_CN
Chinese Traditional	zh_TW
Spanish	es

Changing the administrator username

Complete the following steps to rename a default admin user in Platform UI:

1. Review the current admin and password.

```
oc -n <your-foundational-services-namespace> get secret platform-auth-idp-credentials -o yaml | grep admin |xargs -l
bash -c 'echo -n "$0  " ; echo "$1" | base64 -d && echo'
```

The sample output:

```
admin_password:  IRXkjnsZg8nUUK8S2BBVFJeIOW0wVrhD
admin_username:  cpadmin
```

2. Get the access token. For more information, see Preparing to run API commands.

3. Run the following CURL command to update default admin username:

```
curl -X PUT 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idmgmt/identity/api/v1/users/defaultAdmin' \
    --header 'Content-Type: application/json' \
    --header 'Accept: application/json' \
    --header "Authorization: Bearer $ACCESS_TOKEN" \
    -d "{\"username\": \"${new_admin_username}\"}" \
    --insecure
```

4. Restart the **platform-identity-provider** and **platform-auth-service** pods by deleting them.

    1. Get the **platform-identity-provider** pod name and **platform-auth-service** pod name .

    ```
    oc get pods -n <your-foundational-services-namespace> | grep platform-identity-provider

    oc get pods -n <your-foundational-services-namespace> | grep platform-auth-service
    ```

    2. Delete the **platform-identity-provider** pod and **platform-auth-service** pod.

```
oc delete pod <platform-identity-management-pod-name> -n <your-foundational-services-namespace>

oc delete pod <platform-auth-service-pod-name> -n <your-foundational-services-namespace>
```

5. After the pods restart, login to **cp-console** to test whether the default username has changed.

```
oc get routes -n <your-foundational-services-namespace> |grep -i cp-console
```

Changing the administrator password

The administrator password is stored in a Kubernetes Secret.

Complete the following steps to change the administrator password in Platform UI:

1. Review the current admin and password.

```
oc -n <your-foundational-services-namespace> get secret platform-auth-idp-credentials -o yaml | grep admin |xargs -l bash -c 'echo -n "$0  " ; echo "$1" | base64 -d && echo'
```

The sample output:

```
admin_password:  IRXkjnsZg8nUUK8S2BBVFJeIOW0wVrhD
admin_username:  admin
```

2. Update the **admin_password** field in the **platform-auth-idp-credentials** secret with the new password.

- In the namespace where foundational services is installed, edit the **platform-auth-idp-credentials** secret:

```
oc edit secret platform-auth-idp-credentials -n "${SERVICES_NAMESPACE}"
```

Where, **{SERVICES_NAMESPACE}** is the namespace name where foundational services is installed.

- Search for **admin_password** value in **platform-auth-idp-credentials** secret and replace it with the new password. The new password must be **base64-encoded** password.

3. Restart the **platform-auth-service**, **platform-identity-provider**,**platform-identity-management**, and **usermgmt** pods by deleting them.

    1. Delete the **platform-auth-service** pod.
        - Get the **platform-auth-service** pod name.

        ```
        oc get pods -n <your-foundational-services-namespace> | grep platform-auth-service
        ```

        - Delete the **platform-auth-service** pod.

        ```
        oc delete pod <platform-auth-service-pod-name> -n <your-foundational-services-namespace>
        ```

    2. Delete the **platform-identity-provider** pod.
        - Get the **platform-identity-provider** pod name.

        ```
        oc get pods -n <your-foundational-services-namespace> | grep platform-identity-provider
        ```

        - Delete the **platform-identity-provider** pod.

        ```
        oc delete pod <platform-identity-provider-pod-name> -n <your-foundational-services-namespace>
        ```

    3. Delete the **platform-identity-management** pod.
        - Get the **platform-identity-management** pod name.

        ```
        oc get pods -n <your-foundational-services-namespace> | grep platform-identity-management
        ```

        - Delete the **platform-identity-management** pod.

        ```
        oc delete pod <platform-identity-management-pod-name> -n <your-foundational-services-namespace>
        ```

    4. Delete the **usermgmt** pod.
        - Get the **usermgmt** pod name.

        ```
        oc get pods -n <your-foundational-services-namespace> | grep usermgmt-<pod name>
        ```

        - Delete the **usermgmt** pod.

        ```
        oc delete pod <usermgmt-pod-name> -n <your-foundational-services-namespace>
        ```

After the pod restart, login to **cp-console** to test whether the password has changed.

```
oc get routes -n <your-foundational-services-namespace> |grep -i cp-console
```

The password must follow the defined password rules.

Password rules

Password rules are optional, regular expressions (regex) that are used to set and validate managed password values within a namespace. The **update-secret** command checks password rules before changing the values in the secret. It validates if rules exist on the namespace that the secret is in, and that the secret data element name change contains **pass** or **pwd** in any case combination.

Note: You can use the following regex expressions from the command line. Ensure that you wrap the expressions in single quotation marks:

Table 2. List of rules for Regex expressions

Rule	Regex
Min length 10	`^.{10,}`
Max length 10	`^.{0,10}$`
Length range	`^.{10,20}$`
Require lower	`.*[a-z].*`
Require upper	`.*[A-Z].*`
Require num	`.*[0-9].*`
Require spec	`.*[!@#$%^&*].*`

Customizing the cluster access URL

Customize the Uniform Resource Locator (URL) that you use to log in to IBM Cloud Pak for AIOps console.

- Supported customization formats
- Customize the cluster access URL
  - Customize the cluster access URL with `oc` command

Supported customization formats

The following customization formats are supported:

- `https://<Public IP>:8443/console`
- `https://<Public IP>:8443/console/`
- `https://<Private IP>:8443/console/`
- `https://<Private IP>:custom-port/console/`
- `https://<host name>:8443/console`
- `https://<host name>:custom-port/console`
- `https://localhost:8443/console`
- `https://localhost:<custom port>/console`
- `https://<Regex host name>:8443/console`
- `https://<Regex IP>:8443/console`
- `https://<Regex host name>:<custom port>/console`
- `https://<Regex IP>:<custom port>/console`
- `https://<Regex host name>:<Regex port>/console`
- `https://<Regex IP>:<Regex Port>/console`

Required user type or access level: Administrator

Customize the cluster access URL with `oc` command

Complete the following tasks on the boot node of your cluster.

1. Log in to the boot node as a user with root permissions.
2. Set up the OpenShift client (oc).
3. Copy the content that is in the **`registration-json`** configmap into the file **`registration.yaml`**.

   ```
   oc get cm registration-json -n <your-foundational-services-namespace> -o yaml > registration.yaml
   ```

   The **`registration.yaml`** file content resembles the following code:

   ```
   apiVersion: v1
   data:
     platform-oidc-registration.json: |
       {
       "token_endpoint_auth_method":"client_secret_basic",
       "client_id": "d2a00fc99163f85169ac7c6de758bad1",
       "client_secret": "01661d22bd0b2025fd87e26e994a4894",
       "scope":"openid profile email",
       "grant_types":[
          "authorization_code",
          "client_credentials",
          "password",
          "implicit",
          "refresh_token",
          "urn:ietf:params:oauth:grant-type:jwt-bearer"
       ],
       "response_types":[
          "code",
          "token",
          "id_token token"
       ],
       "application_type":"web",
       "subject_type":"public",
   ```

```
    "post_logout_redirect_uris":[

"https://10.10.25.213:8443/console/logout","https://9.37.239.32:8443/console/logout","https://mycluster.icp:8443/con
sole/logout"    ],
    "introspect_tokens":true,
    "trusted_uri_prefixes":[
       "https://10.10.25.213:8443","https://9.37.239.32:8443","https://mycluster.icp:8443"    ],
    "redirect_uris":[

"https://10.10.25.213:8443/auth/liberty/callback","https://9.37.239.32:8443/auth/liberty/callback","https://mycluste
r.icp:8443/auth/liberty/callback","https://mycluster.icp:8443/oidc/endpoint/OP"    ]
    }
kind: ConfigMap
metadata:
  creationTimestamp: 2018-06-06T11:53:21Z
  name: registration-json
  namespace: <your-foundational-services-namespace>
  resourceVersion: "1255"
  selfLink: /api/v1/namespaces/<your-foundational-services-namespace>/configmaps/registration-json
  uid: 3620b003-6980-11e8-9420-fa163ea0dafe
```

4. Create a **platform-oidc-registration.json** file. Place the file in the **<installation directory>/cluster/cfc-components/** folder.
5. Copy the content that is in the **data:** section of the **registration.yaml** file into the **platform-oidc-registration.json** file. The **platform-oidc-registration.json** file content resembles the following code:

```
{
 "token_endpoint_auth_method":"client_secret_basic",
 "client_id": "d2a00fc99163f85169ac7c6de758bad1",
 "client_secret": "01661d22bd0b2025fd87e26e994a4894",
 "scope":"openid profile email",
 "grant_types":[
    "authorization_code",
    "client_credentials",
    "password",
    "implicit",
    "refresh_token",
    "urn:ietf:params:oauth:grant-type:jwt-bearer"
 ],
 "response_types":[
    "code",
    "token",
    "id_token token"
 ],
 "application_type":"web",
 "subject_type":"public",
 "post_logout_redirect_uris":[

"https://10.10.25.213:8443/console/logout","https://9.37.239.32:8443/console/logout","https://mycluster.icp:8443/con
sole/logout"    ],
 "preauthorized_scope":"openid profile email general",
 "introspect_tokens":true,
 "trusted_uri_prefixes":[
    "https://10.10.25.213:8443","https://9.37.239.32:8443","https://mycluster.icp:8443"    ],
 "redirect_uris":[
 "https://10.10.25.213:8443/auth/liberty/callback","https://9.37.239.32:8443/auth/liberty/callback","https://myclust
er.icp:8443/auth/liberty/callback","https://mycluster.icp:8443/oidc/endpoint/OP"    ]
 }
```

6. Add the following piece of code to the **platform-oidc-registration.json** file:

   ```
   "allow_regexp_redirects":"true",
   ```

   The updated code resembles the following text:

   ```
   {
    "token_endpoint_auth_method":"client_secret_basic",
    "client_id": "d2a00fc99163f85169ac7c6de758bad1",
    "client_secret": "01661d22bd0b2025fd87e26e994a4894",
    "scope":"openid profile email",
    "allow_regexp_redirects":"true",                                    <==========
    "grant_types":[
       "authorization_code",
       "client_credentials",
       "password",
       "implicit",
       "refresh_token",
       "urn:ietf:params:oauth:grant-type:jwt-bearer"
       ...
   ```

7. Add your custom URIs in the **"redirect_uris"** section of the **platform-oidc-registration.json** file. See Supported customization formats for the types of URIs that you can add.

   ```
   "<regexp>:https://<custom IP address or host name>:<custom port>/auth/liberty/callback",
   ```

   Where, you add **<regexp>:** only if you are using a regex in the custom URI.

   Consider the following example URIs that you want to use to access the cluster:

   - Use the master node IP address and any port that starts with **84**. You would then add **"regexp:https://<master node IP address>:84!d!d/auth/liberty/callback"**.

- Use the host name **example.abc.com** and port **4002**. You would then add **"https://example.abc.com:4002/auth/liberty/callback"**.
- Use a variable host name and a dynamic port assignment. You would then add **"regexp:https://example.[a-z]*.com:[0-9]*/auth/liberty/callback"**.

If you added the example custom URIs, the updated code resembles the following text:

```
...
"application_type":"web",
"subject_type":"public",
"post_logout_redirect_uris":[

"https://10.10.25.213:8443/console/logout","https://9.37.239.32:8443/console/logout","https://mycluster.icp:8443/con
sole/logout"    ],
"preauthorized_scope":"openid profile email general",
"introspect_tokens":true,
"trusted_uri_prefixes":[
   "https://10.10.25.213:8443","https://9.37.239.32:8443","https://mycluster.icp:8443"    ],
"redirect_uris":[
   "regexp:https://10.10.25.213:84!d!d/auth/liberty/callback",       <==========
   "https://example.abc.com:4002/auth/liberty/callback",         <==========
   "regexp:https://example.[a-z]*.com:[0-9]*/auth/liberty/callback",          <==========

"https://10.10.25.213:8443/auth/liberty/callback","https://9.37.239.32:8443/auth/liberty/callback","https://mycluste
r.icp:8443/auth/liberty/callback","https://mycluster.icp:8443/oidc/endpoint/OP"    ]
}
```

8. Save and exit the file.
9. Save the client ID, client secret, and access IP to the following variables:
   1. Save the client secret:

      ```
      OAUTH2_CLIENT_REGISTRATION_SECRET=$(oc -n <your-foundational-services-namespace> get secret platform-oidc-
      credentials -o "jsonpath={.data.OAUTH2_CLIENT_REGISTRATION_SECRET}" | base64 --decode)
      ```

   2. Save the client ID:

      ```
      WLP_CLIENT_ID=$(oc -n <your-foundational-services-namespace> get secret platform-oidc-credentials -o "jsonpath=
      {.data.WLP_CLIENT_ID}" | base64 --decode)
      ```

   3. Save the access IP:

      ```
      FIP=<master node IP address>
      ```

10. Apply the changes that you made to the **platform-oidc-registration.json** file.

    ```
    curl -kvv -X PUT -u oauthadmin:$OAUTH2_CLIENT_REGISTRATION_SECRET -H "Content-Type: application/json" -d
    @<installation directory>/cluster/cfc-components/platform-oidc-registration.json
    https://$FIP:8443/oidc/endpoint/OP/registration/$WLP_CLIENT_ID
    ```

11. Update the **registration-json** configmap with the custom URIs. Make the same changes that you made to the **platform-oidc-registration.json**.
12. Open the **registration-json** configmap for editing.

    ```
    oc edit cm registration-json -n <your-foundational-services-namespace>
    ```

13. Add the following piece of code:

    ```
    "allow_regexp_redirects":"true",
    ```

14. Add the custom URIs. If you added the example custom URIs, you would then add the following piece of code:

    ```
    "redirect_uris":[
    "regexp:https://10.10.25.213:84!d!d/auth/liberty/callback",
    "https://example.abc.com:4002/auth/liberty/callback",
    "regexp:https://example.[a-z]*.com:[0-9]*/auth/liberty/callback",
    ```

Adding a logout redirect URL

You can add a logout redirect URL that is displayed after you log out of the Red Hat® OpenShift® Container Platform console.

For more information, see Red Hat® OpenShift® Container Platform documentation.

- For Red Hat® OpenShift® Container Platform version 4.4, see Configuring the web console in OpenShift Container Platform.
- For Red Hat® OpenShift® Container Platform version 4.3, see Configuring the web console in OpenShift Container Platform.

Complete these steps to add a logout redirect URL:

1. Update the Console resource.
2. Update the platform-oidc-registration.json file.

Update the Console resource

Complete these steps to add a logout redirect URL in the **Console** resource:

1. Get the IBM Cloud Pak foundational services URL.

   ```
   oc get route -n <your-foundational-services-namespace> cp-console -o=jsonpath={.spec.host}
   ```

   Following is a sample output:

   ```
   cp-console.apps.hcp.darumtech.net[root@2test-inf ~]
   ```

2. Get the Red Hat® OpenShift® Container Platform console URL.

   ```
   oc get route -n openshift-console console -o jsonpath={.spec.host}
   ```

   Following is a sample output:

   ```
   console-openshift-console.apps.hcp.darumtech.net[root@2test-inf ~]
   ```

3. Add the redirect URLs in the OpenShift console resource.

   1. Open the resource for editing.

      ```
      oc edit console cluster
      ```

      Following is a sample content of the `Console` resource:

      ```
      apiVersion: config.openshift.io/v1
      kind: Console
      metadata:
        annotations:
          release.openshift.io/create-only: "true"
        creationTimestamp: "2020-03-25T15:34:01Z"
        generation: 6
        name: cluster
        resourceVersion: "9119030"
        selfLink: /apis/config.openshift.io/v1/consoles/cluster
        uid: ab896f97-ad65-456e-806b-1f7fbe879345
      spec: {}
      status:
        consoleURL: https://console-openshift-console.apps.hcp.darumtech.net
      ```

   2. Add the logout redirect URL in the `spec` section by using the IBM Cloud Pak foundational services URL and the Red Hat® OpenShift® Container Platform console URL values. Following is the syntax of the logout redirect URL:

```
https://<{{site.data.keyword.cs}} URL>/idprovider/v1/auth/logout?login_url=https://<{{site.data.keyword.ocp}}
console URL>
```

   Following would be the example logout redirect URL:

```
https://cp-console.apps.hcp.darumtech.net/idprovider/v1/auth/logout?login_url=https://console-openshift-
console.apps.hcp.darumtech.net
```

   You would add this redirect URL as shown in the following piece of code:

```
spec:
  authentication:                 <===== Add this section
    logoutRedirect:               <===== Add the logout redirect URL
```

   Following is a sample of the updated `Console` resource:

```
  apiVersion: config.openshift.io/v1
  kind: Console
  metadata:
    annotations:
      release.openshift.io/create-only: "true"
    creationTimestamp: "2020-03-25T15:34:01Z"
    generation: 6
    name: cluster
    resourceVersion: "9119030"
    selfLink: /apis/config.openshift.io/v1/consoles/cluster
    uid: ab896f97-ad65-456e-806b-1f7fbe879345
  spec:
    authentication:
      logoutRedirect: https://cp-console.apps.hcp.darumtech.net/idprovider/v1/auth/logout?login_url=https://console-
openshift-console.apps.hcp.darumtech.net
  status:
    consoleURL: https://console-openshift-console.apps.hcp.darumtech.net
```

Update the platform-oidc-registration.json file

You must add the Red Hat® OpenShift® Container Platform console URL in the platform-oidc-registration.json in the `"redirect_uris"` section.

Configuring access and identity token validity

Change the default setting of the access token and identity token validity.

By default, the access token and identity (ID) token are valid for 12 hours after you sign out of the console or close your browser. You can change this default value as required.

Follow these steps to change the token validity:

1. Edit the **platform-auth-idp** configmap.

   ```
   oc -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

   The file content resembles the following code:

   ```
   # Please edit the following object. Lines beginning with a '#' will be ignored,
   # and an empty file will abort the edit. If an error occurs while saving this file will be
   # reopened with the relevant failures.
   #
   apiVersion: v1
   data:
     AUDIT_ENABLED_IDMGMT: "false"
     AUDIT_ENABLED_IDPROVIDER: "false"
     BASE_AUTH_URL: /v1
     BASE_OIDC_URL: https://127.0.0.1:8443/idauth/oidc/endpoint/OP
     CLUSTER_NAME: mycluster
     HTTP_ONLY: "true"
     IDENTITY_AUTH_DIRECTORY_URL: http://127.0.0.1:3100
     IDENTITY_PROVIDER_URL: http://127.0.0.1:4300
     IDTOKEN_LIFETIME: 12h
     JOURNAL_PATH: /run/systemd/journal
     MASTER_HOST: mycluster.icp
     NODE_ENV: production
     OAUTH2DB_DB_PORT: "3306"
     OIDC_ISSUER_URL: https://mycluster.icp:8443/oidc/endpoint/OP
     SESSION_TIMEOUT: "43200"
     logrotate: |-
       /var/log/audit/*.log {
         su nobody root
       .
       .
       .
   ```

2. Update the values of the **SESSION_TIMEOUT** and **IDTOKEN_LIFETIME** parameters.

   - The **SESSION_TIMEOUT** parameter sets the access token validity, the default value of which is **43200** seconds. Specify a positive integer followed by a unit of time, which can be hours (h), minutes (m), or seconds (s). For example, specify 30 seconds as 30s. You can include multiple values in a single entry. For example, 1m30s is equivalent to 90 seconds.
   - The **IDTOKEN_LIFETIME** parameter sets the ID token validity, the default value of which is **12** hours. Specify a positive integer followed by a unit of time, which can be hours (h), minutes (m), or seconds (s). For example, **14h** or **60000s**. You cannot include multiple values in a single entry. For example, you cannot add **1h30m**.

   After you update and save the changes, you see the following message:

   ```
   configmap "platform-auth-idp" edited
   ```

3. Restart the **platform-auth-service** pod.

   1. Get the **platform-auth-service** pod ID.

      ```
      oc -n <your-foundational-services-namespace> get pods | grep platform-auth-service
      ```

      Following is a sample output of the command:

      ```
      platform-auth-service-t6sfm     4/4     Running       0     1d
      ```

   2. Delete the **platform-auth-service** pod.

      ```
      oc -n <your-foundational-services-namespace> delete pod platform-auth-service-t6sfm
      ```

      After the pod is deleted, you see the following message:

      ```
      pod "platform-auth-service-t6sfm" deleted
      ```

      Wait for a few minutes for the pod to restart.

   3. Verify the status of the pod.

      ```
      oc -n <your-foundational-services-namespace> get pods | grep platform-auth-service-t6sfm
      ```

      The pod gets a new ID after restart.

      ```
      platform-auth-service-5267t                1/4       Running     0          4m
      ```

Changing the security role mappings refresh time interval that is used during authorization

Change the security role mappings refresh time interval that is used during authorization.

By default, the security role mappings refresh time interval is set to 10 minutes. You can change this default value as required.

Follow these steps to change the time interval:

1. Edit the `platform-auth-idp` configmap.

   ```
   oc -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

2. Update the value of the `PDP_REDIS_CACHE_DEFAULT_TTL` parameter, as required. Specify the value in seconds. The default value is `600` seconds.

   After you update and save the changes, you see the following message:

   ```
   configmap "platform-auth-idp" edited
   ```

3. Restart all `platform-identity-management` pods by deleting them.

   ```
   oc -n <your-foundational-services-namespace> delete pod -l k8s-app=platform-identity-management
   ```

   Wait for a few minutes for the pod to restart.

4. Verify the status of the pods.

   ```
   oc -n <your-foundational-services-namespace> get pods | grep platform-identity-management-pdp
   ```

   The status of all the pods must show as `1/1 Running`.

Changing LDAP search cache variable values

Change the default values of the Lightweight Directory Access Protocol (LDAP) search cache variables.

You can change the values of the following variables:

Table 1. LDAP cache parameters

Variable	Default value	Description
`LDAP_SEARCH_CACHE_TIMEOUT`	1200 seconds	The maximum time that the contents of the search results cache are available.
`LDAP_SEARCH_CACHE_ENABLED`	true	A Boolean value to indicate that the search results must be cached or not.
`LDAP_SEARCH_CACHE_SIZE`	2000	The number of search results that are stored in the cache.
`LDAP_SEARCH_CACHE_SIZELIMIT`	2000	The maximum number of results that can be cached for a single LDAP search.
`LDAP_ATTR_CACHE_SIZE`	2000	The number of entities that can be stored in the cache.
`LDAP_ATTR_CACHE_TIMEOUT`	1200 seconds	The maximum time that the contents of the LDAP attribute cache are available.
`LDAP_ATTR_CACHE_SIZELIMIT`	2000	The maximum number of attributes per LDAP entity that are cached.
`LDAP_ATTR_CACHE_ENABLED`	true	A Boolean value to indicate that the entities must be cached or not.

For more information, see LDAP User Registry (ldapRegistry).

Follow these steps to change the variable values:

1. Edit the `platform-auth-idp` configmap.

   ```
   oc -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

   The file content resembles the following code:

   ```
   # Please edit the following object. Lines beginning with a '#' will be ignored,
   # and an empty file will abort the edit. If an error occurs while saving this file will be
   # reopened with the relevant failures.
   #
   apiVersion: v1
   data:
     AUDIT_DETAIL: "false"
     AUDIT_ENABLED_IDMGMT: "false"
     AUDIT_ENABLED_IDPROVIDER: "false"
     .
     .
     LDAP_ATTR_CACHE_ENABLED: "true"
     LDAP_ATTR_CACHE_SIZE: "2000"
     LDAP_ATTR_CACHE_SIZELIMIT: "2000"
     LDAP_ATTR_CACHE_TIMEOUT: 1200s
     LDAP_SEARCH_CACHE_ENABLED: "true"
     LDAP_SEARCH_CACHE_SIZE: "2000"
     LDAP_SEARCH_CACHE_SIZELIMIT: "2000"
     LDAP_SEARCH_CACHE_TIMEOUT: 1200s
     LDAP_SEARCH_CN_ATTR_ONLY: "false"
     LDAP_SEARCH_ID_ATTR_ONLY: "false"
     LDAP_SEARCH_SIZE_LIMIT: "50"
     LDAP_SEARCH_TIME_LIMIT: "5"
     MASTER_HOST: mycluster.icp
     NODE_ENV: production
     .
     .
   "/tmp/kubectl-edit-i5ta2.yaml" 60L, 2204C
   ```

2. Change the LDAP variable values, as required, and save the ConfigMap.

   After you update and save the changes, you see the following message:

   ```
   configmap "platform-auth-idp" edited
   ```

3. Restart the **platform-identity-provider** pod.

    1. Get the **platform-identity-provider** pod ID.

        ```
        oc -n <your-foundational-services-namespace> get pods | grep platform-auth-service
        ```

        Following is a sample output of the command:

        ```
        platform-identity-provider-5b78f                          4/4     Running                3          12d
        ```

    2. Delete the **platform-identity-provider** pod.

        ```
        oc -n <your-foundational-services-namespace> delete pod platform-identity-provider-5b78f
        ```

        After the pod is deleted, you see the following message:

        ```
        pod "platform-identity-provider-5b78f" deleted
        ```

        Wait for a few minutes for the pod to restart.

    3. Verify the status of the pod.

        ```
        oc -n <your-foundational-services-namespace> get pods | grep platform-identity-provider
        ```

        The pod gets a new ID after restart.

        ```
        platform-identity-provider-5267t                1/4       Running     0          4m
        ```

Using a proxy server for outbound traffic

Use a forward proxy to communicate with your external identity provider.

Forward proxy servers isolate the network and provide a control point for outbound requests. The egress traffic is routed through the proxy server, and the proxy server sends it to the destination, which can be an external service outside the firewall. The proxy server also performs the same actions for any response that comes back from the destination. You can configure authentication requirement for the proxy server.

You can use a proxy server only if you are using OpenID Connect (OIDC) to connect your IBM Cloud Pak® with your identity provider (IdP).

Configuring the proxy server

Configure your proxy server to manage traffic for specific namespaces.

The identity management (IM) service uses the following parameters from the proxy configuration to add the proxy settings to IM applications:

- **_userdefined_http_proxy:** A proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be HTTP.
- **_userdefined_https_proxy:** A proxy URL to use for creating HTTPS connections outside the cluster. The URL scheme must be either HTTP or HTTPS.

IBM Cloud Pak requirements

Your IBM Cloud Pak must meet the following requirements:

- IBM Cloud Pak foundational services 4.2 or higher must be installed.
- Your IBM Cloud Pak and the Zen service must be installed.
- The IBM Cloud Pak for Data command-line interface (**cpd-cli**) and resource specification injection (RSI) must be configured. For more information, see Customizing pod specifications with resource specification injection.
- You must install and enable RSI for the namespace where foundational services are installed.

After you complete the requirements, you can inject the proxy configuration parameters into the IM application through RSI configuration.

1. Create a file named **cpfs-im-proxy-envs.json** in the RSI configuration directory, which is usually located under the **cpd-cli** as **cpd-cli-workspace/olm-utils-workspace/work/rsi**. The **cpfs-im-proxy-envs.json** file must contain the following lines. Replace the parameter values as applicable to your environment. Authentication credentials are optional.

    ```
    [ {
        "name":  "_userdefined_http_proxy",
        "value": "http://<username>:<password>@<ip>:<port>"
    }, {
        "name":  "_userdefined_https_proxy",
        "value": "http://<username>:<password>@<ip>:<port>"
    } ]
    ```

2. Run the following RSI patch for IM applications. Replace **$cs-data-ns** with your foundational services data plane namespace.

    ```
    ./cpd-cli manage create-rsi-patch     --cpd_instance_ns=$cs-data-ns     --patch_name="cpfs-im-authsvc-proxy"     --
    description="add proxy settings to im authsvc"     --patch_type=rsi_pod_env_var     --patch_spec=/tmp/work/rsi/cpfs-
    im-proxy-envs.json     --spec_format=set-env     --selector=app.kubernetes.io/instance:platform-auth-service --
    skip_apply=false --state=active
    ```

    ```
    ./cpd-cli manage create-rsi-patch     --cpd_instance_ns=$cs-data-ns     --patch_name="cpfs-im-providersvc-proxy"
    --description="add proxy settings to im providersvc"     --patch_type=rsi_pod_env_var     --
    patch_spec=/tmp/work/rsi/cpfs-im-proxy-envs.json     --spec_format=set-env  --
    selector=app.kubernetes.io/instance:platform-identity-provider --skip_apply=false --state=active
    ```

3. Check the new **platform-auth-service** and **platform-identity-provider** pods to make sure that the proxy values are set as applicable to your environment. If you don't find your environment values despite successfully running the **cpd-cli** command, restart the **platform-auth-service** and **platform-identity-provider** pods and reverify whether the values are set.
4. Configure the public OIDC IdP and log in as you usually do.

Rolling back the RSI proxy configuration from IM applications

To roll back the proxy configuration, first make sure that the RSI injection was successfully completed.

Then, run these commands to deactivate the patch and remove it. Replace **$cs-data-ns** with your foundational services data plane namespace.

```
./cpd-cli manage create-rsi-patch     --cpd_instance_ns=$cs-data-ns     --patch_name="cpfs-im-authsvc-proxy"     --
description="add proxy settings to im authsvc"     --patch_type=rsi_pod_env_var     --patch_spec=/tmp/work/rsi/cpfs-im-
proxy-envs.json     --spec_format=set-env     --selector=app.kubernetes.io/instance:platform-auth-service --
skip_apply=false --state=inactive

./cpd-cli manage delete-rsi-patch \
    --cpd_instance_ns=$cs-data-ns \
    --patch_name="cpfs-im-authsvc-proxy"

./cpd-cli manage create-rsi-patch     --cpd_instance_ns=$cs-data-ns     --patch_name="cpfs-im-providersvc-proxy"     --
description="add proxy settings to im providersvc"     --patch_type=rsi_pod_env_var     --patch_spec=/tmp/work/rsi/cpfs-
im-proxy-envs.json     --spec_format=set-env  --selector=app.kubernetes.io/instance:platform-identity-provider --
skip_apply=false --state=inactive

./cpd-cli manage delete-rsi-patch \
    --cpd_instance_ns=$cs-data-ns \
    --patch_name="cpfs-im-providersvc-proxy"
```

IM sample configuration

Learn how OKTA-IM integration and AZURE-IM integration help in managing the authentication and authorization for the resources.

- SCIM configuration by using IBM Cloud Pak for AIOps UI
- IBM Cloud Pak SCIM OKTA integration
- IBM Cloud Pak SCIM AZURE AD integration
- IBM Cloud Pak JIT support for SCIM OKTA and Azure Active Directory
- Integrating SAML with Azure Active Directory

SCIM configuration by using IBM Cloud Pak for AIOps UI

In IBM Cloud Pak for AIOps UI, you can configure SCIM by selecting the attributes (User and Group attributes) to map the Identity provider (Idp) attributes to the SCIM attributes.

Note: If you want to add custom SCIM attributes, see Adding custom SCIM attributes.

To configure SCIM, complete the following steps:

Pre-requisite: Install the cluster. Once the cluster is installed, the admin username, password, and the URL are displayed.

1. Log in to the console by using the URL and enter the username and password.

   Note: The console can be Platform UI or Common UI.
2. From the navigation menu, click **Identity and access** > **Identity providers**.
3. Click **Create Connection**. The **New LDAP server connection** page is displayed.
4. Select **SCIM configuration**.
5. Click **New attribute** in the SCIM configuration page. The attribute types- **User attribute** and **Group attribute** are displayed.
6. Select **User attribute** and complete the following:
   - Select the attribute from the **Attribute** drop-down.
   - Provide an Idp attribute to map the Idp attribute to the SCIM attribute.
7. Select **Group attribute** and complete the following:
   - Select the attribute from the **Attribute** drop-down.
   - Provide an Idp attribute to map the Idp attribute to the SCIM attribute.

   Similarly, you can create another **User attribute** and **Group attribute** by clicking **New attribute**.

   See the following notes:
   - The attributes in the **Attribute** drop-down are divided into two categories- simple attribute and complex attribute. For example, **Name** is a complex attribute because when you select the **Name** attribute from the drop-down, the **New component attribute** option is displayed. Click **New component**

attribute to select the component attribute from the drop-down. These component attributes can be **familyName**, **givenName**, **userType**, and so on. And, **id** is a simple attribute because it does not have any component attribute.

- If no SCIM attributes mappings are added, the default mapping happens. If you define a custom SCIM attributes mapping, the default mapping will not work and SCIM user search will return only those field values that you have added to the mapping list. Therefore, in such cases, perform the attribute mapping for others SCIM attributes in SCIM configuration section.

8. Click **Create** to update the identity provider.

Editing the SCIM configuration

To edit the SCIM configuration, complete the following steps:

1. In the **Identity Providers** page, a list of SCIM configurations is displayed.
2. Select the name of SCIM configuration to edit and click **Edit connection**. **Edit LDAP connection** dialog box is displayed.
3. In **Edit your LDAP connection** dialog box, click **Edit**. It displays the LDAP connection page that contains SCIM configuration.

   You can edit the SCIM configuration by the following ways:

   - Select **New attribute**, if you want to add new attributes to the SCIM configuration.
   - Delete any existing attribute. For each attribute, there is a **Delete** option in the UI.
   - Change the value of the Idp attribute of any existing attribute.

4. Click **Save**.

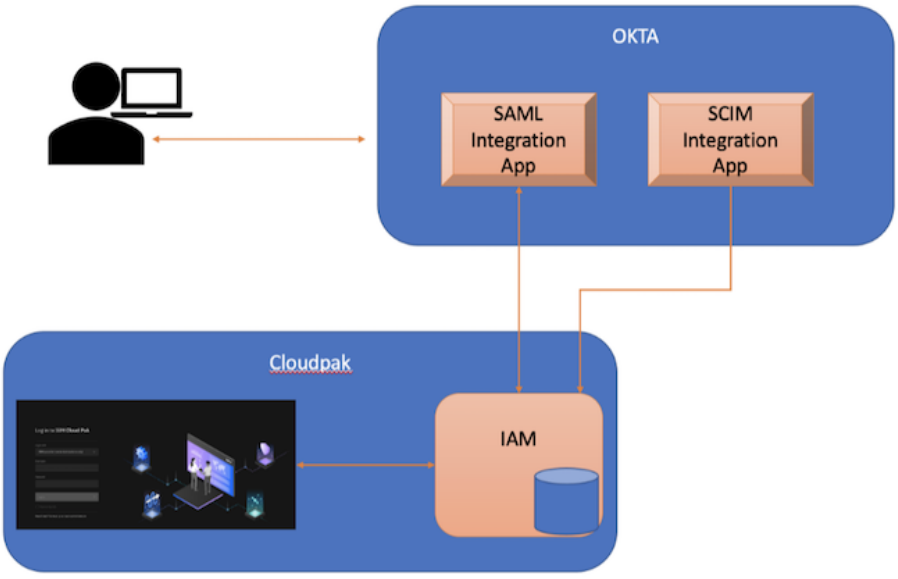
See the following notes:

- The `Group filter` and `Group member ID map` filters do not consider the custom SCIM attribute values. Use LDAP filter values for `Group filter` and `Group member ID map` attributes to map the data. For more information about LDAP filters, see LDAP filters.
- The SCIM users and groups API for LDAP is dependent on the attribute mapping list. You must provide a proper attribute mapping for all the SCIM user and group attributes. If the SCIM attributes mapping is not proper then SCIM API might not be able to fetch users or group details or some attributes might miss in the SCIM users or groups response.

IBM Cloud Pak SCIM Okta integration

You can use SCIM for OKTA-IM integration to manage your IBM Cloud Pak® users and groups. The OKTA-IM integration also helps in managing the authentication and authorization of the resources.

Okta is an enterprise-grade, identity management service that is built for the cloud. However, it is compatible with many on-premises applications. With Okta-IM integration, the organization can manage employees' access to any application or device. Okta is a cloud-based, SaaS-delivered Identity Management (IM) service that provides enterprise-grade identity lifecycle management capabilities to the developers and organizations.



Configuration required for OKTA integration by using SAML SSO and SCIM

See the following notes:

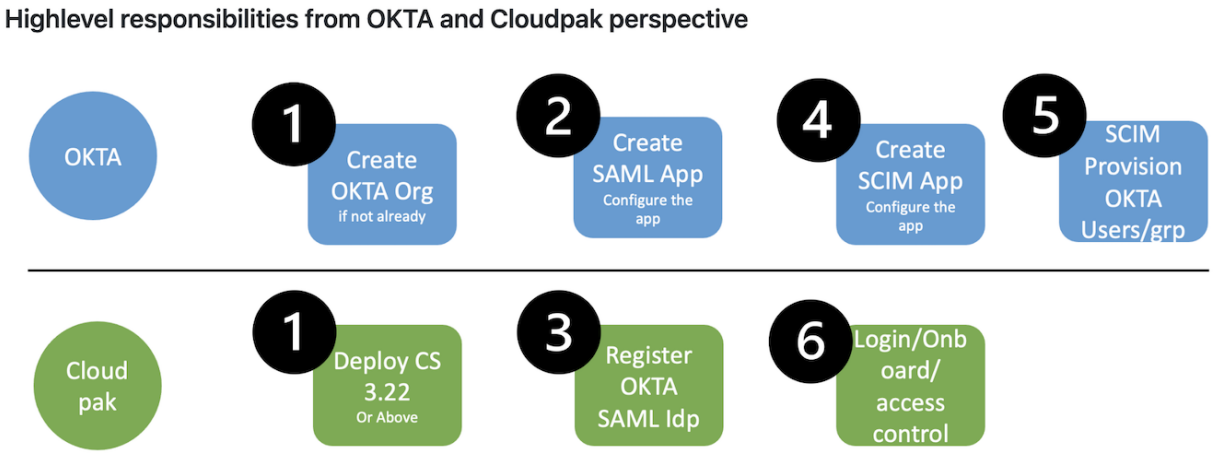
- Use the following process until an IBM-specific application is published in the Okta Integration Network (OIN) catalog.
- You can create a free Okta dev org by signing up on the OKTA Dev portal. Both personal and official accounts are accepted. Once you create a dev account, you can log in to the dev account to get the main Okta dashboard.

See the following prerequisites:

- You must have a valid active OKTA account and org.
- The cp-console route must be displayed with the trusted certificate.
- The cp-console route must be publicly accessible over an internet unless Okta is deployed in a private tenant.

Follow these steps to create an application:

1. Create an Okta Org, if you don't have one. And, deploy the foundational services version 4.x.
2. Create a SAML app and configure the app at Okta. For more information, see Configure Cloud Pak as SAML service provider (SP) at OKTA.
3. Register Okta SAML IdP at your Cloud Pak. For more information, see Configure OKTA as SAML IdP at Cloud Pak.
4. Create a SCIM app. Configure the app at Okta. For more information, see Configure SCIM integration with Cloud Pak SCIM Server.
5. Provision SCIM for Okta users and groups. For more information, see Manage users and Manage groups.
6. Log in or onboard to the Cloud Pak.



Primarily, the following main configurations are required to get OKTA-SCIM integration to work with your Cloud Pak along with SAML SSO.

- Configure Cloud Pak as SAML service provider (SP) at OKTA
- Configure OKTA as SAML IdP at Cloud Pak
- Configure SCIM integration with Cloud Pak SCIM Server

Configuring Cloud Pak as SAML SP at Okta and configuring Okta as SAML IdP at Cloud Pak are performed to make the e2e SSO to work. And, configuring SCIM integration with Cloud Pak SCIM server is performed to enable SCIM provisioning.

Configure Cloud Pak as SAML service provider (SP) at OKTA

Before you begin: You must have the SP **Cloud Pak CommonServices IM liberty** SAML metadata. For more information, see SAML metadata export by using samlmetadata API.

Steps

1. Log in to the OKTA org. You see the Okta dashboard.
2. From the navigation menu, click **Applications** > **Applications**. You see the **Applications** page.
3. Click **Create App Integration**.
4. Select **SAML 2.0**.
5. Click **Next**.
6. Provide the App name.
7. Click **Next**.
8. Fill all the required details for the SAML configuration.
   - **Single sign-on URL:** Get ACS URL from the Liberty SAML metadata. For example, https://cpconsole/ibm/saml20/defaultSP/acs. To get the `cpconsoleroute`, see Getting host details.

- **Audience URI (SP Entity ID):** Get Entity ID from Liberty SAML metadata. For example, https://cpconsole/ibm/saml20/defaultSP. To get the `cpconsoleroute`, see Getting host details.
- **Name ID format:** EmailAddress
- **Application Username:** Okta username

9. Update the **Attribute Statements (optional)** section. It allows Okta to send the attributes in SAML response to Cloud Pak. **Important:** Even though the section name indicates **(optional)**, you must provide at least one SAML attribute. Otherwise, the SAML configuration fails.

   Following are the default **User** attributes:

   - Name: **uid**, Name format: **Basic**, Value: **user.email**
   - Name: **firstName**, Name format: **Basic**, Value: **user.firstName**
   - Name: **lastName**, Name format: **Basic**, Value: **user.last.Name**
   - Name: **emailAddress**, Name format: **Basic**, Value: **user.email**

   The following are the default **Group** attributes:

   - Name: **groups**, Name format: **Basic**, Filter: **Equals** (it can be adopted based on the groupName), Value: **groupName** (groupName should be the Okta groupName)

     Note: While registering the Okta IdP, you can use the same default keys of the **Name** field (uid, firstName, lastName, emailAddress, groups) to specify the values in the `token_attribute_mappings` section. In case you customized the attributes keys, use the customized keys only to specify the values in the `token_attribute_mappings` section. For more information, see SAML attributes statements. The following is an example mapping of the use of default attribute keys to specify the values in `token_attribute_mappings` section:

```
"token_attribute_mappings": {
            "sub": "uid",
            "given_name": "firstName",
            "family_name": "lastName",
            "groups": "groups",
            "email": "emailAddress",
            "first_name": "firstName",
            "last_name": "lastName"
}
```

10. Click **Preview the SAML Assertion** to reconfirm whether the SP metadata is configured correctly.
11. Click **Next** and select an option to specify whether you are a customer or partner.
12. Click **Finish**. You see the **Sign on** tab page.
13. In the **SAML Setup** section, click **View SAML setup instructions**. You then see a page with instructions to configure SAML with your application.
14. From the **Optional** section, copy the IdP metadata and save it to a file named `idp-metadata-xml.xml`.

After you copy the IdP metadata, assign the SAML application to the user or group who need to log in at the Cloud Pak end.

Note: This is a mandatory procedure for the user to be able to log in.

1. From the navigation menu, click **Applications** > **Applications**. You see the **Applications** page.
2. Select the application and users or groups that you want to assign to the application.

If you want to assign existing LDAP users or group to the App, follow the procedure to first integrate OKTA Agent to LDAP Server. Once it is done, the LDAP entities can be imported to OKTA and used for assignment.

Configure Okta as SAML IdP at Cloud Pak

Use the IdP V3 API to register OKTA as SAML IdP. For now, the Okta registration is not supported through UI. Therefore, you need to perform the registration by using API only. However, the process remains the same. For more information about the IdP v3 usage, see IdP V3 registration with Okta.

With the registration IdP v3 API, the required Oauth client is also created. This Oauth client is used for configuring SCIM server Oauth authentication from Okta.

Note: Provide the `redirect_uri` for `oauthClient` registration.

Upon successful registration, you get `clientId` and `clientSecret` name. From the `clientSecret`, both the `clientId` and `clientSecret` can be retrieved.

To be concluded, the IdP v3 Okta registration does the following:

- Register Okta as SAML IdP
- Register Okta to Liberty `oauthClient` while returning the client credentials.

See the following prerequisites:

- Download the `idp-metadata-xml` from Okta.
- Encode the `idp-metadata-xml` content as base64. For example, `cat idp-metadata-xml |base64`.
- You need to pass the `redirect_uri` for OKTA oauth client creation along with IdP V3 registration.

Getting the `redirect_uri`

Based on the environment (a standard org, an EMEA org, and an Okta Preview org), the `redirect_uri` can be one of the following:

- https://system-admin.okta.com/admin/app/cpc/${appName}/oauth/callback
- https://system-admin.okta-emea.com/admin/app/cpc/${appName}/oauth/callback
- https://system-admin.oktapreview.com/admin/app/cpc/${appName}/oauth/callback

Note: Replace the **${appName}** with the variable name for your application instance. If you don't know the variable name for application instance, you can find it as part of the App Embed link from the browser by opening the embed link, or you can also check the **Name** attribute for the application at the **`/Apps API`** endpoint.

For example, if the URL embedded link is https://dev-68798908-admin.okta.com/admin/app/dev-68798908_iamsaml_3/instance/0oa6vnyhkm86B4xha5d7/#tab-general, the **${appName}** will be **dev-68798908_iamsaml_3** and the **`redirect_uri`** is, "redirect_uri": [https://system-admin.okta.com/admin/app/cpc/dev-68798908_iamsaml_3/oauth/callback].

Steps

1. Use the IdP V3 API to register OKTA as SAML IdP.

   The sample for OKTA registration as SAML IdP :

```
curl --location --request POST 'https://cpconsoleroute/idprovider/v3/auth/idsource' \
--header 'Authorization: Bearer $accessToken' \
--header 'Content-Type: application/json' \
--data-raw
'{
   "name": "OKTASAML",
   "description": "OKTASAML",
   "protocol": "saml",
   "type": "okta",
   "idp_config":{
      "token_attribute_mappings": {
        "sub": "uid",
        "given_name": "firstName",
        "family_name": "lastName",
        "groups": "groups",
        "email": "emailAddress"
        "first_name": "firstName",
        "last_name": "lastName"
   },
     "idp_metadata": "Base64-encoded-metadata"
   },
   "jit": false,
   "scim_config":{
      "redirect_url": [
        "okta_redirect_url"
      ],
      "scim_attribute_mappings": {
          "user":{
            "principalName": "userName",
            "name":{
              "givenName": "givenName",
              "familyName": "familyName"
            },
            "displayName": "displayName",
            "emails": [
              {
                 "value": "emails",
                 "type": "home"
               }
               ],
               "id": "id",
               "userName": "userName"
             },
             "group": {
                 "principalName": "displayName",
                 "displayName": "displayName",
                 "id": "displayName"
            }
          }
   }
 }'
```

2. Upon successful registration, you might get the status similar to the following:

```
{
   "status": "success",
   "message": "Identity provider {OKTASAML} is successfully registered with unique identifier : defaultSP,
oktaclient created with clientId: lgz9i024740cyuc0vl6eh3g7f6mv2b1f , client secret: okta-oidcclient-secret"
}
```

   The error message resembles the following:

```
401 "error": "Insufficient user permission for role : rolename"
400 "error": "schema error:schema validation error followed by error"
400 "error": "duplicate : Idp with protocol=saml is already created"
500 "error": "Other errors"
```

   token_attribute_mappings: **`token_attribute_mappings`** is a custom mapping that user can provide based on the SAML attribute mapping that is configured at Okta. User must modify the value, not the key, based on the Okta SAML attribute configuration. For more information about attributes, see Step 9 in Configure Cloud Pak as SAML service provider (SP) at OKTA.

   scim_attribute_mappings: Provide the mapping attributes as in the CURL command. The defined mapping is used in the filter query, creating attributes and the response to be sent.

Note: You can get the client credential details from the secret that was created. These client credential details are sent back as a response.

To get the Oauth client credentials:

1. Log in to the OpenShift Container Platform by using `oc` CLI.
2. Run the following commands:

```
oc get secrets/okta-oidcclient-secret --template={{.data.CLIENT_ID}} | base64 --decode
```

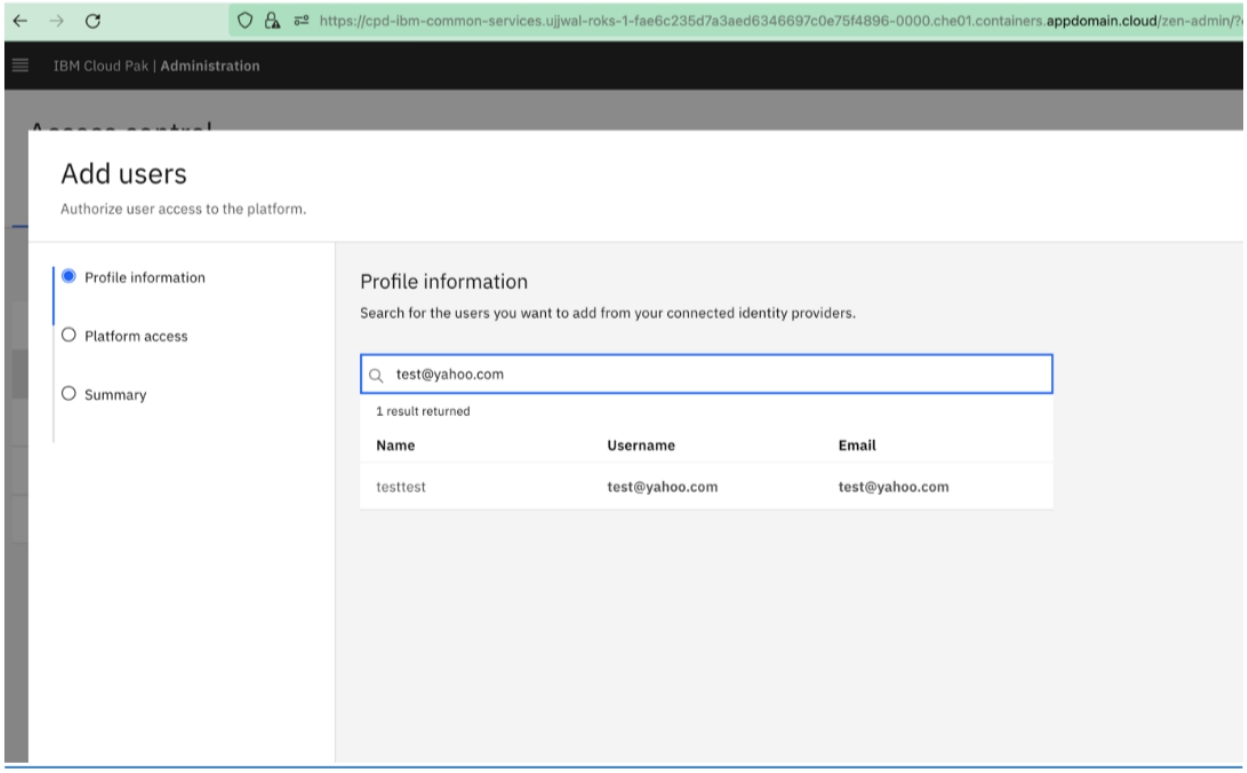
```
oc get secrets/okta-oidcclient-secret --template={{.data.CLIENT_SECRET}} | base64 --decode
```

Once the registration is done, the SAML configuration can be validated by login through the cp-console directly. The default role is **Viewer**.

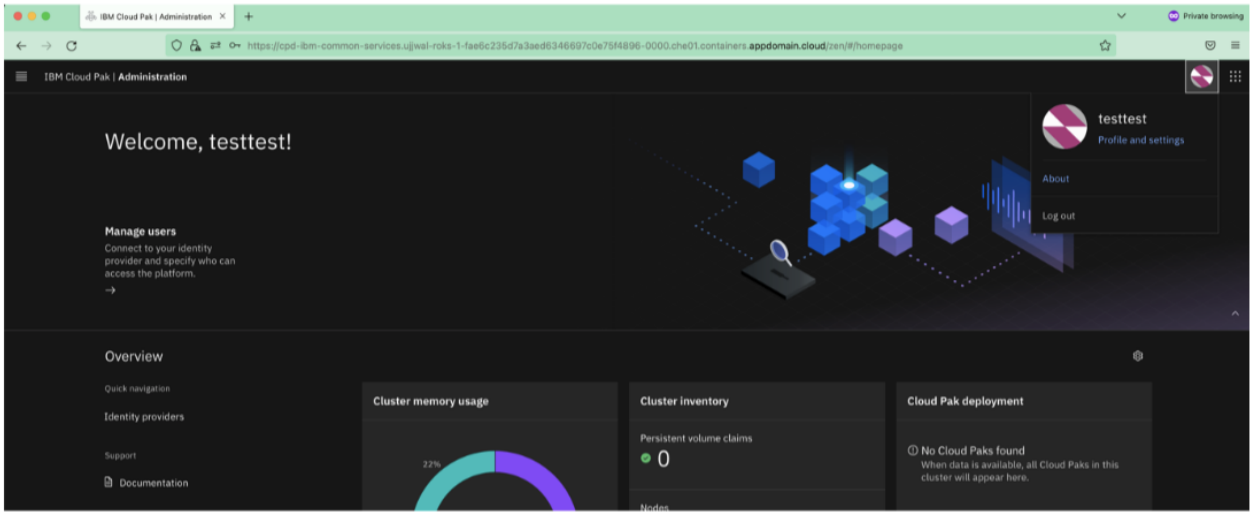
For Platform UI SAML login, you need to first complete the OKTA SCIM integration so that the provisioned user or group can be onboarded at Platform UI end. If you have not already completed the SCIM Integration and provisioned for the users and groups, then complete the steps that are mentioned in Configure SCIM integration with Cloud Pak SCIM server before you proceed further.

To log in to the Platform UI console, follow these steps:

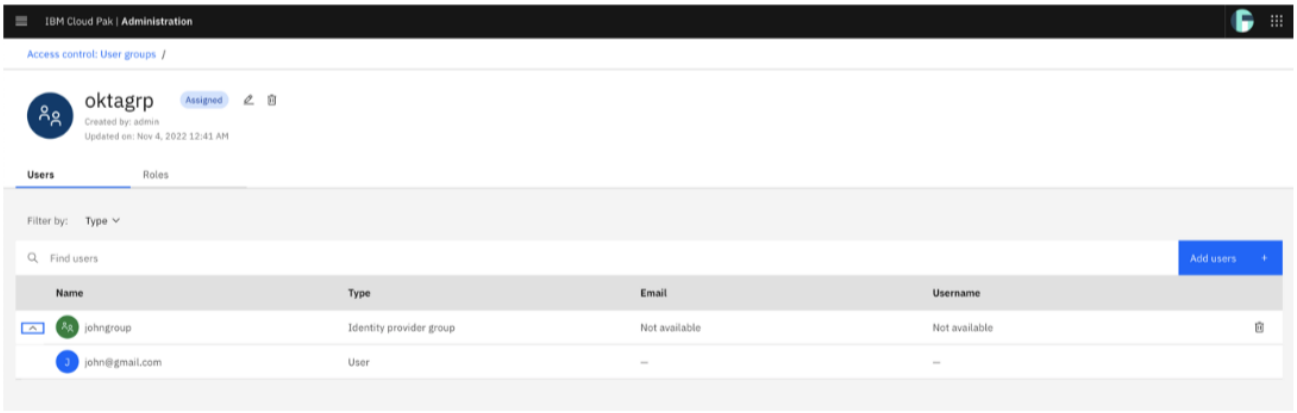
1. Onboard the user to Platform UI from the IM Scim Server. For example, **test@yahoo.com**, a user that was provisioned from OKTA to IM SCIMServer. For more information, see steps that are mentioned in Configure SCIM integration with Cloud Pak SCIM server.



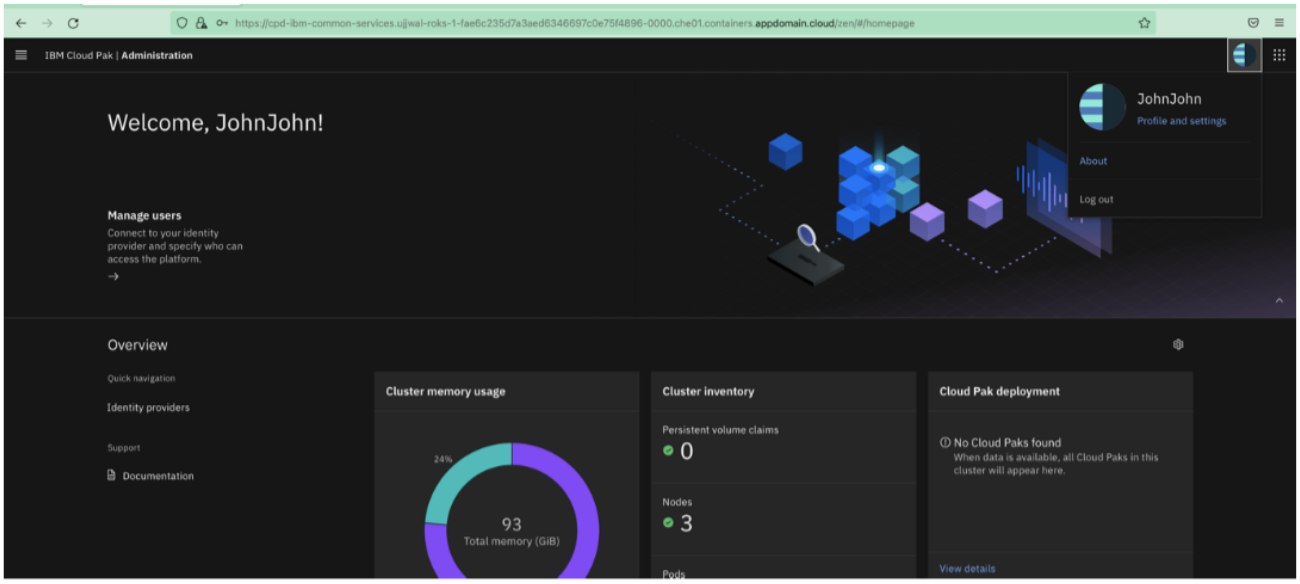
2. Once the user is onboarded, login to the Platform UI console with that user.



3. Similarly, the provisioned group can also be onboarded to the Platform UI console from the IM ScimServer.



Member of the group can log in to the Platform UI console.

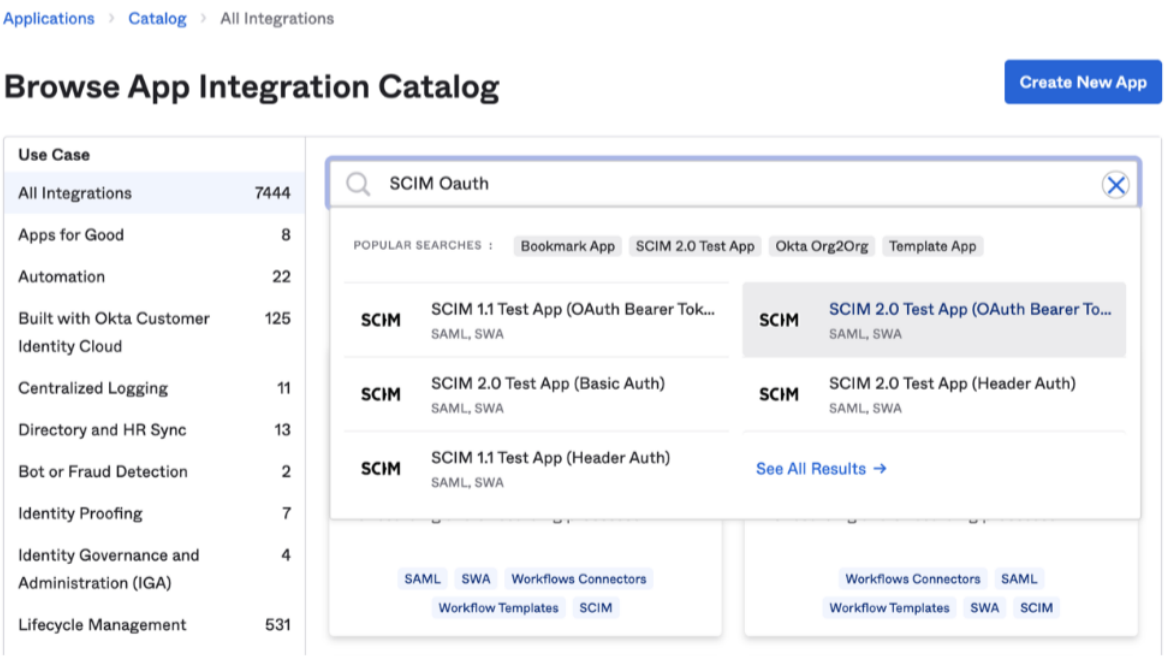


Configure SCIM integration with Cloud Pak SCIM server

Note: You can follow this configuration process until the IM SCIM App is published to the OIN catalog. Later, you can use the proper Oauth configurations with clientID and clientSecret with the IBM Specific Integration App that was created as a part of IdP V3 registration in Configure Okta as SAML IdP at Cloud Pak.

Steps

1. Go to Okta dashboard.
2. Go to **Applications** > **Applications**. The **Application** page appears.
3. Select **Browse App Catalog**. The **Browse App Integration Catalog** page appears.
4. In the search tab, type **SCIM 2.0 Oauth**. Then, select **SCIM 2.0 Test App (OAuth Bearer Token)** from the displayed options.

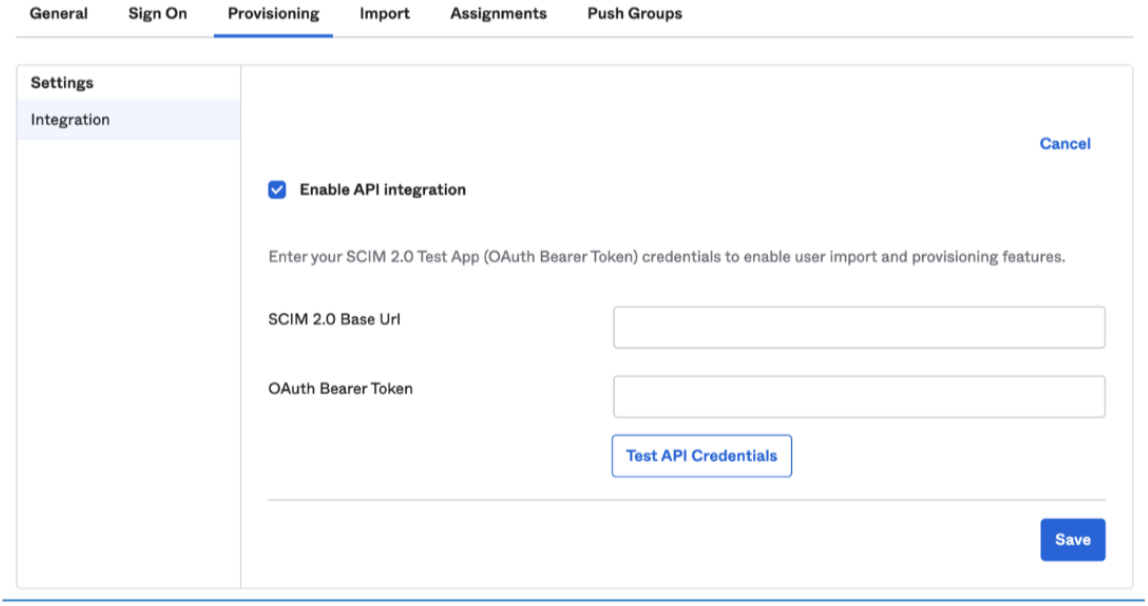


5. Select **Add Integration**.
6. Click **Next**. Then, click **Done**. Ignore the SSO configuration part. Leave it as it is.
7. Go to **Provisioning** > **Integration**.
8. Select **Enable API Integration**.
9. Update the following details:
   - **SCIM 2.0 Base URL**: https://cpconsoleroute/idmgmt/identity/api/v1/scim

   Note: To get `cpconsoleroute` from cluster, see Getting the host details.
   - **OAuth Bearer token**: Update the accessToken as Oauth Bearer Token. To get the OAuth Bearer Token from cluster:
     - Log in to the OpenShift Container Platform cluster by using `oc` CLI.
     - Run the following commands:

       ```
       oc get secrets/platform-auth-scim-credentials --template={{.data.scim_admin_password}} | base64 --decode
       ```

       ```
       curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -d
       "grant_type=password&username=scimadmin&password=passwordfromabove&scope=openid"
       https://cpconsoleroute/idprovider/v1/auth/identitytoken --insecure
       ```



10. Test the API credential.

    See the following notes:

    - It is expected that this validation might fail if you are trying for the first time as the Okta requires at least one user to be returned when the Okta validates the token. Therefore, it is advisable to manually create a user or run the runscope before you test the API credentials.
    - **accessToken** in the following CURL command is the same as the value retrieved from OAuth Bearer token in Step 9 in Configure SCIM integration with Cloud Pak SCIM server.

- Create a user manually.

```
curl --location --request POST 'https://cpconsoleroute/idmgmt/identity/api/v1/scim/users' \
--header 'Authorization: Bearer ${accessToken}' \
--header 'Content-Type: application/json' \
--data-raw '{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
  "userName": "Dummy",
  "name":
  {
    "givenName": "Dummy",
    "middleName": "",
    "familyName": "Dummy"
  },
  "emails":
  [{
    "primary": true,
    "value": "Dummy@ibm.com",
    "type": "work"
  }],
    "displayName": "Dummy",
    "externalId": "Dummy",
    "groups": [],
    "active": true
  }'
```

- To run the RunScope, see Scope of tests.

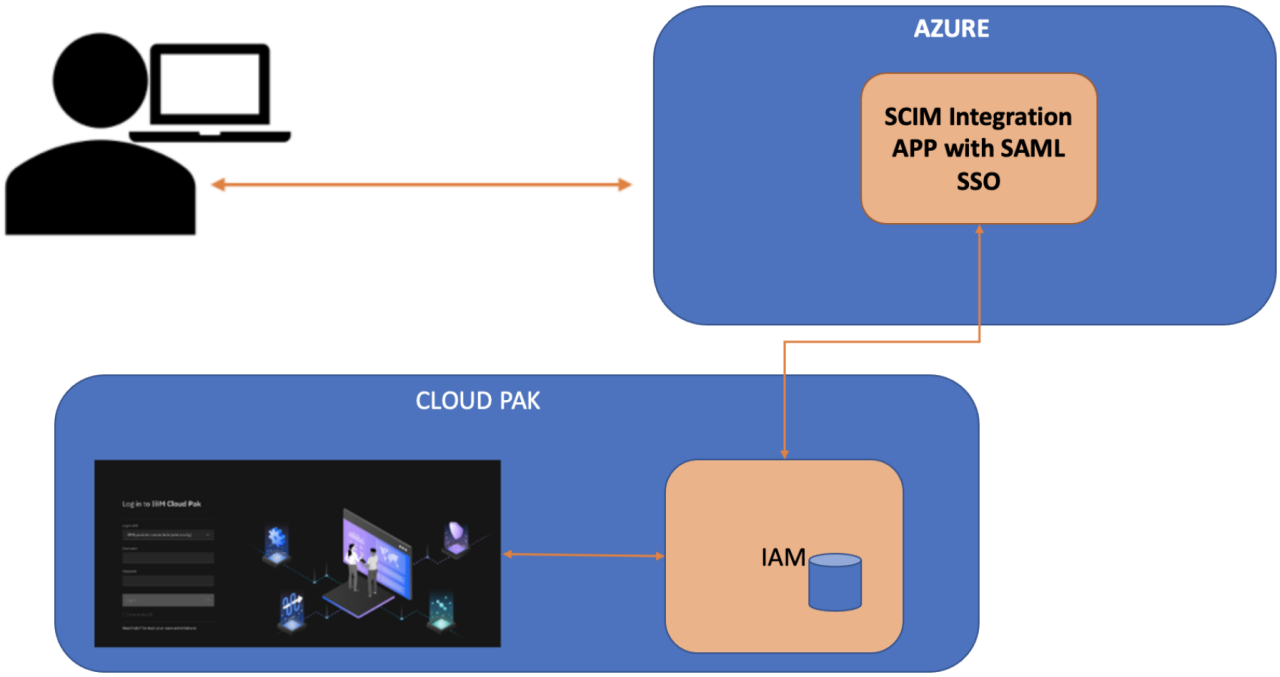
11. Once the token is verified, follow the process to start provisioning for Managing users and Managing groups.

Note: If you want to provision existing LDAP users or groups to the App, follow the procedure to first integrate OKTA Agent to LDAP Server. Once it is done, the LDAP entities can be imported to Okta and used for the assignment.

IBM Cloud Pak SCIM Azure AD integration

From foundational services version 3.19.9, 3.23, and later, Cloud Pak can use the Azure-IM integration by using SCIM to manage the users and groups at the CloudPak end. The Azure-IM integration will also help in managing the authentication and authorization for the resources.

Azure Active Directory (Azure AD) is an enterprise identity service that provides single sign-on, multifactor authentication, and conditional access to protect against cybersecurity attacks.



Configuration required for Azure integration using SSO/SAML and SCIM

See the following notes:

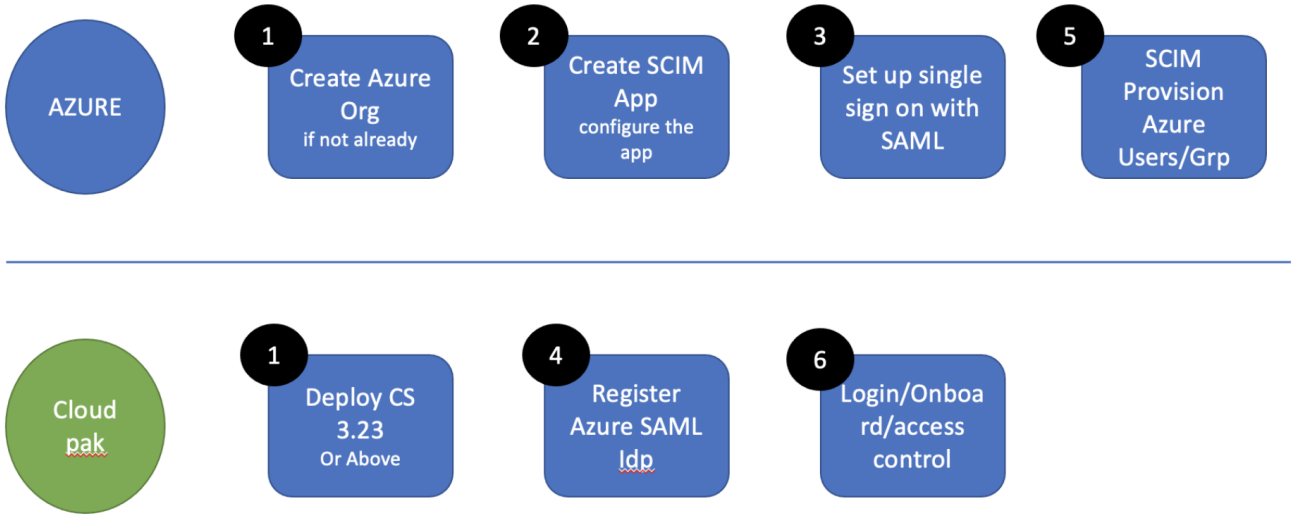
- Follow this process until you have the IBM Specific app published in Azure AD application gallery.
- You can create free Azure dev org by signing up to Azure Dev portal. Both personal and official accounts are accepted. Once dev account is created, you can login to the dev account to get the Azure main dashboard.

See the following prerequisites:

- Cloud Pak must have foundational services version 3.19.9, 3.23, or later.
- You must have a valid active Azure account and org.
- CP console route must be displayed with the trusted certificate.
- CP console route must be publicly accessible over an internet unless Azure is deployed into private tenant.

Follow the steps in sequence:

1. Create Azure Org if you have not created the Azure Org. And, deploy the foundational services version 3.19.9, 3.23, and later.
2. Create the SCIM app and configure the app at Azure. For more information, see Configure CloudPak as SAML service provider (SP) at Azure.
3. Set up single-sign-on with SAML.
4. Register Azure SAML IdP at Cloud Pak. For more information, see Configure Azure as SAML IdP at CloudPak.
5. Perform SCIM provisioning for Azure users and groups. For more information, see Manage users and Manage groups.
6. Login or onboard to the Cloud Pak.



Primarily, the following main configurations are required to get Azure SCIM Integration working with CloudPak along with SAML SSO.

- Configure Cloud Pak as SAML service provider (SP) at Azure
- Configure Azure as SAML IdP at Cloud Pak
- Configure SCIM Integration with Cloud Pak SCIM Server

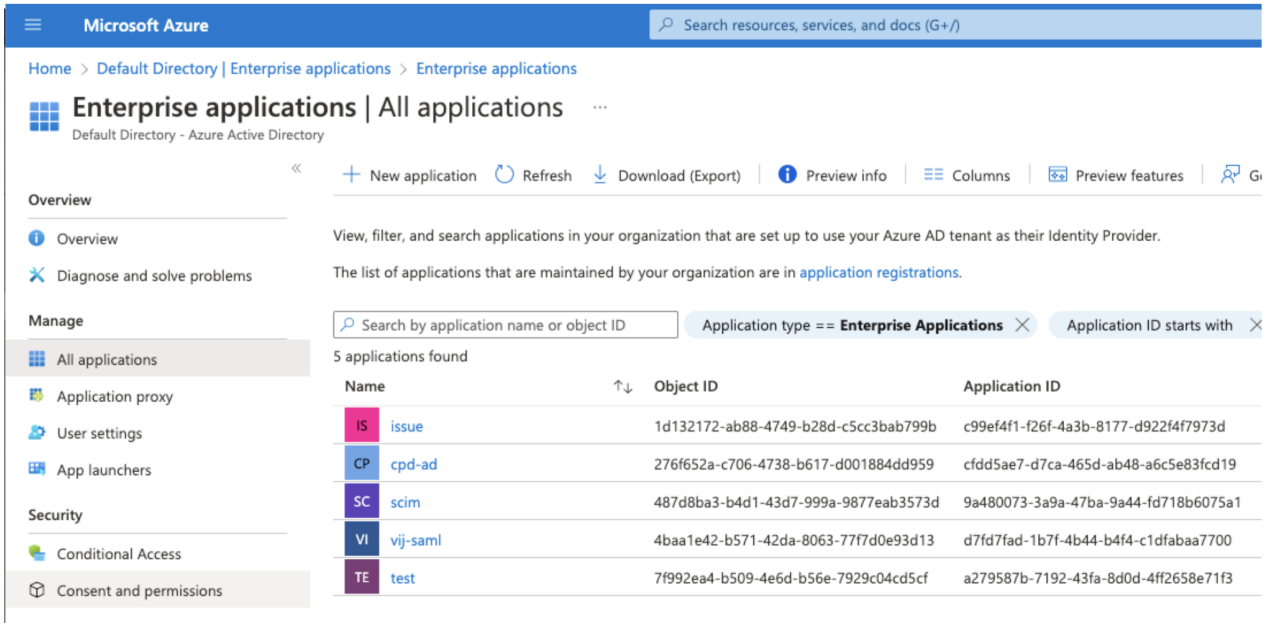
Configuring Cloud Pak as SAML SP at Azure and configuring Azure as SAML IdP at Cloud Pak are performed to make the e2e SSO work. And, configuring SCIM integration with CloudPak SCIM server is performed to enable SCIM provisioning.

Configure Cloud Pak as SAML service provider (SP) at Azure

Before you begin: You must have the SP, **CloudPak CommonServices IM liberty** SAML metadata. You can use the `samlmetadata` API to get the SAML metadata. For more information, see SAML metadata export by using samlmetadata API.

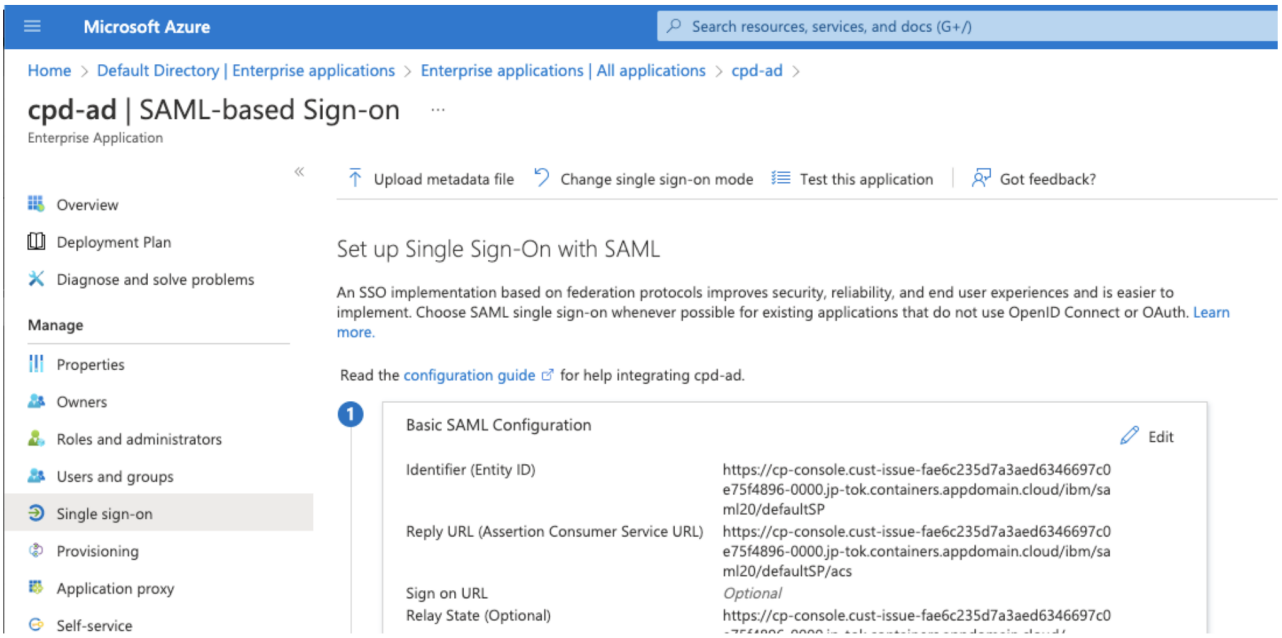
Steps

1. Create a custom FreeStyle enterprise application. Complete the following steps:
    1. Login to Azure org.
    2. Select **Microsoft Entra ID** in the `Azure Services` section.
    3. Click **+ Add** and select **Enterprise application**.
    4. Click **+ Create your own application** and enter the name of your app in the `Input name` section.
    5. Select **Integrate any other application you don't find in the gallery (Non-gallery)** and click **Create**.
2. Select your enterprise application from the list in the **Enterprise applications** page.



3. Click **Get started** in the **Set up single sign on** to configure single sign-on method for your enterprise application.

4. Click **SAML** in the **`Single sign-on`** page.
5. Select **Upload metadata file** option and upload the SP metadata file and click **Add**.



6. Confirm the **Attributes & Claims** names and values.

   The default **User** attributes:

   - Name: **http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name**, Type: **SAML**, Value: **userprincipalname**
   - Name: **http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname**, Type: **SAML**, Value: **user.givenname**
   - Name: **http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname**, Type: **SAML**, Value: **user.surname**
   - Name: **http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress**, Type: **SAML**, Value: **user.mail**

   The default **Group** attributes:

   - Name: **http://schemas.microsoft.com/ws/2008/06/identity/claims/groups**, Type: **SAML**, Filter: **Equals** (it can be adopted based on the groupName), Value: **groupName** (groupName should be the Azure groupName and it should be any of the following: None, All groups, Security groups, Directory roles, Groups assigned to the application)

   Note: While registering the Azure IdP, you can use the same default keys of the **Name** field to specify the values in the **`token_attribute_mappings`** section. The values allows Azure to send the attributes in SAML response to Cloud Pak. In case, you have customized the attributes keys, use the customized keys only to specify the values in the **`token_attribute_mappings`** section. For more information, see SAML authentication with Azure Active Directory. For example, if you are using the default attributes keys to specify the values in **`token_attribute_mappings`** section:

```
"token_attribute_mappings": {

   "sub": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name",
   "given_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname",
   "family_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname",
   "groups": "http://schemas.microsoft.com/ws/2008/06/identity/claims/groups",
   "email": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress",
   "first_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname",
   "last_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"
}
```

7. Once the metadata is saved, go to the SAML certificate section and download **Federation Metadata XML** and save as **`idp-metadata-xml.xml`**.
8. Click **Test this application**.
9. Verify the login by clicking **Test single sign-on with cpd-ad** section.
10. Go to SCIM app and assign the user and/or group who are required to login at Cloud Pak end. To assign the user and/or group:

    - Under **Manage** section, select **Enterprise applications** from the menu. The **Enterprise applications** page appears.
    - Select **Assign users and groups**. Then, select the users and groups that you want to assign.

If you want to assign existing LDAP users or group to the App, use Azure Active Directory (Azure AD) to create a domain in Azure and link the Azure AD to an on-premises AD domain. For more information, see Integrate on-premises AD with Azure. After this, the LDAP entities can be imported to Azure and it can be used for assignment.

Configure Azure as SAML IdP at Cloud Pak

Use the IdP V3 API to register Azure as SAML IdP. For now, the Azure registration is not supported through UI. Therefore, you need to perform the registration using API only. However, the process remains the same. For more information about the IdP v3 usage, see IdP V3 registration with Azure.

With the registration IdP v3 API, the required Oauth client is also created. This Oauth client will be used for configuring SCIM server Oauth authentication from Azure.

Note: Provide the `redirect_uri` for `oauthClient` registration.

Upon successful registration, you will get `clientId` and `clientSecret` name. From the `clientSecret`, both the `clientId` and `clientSecret` can be retrieved.

To be concluded, the IdP v3 Azure registration does the following:

- Register Azure as SAML IdP.
- Register Azure to Liberty `oauthClient` while returning the client credentials.

See the following prerequisites:

- Download the `idp-metadata-xml` from Azure as mentioned in step 7 in Configuring Cloud Pak as SAML service provider (SP) at Azure.
- Encode the `idp-metadata-xml` content as base64. For example, `cat idp-metadata-xml |base64`.
- You need to pass the `redirect_uri` for Azure Oauth client creation along with IdP V3 registration.

Steps

1. Use the IdP V3 API to register Azure as SAML IdP.

   The following sample for Azure registration as SAML IdP. To get `accessToken`, see Get access token by using cpclient_credentials.

```
curl --location --request POST 'https://cpconsoleroute/idprovider/v3/auth/idsource' \
--header 'Authorization: Bearer $accessToken' \
--header 'Content-Type: application/json' \
--data-raw
'{
   "name": "AzureSAML",
   "description": "AzureSAML",
   "protocol": "saml",
   "type": "azuread",
   "idp_config":
    {
      "token_attribute_mappings": {
        "sub": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name",
        "given_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname",
        "family_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname",
        "groups": "http://schemas.microsoft.com/ws/2008/06/identity/claims/groups",
        "email": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
        "first_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname",
        "last_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"
    },
      "idp_metadata": "Base64-encoded-metadata"
    },
    "jit": false,
    "scim_config":
    {
      "redirect_url": ["azure_redirect_url"],
        "scim_attribute_mappings": {
          "user":
          {
            "principalName": "userName",
            "name":
            {
              "givenName": "givenName",
              "familyName": "familyName"
            },
            "displayName": "displayName",
            "emails":
            [
            {
              "value": "emails",
              "type": "home"
            }
            ],
            "id": "id",
            "userName": "userName"
            },
            "group":
            {
             "principalName": "displayName",
             "id": "displayName"
             "displayName": "displayName"
            }
          }
      }
}'
```

2. Upon successful registration, you might get the status similar to the following:

```
{
   "status": "success",
   "message": "Identity provider {AzureSAML} is successfully registered with unique identifier : defaultSP,
azureadclient created with clientId: yx3a5pwtdrmn0sf8ockyf97dg55lswo7 , client secret: azuread-oidcclient-secret"
}
```

   The error message resembles the following:

```
401 "error": "Insufficient user permission for role : rolename"
400 "error": "schema error:schema validation error followed by error"
```

```
400 "error": "duplicate : Idp with protocol=saml is already created"
500 "error": "Other errors"
```

token_attribute_mappings: `token_attribute_mappings` is a custom mapping that user can provide based on the SAML attribute mapping that is configured at Azure. User should modify the values, not the key, based on the Azure SAML attribute configuration. For more information about attributes, see Step 6 in Configure CloudPak as SAML service provider (SP) at Azure.

scim_attribute_mappings: Provide the mapping attributes as in the CURL command. The defined mapping will be used in filter query, creating attributes and the response that is to be sent.

Note: You can get the client credential details from the secret that has been created. This client credential details are sent back as a response.

To get the Oauth client credentials:

- Login to the OpenShift Container platform by using `oc` CLI.
- Run the following commands:

```
oc get secrets/azuread-oidcclient-secret --template={{.data.CLIENT_ID} | base64 -D
```

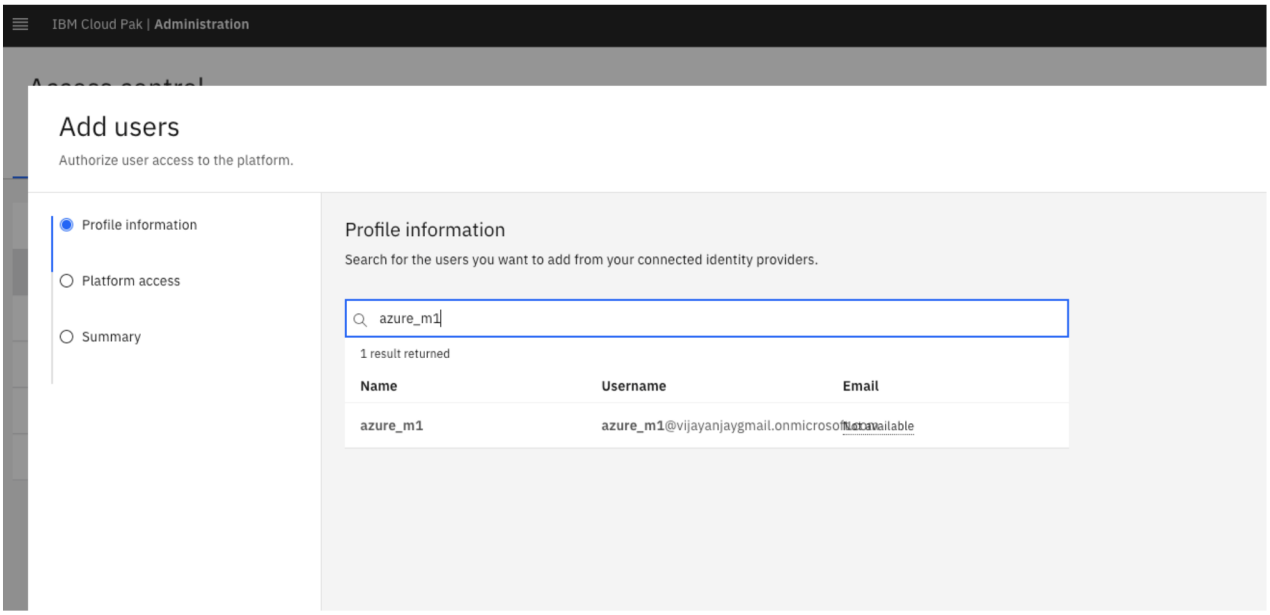
```
oc get secrets/azuread-oidcclient-secret --template={{.data.CLIENT_SECRET}} | base64 -D
```

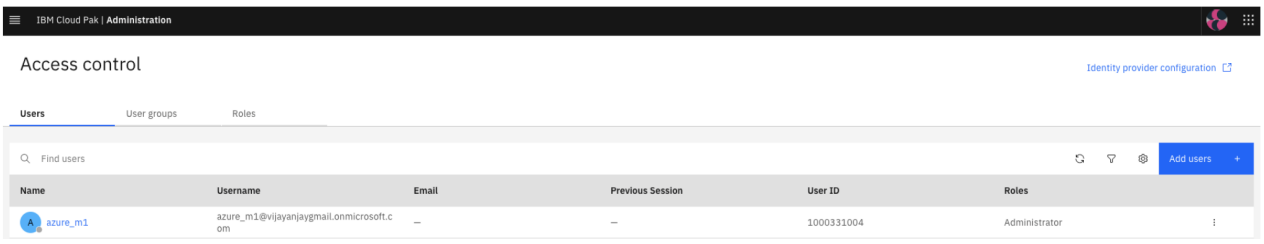
Once the registration is done, the SAML configuration can be validated by login through the cp-console directly. The default role will be **Viewer**.

For Platform UI SAML login, you need to first complete the Azure SCIM integration so that the provisioned user or group can be onboarded at Platform UI end. If you have not already completed the SCIM Integration and provisioned required for the users and groups, then complete the steps that are mentioned in Configure SCIM integration with CloudPak SCIM server before you proceed further.

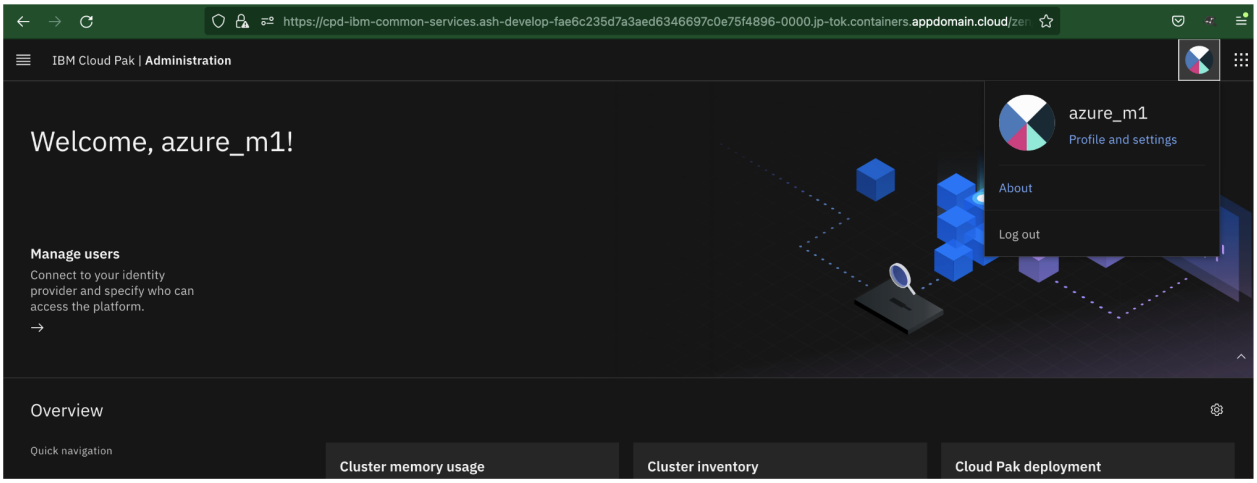
To login Platform UI console, follow these steps:

1. Onboard the user to Platform UI from the IM ScimServer. For example, **azure_m1@vijayanjaygmail.onmicrosoft.com**, a user that has been provisioned from Azure to IM SCIMServer. For more information, see steps that are mentioned in Configure SCIM integration with Cloud Pak SCIM server.

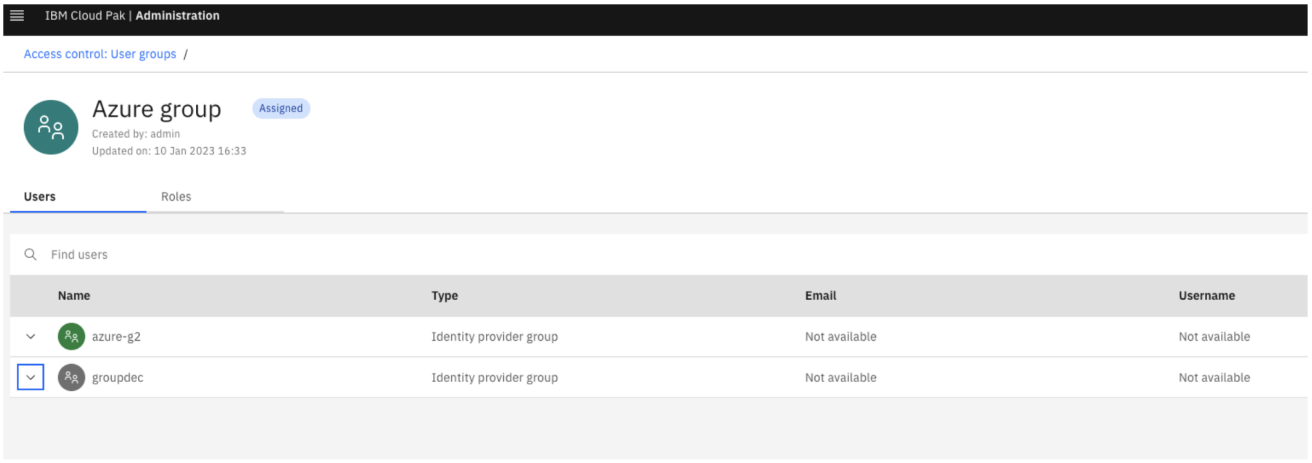




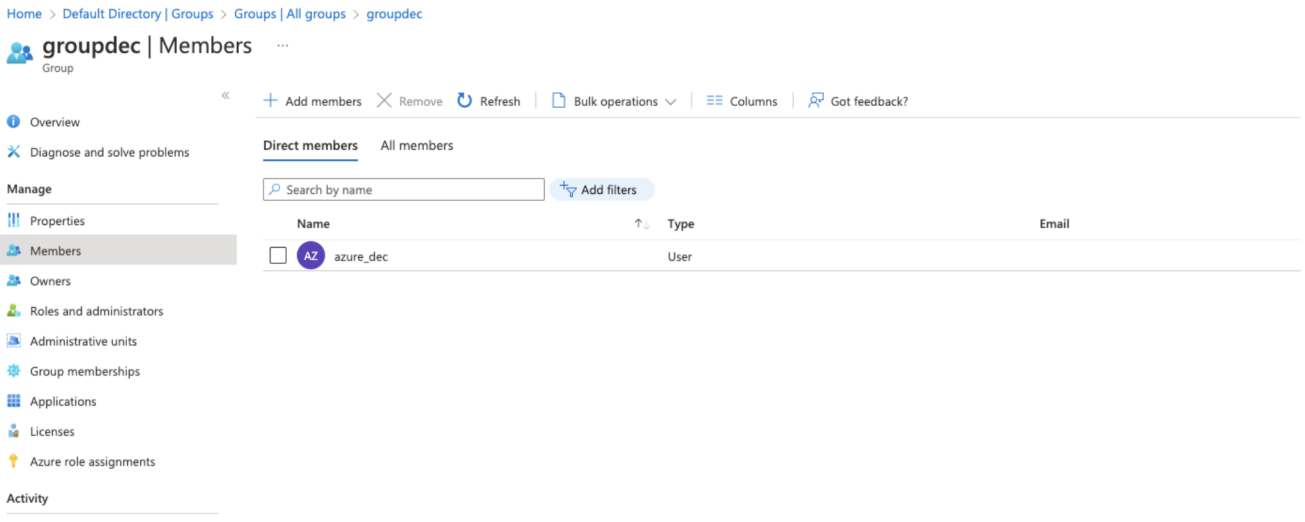
2. Once the user is onboarded, login to the Platform UI console with that user.



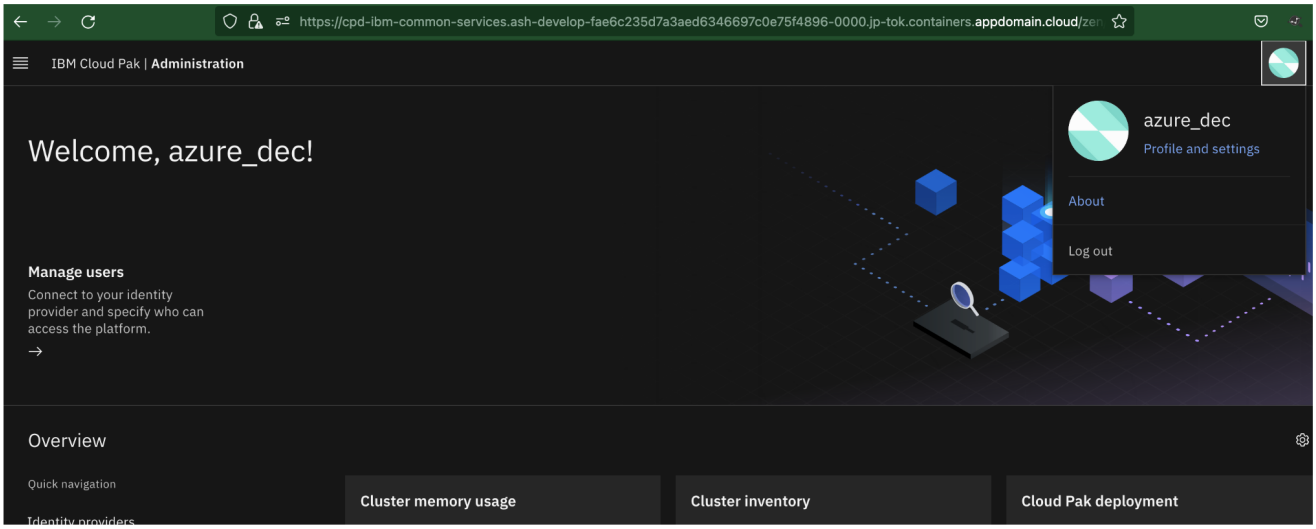
3. Similarly, the provisioned group can also onboarded to Platform UI console from the IM ScimServer.



Azure_dec is the member which is available inside **groupdec**.



Member of the groups can login to the Platform UI console.

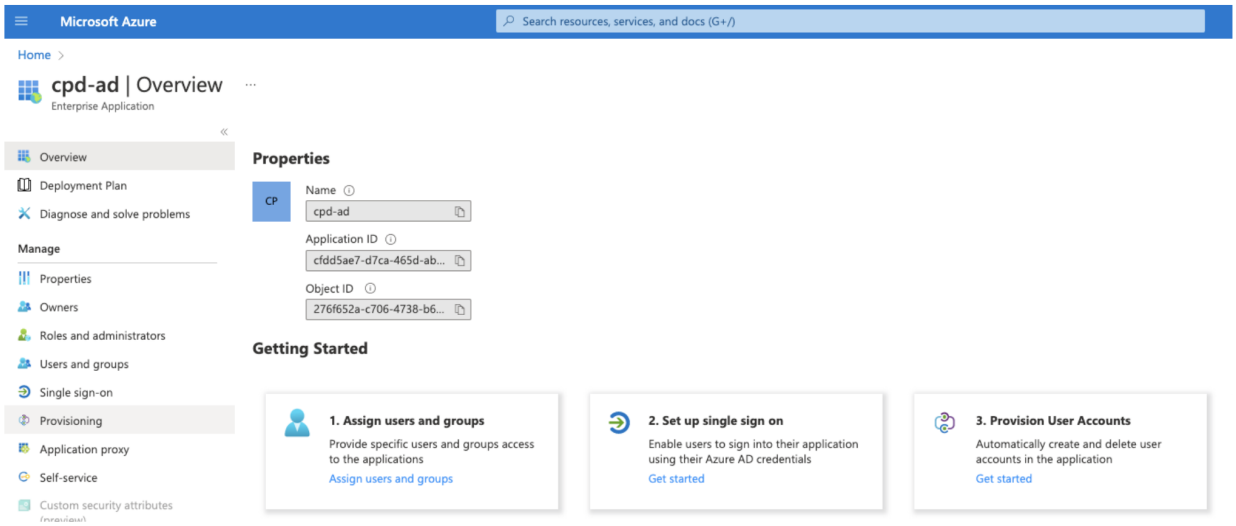


Configure SCIM integration with CloudPak SCIM server

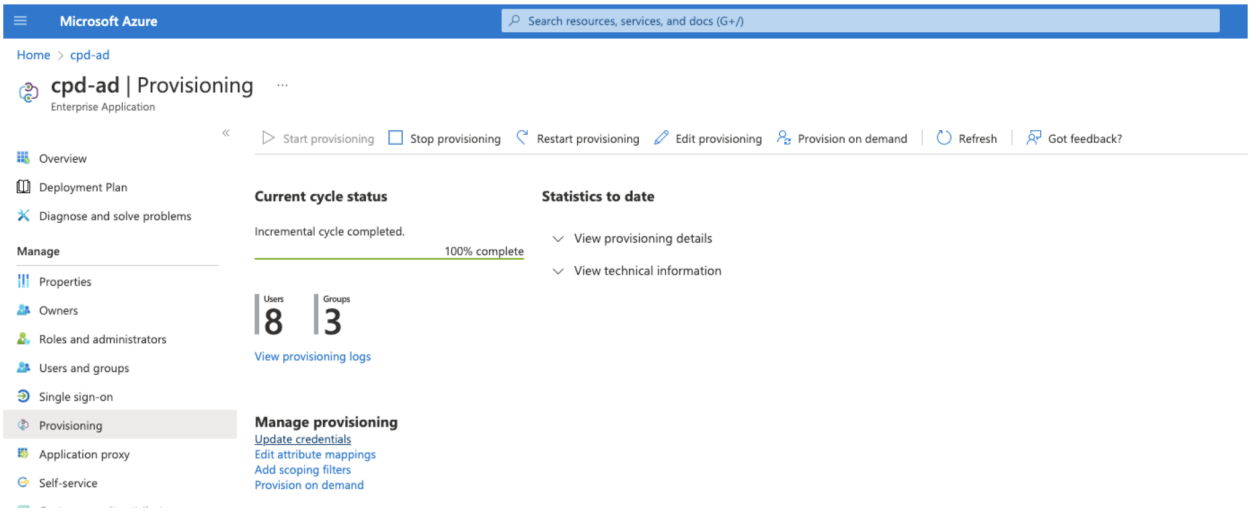
Note: You can follow this configuration process until the IM SCIM App is published to Azure AD app gallery. Later, you can use the proper Oauth configurations with clientID and clientSecret with the IBM Specific Integration App that has been created as a part of IdP V3 registration in Configure Azure as SAML IdP at Cloud Pak.

Steps

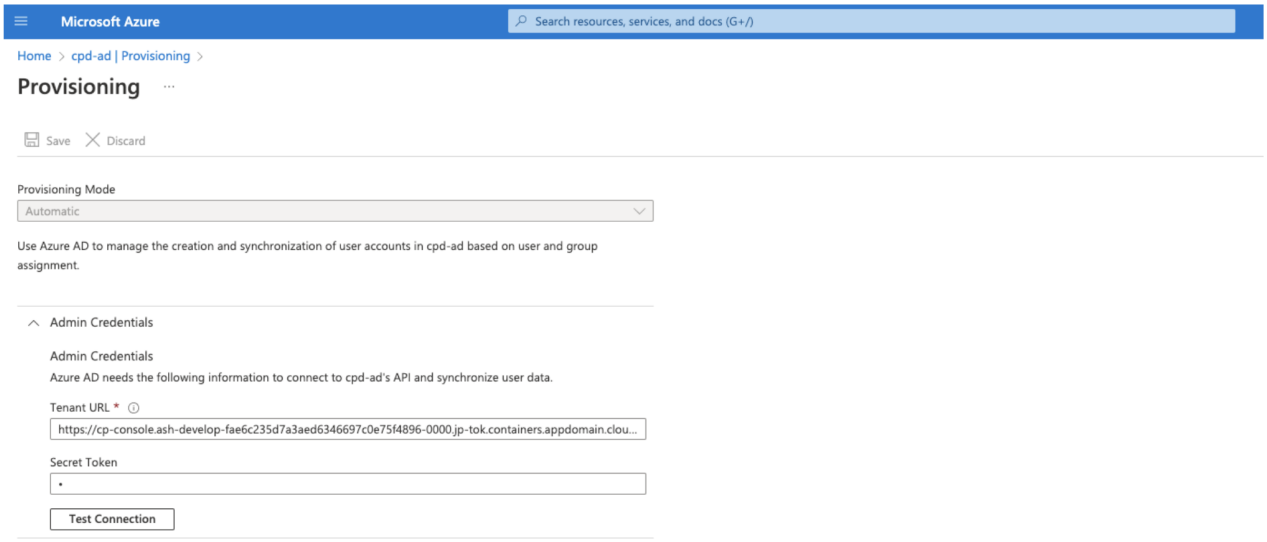
1. Go to Azure Active Directory dashboard .
2. Under **Manage** section, select **Enterprise applications** from the menu. The **Enterprise applications** page appears.
3. Select the enterprise application from the list that is displayed on **Enterprise applications** page. Then, select **Provisioning** from the navigation menu.



4. Click **Update credentials** under **Manage provisioning**.



5. In **Provisioning** page, under **Admin Credentials**, provide **Tenant URL** and **Secret Token**.



- **Tenant URL**: https://cpconsoleroute/idmgmt/identity/api/v1/scim

Note: To get `cpconsoleroute` from cluster, see Getting the host details.

- **Secret token**: Update the Secret Token as Oauth Bearer Token. To get the OAuth Bearer Token from cluster:
  - Login to OpenShift Container Platform cluster using `oc` CLI.
  - Run the following commands:

```
oc get secrets/platform-auth-scim-credentials --template={{.data.scim_admin_password}} | base64 -D

curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -d
"grant_type=password&username=scimadmin&password=passwordfromabove&scope=openid"
https://cpconsoleroute/idprovider/v1/auth/identitytoken --insecure
```

6. Click **Test Connection** to test API Credential. For more information, see Validate a SCIM endpoint.
7. Once the token is verified, follow the process to start provisioning for Managing users and Managing groups.

If you want to provision existing LDAP users or group to the app, follow the step to integrate on-premise AD with Azure.

IBM Cloud Pak foundational services JIT support for SCIM OKTA and Azure Active Directory

You can configure just-in-time (JIT) with Security Assertion Markup Language (SAML) only for OKTA and Azure AD identity providers (IdPs). JIT supports both user and group attributes. You can also use System for Cross-domain Identity Management (SCIM) APIs with attribute mappings. You must correctly configure these attributes at the IdP and service provider (SP). For the attribute-mapping values for IdP and SP, see SAML attribute mappings of IdP and SP.

When you enable JIT, you can use SCIM operations to search users and groups after the user logs in at least once by using SAML IdP. You can then enable their access to the IBM Cloud Pak console.

To enable the access for users and groups before you enable JIT, see Granting access to capabilities within an IBM Cloud Pak.

Configure SAML at IdP and SP ends:

- Configuring SAML at IdP end
- Configuring SAML at SP end

Configuring SAML at IdP end

If you are using the OKTA template application to configure SAML, see Configure Cloud Pak as SAML SP at OKTA. You can provide the attribute mappings in the **Attribute Statements (optional)** section in the application.

Important: Even though the section name indicates **(optional)**, you must provide at least one SAML attribute. Otherwise, the SAML configuration fails.

Configure the SAML IdP attributes such as Name, Name format, and Value at IdP.

Note: The SAML IdP attributes are the user-defined values. You can update these values based on your requirements.

The following table is an example for SAML IdP attributes:

Table 1. Example of SAML IdP attributes

Name	Name format	Value
uid	Basic	user.email
firstName	Basic	user.firstName
lastName	Basic	user.lastName
emailAddress	Basic	user.email
groups	Basic	Matches regex

Configuring SAML at SP end

To register OKTA as SAML IdP, you can either use the console or use the IdP V3 APIs. For more information, see Registering the SAML clients using IdP V3.

Note: It is not necessary to specify `redirect_url` for SAML with OKTA or Azure AD types when JIT is enabled.

IAM supports the following SAML SP attributes for user and group entities:

Table 2. IAM-supported SAML SP attributes

Name	Description
sub	Unique ID
given_name	First name of the user
family_name	Family name of the user
groups	Group name of the user
email	Email ID of the user

The values of the SAML SP attributes must be the same as the user-defined values for SAML IdP attributes.

The following table includes the mapping of the SAML attributes of IdP and SP:

Table 3. SAML attribute mappings of IdP and SP

SAML SP attribute	SAML IdP attribute	Description
sub	uid	Unique ID
given_name	firstName	First name of the user
family_name	lastName	Family name of the user
groups	groups	Group name of the user
email	emailAddress	Email ID of the user

Note: Table 3. SAML attribute mappings of IdP and SP includes the attribute mapping based on Example of SAML IdP attributes and IAM-supported SAML SP attributes tables.

See the following prerequisites:

1. Download the IdP metadata-xml file from the OKTA-SAML integration application. For more information, see IBM Cloud Pak SCIM Okta integration. You can save the file as `idp-metadata-xml.xml`.
2. Encode the IdP metadata-xml file content as base64. See the following example:

   ```
   cat idp-metadata-xml.xml | base64 --decode
   ```

   Following is a sample payload of IdP V3 registration for OKTA with JIT enabled. The example payload is based on attributes from Example of SAML IdP attributes:

   ```
   {
       "name": "OKTASAML",
       "description": "OKTASAML",
       "protocol": "saml",
       "type": "okta",
       "idp_config": {
           "token_attribute_mappings": {
   ```

```
            "sub": "uid",
            "given_name": "firstName",
            "family_name": "lastName",
            "groups": "groups",
            "email": "emailAddress"
        },
        "idp_metadata": "base64 idp metadata"
    },
    "jit": true,
    "scim_config": {
      "scim_attribute_mappings": {
        "user": {
          "principalName": "userName",
          "name": {
            "givenName": "givenName",
            "familyName": "familyName"
          },
          "displayName": "displayName",
          "emails": [
            {
              "value": "emails",
              "type": "home"
            }
          ],
          "id": "id",
          "userName": "userName"
        },
        "group": {
          "principalName": "displayName",
          "id": "displayName"
        }
      }
    }
  }
```

3. Ensure that the values of **`token_attribute_mappings`** such as sub, given_name, family_name, groups, and email are same as the SAML IdP attributes that are configured at **Attribute Statements (optional)** section in OKTA IdP.

   Note: It is mandatory to specify the **`principalName`** for user and groups.

4. The **`scim_attribute_mappings`** is an optional attribute. If you specify the values for **`scim_config.scim_attribute_mappings`**, ensure that the values are the same as the matching attributes that are configured at OKTA IdP.

   If you do not specify the values for **`scim_config.scim_attribute_mappings`**, IAM defaults to the following SCIM attribute mappings for users and groups:

   - SCIM attribute mappings for user:

     ```
     {
        "id": "sub",
        "userName": "sub",
        "principalName": "sub",
        "displayName": "firstName",
        "emails": [
            {
            "value": "email",
            "type": "primary"
            }],
        "familyName": "lastName",
        "givenName": "firstName",
        "externalId": "sub"
     }
     ```

   - SCIM attribute mappings for groups:

     ```
     {
        "id" : "_id",
        "displayName" : "_id",
        "groupName" : "_id",
        "principalName" : "_id",
     }
     ```

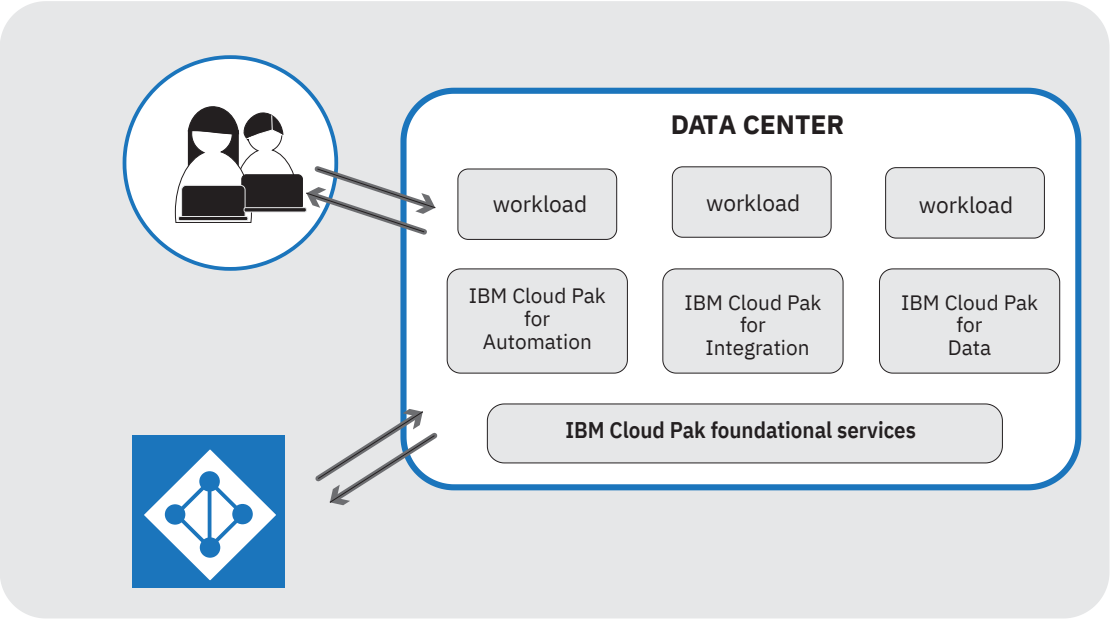
     Restriction: Ensure that you manually remove the user or group from your IBM Cloud Pak when you remove the user or group from OKTA.

Integrating IBM Cloud Pak SAML with Azure Active Directory

Learn how to integrate the IBM Cloud Pak® SAML with Azure Active Directory.

IBM Cloud Pak foundational services provides single sign-on (SSO) to IBM Cloud Paks and to workloads that run on the IBM Cloud Paks.

The SAML authentication flow supports automated onboarding by just-in-time (JIT) provisioning of users. You can configure SSO with Azure Active Directory, and grant user access to Cloud Pak Platform based on user's group membership from Azure Active Directory. After you complete the configuration, users who belong to an Azure Active Directory group are granted access to the IBM Cloud Pak.



To integrate the SAML with Azure Active Directory, complete the following steps:

- Prerequisites
- Configuring SSO with Azure Active Directory
- Configuring Azure as SAML IdP at IBM Cloud Pak®
- Configuring JIT support for Azure Active Directory integration

Prerequisites

To complete the configuration, you must meet the following requirements:

- Subscribe to Microsoft Azure.
- Be able to run `oc` or `kubectl` commands with admin access to the Red Hat® OpenShift® Container Platform cluster.
- Install foundational services with your IBM Cloud Pak.
- Grant a user administrator access to your IBM Cloud Pak.

Configuring SSO with Azure Active Directory

Complete the steps in the following sections to configure SSO with Azure Active Directory.

Retrieve the SAML metadata file from your IBM Cloud Pak

Complete these steps to download the SAML metadata file:

1. Retrieve your IBM Cloud Pak login URL by checking the foundational services route. For example, by using the following command:

   ```
   oc get route -n <your-foundational-services-namespace>
   ```

2. Log in to your IBM Cloud Pak console as an admin user.
3. Click **Identity providers** > **New connection**.
4. Select **SAML 2.0**.
5. On the **New SAML connection** page, click **Download metadata**.

Configure your Azure Active Directory application with SAML SSO

Complete the following tasks in the Microsoft Azure portal.

Create your own Azure Active Directory application. Add users and groups to your application. Set up SSO

Create your own Azure Active Directory application

1. Log in to the Microsoft Azure portal.
2. From the Azure menu, select **Azure Active Directory**.
3. From the navigation menu, select **Enterprise applications**.
4. Create you application.
   1. Click **Create your own application**.
   2. Enter the name of your application.
   3. Select the **Integrate any other application you don't find in the gallery (Non-gallery)** option.
   4. Click **Create** and wait a few seconds. The Overview page of your application opens.

Add users and groups to your application

1. On the Overview page, select the **1. Assign users and groups** tile from the Getting started section.
2. Click **Add user/group**.
3. On the Add assignment page, under Users and groups, click **None selected**. The list of users and groups opens in a **Users and groups** window.
4. Select the users and groups that you want to add to your application from the list, and click **Select**.
5. On the Add assignment page, click **Assign**.
6. Go back to your application overview.

Set up SSO

1. On the Overview page, select the **2. Set up single sign on** tile from the Getting started section.
2. Choose **SAML**.
3. Provide basic SAML configuration.
    1. On the Set up Single Sing-On with SAML page, click **Upload metadata file**.
    2. Select the SAML metadata file that you downloaded in the Retrieve the SAML metadata file section.
    3. Click **Add**.
       After you upload the metadata file, the Basic SAML configuration window opens. The following fields are automatically filled from the metadata file: Identifier, Reply URL, and Logout Url.
    4. Click **Save**, and go back to the **Set up Single Sing-On with SAML** page.
4. Configure group claim.
    1. On the Single Sing-On with SAML page, under **Attributes and Claims**, click **Edit**.
    2. Click **Add a group claim**. For more information, see Configure group claims for applications with Azure Active Directory in Microsoft documentation.
       Note: If an existing application expects to consume group information by using claims, Azure Active Directory can be configured with different claim formats.
    3. Select **Groups assigned to the application** as the groups that should be returned in the claim, and click **Save**.
5. Configure the SAML signing certificate. On the Single Sing-On with SAML page, under SAML Signing Certificate, click **Download** to save the **Federation Metadata XML** to your local drive.

Upload the Federation Metadata XML configuration to your IBM Cloud Pak

Complete these steps from your IBM Cloud Pak console:

1. Log in to your console as an admin user.
2. Click **Identity providers**.
3. Click the SAML connection that you created in Retrieve the SAML metadata file from your IBM Cloud Pak section.
4. Click **Token attribute mapping**.
5. Update the token attribute-mapping fields from the Azure Active Directory Federation Metadata XML file that you downloaded in the Set up SSO section.
6. Click **Create**.
7. From the **Workloads** page, restart all pods with `platform-` and `usermgmt-` in their name.

Test SAML login from the IBM Cloud Pak

1. To get your IBM Cloud Pak console URL, run the following command:

   ```
   oc get route -n <your-foundational-services-namespace>
   ```

2. Copy the URL and paste it into your browser.
3. On the login screen, select **Enterprise SAML**. Use your Microsoft email to log in, for example, user@cloudpak.onmicrosoft.com. You can log in. However, your view of the Administration panel is limited. When you click at your profile avatar, you see that you have a viewer access role.

Grant an admin role

Use the Zen console to assign the admin role to users.

Grant access to capabilities within an IBM Cloud Pak

You can customize access and grant users access to capabilities within the IBM Cloud Pak. Each capability can have a granular access control policy to control who can access different resources or APIs.

You can grant access to Cloud Pak Platform to users. By default, Cloud Pak Platform comes with two roles:

- User - Basic role that gives minimal access to the platform.
- Administrator - Admin role that gives access to the entire Cloud Pak Platform.

The user needs a User role to access the IBM Cloud Pak capabilities.

Complete the following steps to grant a User role for Cloud Pak Platform. The procedure is based on APIs.

1. Set the project to the namespace where foundational services pods are running. Use the namespace where `platform-auth-service-xxxx` pod is running.

   ```
   oc project <your-foundational-services-namespace>
   ```

2. Export the `CPHOST`, `ZENHOST`, `USERNAME`, and `PASSWORD` environment variables.

```
export CPHOST=https://$(oc get route cp-console -o jsonpath="{.spec.host}")
export ZENHOST=https://$(oc get route cpd -o jsonpath="{.spec.host}")
export USERNAME=`oc get secret platform-auth-idp-credentials -o json | jq -r .data.admin_username| base64 -d`
export PASSWORD=`oc get secret platform-auth-idp-credentials -o json | jq -r .data.admin_password| base64 -d`
```

3. Get the IBM Cloud Pak access token for the admin user with the following command:

```
IAM_TOKEN=$(curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -d
"grant_type=password&username=${USERNAME}&password=${PASSWORD}&scope=openid"
"${CPHOST}/idprovider/v1/auth/identitytoken" 2> /dev/null | jq -r ".access_token")
```

4. Extract and set the **$ZENTOKEN** value from the **$IAMTOKEN** value that you got from the previous step.

```
ZENTOKEN=$(curl -k -X GET "${ZENHOST}/v1/preauth/validateAuth" --header "username: ${USERNAME}" --header "iam-token:
${IAM_TOKEN}" 2> /dev/null | jq -r '.accessToken')
```

5. Create a user group with the role that you want to assign to the user.

```
curl -k -s -X POST "${ZENHOST}/usermgmt/v2/groups" --header 'Content-Type: application/json' --header 'Accept:
application/json' --header "Authorization: Bearer ${ZENTOKEN}" -d '{"name": "azuregroup","description": "Group
default Azure group","role_identifiers": ["zen_user_role"]}'
```

Note: To create an admin group, specify **"zen_administrator_role"** in the role identifiers field.

Verify that the group is listed, and get the group ID:

```
curl -k -s -X GET "${ZENHOST}/usermgmt/v2/groups" --header 'Content-Type: application/json' --header 'Accept:
application/json' --header "Authorization: Bearer ${ZENTOKEN}" | jq -r
```

Save the Azure group ID, you will need it in the next step.

Note: Azure SAML response contains the IDs or names of all groups to which a user belongs. You can check the Azure group ID or group name in the Microsoft Azure portal. Go to the **Groups** page, and find the **Object Id** or group name of the specific group.

6. Add the Azure group to the **LDAP**, **SAML**, or **OIDC** user group. You can specify the group ID or group name of the **LDAP**, **SAML**, or **OIDC** user group in the **"idp_groups"** parameter.

```
curl -k -s -X POST "${ZENHOST}/usermgmt/v2/groups/<azuregroup_ID>/members" --header 'Content-Type: application/json'
--header 'Accept: application/json' --header "Authorization: Bearer ${ZENTOKEN}" -d '{"idp_groups": ["group ID or
group name of the user group"]}'
```

In the command, the **<azuregroup_ID>** is the Azure group ID that you got in the previous step.

Configuration is completed.

7. Verify whether the user who is a member of the Azure group can log in to Cloud Pak Platform with proper permissions. Open Cloud Pak Platform in your browser by using the URL of $CPHOST.

Configuring Azure as SAML IdP at CloudPak

To configure azure as SAML IdP, see Configure Azure as SAML IdP at CloudPak.

Configuring JIT support for Azure Active Directory integration

To configure JIT support for Azure Active Directory integration, see IBM Cloud Pak JIT support for SCIM OKTA and Azure AD.

IM common service EDB PostgreSQL database backup and restore

EDB PostgreSQL is a database that is used with the Identity Management (IM) service.

Learn how to back up, restore, and troubleshoot EDB PostgreSQL database.

- Backing up EDB PostgreSQL database
- Restoring EDB PostgreSQL database
- Backing up and restoring the common services EDB PostgreSQL database using pg_dump

Backing up common services EDB PostgreSQL database

Set the EDB backup deployment to store the Identity Management (IM) data in the **common-service-db** backup resources.

Prerequisites

- Download the CASE package and extract the **installer-scripts** directory. For more information, see Downloading scripts for additional configuration from specific version CASE bundle.
- Set up any Amazon S3-compatible storage. For example, you can create a bucket in IBM Cloud Object Storage. For more information, see IBM Cloud Object Storage.

- When you add a service credential to the bucket, include the hash-based message authentication code (HMAC). For more information, see Service credentials. From the **Cloud Object Storage** navigation menu, gather the following information:
  - **`access key id`**, which can be found on the **Service credentials** page that is associated with the bucket.
  - **`secret access key`**, which can be found on the **Service credentials** page that is associated with the bucket.
  - **`bucket name`**, which can be found on the **Buckets** page.
  - **`bucket region`**, which can be found on the **Buckets** page.
  - **`root directory name`**, which is the path in the bucket where you want to store the backups.
  - **`s3 URL`**, which is the endpoint URL of the bucket.

    Note: The endpoint URL must start with **`http://`** or **`https://`**.
- Set up Velero on the new cluster. For more information, see Create the backup resources section. If the cluster being backed up or restored to uses s390x for architecture, any velero CLI commands must be run on an alternate cluster that does not use s390x and has **`oc`** access to the original (usually using **`oc login`**). Velero CLI does not yet support s390x.
- If the Velero or OADP is not configured in your cluster, you can backup the EDB PostgreSQL database using **`pg_dump`** CLI method. For more information, see Back up EDB database using **`pg_dump`** CLI method.

Back up EDB database (common-service-db)

Complete the following steps to back up EDB database for IM:

1. Get the common-service-db backup resources.

```
wget https://raw.githubusercontent.com/IBM/ibm-common-service-operator/scripts-adopter/velero/schedule/common-service-db/cs-db-backup-deployment.yaml
wget https://raw.githubusercontent.com/IBM/ibm-common-service-operator/scripts-adopter/velero/schedule/common-service-db/cs-db-backup-pvc.yaml
wget https://raw.githubusercontent.com/IBM/ibm-common-service-operator/scripts-adopter/velero/schedule/common-service-db/cs-db-br-script-cm.yaml
wget https://raw.githubusercontent.com/IBM/ibm-common-service-operator/scripts-adopter/velero/schedule/common-service-db/cs-db-role.yaml
wget https://raw.githubusercontent.com/IBM/ibm-common-service-operator/scripts-adopter/velero/schedule/common-service-db/cs-db-rolebinding.yaml
wget https://raw.githubusercontent.com/IBM/ibm-common-service-operator/scripts-adopter/velero/schedule/common-service-db/cs-db-sa.yaml
```

2. Update the backup files.
   - Replace **`<cs-db namespace>`** with the namespace where **`common-service-db`** instance is running.
   - Replace the **`<storage class>`** with the storage class that the current IM deployment uses.
3. Add the PVC to the cluster.

```
oc apply -f cs-db-backup-pvc.yaml
```

4. Add the **`cs-db-br-script-cm.yaml`** to the correct namespace

```
oc apply -f cs-db-br-script-cm.yaml
```

5. Give the common-service-db backup necessary permissions

```
oc apply -f cs-db-sa.yaml
```

```
oc apply -f cs-db-role.yaml
```

```
oc apply -f cs-db-rolebinding.yaml
```

6. Add the deployment to the cluster.

```
oc apply -f cs-db-backup-deployment.yaml
```

Restoring EDB PostgreSQL database

Deploy the EDB PostgreSQL database backup deployment to restore the Identity Management (IM) data from the **`common-service-db`** backup resources.

Prerequisites

- Download the CASE package and extract the **`installer-scripts`** directory. For more information, see Downloading scripts for additional configuration from specific version CASE bundle.
- Set up any Amazon S3-compatible storage. For example, you can create a bucket in IBM Cloud Object Storage. For more information, see IBM Cloud Object Storage.
- When you add a service credential to the bucket, include the hash-based message authentication code (HMAC). For more information, see Service credentials. From the **Cloud Object Storage** navigation menu, gather the following information:

- `access key id`, which can be found on the **Service credentials** page that is associated with the bucket.
- `secret access key`, which can be found on the **Service credentials** page that is associated with the bucket.
- `bucket name`, which can be found on the **Buckets** page.
- `bucket region`, which can be found on the **Buckets** page.
- `root directory name`, which is the path in the bucket where you want to store the backups.
- `s3 URL`, which is the endpoint URL of the bucket.

  Note: The endpoint URL must start with `http://` or `https://`.

- Set up Velero on the new cluster. For more information, see Create the backup resources section.

If the cluster being backed up or restored to uses s390x for architecture, any velero CLI commands must be run on an alternate cluster that does not use s390x and has `oc` access to the original (usually using `oc login`). Velero CLI does not yet support s390x.

- If the Velero or OADP is not configured in your cluster, you can restore the EDB PostgreSQL database using `pg_restore` CLI method. For more information, see Restore EDB database using `pg_restore` CLI method.

Restore EDB PostgreSQL database (common-service-db)

Complete the following steps to restore IM data from `common-service-db` backup resources.

1. Get the restore object.

   ```
   wget https://raw.githubusercontent.com/IBM/ibm-common-service-operator/scripts-adopter/velero/restore/restore-cs-db.yaml
   ```

2. Substitute the `__BACKUP_NAME__` with the name of the backup resource that you created previously.

   ```
   vi restore-cs-db.yaml
   ```

3. Restore the `cs-db` data.

   ```
   oc apply -f restore-cs-db.yaml
   ```

4. Check restore progress. Proceed with the next step after restore process is complete.

   ```
   velero restore get
   ```

5. Check logs of the velero restore to verify that the data was restored.

   ```
   velero restore logs restore-zen5-data
   ```

Troubleshooting restoring EDB PostgreSQL database

If the logs or the data indicate that the restore was not successful, complete the following steps:

1. Get the restore job:

   ```
   wget https://raw.githubusercontent.com/IBM/ibm-common-service-operator/scripts-adopter/velero/restore/common-service-db/cs-db-restore-job.yaml
   ```

2. Replace `<cs-db namespace>` with the namespace where common-service-db instance is running.

3. Delete the existing cs-db-backup deployment and cs-db-backup pod.

   ```
   oc delete deploy cs-db-backup -n <namespace>
   ```

4. Run the restore job.

   ```
   oc apply -f cs-db-restore-job.yaml
   ```

Adding users for console access

When you add a user to the platform for accessing the IBM Cloud Pak for AIOps console and tools, a user profile (or record) is created for the user.

When a user is given access to the platform, a profile is created for the user. This profile lists the user's roles and permissions. The user can also now access the Cloud Pak for AIOps console.

All users that are added to the platform are also automatically included within a default *All users* group. This group is used to give all users access to the Cloud Pak for AIOps console and common features, such as their profile and settings. You cannot edit or delete this group.

You can add a user to the platform in the following ways:

- You can give a user from a connected LDAP service access by directly assigning that user a role.
- You can include a user from a connected LDAP service into a user group. When you add a user to a user group, the platform automatically assigns the user each role that is assigned to the group.

- You can give all members of an LDAP group access to the platform by adding that LDAP group to a user group.

Important:

- You cannot create a user from the **Access control** page of your cluster Cloud Pak for AIOps console. You can use the **Access control** page to add only an existing LDAP user to your cluster.
- You must use the configured identity provider (LDAP) to manage users that are added from that provider system. For instance, to change or reset the password for a user, you must complete that change in your configured provider system. You cannot change or reset a user password with the Cloud Pak for AIOps console.

  To see your configured identity providers, navigate to **Administration** > **Identity providers** in the Cloud Pak for AIOps console.

Before you begin

- **Required permissions**: To manage access to the Cloud Pak for AIOps console, you must have one of the following permissions:
  - **Administer platform**
  - **Manage users**
- Connect to an LDAP service. You need to connect to an LDAP service as an identity provider to create the users that you want to add IBM Cloud Pak for AIOps.
  - To configure an LDAP connection, you must be an Administrator with permissions to manage users. For more information, see Configuring an LDAP connection.
  - Any user that needs access to IBM Cloud Pak® for AIOps must exist in your LDAP service with their **`user name`** and **`email`** attributes set. If these attributes are not set for a user, you might not be able to find and add the user to IBM Cloud Pak® for AIOps.
- **Required information for creating a user**: You need at least one of the following types of information about a user from your connected LDAP service to add the user to IBM Cloud Pak for AIOps:
  - Full name: If you use this information when adding a user, you must enter the name exactly as it is specified in the LDAP service.
  - Username: This value maps to the field that you specified for the **User search field** in the LDAP configuration
  - Email address: If you use this information when adding a user, enter the email address exactly as it is specified in the LDAP service.
- **User groups**: You can create user groups to simplify the process of managing large groups of users. User groups make it easier to manage a large number of users with similar access requirements.

  By default, an All users group is included. As the name suggests, all users are automatically included in this group. The group is used to give all platform users access to the console and common features, such as their profile and settings. You cannot edit or delete this group.

  For more information, see Managing user groups.

Procedure

To give users access to the console:

1. Log in to the IBM Cloud Pak for AIOps console as an administrator with permissions to manage users.
2. From the navigation menu, select **Administration** > **Access control**.
3. From the *Users* tab, click **Add user**.
4. Search for the user's profile information by entering one of the following details about the user:
   - Full name
   - Username
   - Email

   Limitations:
   - You cannot select a specific LDAP service to use for adding users. All connected LDAP services are searched for the specified user ID, username, or email.
   - You cannot use an external lookup service for finding and adding users from the LDAP service to IBM Cloud Pak for AIOps.
   - You need to know the user ID, user name, or email of a user within the connected LDAP service to find and add that user.
5. Click **Next**.
6. Set the platform access permissions for the user. Privileges on the platform are controlled by permissions. Users are granted permissions through role assignment. Users can be assigned roles directly or inherit roles from user groups that they are included within. Users within a user group inherit all roles and permissions that associated with the user group.

   Select either of the options to set permissions for the user:

   - **Assign roles directly**

     Select this option to directly assign one or more existing roles or create a role for the user.

     - Optional. Create a new role to assign to the user. For more information about existing roles and permissions, or about creating a role, see Roles and permissions.
     - Select the checkbox for each role that you want to assign to the user. If you created a new role for the user, you must still select the checkbox for that role.

- Click **Next**.

- **Add to user group**

  Select this option to add a user to an existing user group or groups. For more information about user groups, including how to create a group, see Managing user groups.

  - Select the checkbox for each user group or groups that you want to include the user within.
  - Click **Next**.

7. Review the *Summary* of the user profile, user groups, roles, and access. When finished, click **Add**.

What to do next

- Edit the user details and roles, such as to add more roles or to add the user to a user group. When you are editing a user, you can complete the following changes:

  - View, add, and remove roles
  - View, add, and remove permissions
  - Add the user to a user group
  - Remove the user from a group
  - Change the user full name
  - Change the user email
  - Delete the user from IBM Cloud Pak for AIOps

  You can also add a user group, such as to add and manage roles for all users that you include in the group.

- Login as the newly added user

  As a user directly added from the LDAP service or from an added user group, log in to the console and verify that the expected roles and permissions exist.

  Upon login, either your *Home* page or the *Administrator panel* is displayed. Users with an administrator role view the *Administrator panel* upon login, while all other users view their *Home* page. From the main navigation menu, browse through your available tools and pages to ensure that you can view the correct data and complete the tasks for your assigned permissions.

Managing user groups

You can create user groups to simplify the process of managing large groups of users.

User groups make it easier to manage a large number of users with similar access requirements. For example, if you know that 20 different users are going to collaborate on a project and they all need the same role, you can add them to a group that is assigned that role. If a member of the group leaves the company, you can remove the user from the group, rather than looking for all of the assets that the user has access to.

Before you begin

- **Required permissions**: To manage groups, you must have one of the following permissions:

  - **Administer platform**
  - **Manage groups**

About this task

You can create and edit groups from the **User groups** tab of the **Access control** page.

Important:

- You can either create assigned groups or dynamic groups

  - If you select the assigned option, you can select the users you want to include.
  - If you select the dynamic option, you can create attribute-based rules to determine which users are automatically added to the group.
- By default, an **All users** group is included. As the name suggests, all users are automatically included in this group. The group is used to give all platform users access to the console. You cannot edit or delete this group.

Procedure

You can create an user group with one of the following methods:

- Creating an user group with the **Assigned** option
- Creating an user group with the **Dynamic** option

Creating an user group with the Assigned option

1. Log in to the console.
2. From the navigation menu, select **Administration** > **Access control**.

3. Open the **User groups** tab.
4. Click **New user group**.
5. Enter a name and, optionally, a description for the role.
6. Under Membership type, select **Assigned**.

   Note: With assigned membership, you select the users that you want to add to the group.
7. Click **Next**.
8. Specify the users to include in the group. The available options depend on whether the environment is connected to use an identity provider for authentication, such as an LDAP server.

   - **Existing users** - Find and add existing platform users.
   - **Identity provider users** - Find and add a user from a connected identity provider, such as an LDAP server.
   - **Identity provider groups** - Find and add a user group from a connected identity provider, such as an LDAP server. If IM integration is enabled, choose this option.

   After you find the user or group, select the user or group to add them to the new user group. Repeat this action to add more users or groups.

   Notes:

   - When searching for a user, you can search by the user's name, email, or username.
   - If you have the **Manage users** permission and you do not see the user that you want to add to the group, first create the user.
   - If you add users from an LDAP group, the users might not immediately be added to the user group. If you do not see a user within the group, log in to the console as that user. Upon login, the Platform UI detects that the user belongs to an LDAP group and creates the user profile. User profiles for LDAP group users are created only when the users log in to the console.
9. Click **Next**.
10. Select one or more roles that you want to assign to this group.
11. Click **Next**.
12. Review the summary. If the values are correct, click **Create**.

Creating an user group with the Dynamic option

1. Log in to the console.
2. From the navigation menu, select **Administration** > **Access control**.
3. Open the **User groups** tab.
4. Click **New user group**.
5. Enter a name for the user group. You can optionally include a description for the role.
6. Select **Dynamic** in the **Membership type** section.

   Note: You can create attribute-based rules to determine which users are automatically added to the group with dynamic membership.
7. Click **Next**.
8. Define the rule in **Membership rule** to add the users automatically in the group based on the attributes that are assigned to the users on the identity provider.

   1. Select one of the following conditions:

      - Select **All conditions (AND)** to include the users with all the specified attribute, operator, and values.
      - Select **Any conditions (OR)** to include the users with a minimum of one specified attribute, operator, and values.
   2. Select the **Attribute** and **Operator** from the drop-down list and enter the value based on the identity provider for the defined conditions in the **Value** field.

      - **Attribute:** Select **Location**, **Nationality**, **Organization**, or **User type**.
      - **Operator:** Select **Equal**, **Not equal**, **Match**, or **Not match**.
      - **Value:** Enter the value based on the identity provider for the defined conditions.

      If you need to add more than one conditions, click **Add conditions**.
9. Click **Next**.
10. Select at least one role to assign to the new user group. For example, you can assign **Administrator** and/or **User** to the user group. Alternatively, you can **create a new role** and assign the new role you created.
11. Click **Next**.
12. Review the summary. If the values are correct, click **Create**.

Results

You can now use the group to give users access to various assets on the platform.

Viewing your user group assignment as a user

You can see what user groups you are included within from your user profile:

1. Log in to the IBM Cloud Pak for AIOps console and click your avatar in the upper right of the toolbar.
2. Click **Profile and settings**.
3. Click the **User groups** tab.

The groups that you are associated with are listed in the table.

Roles and permissions

For accessing and completing actions with IBM Cloud Pak® for AIOps, users require specific roles and permissions. Review the different available roles, permissions, and the associated actions that users can complete with the roles and permissions. These roles and permissions help you get your users set up to begin day to day operations.

- Predefined roles
- Creating a role
- Permissions
- Assigning roles and permissions to users
- Viewing your permissions
- Red Hat OpenShift Container Platform roles

Predefined roles

A role defines the permissions that a user or group has. You can edit the roles that are available by default roles or create roles if the default set of permissions for a role does not align with your business needs.

The following roles are available by default:

- Administrator
- Automation Administrator
- Automation Analyst
- Automation Developer (AIOps Developer role)
- Automation Operator
- Service Administrator
- User

Notes:

- **Service Administrator**: Users with the *Service Administrator* role do not have permission to add or update a role, or view the details of a user's assign role. If a user with the Service Administrator role selects to view details about a role when viewing details about a user group, a 401 error page is instead displayed.
- **Automation Analyst** and **User**: These predefined roles are available by default but do not include any permissions for completing actions within IBM Cloud Pak for AIOps.

  These default roles are used within the IBM® Automation family of offerings, which includes IBM Cloud Pak for AIOps, however, these roles are not used within IBM Cloud Pak for AIOps. If more IBM Cloud Paks are installed, these roles can have permissions for completing actions within tools for those other IBM Cloud Paks.

  As these roles do not include or provide any permissions within IBM Cloud Pak for AIOps, do not assign the roles to users within IBM Cloud Pak for AIOps.
- Not all permissions are associated with a predefined role. A user with with a permission to manage roles, such as the Administer platform permission, can assign permissions to a role when needed.

Default user

The default user (`admin`) is automatically assigned the following roles when the roles are added to the platform:

- Administrator

Creating a role

If the existing default roles are not sufficient for your needs, you can create a role for your users. In addition, when you are adding a user, you can choose to create a role for that user instead of assigning an existing role.

The settings for creating a role are the same regardless of the path you select to create the role.

To directly create a role, complete the following steps:

1. Log in to the console as an administrator with permissions to manage users.

   Required permissions: To manage access to the Cloud Pak for AIOps console, you must have one of the following permissions:

   - **Administer platform**
   - **Manage platform roles**
2. From the main navigation, click **Administration** > **Access control**.
3. Select the **Roles** tab and click **New role**.

4. Enter the details for the new role:
    - Name - Enter a distinctive and descriptive name for the role.
    - Description - Enter an optional description for the role to define the role, such as to identify the objective or responsibilities for users with the role.
5. Click **Next**.
6. Select the permissions for the role. You can use the *Find permissions* search field to find and filter the list of permissions.
7. Click **Next**.
8. Review the details for the role and click **Create**.

Permissions

The following table describes the actions that are associated with each permission.

AI models

Table: AI models permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage AI models	(1)			X	X	X
View AI models status	(2)		X	X	X	X

1. Manage AI models

   Users with this permission can complete the following actions:

   - Gather and prepare AI training data
   - Train AI models
   - Deploy AI models
2. View AI models status

   Users with this permission can complete the following actions:

   - View AI model status

Resource management (Applications)

Table: Resource management permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage applications	(1)		X	X	X	X
Manage resource group templates	(2)			X	X	X
Manage topology comments	(3)	X	X	X	X	X
View topologies	(4)	X	X	X	X	X

1. Manage applications

   Users with this permission can manage applications and related settings. Users with this permission can complete the following actions:

   - View applications and resources
   - Create applications
   - Edit applications
   - Delete applications
2. Manage resource group templates

   Users with this permission can manage templates that define how resources are organized into resource groups. Users with this permission can complete the following actions:

   - View resource group templates
   - Create resource group templates
   - Edit resource group templates
   - Delete resource group templates
3. Manage topology comments

   Users with this permission can manage comments on resources included in a topology. Users with this permission can complete the following actions:

   - View topology comments
   - Create topology comments
4. View topologies

   Users with this permission can view application topologies. Users with this permission can complete the following actions:

   - View topologies

Basic operations

Table: Basic operations permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
View operational data	(1)	X	X	X	X	X
Manage operational data	(2)	X	X	X	X	X
Use AIOps Insights	(3)	X	X	X	X	X

1. View operational data

   Users with this permission have the following read-only access in IBM Cloud Pak® for AIOps:

   - View alerts
   - View incidents
   - View alert filters
   - View incident filters
   - View alert views
   - View incident views
2. Manage operational data

   In addition to the View operational data permissions, users with this permission can perform the following actions:

   - Update alerts (Execute right-click actions)
   - Update incidents (Change priority, change state, assign, etc.)
   - Add/Edit/Delete alerts filters
   - Add/Edit/Delete alerts views
   - Add/Edit/Delete incidents filters
   - Add/Edit/Delete incidents views
   - Add/Edit/Delete right-click actions
3. Use AIOps Insights

   Use the AIOps Insights dashboard to view productivity metrics such as mean time to restore, incident activity, noise reduction, and runbook usage.

   - Use AIOps Insights

Integrations

Table: Integrations permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage integrations	(1)			X	X	X
View integrations	(2)		X	X	X	X

1. Manage integrations

   Users with this permission can manage integrations to infrastructure, log, event, source code, and chat management tools. Users with this permission can complete the following actions:

   - View integrations
   - Create integrations
   - Edit integrations
   - Delete integrations
2. View integrations

   Users with this permission can view integrations to infrastructure, log, event, source code, and chat management tools. Users with this permission can complete the following actions:

   - View integrations

Operational policies

This permission is required for viewing policy-related information that can be displayed within the details for an application incident.

Attention: To access **Automations** in the main navigation menu, the *Use runbooks* permission is required in conjunction with the Operational policy roles. For more information, see Runbook permissions.

Table: Operational policies permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Delete operational policies	(1)	X		X	X	X
Edit operational policies	(2)	X		X	X	X
View operational policies	(3)	X	X	X	X	X

1. Delete operational policies

   Users with this permission can delete policies for promoting alerts to incidents. Users with this permission can complete the following actions:

   - Delete operational policies
2. Edit operational policies

Users with this permission can edit policies for promoting alerts to incidents. Users with this permission can complete the following actions:

- Edit operational policies

3. View operational policies

   Users with this permission can view policies for promoting alerts to incidents. User with this permission can complete the following actions:

   - View operational policies

Platform administration

Permissions in this category enable an administrator to configure, customize, monitor, and manage the platform.

Only the Administer platform permission is associated with a predefined role (Adminstrator). This permission includes all other platform administration permissions.

Table: Platform administration permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Administer platform	(1)					X
Manage configurations	(2)					
Manage platform health	(3)					
View platform health	(4)					

1. Administer platform

   This permission offers the most comprehensive set of actions for managing and monitoring the platform. Users with this permission have elevated privileges and can grant or revoke all permissions, including other administrative permissions. Users with this permission can complete the same actions as users that have the following permissions:

   - Manage configurations (Administer platform)
   - Manage platform health (Administer platform)
   - Manage platform roles (User administration)
   - Manage users (User administration)
   - Manage user groups (User administration)
   - Manage service instances (Service instances)

2. Manage configurations

   Users with this permission can customize the platform, integrate the platform with other applications, and enable connections to unsupported data sources. Users with this permission can access the Customizations page, the Configurations page, and the JDBC drivers tab on the Platform connections page. Some actions require specific services to be installed. Users with this permission can complete the following actions:

   - Configure connection to SMTP server

     Note: Requires Watson Studio or Watson Knowledge Catalog. An SMTP connection enables the platform to send emails.

   - Configure integration with IBM Guardium appliances

     Note: Requires Watson Knowledge Catalog. Use IBM Guardium to audit access to sensitive data on remote databases.

   - Configure connections to Hadoop clusters

     Note: Requires Execution Engine for Apache Hadoop.

   - Customize branding
   - Enable and disable home page cards
   - Enable and disable default support links
   - Add and delete custom support links
   - Enable and disable guided tours
   - Import JDBC drivers

     Note: Requires the common core services. JDBC drivers enable users to connect to unsupported data sources.

3. Manage platform health

   Users with this permission can monitor resource use, set quotas and alerts, manage workloads to maintain the health of the platform, and gather diagnostic data when problems occur. Users with this permission can access the Monitoring page and the Diagnostics page. Users with this permission can complete the following actions:

   - Monitor workloads and resource use

     Note: Workloads include active runtimes and jobs. Resources include memory and vCPU use across services, instances, and environments.

   - Stop any runtime environment

     Note: Environments are hardware and software configurations defined for running analytical assets or jobs.

- View pod status, details, and logs

  Note: Services are composed of Kubernetes pods. A pod is an instance of a process that runs on the cluster.

- Restart pods
- View platform quotas and service quotas

  Note: A quota specifies the maximum amount of memory or vCPU that the platform or a service should use.

- View event history and alerts
- Set and edit platform resource quotas
- Set and edit individual service resource quotas
- Create and run diagnostics jobs

  Note: A diagnostic job gathers information that can be used for troubleshooting problems.

- Delete diagnostics jobs

4. View platform health

   Users with this permission can monitor resource use and workloads across the platform to gauge the health of the platform. Users with this permission have read-only access to the Monitoring page. Users with this permission can complete the following actions:

   - Monitor workloads and resource use

     Note: Workloads include active runtimes and jobs. Resources include memory and vCPU use across services, instances, and environments.

   - View pod status, details, and logs

     Note: Services are composed of Kubernetes pods. A pod is an instance of a process that runs on the cluster.

   - View platform quotas and service quotas

     Note: A quota specifies the maximum amount of memory or vCPU that the platform or a service should use.

   - View event history and alerts

Profiles

Table: Profiles permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage profiles	(1)					X

1. Manage profiles.

   Users with this permission can complete the following actions using filtering and redacting:

   - Limit the access a user/group has to see alerts
   - Limit the access a user/group has to see incidents
   - Limit the access a user/group has to see topology/inventory

Runbooks

Table: Runbooks permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Administer runbooks and runbook application	(1)			X	X	X
Author and manage runbooks	(2)			X	X	X
Author runbooks	(3)	X		X	X	X
Use runbooks	(4)	X	X	X	X	X
View runbooks	(5)	X	X	X	X	X

1. Administer runbooks and runbook application

   Users with this permission can complete the following actions:

   - View draft and published runbooks
   - Run draft and published runbooks
   - Create draft runbooks
   - Edit draft runbooks
   - Approve draft runbooks
   - Delete draft and published runbooks
   - Edit integrations
   - Edit global settings
   - Edit group ownership

2. Author and manage runbooks

Users with this permission can complete the following actions:

- View draft and published runbooks
- Run draft and published runbooks
- Create draft runbooks
- Edit draft runbooks
- Approve draft runbooks
- Delete draft and published runbooks

3. Author runbooks

   Users with this permission can complete the following actions:

   - View draft and published runbooks
   - Run draft and published runbooks
   - Create draft runbooks
   - Edit draft runbooks

4. Use runbooks

   Users with this permission can complete the following actions:

   - View published runbooks
   - Run published runbooks

5. View runbooks

   Users with this permission can complete the following actions:

   - View published runbooks

Secure tunnel

Table: Secure tunnel permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Install tunnel connectors	(1)			X	X	X
Manage tunnel applications	(2)			X	X	X
Manage tunnel connections	(3)			X	X	X
Manage tunnel templates	(4)			X	X	X
Renew certificates	(5)			X	X	X
View tunnel audit logs	(6)			X	X	X

1. Install tunnel connectors

   Users with this permission can install tunnel connectors on remote clusters or hosts. Users with this permission can complete the following actions:

   - Install tunnel connectors

2. Manage tunnel applications

   Users with this permission can manage application traffic at the address or port level. Users with this permission can complete the following actions:

   - View application mappings
   - Create application mappings
   - Edit application mappings
   - Delete application mappings

3. Manage tunnel connections

   Users with this permission can manage connections to remote clusters or hosts. Users with this permission can complete the following actions:

   - View tunnel connections
   - Create tunnel connections
   - Edit tunnel connections
   - Delete tunnel connections

4. Manage tunnel templates

   Users with this permission can manage templates that simplify the creation of tunnel connections. Users with this permission can complete the following actions:

   - View tunnel templates
   - Create tunnel templates
   - Edit tunnel templates
   - Delete tunnel templates

5. Renew certificates

   Users with this permission can renew certificates for tunnel connections. Users with this permission can complete the following actions:

   - Renew certificates for tunnel connections.

6. View tunnel audit logs

Users with this permission can view a history of user-driven operations and tunnel connection events. Users with this permission can complete the following actions:

- View tunnel audit logs

Service instances

The service instance permissions are included through the underlying IBM Cloud Pak foundational services that is installed with IBM Cloud Pak for AIOps. These permissions are not used within IBM Cloud Pak® for AIOps:

- Create service instances
- Manage service instances

If more IBM Cloud Paks are installed, these permissions can be required for completing actions within tools for those other IBM Cloud Paks.

As these permissions are not used within IBM Cloud Pak for AIOps, do not assign the permissions to any roles that you create unless they are required for use with another installed IBM Cloud Pak.

Topology administration

Table: Topology administration permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage advanced topology settings	(1)			X	X	X
Manage topology presentation	(2)			X	X	X
Manage topology rules	(3)			X	X	X
Manage topology tools	(4)			X	X	X

1. Manage advanced topology settings

   Users with this permission can define global preferences for creating and using topologies. Users with this permission can complete the following actions:

   - View advanced topology settings
   - Edit advanced topology settings
2. Manage topology presentation

   Users with this permission can define how topologies are presented to end users. Users with this permission can complete the following actions:

   - View topology presentation settings
   - Create topology presentation settings
   - Edit topology presentation settings
   - Delete topology presentation settings
3. Manage topology rules

   Users with this permission can define how resources are processed to enable event correlation, tagging of resources, and the creation of end-to-end topologies. Users with this permission can complete the following actions:

   - View topology rules
   - Create topology rules
   - Edit topology rules
   - Delete topology rules
4. Manage topology tools

   Users with this permission can define custom actions for resources that are included within topologies. Users with this permission can complete the following actions:

   - View topology tools
   - Create topology tools
   - Edit topology tools
   - Delete topology tools

User administration

User administration permissions enable an administrator to manage users, groups, and roles. These permissions apply to the platform. Service instances and workspaces such as projects, catalogs, and deployment spaces have their own access controls.

Table: User administration permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage platform roles	(1)					
Manage user groups	(2)			X		
Manage users	(3)			X		

1. Manage platform roles

   Users with the *Manage platform roles* permission can access the Roles tab on the Access control page to modify platform roles or create custom roles. Roles determine the permissions that a user or user group has. This permission does not apply to service instances or assets, such as projects, catalogs, and deployment spaces. Users with this permission can complete the following actions:

   - Create platform roles

- Edit platform roles
- Delete platform roles

2. Manage user groups

   Users with the *Manager user groups* permission can access the User groups tab on the Access control page to manage user groups. User groups make it easier to manage the roles (and permissions) of users with similar access requirements. Users with this permission can complete the following actions:

   - Create user groups
   - Edit user groups
   - Delete user groups
   - Assign roles to user groups
   - Remove roles from user groups

3. Manage users

   Users with the *Manager users* permission can access the Users tab on the Access control page, and can onboard users to the platform. Users with this permission can complete the following actions:

   - Add users
   - Edit user profiles
   - Assign roles to users
   - Remove roles from users
   - Remove users

What permissions do I have?

You can see what permissions you have from your profile. Your permissions are determined by the roles that are assigned to you.

To see what roles are assigned to you:

1. Log in to the IBM Cloud Pak for AIOps console and click your avatar in the upper right of the toolbar.
2. Click **Profile and settings**.
3. Click the **Roles** tab.

The permissions that are associated with your role (or roles) are listed in the **Enabled permissions** column.

Adding users

To add users, you need to have an external LDAP server set up for creating user profiles and authentication. When a user record is included within the LDAP server, you can add the user account within the IBM Cloud Pak for AIOps console. You can add a user directly or as part of an LDAP group. For more information, see Adding users.

Assigning roles and permissions

When you add a user or group, you must specify the role and permissions that they have. These roles and permissions can be granted to users through a combination of the Cloud Pak for AIOps console and an external LDAP server for managing user authentication.

Red Hat OpenShift Container Platform roles

To add and manage users and roles for accessing and using the Red Hat OpenShift Container Platform web console, you need to use the Red Hat role-based access control (RBAC). For more information, see the Red Hat OpenShift topic Using RBAC to define and apply permissions.

Profiles

Profiles can be assigned to user groups to control the visibility of incidents, alerts, applications, resource groups, and resources to those users. These profiles can combine restriction filters with redaction policies to form the final redaction criteria that is applied to a user group.

A profile can be a *Group* profile or a *Base* profile. By default, there is always a **Base** profile which applies to all users. You can use this profile to assign restriction filters and redaction policies to all users. You can use one or more **Group** profiles to assign restriction filters and redaction policies to specific groups of users. *Group* profiles can augment or override the settings in the *Base* profile. The Base profile should be configured to show the minimum set of entities or properties that users in the system can see, so that the filter and policies should be aiming to remove everything from the views, while the group profiles aim to add things back into the views. A profile can be enabled, even when the underlying filters and policies are disabled.

Restriction filters control the visibility of incidents, alerts, applications, resource groups, and resources. These filters are assigned to users through profiles. You can use restriction filters to control how information is distributed to your users. Restriction filtering works by enabling you to specify that under certain situations, users should only see particular alerts, incidents, applications, resource groups or resources based on particular conditions.

Notes:

- Restriction filters do not hide resources in topology graph visualisations, so a combination of filter and redaction is required to make sure that no data is exposed.
- If no filter or policy shows it might be due to none being available for that data type.

To create and apply a restriction filter, you need to create a filter from either the *Incidents*, *Alerts*, or *Resource management* (application, resource, resource group) page and save the filter as a restriction filter. After creating a restriction filter, you can then include it in the Base profile or an appropriate group profile. For more

information, see Creating restriction filters.

Redaction policies work alongside filtering and allow you to specify the visibility and content of the properties of incidents, alerts, applications, resource groups, and resources, and are assigned to users through profiles. The policies are defined using Rego, a programming language created specifically for redaction policies, which allows you to manipulate the end result output either by hiding or showing all the properties or reacting or modifying particular properties. For more information, see Creating redaction policies.

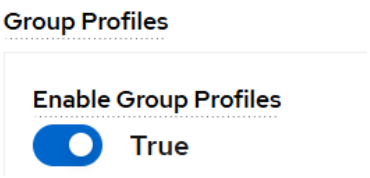
If you need to create or manage a user group before applying any profile, see Managing user groups.

- Prerequisites
- Creating a group profile
- Viewing a profile
- Enabling or disabling a profile
- Editing a profile
- Duplicating a profile
- Deleting a profile

Prerequisites

To work with profiles, and their associated restriction filters and redaction policies, you must possess an administrative user account with the **Manage profiles** role. For more information, see Roles and permissions.

Note: In the installation resource you are using you need to set the *Enable Group Profiles* radio button to *True*.



Creating a group profile

1. From the main navigation menu, go to **Administration** > **Profiles**.

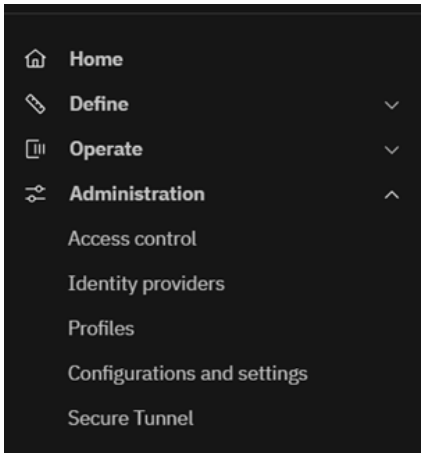


Figure. Profiles list

By default, the **Profiles** tab for the **Profiles** page is displayed.

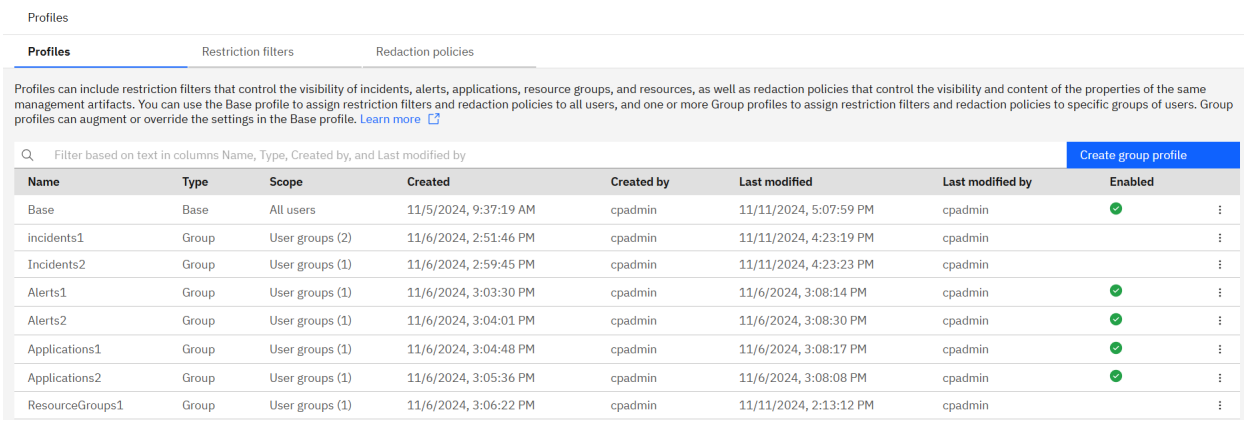


Figure. Profiles list

2. Click **Create group profile**. The options for creating a **New profile** are displayed.

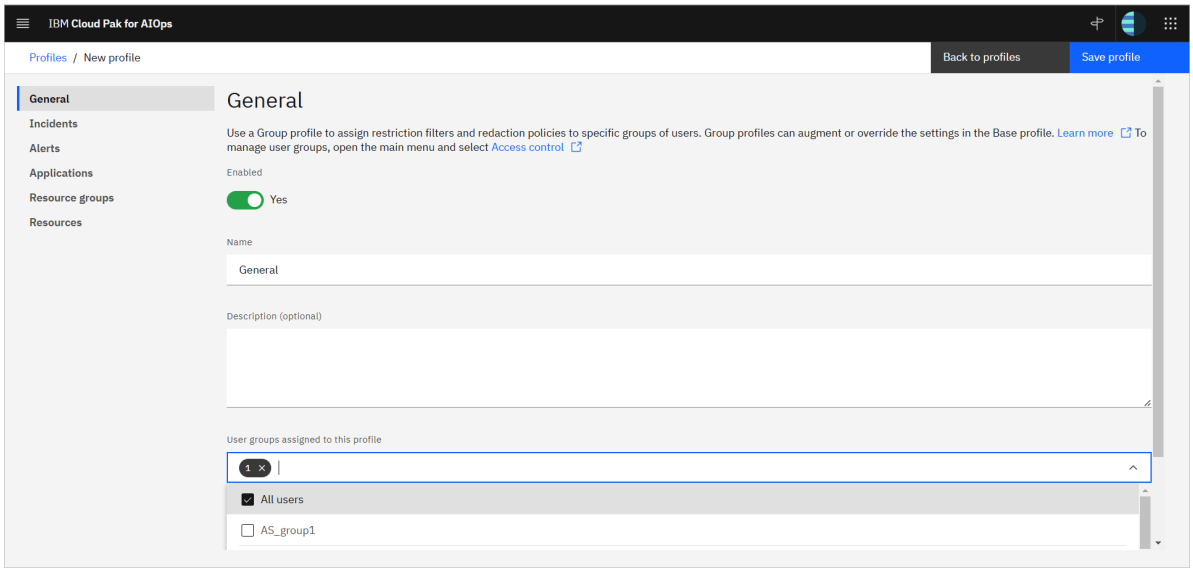


Figure. New profile creation

3. On the **General** details page, enter the following details:

   - Select whether to enable the profile by switching the value of **Enabled** from **No** to **Yes**. You can enable or disable a profile later as needed.
   - Enter a **Name** for the profile.
   - Enter a **Description (optional)** for the profile
   - Select the **User groups assigned to this profile**. Selected user groups display in a **User group** list.

4. Click through each of the following pages to select any required restriction filters or redaction policies to control the visibility of properites and content:

   - **Incidents**
   - **Alerts**
   - **Applications**
   - **Resource groups**
   - **Resources**
   - To control visibility, select a **Restriction filter**. For example, on the **Incidents** page, this selection is in the section **Visibility of Incidents**. The selection is in a similar section on each page.

     After selecting the filter, switch the toggle to enable the filter.

     Note: If you need to create a restriction filter, you must go to the console page for that feature to create the filter. For more information, see Creating restriction filters. Ensure that you save the filter as a restriction filter.

     To view general details for a selected restriction filter, click the *Information* icon. The following details are displayed for the filter:

     - Name
     - Created (date)
     - Created by
     - Last modified (date)
     - Last modified by
   - To control visibility and content, select a **Redaction policy**. For example, on the **Incidents** page, this selection is in the section **Visibility and content of Incident properties**. The selection is in a similar section on each page.

     After selecting the filter, switch the toggle to enable the policy. Once a filter/policy has been selected from the list the toggle automatically switches to *on* but it can be set to *off* if required.

     Notes:

     - If you need to edit the selected policy, click the **Edit selected redaction policy** icon.
     - If you need to create a redaction policy, complete one of the following actions:
       - Click the **Add new redaction policy** icon.
       - Select a policy and click the **Duplicate selected redaction policy** icon.

     For more information about creating and editing redaction policies, see Creating redaction policies. Redaction policies must be authored in the Rego language.

     To view general details for a selected restriction filter, click the *Information* icon. The following details are displayed for the filter:

- Name
- Created (date)
- Created by
- Last modified (date)
- Last modified by

5. Click **Save profile**. Your new profile is saved and you are returned to the **Profiles** page. The new profile is now included in the list of existing profiles.

Viewing a profile

To view an existing group profile or the base profile, go to the **Profiles** page in the UI:

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles and the following general details for those profiles:

   - **Name**:
   - **Type**:
   - **Scope**:
   - **Created**:
   - **Created by**:
   - **Last modified**:
   - **Last modified by**:
   - **Enabled**:

2. (Optional) If needed, you can filter the list of profiles to help you find a specific profile.

Enabling or disabling a profile

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles.
2. Find the profile that you want to enable or disable. If needed, you can filter the list of profiles to help you find a specific profile.
3. For that profile, click the *Options* (3-dot) menu and click either **Enable** or **Disable**.

   A dialog requesting you to confirm your choice is displayed. This dialog indicates the number of affected user groups for changes to the selected profile. Click **Enable profile** or **Disable profile** to confirm your choice.

Editing a profile

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles.
2. Find the profile that you want to edit. If needed, you can filter the list of profiles to help you find a specific profile.
3. For that profile, click the *Options* (3-dot) menu and click **Edit selected**.
4. Complete any required changes and click **Save profile**. For details on the available settings, see Creating a group profile.

   Note: If you want to edit the **Base** profile, the available settings to edit are the same as a group profile.

Duplicating a profle

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles.
2. Find the profile that you want to duplicate. If needed, you can filter the list of profiles to help you find a specific profile.

   Note: You cannot duplicate the **Base** profile.

3. For that profile, click the *Options* (3-dot) menu and click **Duplicate**.
4. Enter or select the required settings for the profile and click **Save profile**. For details on the available settings, see Creating a group profile.

Deleting a profile

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles.
2. Find the profile that you want to delete. If needed, you can filter the list of profiles to help you find a specific profile.
3. For that profile, click the *Options* (3-dot) menu and click **Delete**.

   A dialog requesting you to confirm the deletion is displayed. Click **Delete** to confirm and delete the profile.

Restriction filters

Restriction filters control the visibility of incidents, alerts, applications, resource groups, and resources. These filters are assigned to users through profiles. When creating the filter the name must be unique to the system, including normal filters, and then if editing the filter the name cannot be changed.

You can use restriction filters to control how information is distributed to your users. Restriction filtering works by enabling you to specify that under certain situations, the users should only see particular alerts, incidents, applications, resource groups or resources based on particular conditions.

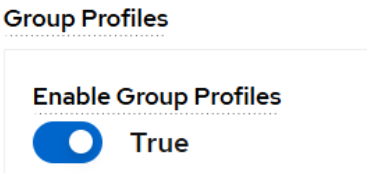
To create and apply a restriction filter, you need to create a filter from either the *Incidents*, *Alerts*, or *Resource management* (application, resource, resource group) page and save the filter as a restriction filter. After creating a restriction filter, you can include it in the Base profile (applied to all users) or an appropriate group profile (applied to specified user groups).

- Prerequisites
- Creating a restriction filter
- Viewing restriction filters
- Editing restriction filters

Prerequisites

To work with profiles, and their associated restriction filters and redaction policies, you must possess an administrative user account with the **Manage profiles** role. For more information, see Roles and permissions.

Note: In the installation resource you are using you need to set the *Enable Group Profiles* radio button to *True*.



Creating a restriction filter

1. Go to the appropriate UI page for the filter that you want to create:
   - For an incident-related filter, go to the **Operate** > **Incidents** page.
   - For an alert related filter, go to the **Operate** > **Alerts** page.
   - For an application, resource group, or resource-related filter, go to the **Operate** > **Resource management** page.
2. Follow the steps for creating a new filter, but save the filter as a restriction filter. For details on creating a filter see:
   - For an incident-related filter, see Searching and filtering incidents.
   - For an alert related filter, see Searching and filtering alerts.
   - For an application, resource group, or resource related filter, see Resource management: Searching and filtering.
3. When you have finished selecting your options for the filter, click the three dots icon next to the Filter conditions. From the options that are listed, click **Save as new restriction filter**.

   The following images show where to find the *Save as new restriction filter* option for creating a filter from each of the **Incidents**, **Alerts**, **Resource management** pages.

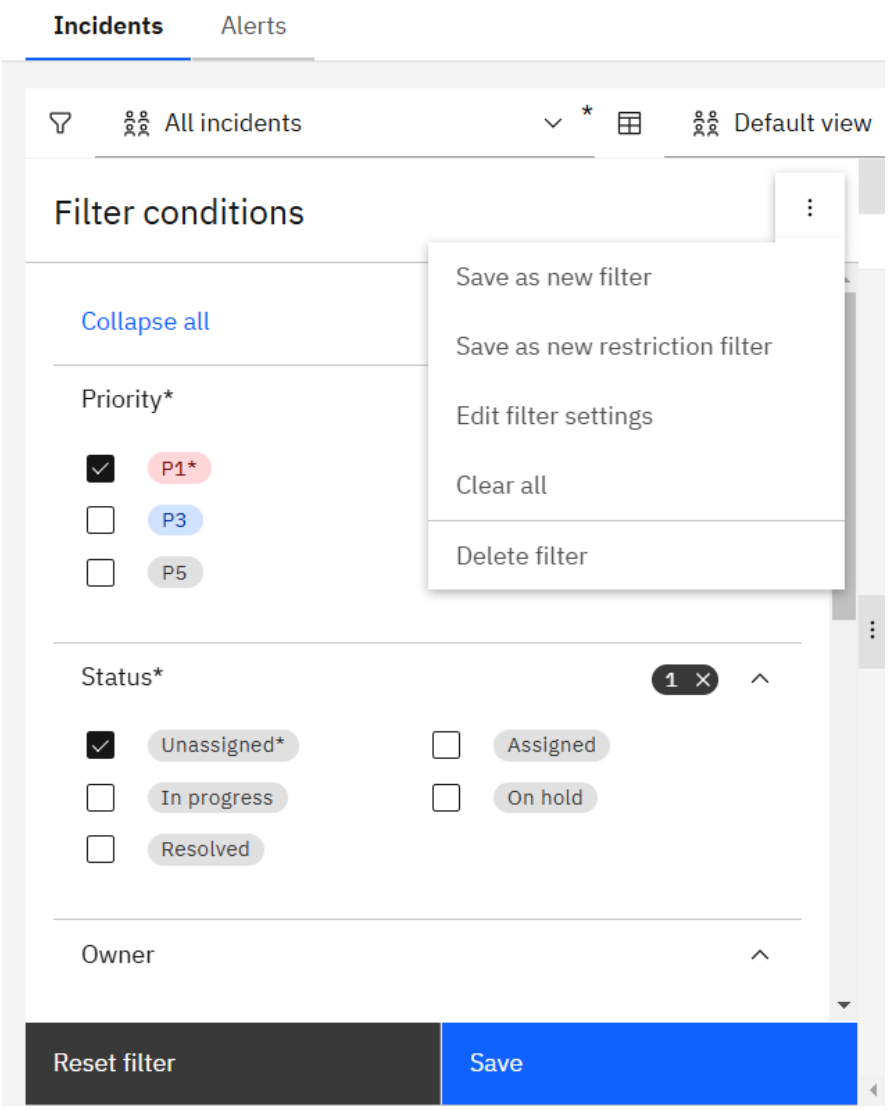


Figure. Example showing filters being applied for incidents

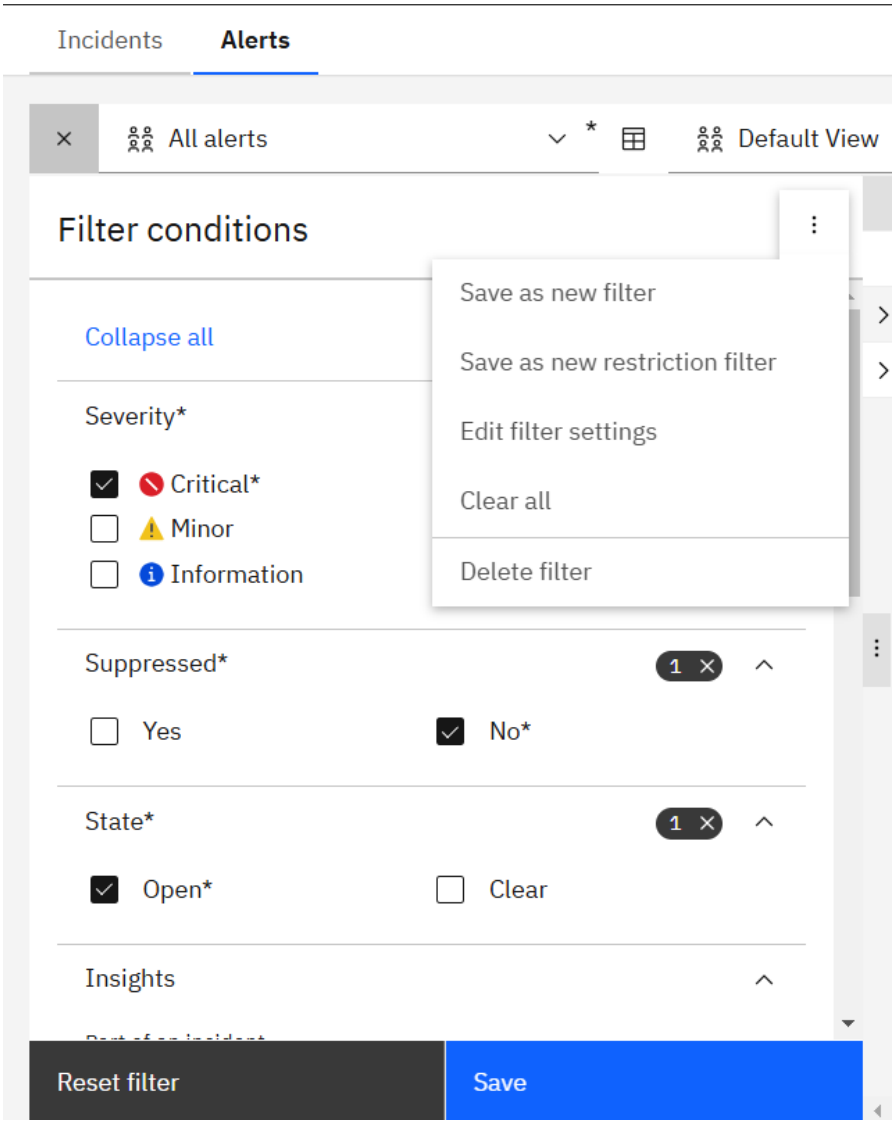


Figure. Example showing filters being applied for alerts

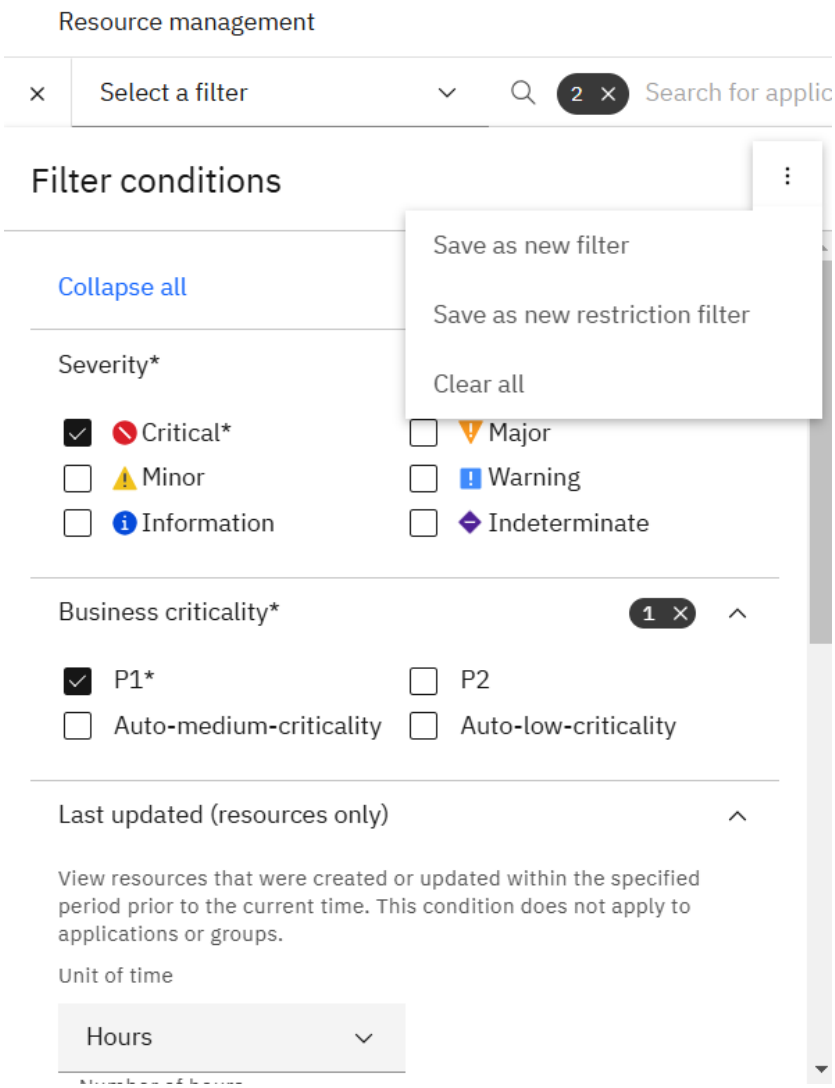


Figure. Example showing filters being applied for resource management

4. On the **Save as new restriction filter** dialog, complete the following settings:

- Enter the name and description for the filter.
- Select any user groups that need to be assigned the filter. For instance, assign the filter to any groups that are assigned *Manage Profiles* permission and responsible for creating and managing access control profiles.

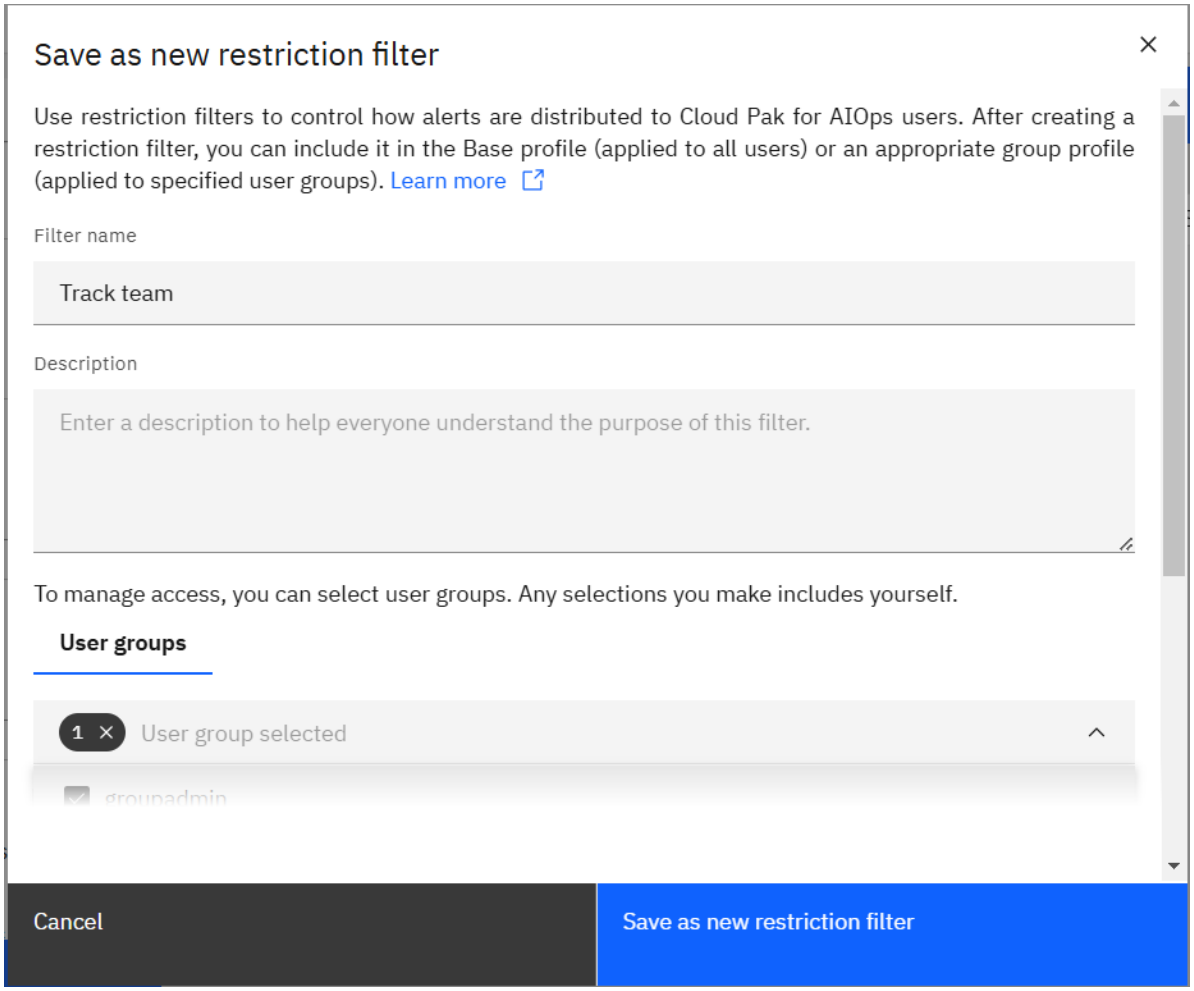


Figure. Save new restriction filter dialog

5. When you have finished entering and selecting the details for the filter, click **Save as new restriction filter**.

   Your new filter is created and listed on the **Restriction filters** tab of the **Administration** > **Profiles** page.

6. From the **Restriction filters** tab, click the **Profiles** tab and apply your created filter to a profile. For more information, see Managing profiles.

Viewing restriction filters

To view restriction filters go to the **Administration** > **Profiles** page. Then, click the **Restriction filters** tab. The list of existing filters is displayed, with the following details for each filter:

- Name
- Data types
- Created (date)
- Created by
- Last modified (date)
- Last modified by

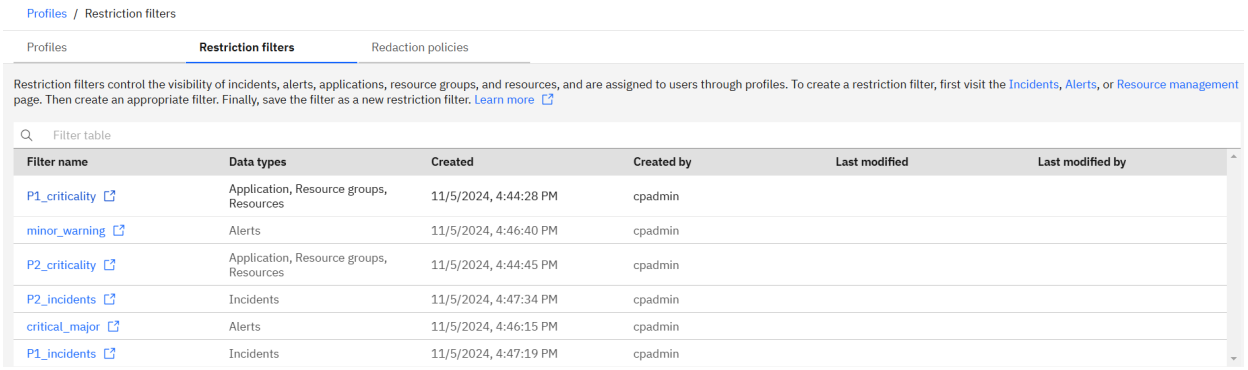


Figure. Restriction filters list

Editing restriction filters

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed.
2. Click the **Restriction filters** tab. The list of existing filters is displayed, with the following details for each filter:

- Name
- Data types
- Created (date)
- Created by
- Last modified (date)
- Last modified by

3. Find the filter that you want to edit. If needed, you can filter the list to help you find a specific filter.
4. For that restriction filter, click the value for the **Filter name**. The corresponding **Incidents**, **Alerts**, or **Resource management** page for editing the filter opens.
5. Click the Filter icon to open the filter side panel. Complete any required changes and click **Save**. profile**. For details on the available settings, see the appropriate documentation:
    - For an incident-related filter, see Searching and filtering incidents.
    - For an alert related filter, see Searching and filtering alerts.
    - For an application, resource group, or resource-related filter, see Resource management: Searching and filtering.

Redaction policies

You can use redaction policies to hide, show, and change the properties (fields) of management artifacts such as incidents, alerts, applications, resource groups, and resources to users.

Redaction policies work alongside filtering to define the access control of data for your users. You can create, edit and duplicate direct from the profile definitions. If you need to hide or show management artifacts, as opposed to the properties of management artifacts, use restriction filters For more information, see Creating restriction filters.

Base policies are the most redactive policies, comprising the minimum set of properties every user in the system is allowed to see. Group policies act as a way of additionally including extra properties for certain users. If a user has multiple group profiles assigned with different policies then the end JSON they see will be the combination of all policies. If the base policy option is disabled it might be due to no base policy being currently enabled.

Redaction policies must be authored using Rego, a programming language created specifically for policy-based control of IT environments. You can author your policies to manipulate the end result output either by hiding or showing all the properties or reacting or modifying particular properties.

For authoring a redaction policy in the Rego policy language, you can use the authoring tools that are included with the console **Profiles** page. These authoring tools include syntax checking and linting capabilities to help you create your policy source. For linting error notifications in the line number margin appear with details of the error.

Redaction policies are implemented using Open Policy Agent, which is an Apache v2.0 licensed CNCF project. You can use existing Open Policy Agent documentation to help you author your own policies. For more information, see the Open Policy Agent documentation.

Note: If a custom tool or styling is being applied using specific properties, you must ensure these are included in the final payload and not redacted, as otherwise the UI will not work.

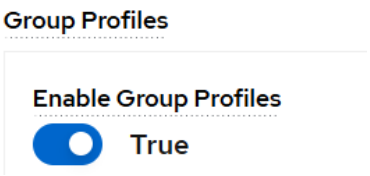
You can apply a created redaction policy to all users, or specific user groups through a profile. To apply a policy to all users, apply the policy to the *Base* profile. To apply a policy to one or more specific user groups, use a *Group* profile. For more information, see Managing profiles.

- Prerequisites
- Creating a redaction policy
    - testing a policy
- Viewing a redaction policy
- Editing a redaction policy
- Duplicating a redaction policy
- Deleting a redaction policy

Prerequisites

To work with profiles, and their associated restriction filters and redaction policies, you must possess an administrative user account with the **Manage profiles** role. For more information, see Roles and permissions.

Note: In the installation resource you are using you need to set the *Enable Group Profiles* radio button to *True*.



Creating a redaction policy

You can create duplicate, edit, and delete redaction policies from the console **Profiles** page.

1. Go to the **Administration** > **Profiles** page. Then, click the **Redaction policies** tab. The list of existing policies is displayed.

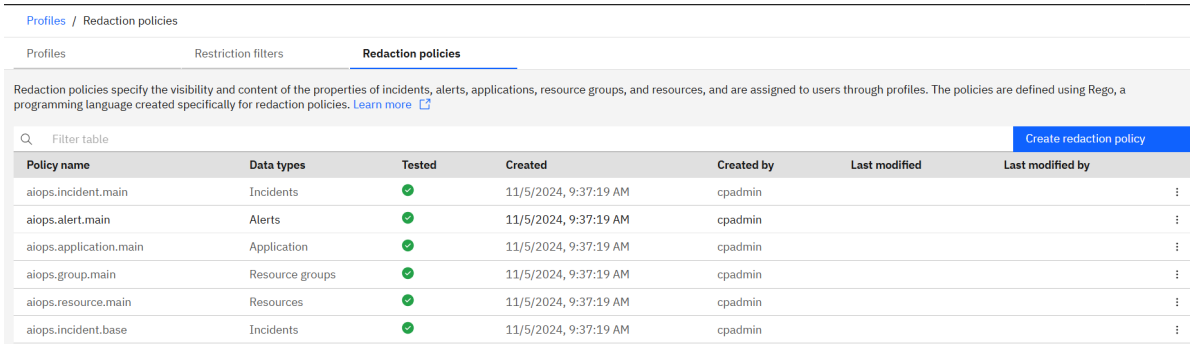


Figure. Redaction policy tab

2. Click **Create redaction policy**. The options and authoring tools for creating a **New redaction policy** are displayed. From this page, you can author your policy source, apply it to a particular data type, test it, and save it.

You can also reach this page when creating or editing a profile.

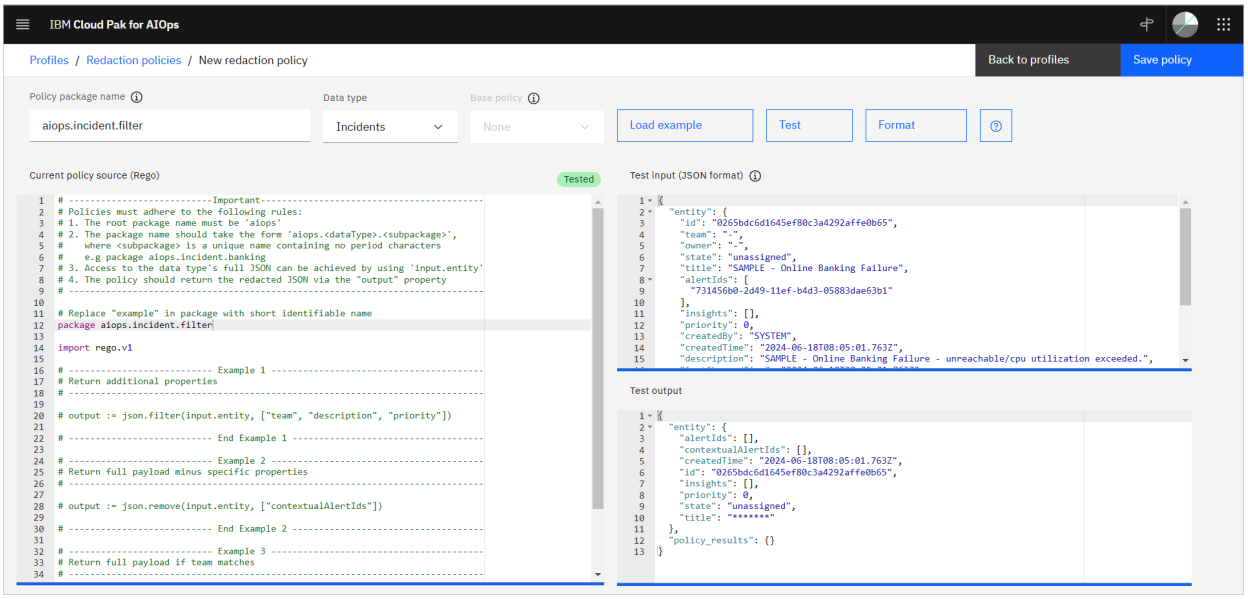


Figure. Redaction policy creation

3. Enter or select the following mandatory details for the policy:

- **Policy package name**: The name must be in the form **`aiops.<data_type>.<policy_name>`**, where **`<data_type`** is either **`incident`**, **`alert`**, **`application`**, **`group`**, or **`resource`**. For example, **`aiops.resource.myPolicy`**.

  The value for the *Policy package name* must match the value for the package name that is include in the policy source.

- **Data type**: Select the type of management artifact that the policy affects. The value can be **`Incidents`**, **`Alerts`**, **`Applications`**, **`Resource groups`**, or **`Resources`**.

  Note: If the redaction policy editor is opened when you are creating or editing a profile, you are not be able to change the data type.

- **Base policy**: Select a base policy when the current policy will be included in a *Group* profile, and there will be a corresponding policy in the *Base* profile. The base policy source will be displayed above the current policy source.

  When you run a test, the test is run against the base policy and the current policy. If a base policy is not selected, tests are run against the current policy alone.

When testing with a base policy the output includes a section called *policy_results*, which shows the individual contributions from each policy.

4. (Optional) Click **Load example** to load an example policy and companion test data to view how a policy and test input can be structured.

   To test the example, click **Test**. The *Test* button will only be clickable once the policy has been given a valid package name. You can then view the results in the **Test output** area.

   Note: For a test policy the entity JSON should be placed under the entity as this will match the input format in the policy.

5. Author your policy source within the **Current policy source (Rego)** area.

   For examples of how to define different filter conditions, see Example redaction policies.

6. Click **Format** to properly indent each line of your policy source. A red error will show at the top of the file if formatting is required.

7. Test your policy source. To test a policy load representative test data. You can include the JSON representation of an incident, alert, application, resource group, or resource.

   When you have your input data entered within the **Test input (JSON format)** area, click **Test**. Review the results in the **Test output** area.

   Note: If a base policy is selected, the base policy and the current policy are included in tests. To exclude the base policy from tests, set **Base policy** to *None*.

   The following examples show how to get test data for existing management artifacts:

   Example: Getting incident test data:

   1. Open the main menu and select **Operate** > **Incidents**.
   2. Open an appropriate incident.
   3. Click **View raw**.
   4. Copy the JSON code to the clipboard.
   5. In the Test input field of the redaction policy editor, include the JSON representation of the incident as an entity, for example:

      ```
      {
       "entity": {
         "id": "12345678-0005-2024-0920-025539000001",
         "state": "unassigned",
         …
        }
      }
      ```

   Example: Getting alert test data:

   1. Open the main menu and select **Operate** > **Alerts**.
   2. Open an appropriate alert.
   3. In the Alert details panel, select the **Raw view**.
   4. Copy the JSON code to the clipboard.
   5. In the Test input field of the redaction policy editor, include the JSON representation of the alert as an entity, for example:

      ```
      {
      "entity": {
        "acknowledged": false,
        "deduplicationKey": "12345678-0005-2024-0920-025539000020",
        …
       }
      }
      ```

   Example: Getting application, resource groups, or resources test data:

   1. Open the main menu and select **Operate** > **Resource management**.
   2. On the Resource management page, select the **Applications** tab, **Resource groups** tab, or **Resources** tab.
   3. Locate an appropriate application/service, resource group, or resource and click More details.
   4. In the details window, set **Show JSON** to On.
   5. Copy the JSON code to the clipboard.
   6. In the Test input field of the redaction policy editor, include the JSON representation of the application/service, resource group, or resource as an entity, for example:

      ```
      {
       "entity": {
         "_id": "rSRqQX6NTVywShGIjhm-Kg",
         "businessCriticality": "Tier 1",
         …
        }
      }
      ```

8. When completed and tested, click **Save policy**.

Your new policy is created and listed on the **Redaction policy** tab of the **Administration** > **Profiles** page.

9. From the **Redaction policies** tab, click the **Profiles** tab and apply your created policy to one or more profiles. For more information, see Managing profiles.

Tips: If a policy is not working as expected, complete the following actions:

- Leverage the test capabilities included in the redaction policy editor.
- Ensure that the policy is selected and enabled in the appropriate profiles.
- Ensure that the relevant profiles are enabled.
- If a policy is included in one or more group profiles, ensure that the profiles include the appropriate user groups.

Viewing redaction policies

To view restriction filters go to the **Administration** > **Profiles** page. Then, click the **Redaction policies** tab. The list of existing policies is displayed, with the following details for each policy:

- Policy Name
- Data types
- Tested
- Created (date)
- Created by
- Last modified (date)
- Last modified by

Editing a redaction policy

It is important to note that the main policy should never be modified unless done with the help of support. The main policy is responsible for combining all policies together. It also ensures that the required properties are always included, and should not be removed as they are used by the UI to render screens correctly.

1. From the main navigation menu, go to **Administration** > **Profiles**. Click the **Redaction policies** tab.
2. Find the policy that you want to edit. If needed, you can filter the list of policies to help you find a specific policy.
3. For that policy, click the *Options* (3-dot) menu and click **Edit selected**.
4. Complete any required changes and click **Save policy**.

Duplicating a redaction policy

1. From the main navigation menu, go to **Administration** > **Profiles**. Click the **Redaction policies** tab.
2. Find the policy that you want to duplicate. If needed, you can filter the list of policies to help you find a specific policy.
3. For that policy, click the *Options* (3-dot) menu and click **Duplicate**.
4. Change the policy as needed and click **Save profile**.

Deleting a redaction policy

1. From the main navigation menu, go to **Administration** > **Profiles**. Click the **Redaction policies** tab.
2. Find the policy that you want to delete. If needed, you can filter the list of policies to help you find a specific policy.
3. For that policy, click the *Options* (3-dot) menu and click **Delete**.

   A dialog requesting you to confirm the deletion is displayed. Click **Delete** to confirm and delete the policy.

   Note: You cannot delete a policy if is being used in the profile.

Example redaction policies

The following code samples shows the various different ways to define code filters in Rego:

Full JSON example

```
{
    "entity": {
      "_id": "imSB3oo-RcWqjFhX-55Qag",
      "name": "delaware10-le4-cs29.na.test.lab",
      "tags": [
        "1.3.6.1.4.1.9.1.217",
        "B510/3D32 3rd Floor Lab",
        "CiscoCat29xx",
        "joe@ibm.com"
      ],
      "businessCriticality": "Gold",
      "manual": 0,
      "cdmType": 2,
      "sysName": "delaware10-le4-cs29.na.test.lab",
      "services": "datalink(2)",
```

```
      "uniqueId": "Example:delaware10-le4-cs29.na.test.lab",
      "className": "CiscoCat29xx",
      "sysContact": "joe@ibm.com",
      "vertexType": "resource",
      "description": "Description text here",
      "entityTypes": [
        "host",
        "switch"
      ],
      "matchTokens": [
        "delaware10-le4-cs29.na.test.lab",
        "delaware10-le4-cs29",
        "entityid.23492249"
      ],
      "sysLocation": "B510/3D32 3rd Floor Lab",
      "sysObjectId": "1.3.6.1.4.1.9.1.217",
      "cdmAdminState": 0,
      "accessProtocol": "IPv4",
      "interfaceCount": 27,
      "isIpForwarding": "not-forwarding",
      "sysDescription": "System description",
      "accessIPAddress": "172.30.238.10",
      "entityChangeTime": "2019-07-19T15:17:34.000Z",
      "entityCreateTime": "2019-07-19T15:17:34.000Z",
      "accessScopeTokens": [
        "file.observer:itnm.txt"
      ],
      "lastBroadcastTime": "2024-03-21T11:11:33.927Z",
      "entityDiscoveryTime": "2019-07-19T15:04:42.000Z"
    }
  }
```

Equals filter example

```
output := input.entity if {
    input.entity.cdmType == 2
}
```

Is not equal filter example

```
output := input.entity if {
    not input.entity.cdmType == 2
}
```

Contains filter example

```
output := input.entity if {
    contains(input.entity.name, "ware")
}
```

Does not contain filter example

```
output := input.entity if {
    not contains(input.entity.name, "ware")
}
```

Starts with filter example

```
output := input.entity if {
    startswith(input.entity.name, "del")
}
```

Ends with filter example

```
output := input.entity if {
    endswith(input.entity.name, ".lab")
}
```

Empty filter example

```
output := input.entity if {
    not input.entity.cdmType
}
```

Is not empty filter example

```
output := input.entity if {
    input.entity.cdmType
}
```

Matches filter example

```
output := input.entity if {
    regex.match(`^delaware`, input.entity.name)
}
```

Does not match filter example

```
output := input.entity if {
    not regex.match(`^delaware`, input.entity.name)
}
```

Check value matches a value in a set filter example

Requires knowing the criticalities to allow

```
output := input.entity if {
    input.entity.businessCriticality in ["Gold", "Silver", "Bronze"]
}
```

Includes any filter example

Check a list of provided types

```
output := input.entity if {
    some type in input.entity.entityTypes
    type in ["host", "otherType", "otherType2"]
}
```

Includes all filter example

```
output := input.entity if {
    every type in input.entity.entityTypes {
        type in ["host", "switch"]
    }
}
```

Complex combination

```
output := input.entity if {
    # When dealing with a host
    "host" in input.entity.entityTypes

    # Must start with del to show properties
    startswith(input.entity.name, "del")

    # Must include the tag "joe@ibm.com"
    some type in input.entity.tags
    type in ["joe@ibm.com"]
}
```

Complex definition to handle many different types

```
output := input.entity if {
    # When dealing with a host
    "host" in input.entity.entityTypes

    # Must start with del to show properties
    startswith(input.entity.name, "del")

    # Must include the tag "joe@ibm.com"
    some type in input.entity.tags
    type in ["joe@ibm.com"]
} else := input.entity if {
    # When dealing with a people
    "people" in input.entity.entityTypes

    # Must start with del to show properties
    startswith(input.entity.team, "uk_")
} else := input.entity if {
    # When dealing with a vms
    "vm" in input.entity.entityTypes

    # Must start with del to show properties
    regex.match(`^delaware`, input.entity.name)
}
```

Profiles

Profiles can be assigned to user groups to control the visibility of incidents, alerts, applications, resource groups, and resources to those users. These profiles can combine restriction filters with redaction policies to form the final redaction criteria that is applied to a user group.

A profile can be a *Group* profile or a *Base* profile. By default, there is always a **Base** profile which applies to all users. You can use this profile to assign restriction filters and redaction policies to all users. You can use one or more **Group** profiles to assign restriction filters and redaction policies to specific groups of users. *Group* profiles can augment or override the settings in the *Base* profile. The Base profile should be configured to show the minimum set of entities or properties that users in the system can see, so that the filter and policies should be aiming to remove everything from the views, while the group profiles aim to add things back into the views. A profile can be enabled, even when the underlying filters and policies are disabled.

Restriction filters control the visibility of incidents, alerts, applications, resource groups, and resources. These filters are assigned to users through profiles. You can use restriction filters to control how information is distributed to your users. Restriction filtering works by enabling you to specify that under certain situations, users should only see particular alerts, incidents, applications, resource groups or resources based on particular conditions.

Notes:

- Restriction filters do not hide resources in topology graph visualisations, so a combination of filter and redaction is required to make sure that no data is exposed.
- If no filter or policy shows it might be due to none being available for that data type.

To create and apply a restriction filter, you need to create a filter from either the *Incidents*, *Alerts*, or *Resource management* (application, resource, resource group) page and save the filter as a restriction filter. After creating a restriction filter, you can then include it in the Base profile or an appropriate group profile. For more information, see Creating restriction filters.

Redaction policies work alongside filtering and allow you to specify the visibility and content of the properties of incidents, alerts, applications, resource groups, and resources, and are assigned to users through profiles. The policies are defined using Rego, a programming language created specifically for redaction policies, which allows you to manipulate the end result output either by hiding or showing all the properties or reacting or modifying particular properties. For more information, see Creating redaction policies.

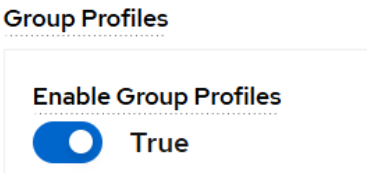
If you need to create or manage a user group before applying any profile, see Managing user groups.

- Prerequisites
- Creating a group profile
- Viewing a profile
- Enabling or disabling a profile
- Editing a profile
- Duplicating a profile
- Deleting a profile

Prerequisites

To work with profiles, and their associated restriction filters and redaction policies, you must possess an administrative user account with the **Manage profiles** role. For more information, see Roles and permissions.

Note: In the installation resource you are using you need to set the *Enable Group Profiles* radio button to *True*.



Creating a group profile

1. From the main navigation menu, go to **Administration** > **Profiles**.

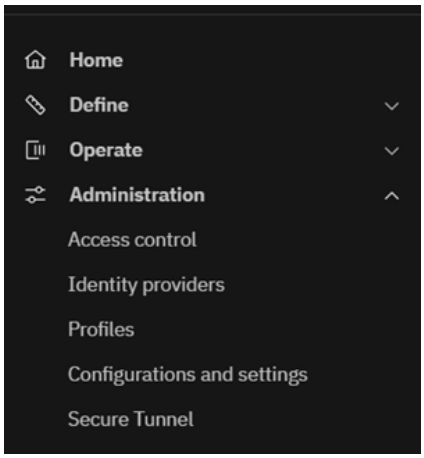


Figure. Profiles list

By default, the **Profiles** tab for the **Profiles** page is displayed.

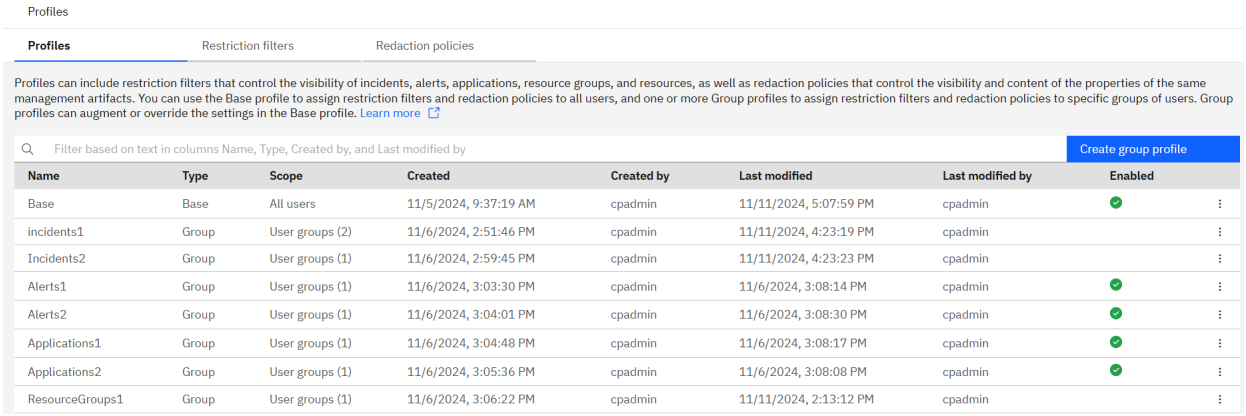


Figure. Profiles list

2. Click **Create group profile**. The options for creating a **New profile** are displayed.

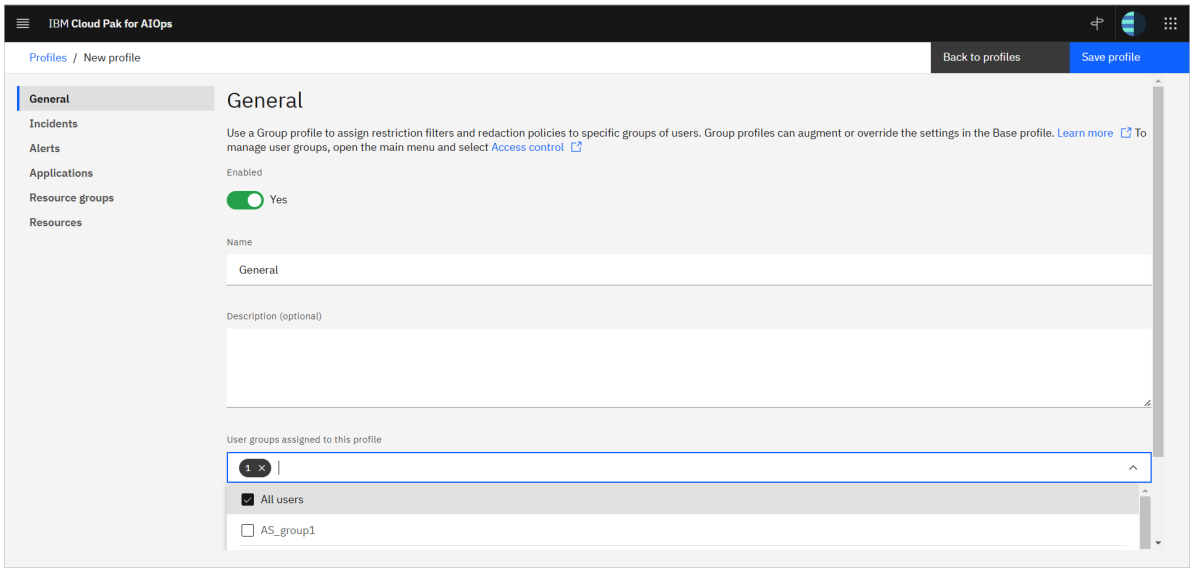


Figure. New profile creation

3. On the **General** details page, enter the following details:

   - Select whether to enable the profile by switching the value of **Enabled** from **No** to **Yes**. You can enable or disable a profile later as needed.
   - Enter a **Name** for the profile.
   - Enter a **Description (optional)** for the profile
   - Select the **User groups assigned to this profile**. Selected user groups display in a **User group** list.

4. Click through each of the following pages to select any required restriction filters or redaction policies to control the visibility of properites and content:

   - **Incidents**
   - **Alerts**
   - **Applications**
   - **Resource groups**
   - **Resources**
   - To control visibility, select a **Restriction filter**. For example, on the **Incidents** page, this selection is in the section **Visibility of Incidents**. The selection is in a similar section on each page.

     After selecting the filter, switch the toggle to enable the filter.

     Note: If you need to create a restriction filter, you must go to the console page for that feature to create the filter. For more information, see Creating restriction filters. Ensure that you save the filter as a restriction filter.

     To view general details for a selected restriction filter, click the *Information* icon. The following details are displayed for the filter:

     - Name
     - Created (date)
     - Created by
     - Last modified (date)

- Last modified by

- To control visibility and content, select a **Redaction policy**. For example, on the **Incidents** page, this selection is in the section **Visibility and content of Incident properties**. The selection is in a similar section on each page.

  After selecting the filter, switch the toggle to enable the policy. Once a filter/policy has been selected from the list the toggle automatically switches to *on* but it can be set to *off* if required.

  Notes:

  - If you need to edit the selected policy, click the **Edit selected redaction policy** icon.
  - If you need to create a redaction policy, complete one of the following actions:
    - Click the **Add new redaction policy** icon.
    - Select a policy and click the **Duplicate selected redaction policy** icon.

  For more information about creating and editing redaction policies, see Creating redaction policies. Redaction policies must be authored in the Rego language.

  To view general details for a selected restriction filter, click the *Information* icon. The following details are displayed for the filter:

  - Name
  - Created (date)
  - Created by
  - Last modified (date)
  - Last modified by

5. Click **Save profile**. Your new profile is saved and you are returned to the **Profiles** page. The new profile is now included in the list of existing profiles.

Viewing a profile

To view an existing group profile or the base profile, go to the **Profiles** page in the UI:

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles and the following general details for those profiles:

   - **Name**:
   - **Type**:
   - **Scope**:
   - **Created**:
   - **Created by**:
   - **Last modified**:
   - **Last modified by**:
   - **Enabled**:

2. (Optional) If needed, you can filter the list of profiles to help you find a specific profile.

Enabling or disabling a profile

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles.
2. Find the profile that you want to enable or disable. If needed, you can filter the list of profiles to help you find a specific profile.
3. For that profile, click the *Options* (3-dot) menu and click either **Enable** or **Disable**.

   A dialog requesting you to confirm your choice is displayed. This dialog indicates the number of affected user groups for changes to the selected profile. Click **Enable profile** or **Disable profile** to confirm your choice.

Editing a profile

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles.
2. Find the profile that you want to edit. If needed, you can filter the list of profiles to help you find a specific profile.
3. For that profile, click the *Options* (3-dot) menu and click **Edit selected**.
4. Complete any required changes and click **Save profile**. For details on the available settings, see Creating a group profile.

   Note: If you want to edit the **Base** profile, the available settings to edit are the same as a group profile.

Duplicating a profle

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles.
2. Find the profile that you want to duplicate. If needed, you can filter the list of profiles to help you find a specific profile.

   Note: You cannot duplicate the **Base** profile.

3. For that profile, click the *Options* (3-dot) menu and click **Duplicate**.
4. Enter or select the required settings for the profile and click **Save profile**. For details on the available settings, see Creating a group profile.

Deleting a profile

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed. This tab lists the existing profiles.
2. Find the profile that you want to delete. If needed, you can filter the list of profiles to help you find a specific profile.
3. For that profile, click the *Options* (3-dot) menu and click **Delete**.

   A dialog requesting you to confirm the deletion is displayed. Click **Delete** to confirm and delete the profile.

Restriction filters

Restriction filters control the visibility of incidents, alerts, applications, resource groups, and resources. These filters are assigned to users through profiles. When creating the filter the name must be unique to the system, including normal filters, and then if editing the filter the name cannot be changed.

You can use restriction filters to control how information is distributed to your users. Restriction filtering works by enabling you to specify that under certain situations, the users should only see particular alerts, incidents, applications, resource groups or resources based on particular conditions.

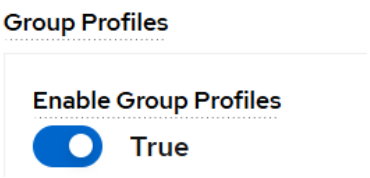
To create and apply a restriction filter, you need to create a filter from either the *Incidents*, *Alerts*, or *Resource management* (application, resource, resource group) page and save the filter as a restriction filter. After creating a restriction filter, you can include it in the Base profile (applied to all users) or an appropriate group profile (applied to specified user groups).

- Prerequisites
- Creating a restriction filter
- Viewing restriction filters
- Editing restriction filters

Prerequisites

To work with profiles, and their associated restriction filters and redaction policies, you must possess an administrative user account with the **Manage profiles** role. For more information, see Roles and permissions.

Note: In the installation resource you are using you need to set the *Enable Group Profiles* radio button to *True*.



Creating a restriction filter

1. Go to the appropriate UI page for the filter that you want to create:
   - For an incident-related filter, go to the **Operate** > **Incidents** page.
   - For an alert related filter, go to the **Operate** > **Alerts** page.
   - For an application, resource group, or resource-related filter, go to the **Operate** > **Resource management** page.
2. Follow the steps for creating a new filter, but save the filter as a restriction filter. For details on creating a filter see:
   - For an incident-related filter, see Searching and filtering incidents.
   - For an alert related filter, see Searching and filtering alerts.
   - For an application, resource group, or resource related filter, see Resource management: Searching and filtering.
3. When you have finished selecting your options for the filter, click the three dots icon next to the Filter conditions. From the options that are listed, click **Save as new restriction filter**.

   The following images show where to find the *Save as new restriction filter* option for creating a filter from each of the **Incidents**, **Alerts**, **Resource management** pages.

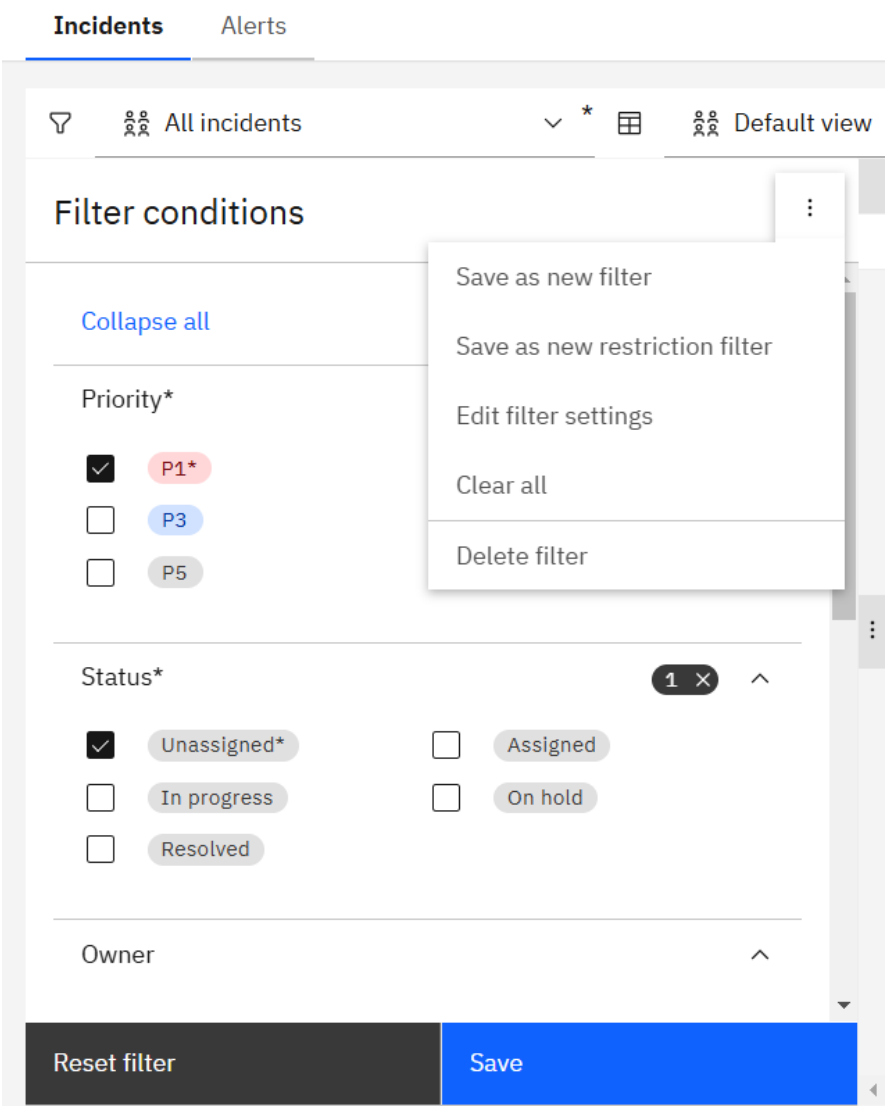


Figure. Example showing filters being applied for incidents

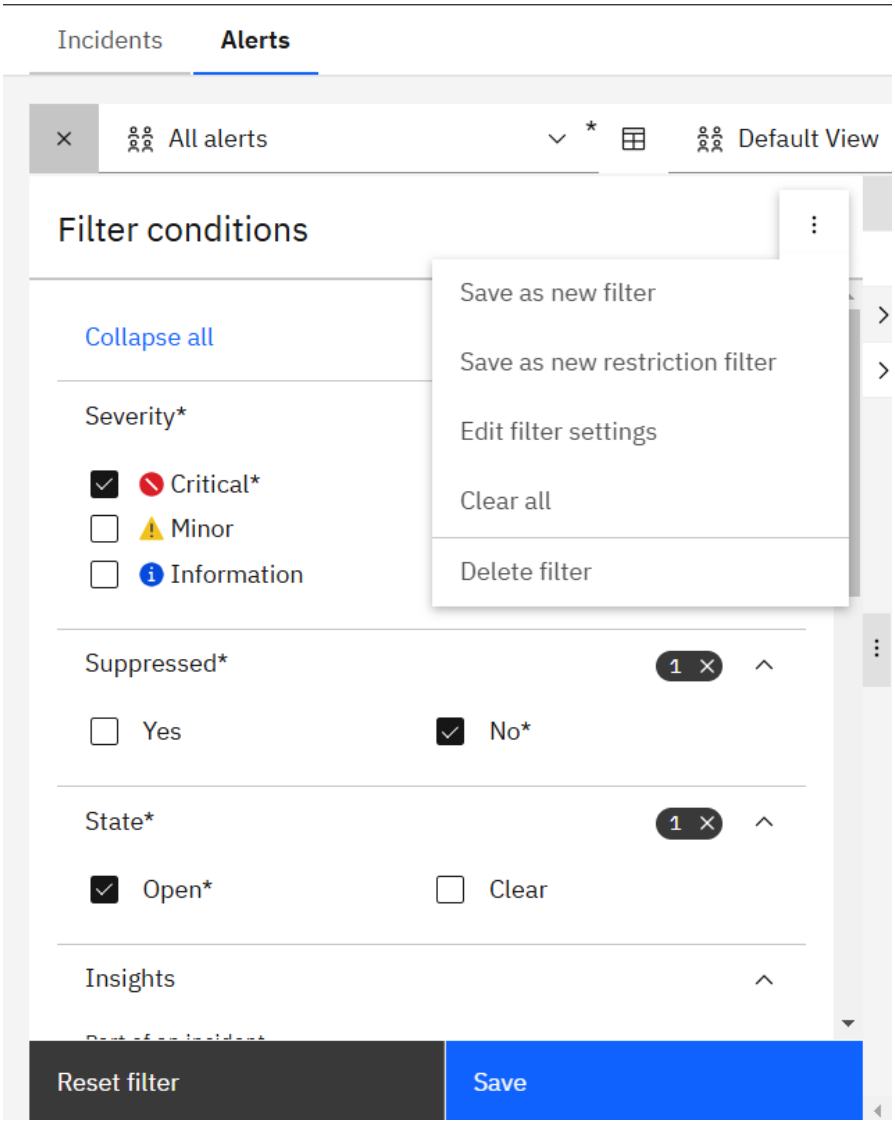


Figure. Example showing filters being applied for alerts

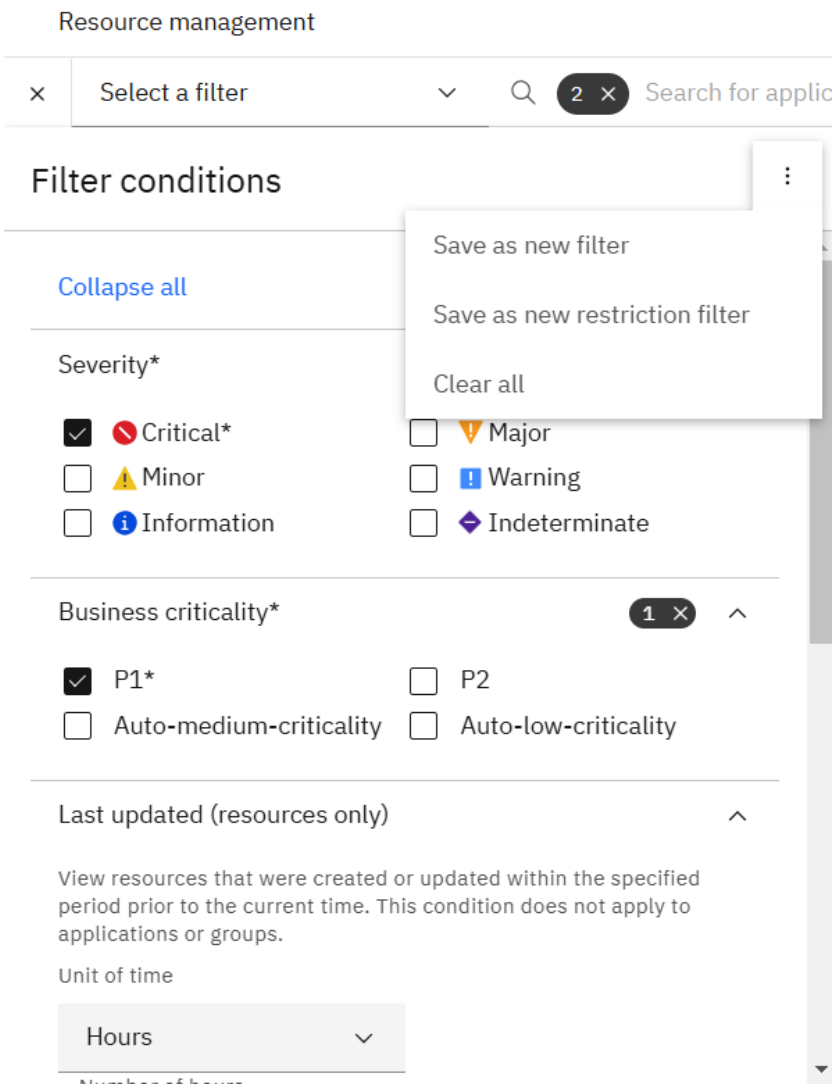


Figure. Example showing filters being applied for resource management

4. On the **Save as new restriction filter** dialog, complete the following settings:

- Enter the name and description for the filter.
- Select any user groups that need to be assigned the filter. For instance, assign the filter to any groups that are assigned *Manage Profiles* permission and responsible for creating and managing access control profiles.

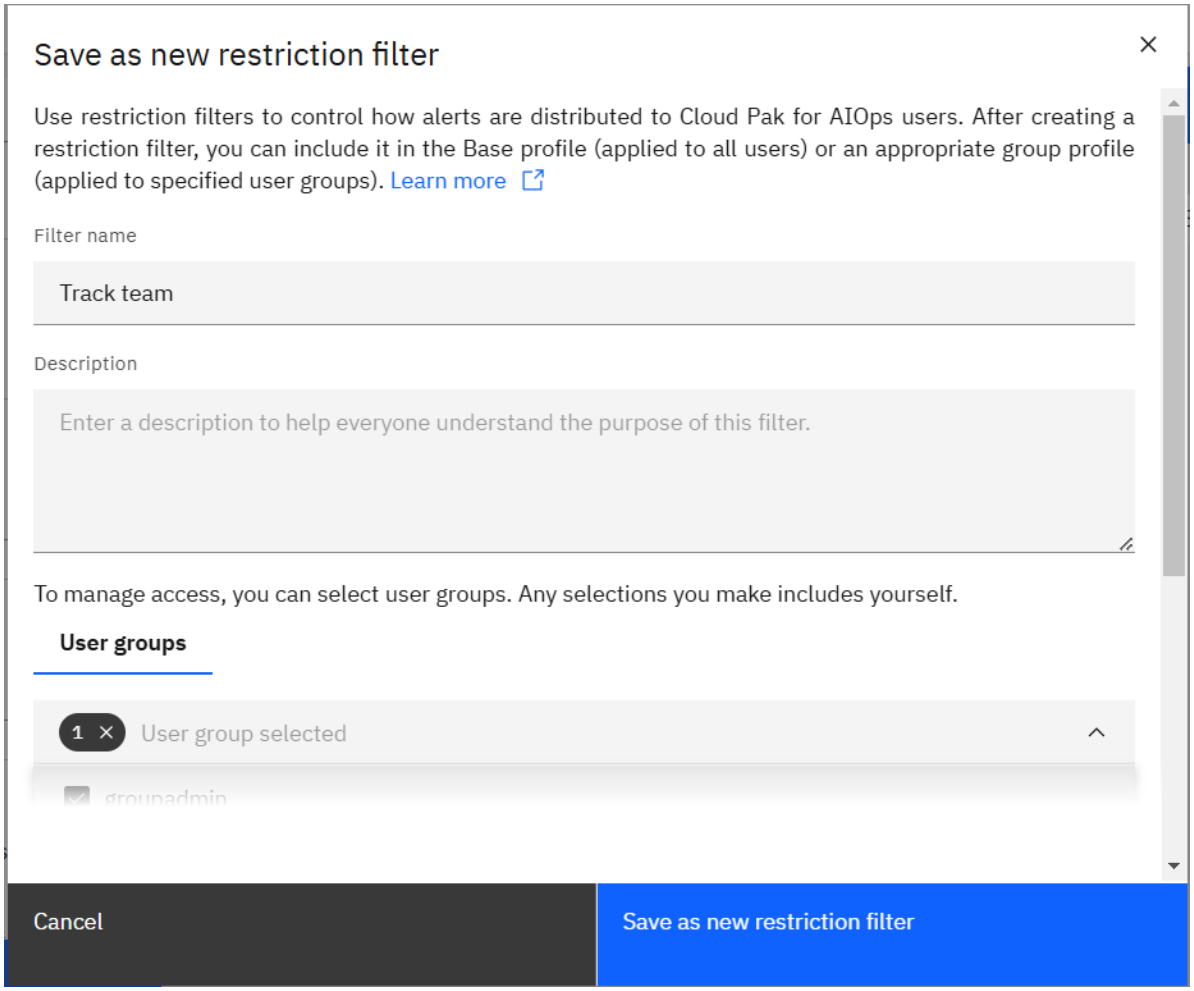


Figure. Save new restriction filter dialog

5. When you have finished entering and selecting the details for the filter, click **Save as new restriction filter**.

   Your new filter is created and listed on the **Restriction filters** tab of the **Administration** > **Profiles** page.

6. From the **Restriction filters** tab, click the **Profiles** tab and apply your created filter to a profile. For more information, see Managing profiles.

Viewing restriction filters

To view restriction filters go to the **Administration** > **Profiles** page. Then, click the **Restriction filters** tab. The list of existing filters is displayed, with the following details for each filter:

- Name
- Data types
- Created (date)
- Created by
- Last modified (date)
- Last modified by

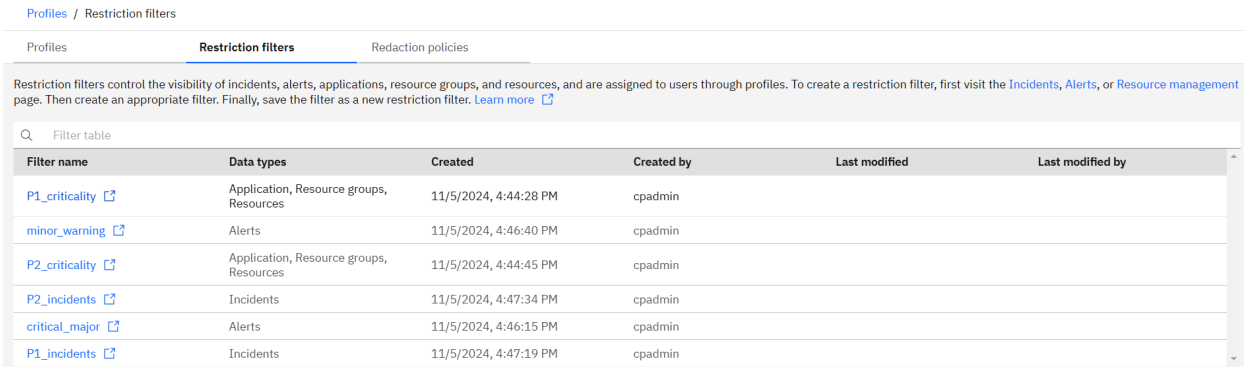


Figure. Restriction filters list

Editing restriction filters

1. From the main navigation menu, go to **Administration** > **Profiles**. By default, the **Profiles** tab for the page is displayed.
2. Click the **Restriction filters** tab. The list of existing filters is displayed, with the following details for each filter:

- Name
- Data types
- Created (date)
- Created by
- Last modified (date)
- Last modified by

3. Find the filter that you want to edit. If needed, you can filter the list to help you find a specific filter.
4. For that restriction filter, click the value for the **Filter name**. The corresponding **Incidents**, **Alerts**, or **Resource management** page for editing the filter opens.
5. Click the Filter icon to open the filter side panel. Complete any required changes and click **Save**. profile**. For details on the available settings, see the appropriate documentation:
    - For an incident-related filter, see Searching and filtering incidents.
    - For an alert related filter, see Searching and filtering alerts.
    - For an application, resource group, or resource-related filter, see Resource management: Searching and filtering.

Redaction policies

You can use redaction policies to hide, show, and change the properties (fields) of management artifacts such as incidents, alerts, applications, resource groups, and resources to users.

Redaction policies work alongside filtering to define the access control of data for your users. You can create, edit and duplicate direct from the profile definitions. If you need to hide or show management artifacts, as opposed to the properties of management artifacts, use restriction filters For more information, see Creating restriction filters.

Base policies are the most redactive policies, comprising the minimum set of properties every user in the system is allowed to see. Group policies act as a way of additionally including extra properties for certain users. If a user has multiple group profiles assigned with different policies then the end JSON they see will be the combination of all policies. If the base policy option is disabled it might be due to no base policy being currently enabled.

Redaction policies must be authored using Rego, a programming language created specifically for policy-based control of IT environments. You can author your policies to manipulate the end result output either by hiding or showing all the properties or reacting or modifying particular properties.

For authoring a redaction policy in the Rego policy language, you can use the authoring tools that are included with the console **Profiles** page. These authoring tools include syntax checking and linting capabilities to help you create your policy source. For linting error notifications in the line number margin appear with details of the error.

Redaction policies are implemented using Open Policy Agent, which is an Apache v2.0 licensed CNCF project. You can use existing Open Policy Agent documentation to help you author your own policies. For more information, see the Open Policy Agent documentation.

Note: If a custom tool or styling is being applied using specific properties, you must ensure these are included in the final payload and not redacted, as otherwise the UI will not work.

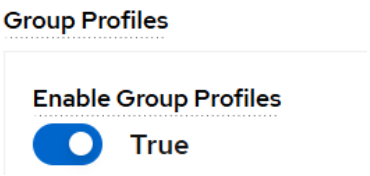
You can apply a created redaction policy to all users, or specific user groups through a profile. To apply a policy to all users, apply the policy to the *Base* profile. To apply a policy to one or more specific user groups, use a *Group* profile. For more information, see Managing profiles.

- Prerequisites
- Creating a redaction policy
    - testing a policy
- Viewing a redaction policy
- Editing a redaction policy
- Duplicating a redaction policy
- Deleting a redaction policy

Prerequisites

To work with profiles, and their associated restriction filters and redaction policies, you must possess an administrative user account with the **Manage profiles** role. For more information, see Roles and permissions.

Note: In the installation resource you are using you need to set the *Enable Group Profiles* radio button to *True*.



Creating a redaction policy

You can create duplicate, edit, and delete redaction policies from the console **Profiles** page.

1. Go to the **Administration** > **Profiles** page. Then, click the **Redaction policies** tab. The list of existing policies is displayed.

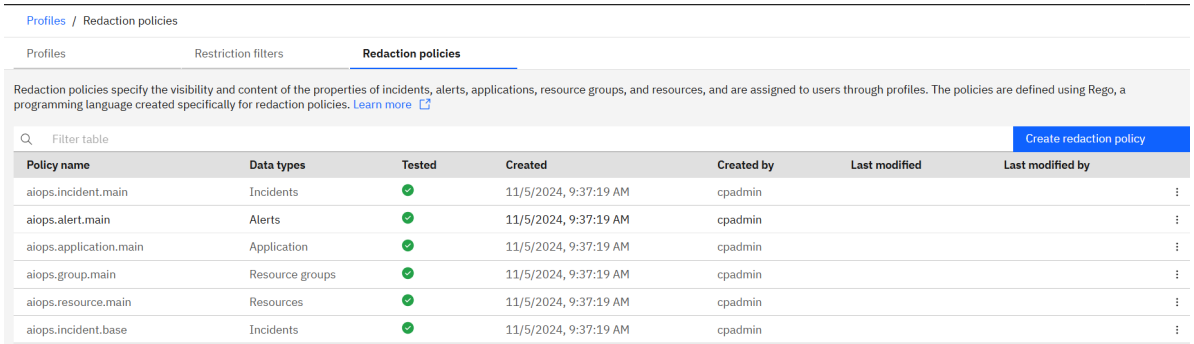


Figure. Redaction policy tab

2. Click **Create redaction policy**. The options and authoring tools for creating a **New redaction policy** are displayed. From this page, you can author your policy source, apply it to a particular data type, test it, and save it.

You can also reach this page when creating or editing a profile.

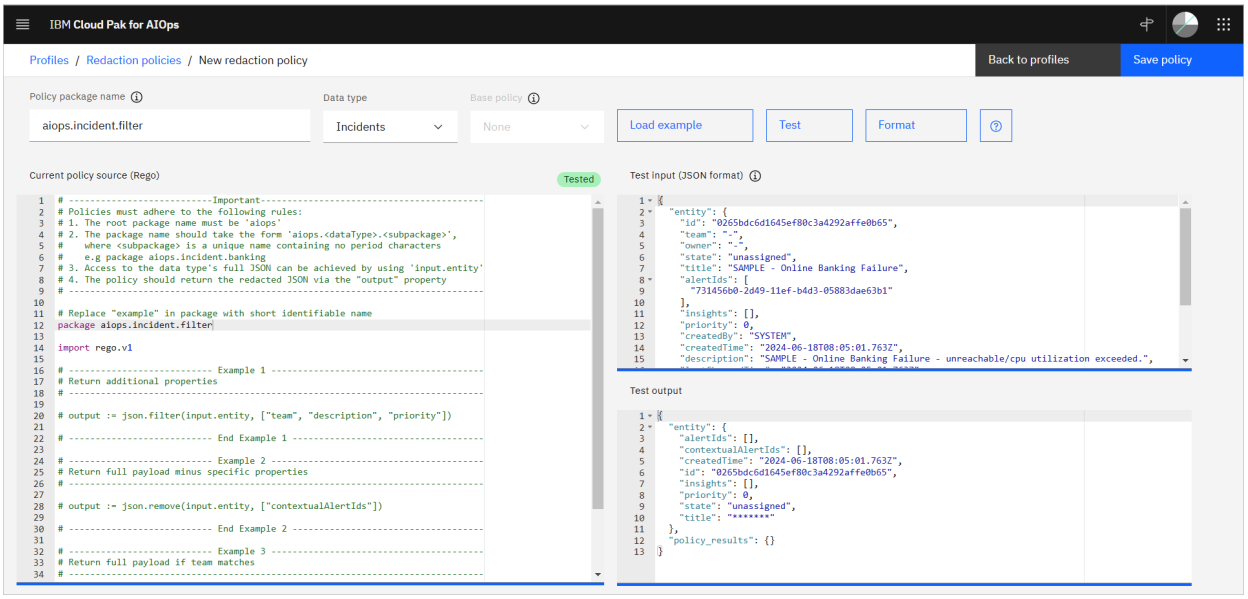


Figure. Redaction policy creation

3. Enter or select the following mandatory details for the policy:

- **Policy package name**: The name must be in the form **`aiops.<data_type>.<policy_name>`**, where **`<data_type`** is either **`incident`**, **`alert`**, **`application`**, **`group`**, or **`resource`**. For example, **`aiops.resource.myPolicy`**.

  The value for the *Policy package name* must match the value for the package name that is include in the policy source.

- **Data type**: Select the type of management artifact that the policy affects. The value can be **`Incidents`**, **`Alerts`**, **`Applications`**, **`Resource groups`**, or **`Resources`**.

  Note: If the redaction policy editor is opened when you are creating or editing a profile, you are not be able to change the data type.

- **Base policy**: Select a base policy when the current policy will be included in a *Group* profile, and there will be a corresponding policy in the *Base* profile. The base policy source will be displayed above the current policy source.

  When you run a test, the test is run against the base policy and the current policy. If a base policy is not selected, tests are run against the current policy alone.

When testing with a base policy the output includes a section called *policy_results*, which shows the individual contributions from each policy.

4. (Optional) Click **Load example** to load an example policy and companion test data to view how a policy and test input can be structured.

   To test the example, click **Test**. The *Test* button will only be clickable once the policy has been given a valid package name. You can then view the results in the **Test output** area.

   Note: For a test policy the entity JSON should be placed under the entity as this will match the input format in the policy.

5. Author your policy source within the **Current policy source (Rego)** area.

   For examples of how to define different filter conditions, see Example redaction policies.

6. Click **Format** to properly indent each line of your policy source. A red error will show at the top of the file if formatting is required.

7. Test your policy source. To test a policy load representative test data. You can include the JSON representation of an incident, alert, application, resource group, or resource.

   When you have your input data entered within the **Test input (JSON format)** area, click **Test**. Review the results in the **Test output** area.

   Note: If a base policy is selected, the base policy and the current policy are included in tests. To exclude the base policy from tests, set **Base policy** to *None*.

   The following examples show how to get test data for existing management artifacts:

   Example: Getting incident test data:

   1. Open the main menu and select **Operate** > **Incidents**.
   2. Open an appropriate incident.
   3. Click **View raw**.
   4. Copy the JSON code to the clipboard.
   5. In the Test input field of the redaction policy editor, include the JSON representation of the incident as an entity, for example:

      ```
      {
       "entity": {
         "id": "12345678-0005-2024-0920-025539000001",
         "state": "unassigned",
         …
        }
      }
      ```

   Example: Getting alert test data:

   1. Open the main menu and select **Operate** > **Alerts**.
   2. Open an appropriate alert.
   3. In the Alert details panel, select the **Raw view**.
   4. Copy the JSON code to the clipboard.
   5. In the Test input field of the redaction policy editor, include the JSON representation of the alert as an entity, for example:

      ```
      {
      "entity": {
        "acknowledged": false,
        "deduplicationKey": "12345678-0005-2024-0920-025539000020",
        …
       }
      }
      ```

   Example: Getting application, resource groups, or resources test data:

   1. Open the main menu and select **Operate** > **Resource management**.
   2. On the Resource management page, select the **Applications** tab, **Resource groups** tab, or **Resources** tab.
   3. Locate an appropriate application/service, resource group, or resource and click More details.
   4. In the details window, set **Show JSON** to On.
   5. Copy the JSON code to the clipboard.
   6. In the Test input field of the redaction policy editor, include the JSON representation of the application/service, resource group, or resource as an entity, for example:

      ```
      {
       "entity": {
         "_id": "rSRqQX6NTVywShGIjhm-Kg",
         "businessCriticality": "Tier 1",
         …
        }
      }
      ```

8. When completed and tested, click **Save policy**.

Your new policy is created and listed on the **Redaction policy** tab of the **Administration** > **Profiles** page.

9. From the **Redaction policies** tab, click the **Profiles** tab and apply your created policy to one or more profiles. For more information, see Managing profiles.

Tips: If a policy is not working as expected, complete the following actions:

- Leverage the test capabilities included in the redaction policy editor.
- Ensure that the policy is selected and enabled in the appropriate profiles.
- Ensure that the relevant profiles are enabled.
- If a policy is included in one or more group profiles, ensure that the profiles include the appropriate user groups.

Viewing redaction policies

To view restriction filters go to the **Administration** > **Profiles** page. Then, click the **Redaction policies** tab. The list of existing policies is displayed, with the following details for each policy:

- Policy Name
- Data types
- Tested
- Created (date)
- Created by
- Last modified (date)
- Last modified by

Editing a redaction policy

It is important to note that the main policy should never be modified unless done with the help of support. The main policy is responsible for combining all policies together. It also ensures that the required properties are always included, and should not be removed as they are used by the UI to render screens correctly.

1. From the main navigation menu, go to **Administration** > **Profiles**. Click the **Redaction policies** tab.
2. Find the policy that you want to edit. If needed, you can filter the list of policies to help you find a specific policy.
3. For that policy, click the *Options* (3-dot) menu and click **Edit selected**.
4. Complete any required changes and click **Save policy**.

Duplicating a redaction policy

1. From the main navigation menu, go to **Administration** > **Profiles**. Click the **Redaction policies** tab.
2. Find the policy that you want to duplicate. If needed, you can filter the list of policies to help you find a specific policy.
3. For that policy, click the *Options* (3-dot) menu and click **Duplicate**.
4. Change the policy as needed and click **Save profile**.

Deleting a redaction policy

1. From the main navigation menu, go to **Administration** > **Profiles**. Click the **Redaction policies** tab.
2. Find the policy that you want to delete. If needed, you can filter the list of policies to help you find a specific policy.
3. For that policy, click the *Options* (3-dot) menu and click **Delete**.

   A dialog requesting you to confirm the deletion is displayed. Click **Delete** to confirm and delete the policy.

   Note: You cannot delete a policy if is being used in the profile.

Example redaction policies

The following code samples shows the various different ways to define code filters in Rego:

Full JSON example

```
{
    "entity": {
      "_id": "imSB3oo-RcWqjFhX-55Qag",
      "name": "delaware10-le4-cs29.na.test.lab",
      "tags": [
        "1.3.6.1.4.1.9.1.217",
        "B510/3D32 3rd Floor Lab",
        "CiscoCat29xx",
        "joe@ibm.com"
      ],
      "businessCriticality": "Gold",
      "manual": 0,
      "cdmType": 2,
      "sysName": "delaware10-le4-cs29.na.test.lab",
      "services": "datalink(2)",
```

```
      "uniqueId": "Example:delaware10-le4-cs29.na.test.lab",
      "className": "CiscoCat29xx",
      "sysContact": "joe@ibm.com",
      "vertexType": "resource",
      "description": "Description text here",
      "entityTypes": [
        "host",
        "switch"
      ],
      "matchTokens": [
        "delaware10-le4-cs29.na.test.lab",
        "delaware10-le4-cs29",
        "entityid.23492249"
      ],
      "sysLocation": "B510/3D32 3rd Floor Lab",
      "sysObjectId": "1.3.6.1.4.1.9.1.217",
      "cdmAdminState": 0,
      "accessProtocol": "IPv4",
      "interfaceCount": 27,
      "isIpForwarding": "not-forwarding",
      "sysDescription": "System description",
      "accessIPAddress": "172.30.238.10",
      "entityChangeTime": "2019-07-19T15:17:34.000Z",
      "entityCreateTime": "2019-07-19T15:17:34.000Z",
      "accessScopeTokens": [
        "file.observer:itnm.txt"
      ],
      "lastBroadcastTime": "2024-03-21T11:11:33.927Z",
      "entityDiscoveryTime": "2019-07-19T15:04:42.000Z"
    }
  }
```

Equals filter example

```
output := input.entity if {
    input.entity.cdmType == 2
}
```

Is not equal filter example

```
output := input.entity if {
    not input.entity.cdmType == 2
}
```

Contains filter example

```
output := input.entity if {
    contains(input.entity.name, "ware")
}
```

Does not contain filter example

```
output := input.entity if {
    not contains(input.entity.name, "ware")
}
```

Starts with filter example

```
output := input.entity if {
    startswith(input.entity.name, "del")
}
```

Ends with filter example

```
output := input.entity if {
    endswith(input.entity.name, ".lab")
}
```

Empty filter example

```
output := input.entity if {
    not input.entity.cdmType
}
```

Is not empty filter example

```
output := input.entity if {
    input.entity.cdmType
}
```

Matches filter example

```
output := input.entity if {
    regex.match(`^delaware`, input.entity.name)
}
```

Does not match filter example

```
output := input.entity if {
    not regex.match(`^delaware`, input.entity.name)
}
```

Check value matches a value in a set filter example

Requires knowing the criticalities to allow

```
output := input.entity if {
    input.entity.businessCriticality in ["Gold", "Silver", "Bronze"]
}
```

Includes any filter example

Check a list of provided types

```
output := input.entity if {
    some type in input.entity.entityTypes
    type in ["host", "otherType", "otherType2"]
}
```

Includes all filter example

```
output := input.entity if {
    every type in input.entity.entityTypes {
        type in ["host", "switch"]
    }
}
```

Complex combination

```
output := input.entity if {
    # When dealing with a host
    "host" in input.entity.entityTypes

    # Must start with del to show properties
    startswith(input.entity.name, "del")

    # Must include the tag "joe@ibm.com"
    some type in input.entity.tags
    type in ["joe@ibm.com"]
}
```

Complex definition to handle many different types

```
output := input.entity if {
    # When dealing with a host
    "host" in input.entity.entityTypes

    # Must start with del to show properties
    startswith(input.entity.name, "del")

    # Must include the tag "joe@ibm.com"
    some type in input.entity.tags
    type in ["joe@ibm.com"]
} else := input.entity if {
    # When dealing with a people
    "people" in input.entity.entityTypes

    # Must start with del to show properties
    startswith(input.entity.team, "uk_")
} else := input.entity if {
    # When dealing with a vms
    "vm" in input.entity.entityTypes

    # Must start with del to show properties
    regex.match(`^delaware`, input.entity.name)
}
```

Setting your preferences

You can customize the console to work for you.

You can customize your:

- Home page
- Profile photo

Customizing the contents of your home page

You can customize the cards that are displayed on your home page and the links that are displayed in the **Quick navigation** section.

To customize your home page:

Click the **Settings** icon:

- To customize the cards on your home page:
    1. Select **Personalize cards**:
        - You can add new cards from the list of **Available cards**.
        - You can drag the cards in the **Overview** section to reorder them.
    2. Click **Save and close**.
- To customize the links in the **Quick navigation**:
    1. Select **Personalize navigation**:
        - You can select up to 10 links from the **Add links** list.
        - You can drag the links in the **Quick navigation** section to reorder them.
    2. Click **Save and close**.

Changing your profile photo

Your profile photo is displayed next to projects that you create, notifications about actions that you've taken, and more. Your profile photo can make it easier for your coworkers to identify content that you contributed to.

To change your profile photo:

1. Click your avatar in the toolbar.
2. Click **Profile and settings**.
3. Click the **Replace photo** icon next to the current image.
4. Browse for or drag and drop the image that you want to use. The image can be up to 1 MB and must be a JPG or PNG file.
5. Click **Save**.

You might need to refresh your browser for the change to take effect.

Secret rotation

Find out about the secure operation of the following operators, and the procedure you must follow if you update secrets that are owned or used by these operators.

- Issue Resolution Core operator secret usage
- Issue Resolution Lifecycle operator secret usage

Issue Resolution Core operator secret usage

Find out about the secure operation of the **`Issue Resolution Core`** operator.

Secrets owned by the Issue Resolution Core operator

The **`Issue Resolution Core`** operator's secrets are created automatically when IBM Cloud Pak® for AIOps is installed. Alternatively, you can manually create some or all of these secrets before IBM Cloud Pak for AIOps is installed, with values of your own choosing. If these secrets exist before IBM Cloud Pak for AIOps is installed, then they are not modified (although new properties might be added).

The following table describes the secrets that the **`Issue Resolution Core`** operator owns. These secrets contain credentials, encryption keys, and other secrets. Some values can be changed after installation, which might necessitate restarting pods that have access to these secrets, or require additional steps to run. Details are provided in the table.

Table 1. Secrets that belong to the Issue Resolution Core operator

Secret	Property	Type	Purpose	Rotation	Example commands
`aiops-ir-core-cem-users-secret`	**`clientid`**	hex15	OAuth client ID	Change the secret. **`aiops-akora-ui`**, **`aiops-ir-core-cem-users`**, **`aiops-ir-core-esarchiving`**, **`aiops-ir-core-ncodl-api`**, **`aiops-ir-core-rba-rbs`** and **`aiops-ir-core-usercfg`** will restart automatically.	`oc set data secret aiops-ir-core-cem-users-secret --from-literal=clientid=$(openssl rand --hex 15)`

Secret	Property	Type	Purpose	Rotation	Example commands
`aiops-ir-core-cem-users-secret`	`clientsecret`	hex15	OAuth client secret	Change the secret. `aiops-akora-ui`, `aiops-ir-core-cem-users`, `aiops-ir-core-esarchiving`, `aiops-ir-core-ncodl-api`, `aiops-ir-core-rba-rbs` and `aiops-ir-core-usercfg` will restart automatically.	`oc set data secret aiops-ir-core-cem-users-secret --from-literal=clientsecret=$(openssl rand --hex 15)`
`aiops-ir-core-cem-users-secret`	`oidcclientid`	hex15	Not used in AIOps	This secret can be changed without effect.	
`aiops-ir-core-cem-users-secret`	`oidcclientsecret`	hex15	Not used in AIOps	This secret can be changed without effect.	
`aiops-ir-core-cem-users-secret`	`password`	hex15	Basic auth for `https://aiops-ir-core-cem-users:6002`	Change the secret. `aiops-akora-ui`, `aiops-ir-core-cem-users`, `aiops-ir-core-esarchiving`, `aiops-ir-core-ncodl-api`, `aiops-ir-core-rba-rbs` and `aiops-ir-core-usercfg` will restart automatically.	`oc set data secret aiops-ir-core-cem-users-secret --from-literal=password=$(openssl rand --hex 15)`
`aiops-ir-core-cem-users-secret`	`username`	hex15	Basic auth for `https://aiops-ir-core-cem-users:6002`	Change the secret. `aiops-akora-ui`, `aiops-ir-core-cem-users`, `aiops-ir-core-esarchiving`, `aiops-ir-core-ncodl-api`, `aiops-ir-core-rba-rbs` and `aiops-ir-core-usercfg` will restart automatically.	`oc set data secret aiops-ir-core-cem-users-secret --from-literal=username=$(openssl rand --hex 15)`
`aiops-ir-core-couchdb`	`admin_password`	hex15	Basic auth for `https://aiops-ir-core-couchdb-api`	Change the secret and restart `c-example-couchdbcluster-m`	`oc set data secret aiops-ir-core-couchdb --from-literal=admin_password=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`
`aiops-ir-core-couchdb`	`couchdb_auth_secret`	hex15	Internal CouchDB use	Change the secret and restart `c-example-couchdbcluster-m`	`oc set data secret aiops-ir-core-couchdb --from-literal=couchdb_auth_secret=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`
`aiops-ir-core-couchdb`	`csrf_secret`	hex15	Internal CouchDB use	Change the secret and restart `c-example-couchdbcluster-m`	`oc set data secret aiops-ir-core-couchdb --from-literal=csrf_secret=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`
`aiops-ir-core-couchdb`	`erlang_cookie`	hex15	Internal CouchDB use	Change the secret and restart `c-example-couchdbcluster-m`	`oc set data secret aiops-ir-core-couchdb --from-literal=erlang_cookie=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`
`aiops-ir-core-couchdb`	`ibm_password`	hex15	Internal CouchDB use	Change the secret and restart `c-example-couchdbcluster-m`	`oc set data secret aiops-ir-core-couchdb --from-literal=ibm_password=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`
`aiops-ir-core-model-secret`	`hkeyname`	ts	Identifier for `hkeyvalue` value	See Key rotation.	
`aiops-ir-core-model-secret`	`hkeyvalue`	hex16	HMAC key for sensitive data in CouchDB	See Key rotation.	
`aiops-ir-core-model-secret`	`oldhkeys`	hex16	Previous HMAC keys for sensitive data in CouchDB	See Key rotation.	
`aiops-ir-core-model-secret`	`keyname`	ts	Identifier for `keyvalue` value	See Key rotation.	
`aiops-ir-core-model-secret`	`keyvalue`	hex32	Encryption key for sensitive data in CouchDB	See Key rotation.	
`aiops-ir-core-model-secret`	`oldkeys`	hex32	Previous encryption keys for sensitive data in CouchDB	See Key rotation.	

Secret	Property	Type	Purpose	Rotation	Example commands
aiops-ir-core-ncodl-api-secret	password	hex15	Basic auth for https://aiops-ir-core-ncodl-api:10011	Change the secret. - Restart cp4waiops-metricsprocessor. - aiops-ir-core-ncodl-api restarts automatically.	oc set data secret aiops-ir-core-ncodl-api-secret --from-literal=password=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor
aiops-ir-core-ncodl-api-secret	username	hex15	Basic auth for https://aiops-ir-core-ncodl-api:10011	Change the secret. - Restart cp4waiops-metricsprocessor. - aiops-ir-core-ncodl-api restarts automatically.	oc set data secret aiops-ir-core-ncodl-api-secret --from-literal=username=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor
aiops-ir-core-ncodl-std-secret	password	hex15	Basic auth for https://aiops-ir-core-ncodl-std:10011	Change the secret. -Restart aimanager-aio-ai-platform-api-server, aimanager-aio-change-risk, aimanager-aio-chatops-orchestrator, aimanager-aio-chatops-slack-integrator, aimanager-aio-chatops-teams-integrator, aimanager-aio-controller, aimanager-aio-cr-api, aimanager-aio-log-anomaly-detector, aimanager-aio-log-anomaly-feedback-learning, aimanager-aio-oob-recommended-actions, aimanager-aio-similar-incidents-service. - aiops-ir-analytics-probablecause, aiops-ir-core-api, aiops-ir-core-ncodl-api, aiops-ir-core-ncodl-if, aiops-ir-core-ncodl-jobmgr, aiops-ir-core-ncodl-std, and aiops-ir-core-rba-as restart automatically.	oc set data secret aiops-ir-core-ncodl-std-secret --from-literal=password=$(openssl rand -hex 15) && oc delete pod -l app.kubernetes.io/component=ai-platform-api-server && oc delete pod -l app.kubernetes.io/component=change-risk && oc delete pod -l app.kubernetes.io/component=chatops-orchestrator && oc delete pod -l app.kubernetes.io/component=chatops-slack-integrator && oc delete pod -l app.kubernetes.io/component=chatops-teams-integrator && oc delete pod -l app.kubernetes.io/component=controller,app.kubernetes.io/name=aio && oc delete pod -l app.kubernetes.io/component=cr-api && oc delete pod -l app.kubernetes.io/component=log-anomaly-detector && oc delete pod -l app.kubernetes.io/component=log-anomaly-feedback-learning && oc delete pod -l app.kubernetes.io/component=oob-recommended-actions && oc delete pod -l app.kubernetes.io/component=similar-incidents-service
aiops-ir-core-ncodl-std-secret	username	fixed	Basic auth for https://aiops-ir-core-ncodl-std:10011	The username is system. Do not change, included for future flexibility.	
aiops-ir-core-omni-secret	OMNIBUS_PROBE_PASSWORD	hex15	Probe password for ObjectServer	See ObjectServer passwords.	
aiops-ir-core-omni-secret	OMNIBUS_ROOT_PASSWORD	hex15	Root password for ObjectServer	See ObjectServer passwords.	
aiops-ir-core-postgresql	caCertificate	x509	Root CA certificate for PostgreSQL	This value must be copied from <instance>-edb-secret.	
aiops-ir-core-postgresql	dbname	fixed	Name of event database in PostgreSQL	The database name is aiops_irb. This secret must not be changed and is included for future flexibility.	
aiops-ir-core-postgresql	host	hostname	Service name of PostgreSQL cluster	This value must be copied from <instance>-edb-secret.	

Secret	Property	Type	Purpose	Rotation	Example commands
aiops-ir-core-postgresql	**password**	hex15	Password for PostgreSQL event runtime user	Change the secret. - **aiops-ir-core-ncodl-jobmgr** and **aiops-ir-core-ncodl-std** restart automatically.	**oc set data secret aiops-ir-core-postgresql --from-literal=password=$(openssl rand -hex 15)**
aiops-ir-core-postgresql	**port**	port	Port for PostgreSQL cluster service	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql	**schema**	fixed	Name of event schema in PostgreSQL	The schema name is **aiops_irb**. This secret must not be changed and is included for future flexibility.	
aiops-ir-core-postgresql	**username**	fixed	Name of PostgreSQL event runtime user	The username is **aiops_irb**. This secret must not be changed and is included for future flexibility.	
aiops-ir-core-postgresql-admin	**caCertificate**	x509	Root CA certificate for PostgreSQL	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-admin	**dbname**	fixed	Name of event database in PostgreSQL	The database name is **aiops_irb**. This secret must not be changed and is included for future flexibility.	
aiops-ir-core-postgresql-admin	**host**	hostname	Service name of PostgreSQL cluster	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-admin	**password**	hex15	Password for PostgreSQL event admin user	Change the secret. No pod restarts are required.	**oc set data secret aiops-ir-core-postgresql-admin --from-literal=password=$(openssl rand -hex 15)**
aiops-ir-core-postgresql-admin	**port**	port	Port for PostgreSQL cluster service	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-admin	**schema**	fixed	Name of event schema in PostgreSQL	The schema name is **aiops_irb**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-admin	**username**	fixed	Name of PostgreSQL event admin user	The username is **aiops_irb_admin**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc	**caCertificate**	x509	Root CA certificate for PostgreSQL	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-uc	**dbname**	fixed	Name of event database in PostgreSQL	The database name is **aiops_irbuc**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc	**host**	hostname	Service name of PostgreSQL cluster	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-uc	**password**	hex15	Password for PostgreSQL event runtime user	Change the secret. - **aiops-ir-core-usercfg** restarts automatically.	**oc set data secret aiops-ir-core-postgresql-uc --from-literal=password=$(openssl rand -hex 15)**
aiops-ir-core-postgresql-uc	**port**	port	Port for PostgreSQL cluster service	This value must be copied from **<instance>-edb-secret**.	

Secret	Property	Type	Purpose	Rotation	Example commands
aiops-ir-core-postgresql-uc	**schema**	fixed	Name of event schema in PostgreSQL	The schema name is **aiops_irbuc**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc	**username**	fixed	Name of PostgreSQL event runtime user	The username is **aiops_irbuc**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc-admin	**caCertificate**	x509	Root CA certificate for PostgreSQL	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-uc-admin	**dbname**	fixed	Name of event database in PostgreSQL	The database name is **aiops_irbuc**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc-admin	**host**	hostname	Service name of PostgreSQL cluster	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-uc-admin	**password**	hex15	Password for PostgreSQL event admin user	Change the secret. No pod restarts are required.	**oc set data secret aiops-ir-core-postgresql-uc-admin --from-literal=password=$(openssl rand -hex 15)**
aiops-ir-core-postgresql-uc-admin	**port**	port	Port for PostgreSQL cluster service	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-uc-admin	**schema**	fixed	Name of event schema in PostgreSQL	The schema name is **aiops_irbuc**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc-admin	**username**	fixed	Name of PostgreSQL event admin user	The username is **aiops_irbuc_admin**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-rba-devops-secret	**password**	hex15	Basic auth for **https://aiops-ir-core-rba-as:3080** and **https://aiops-ir-core-rba-rbs:3005**	Change the secret. - Restart **cp4waiops-metricsprocessor**. - **aiops-ir-core-rba-as** and **aiops-ir-core-rba-rbs** restart automatically.	**oc set data secret aiops-ir-core-rba-devops-secret --from-literal=password=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor**
aiops-ir-core-rba-devops-secret	**username**	hex15	Basic auth for **https://aiops-ir-core-rba-as:3080** and **https://aiops-ir-core-rba-rbs:3005**	Change the secret. - Restart **cp4waiops-metricsprocessor**. - **aiops-ir-core-rba-as** and **aiops-ir-core-rba-rbs** restart automatically.	**oc set data secret aiops-ir-core-rba-devops-secret --from-literal=username=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor**
aiops-ir-core-rba-jwt-secret	**secret**	hex15	Key used to sign runbook JSON Web Tokens	Change the secret. - Restart **aiops-akora-ui**, **aiops-ir-core-rba-as**, **aiops-ir-core-rba-rbs**, **connector-orchestrator**.	**oc set data secret aiops-ir-core-rba-jwt-secret --from-literal=secret=$(openssl rand -hex 15) && oc delete pod -l app.kubernetes.io/component=aiops-aiops-akora-ui && oc delete pod -l app.kubernetes.io/component=rba-as && oc delete pod -l app.kubernetes.io/component=rba-rbs && oc delete pod -l app=connector-orchestrator**
aiops-ir-core-usercfg-creds	password	hex15	Basic auth for **https://aiops-ir-core-usercfg:10011**	Change the secret. **aiops-ir-core-ncodl-api** and **aiops-ir-core-usercfg** will restart automatically.	**oc set data secret aiops-ir-core-usercfg-creds --from-literal=password=$(openssl rand --hex 15)**
aiops-ir-core-usercfg-creds	username	hex15	Basic auth for **https://aiops-ir-core-usercfg:10011**	Change the secret. **aiops-ir-core-ncodl-api** and **aiops-ir-core-usercfg** will restart automatically.	**oc set data secret aiops-ir-core-usercfg-creds --from-literal=username=$(openssl rand --hex 15)**

Where **<instance>** is the name of your IBM Cloud Pak for AIOps instance, for example **ibm-cp-aiops**.

The **Type** for each secret in the table has one of the following entries:

- hex15, hex16, hex32 - 15, 16, or 32-byte number encoded in hexadecimal (30, 32 or 64 hex digits).
- ts - timestamp value that is used as an identifier for another value. The timestamp is not significant and alternative strings can be used. Some restrictions on the characters might exist. ASCII alphanumeric characters are recommended.
- fixed - do not change this value. The value is included in a secret for future flexibility.
- x509 - an X.509 certificate
- hostname - the DNS hostname of a service.
- port - a TCP port number

Secrets that are used by the Issue Resolution Core operator

The following table describes secrets that are owned by another operator but that are used by the Issue Resolution Core operator. The table shows the Issue Resolution Core components that need to be restarted if these secrets are rotated.

Table 2. Secrets that are used by the Issue Resolution Core operator

Secret	Owner	Purpose	Automatically restarted
`aiops-topology-cassandra-auth-secret`	`asm-operator`	Cassandra contact details	`aiops-ir-core-archiving`, `aiops-ir-core-esarchiving`
`<instance>-couchdb-secret`	`ibm-aiops-orchestrator`	CouchDB contact details	`aiops-ir-core-cem-users`, `aiops-ir-core-rba-as`, `aiops-ir-core-rba-rbs`
`<instance>-es-admin-user-connection-secret`	`ibm-aiops-orchestrator`	Elasticsearch contact details	`aiops-ir-core-datarouting`, `aiops-ir-core-ncodl-api`
`<instance>-kafka-secret`	`ibm-aiops-orchestrator`	Kafka contact details	`aiops-ir-core-api`, `aiops-ir-core-archiving`, `aiops-ir-core-datarouting`, `aiops-ir-core-esarchiving`, `aiops-ir-core-ncodl-if`, `aiops-ir-core-ncodl-jobmgr`, `aiops-ir-core-ncodl-std`, `aiops-ir-core-rba-rbs`
`<instance>-redis-secret`	`ibm-aiops-orchestrator`	Redis contact details	`aiops-ir-core-cem-users`, `aiops-ir-core-esarchiving`, `aiops-ir-core-rba-rbs`
`internal-nginx-svc-tls`	`ibm-zen-operator`	Trusted CA for Zen	`aiops-ir-core-cem-users`
`internal-tls`	`ibm-zen-operator`	Trusted CA for Zen	`aiops-ir-core-cem-users`
`zen-service-broker-secret`	`ibm-zen-operator`	Token for accessing Zen API	`aiops-ir-core-cem-users`

Where **<instance>** is the name of your IBM Cloud Pak for AIOps instance, for example **ibm-cp-aiops**.

Key rotation

Warning: The secret **aiops-ir-core-model-secret** contains an encryption key and a HMAC key that is used to protect sensitive data that is stored in CouchDB. Care must be taken when rotating these keys since data that is protected with previous keys cannot be recovered if the keys are lost.

Replacement values for **keyvalue** must be 16 bytes and in hexadecimal format (32 digits). Replacement values for **hkeyvalue** must be 32 bytes and in hexadecimal format (64 digits).

Each key has an associated identifier. Encrypted data is stored along with the identifier of the key that was used to protect it. The identifier must be changed when and only when the key is changed. When the operator creates a new key, it uses the current time as the identifier (for example **2022-06-13T10:58:10Z**). This is a convention but other string formats may be used. If another format is used, there might be some restrictions on the characters that can be used. ASCII alphanumeric characters are recommended.

When rotating keys, all old (key, identifier) pairs must be stored in the **oldhkeys** and **oldkeys** properties so that data that is protected with the old keys can be used. The current key is always used when data is stored or modified. The following examples show the use of the old keys' properties.

1. Secret created by the operator.

```
{
  "hkeyname": "2022-06-13T10:58:10Z",
  "hkeyvalue": "2905c99e41fd6594de9e8632382da185eec94cea545cbd96082552274a454d2b",
  "keyname": "2022-06-13T10:58:10Z",
  "keyvalue": "5cc068f45904c0ff998213941684790e"
}
```

2. Secret after the first rotation.

   The identifiers **myfirsthkey** and **myfirstkey** were used. Time based could have been used.

```
{
  "hkeyname": "myfirsthkey",
  "hkeyvalue": "7539eb64abdae09a500659ca0593140babd00046b175c3a83e3ed28f8d79daca",
  "oldhkeys": "{\"keys\":[{\"name\":\"2022-06-
13T10:58:10Z\",\"value\":\"2905c99e41fd6594de9e8632382da185eec94cea545cbd96082552274a454d2b\"}]}",
  "keyname": "myfirstkey",
  "keyvalue": "211812c9750778c0175bca4a11cc1e3b",
  "oldkeys": "{\"keys\":[{\"name\":\"2022-06-13T10:58:10Z\",\"value\":\"5cc068f45904c0ff998213941684790e\"}]}"
}
```

3. Secret after the second rotation.

The identifiers **mysecondhkey** and **mysecondkey** were used. Time based could have been used.

```
{
  "hkeyname": "mysecondhkey",
  "hkeyvalue": "07a290a68d6ff7eb45b7febbfd8c5079da5ee40c068f2e1bb87293c22c39b712",
  "oldhkeys": "{\"keys\":[{\"name\":\"2022-06-
13T10:58:10Z\",\"value\":\"2905c99e41fd6594de9e8632382da185eec94cea545cbd96082552274a454d2b\"},
{\"name\":\"myfirsthkey\",\"value\":\"7539eb64abdae09a500659ca0593140babd00046b175c3a83e3ed28f8d79daca\"}]}",
  "keyname": "mysecondkey",
  "keyvalue": "f01b36adba59c50841a1444cee8eb09e",
  "oldkeys": "{\"keys\":[{\"name\":\"2022-06-13T10:58:10Z\",\"value\":\"5cc068f45904c0ff998213941684790e\"},
{\"name\":\"myfirstkey\",\"value\":\"211812c9750778c0175bca4a11cc1e3b\"}]}"
}
```

If the **oldhkeys** and **oldkeys** properties exist, they must hold the JSON representation of an object with a single property (the name does not matter, this example used **keys**) that has an array of objects. Each of the inner objects has a **name** and a **value** property holding the identifier and value of a previous key.

```
oc delete pod -l app.kubernetes.io/component=rba-as && oc delete pod -l app.kubernetes.io/component=rba-rbs
```

ObjectServer passwords

Use the following steps to change the ObjectServer password. Ensure that you retain a copy of the original password until these steps are complete.

1. Run the following command to obtain the current passwords from the **aiops-ir-core-omni-secret** secret.

   For example:

   ```
   oc get secret aiops-ir-core-omni-secret -o json | jq -r '.data.OMNIBUS_PROBE_PASSWORD|@base64d'`
   oc get secret aiops-ir-core-omni-secret -o json | jq -r '.data.OMNIBUS_ROOT_PASSWORD|@base64d'`
   ```

2. Open a session on the primary ObjectServer pod.

   ```
   oc rsh aiops-ir-core-ncoprimary-0
   ```

3. Log in to the OMNIbus query tool **nco_sql**, and supply the root password when prompted.

   ```
   $NCHOME/omnibus/bin/nco_sql -u root -s AGG_P_C
   ```

4. If you want to change the root password, enter the following command.

   ```
   alter user 'root' set password '<NEWPASSWORD>';
   go
   ```

   Where **<NEWPASSWORD>** is the password that you want to set.

5. If you also want to change the probe password, enter the following command.

   ```
   alter user 'probe' set password '<PROBE_PASSWORD>';
   go
   ```

   Where **<PROBE_PASSWORD>** is the password that you want to set.

6. Test the new root password by entering **quit** to leave the SQL prompt, and then log in again and supply the new root password when prompted.

   ```
   $NCHOME/omnibus/bin/nco_sql -u root -s AGG_P_C
   ```

7. Exit the **oc rsh** session.

8. Wait for 60 seconds to ensure that the password updates have been transferred to the backup ObjectServer.

9. Remote shell into the backup ObjectServer, and test the new root password.

   ```
   oc rsh -c objserv aiops-ir-core-ncobackup-0`
   ```

10. Log in to the OMNIbus query tool **nco_sql**, and supply the root password when prompted.

    ```
    $NCHOME/omnibus/bin/nco_sql -u root -s AGG_B_C
    ```

11. Exit the **oc rsh** session.

12. Update the **aiops-ir-core-omni-secret** with one or more new passwords.

    For example,

    ```
    oc set data secret aiops-ir-core-omni-secret --from-literal=OMNIBUS_PROBE_PASSWORD=<PROBE_PASSWORD>
    oc set data secret aiops-ir-core-omni-secret --from-literal=OMNIBUS_ROOT_PASSWORD=<NEWPASSWORD>
    ```

    Where

    - **<NEWPASSWORD>** is the new root password
    - **<PROBE_PASSWORD>** is the new probe password

13. If you changed the root password, run the following command to restart **aiops-ir-core-ncodl-if**

    ```
    oc delete pod -l app.kubernetes.io/component=ncodl-if
    ```

Issue Resolution Lifecycle operator secret usage

Find out about the secure operation of the **Issue Resolution Lifecycle** operator.

Secrets owned by the Issue Resolution Lifecycle operator

The **Issue Resolution Lifecycle** operator's secrets are created automatically when IBM Cloud Pak® for AIOps is installed. Alternatively, you can manually create some or all of these secrets before IBM Cloud Pak for AIOps is installed, with values of your own choosing. If these secrets exist before IBM Cloud Pak for AIOps is installed, then they are not modified (although new properties might be added).

The following table describes the secrets that the **Issue Resolution Lifecycle** operator owns. These secrets contain credentials, encryption keys, and other secrets. Some values can be changed after installation, which might necessitate restarting pods that have access to these secrets, or require additional steps to run. Details are provided in the table.

Table 1. Secrets that belong to the Issue Resolution Lifecycle operator

Secret	Property	Type	Purpose	Rotation	Example commands
aiops-ir-lifecycle-flink-admin-user	**password**	hex15	Basic auth for **https://aiops-ir-lifecycle-flink-api:9081**	Change the secret. **aiops-ir-lifecycle-flink** will restart automatically.	**oc set data secret aiops-ir-lifecycle-flink-admin-user --from-literal=password=$(openssl rand --hex 15)**
aiops-ir-lifecycle-flink-admin-user	**username**	fixed	Basic auth for **https://aiops-ir-lifecycle-flink-api:9081**	The user name is flink-admin, this should not be changed and is included for future flexibility.	
aiops-ir-lifecycle-flink-internal-keystore	**password**	hex15	Password used for key store within pods.	Change the secret. **aiops-ir-lifecycle-flink** and **aiops-ir-lifecycle-flink-taskmanager** will restart automatically.	**oc set data secret aiops-ir-lifecycle-flink-internal-keystore --from-literal=password=$(openssl rand --hex 15)**
aiops-ir-lifecycle-flink-internal-truststore	**password**	hex15	Password used for trust store within pods.	Change the secret. **aiops-ir-lifecycle-flink** and **aiops-ir-lifecycle-flink-taskmanager** will restart automatically.	**oc set data secret aiops-ir-lifecycle-flink-internal-truststore --from-literal=password=$(openssl rand --hex 15)**
aiops-ir-lifecycle-flink-rest-keystore	**password**	hex15	Password used for key store within pods.	Change the secret. **aiops-ir-lifecycle-flink** and **aiops-ir-lifecycle-flink-taskmanager** will restart automatically.	**oc set data secret aiops-ir-lifecycle-flink-rest-keystore --from-literal=password=$(openssl rand --hex 15)**
aiops-ir-lifecycle-flink-rest-truststore	**password**	hex15	Password used for trust store within pods.	Change the secret. **aiops-ir-lifecycle-flink** and **aiops-ir-lifecycle-flink-taskmanager** will restart automatically.	**oc set data secret aiops-ir-lifecycle-flink-rest-truststore --from-literal=password=$(openssl rand --hex 15)**
aiops-ir-lifecycle-flink-truststore	**password**	hex15	Password used for trust store within pods.	Change the secret. **aiops-ir-lifecycle-flink** and **aiops-ir-lifecycle-flink-taskmanager** will restart automatically.	**oc set data secret aiops-ir-lifecycle-flink-truststore --from-literal=password=$(openssl rand --hex 15)**
aiops-ir-lifecycle-policy-registry-svc	**password**	hex15	Basic authentication for **https://aiops-ir-lifecycle-policy-registry-svc:5601**	Change the secret. - Restart **cp4waiops-metricsprocessor**. - **aiops-ir-lifecycle-policy-registry-svc** restarts automatically.	**oc set data secret aiops-ir-lifecycle-policy-registry-svc --from-literal=password=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor**
aiops-ir-lifecycle-policy-registry-svc	**username**	fixed	Basic authentication for **https://aiops-ir-lifecycle-policy-registry-svc:5601**	The username is **system** and must not be changed. It is included for future flexibility.	

The **Type** for each secret in the table has one of the following entries:

- hex15 - 15-byte number encoded in hexadecimal (30 hex digits).
- fixed - do not change this value. The value is included in a secret for future flexibility.

Secrets that are used by the Issue Resolution Lifecycle operator

The following table describes secrets that are owned by another operator but that are used by the Issue Resolution Lifecycle operator. The table shows the Issue Resolution Lifecycle components that are automatically restarted if these secrets are rotated, no manual component restarts are required.

Table 2. Secrets that are used by the Issue Resolution Lifecycle operator

Secret	Owner	Purpose	Automatically restarted
`aimanager-ibm-minio-access-secret`	`AIManager/aimanager`	Credentials for Minio S3 storage	`aiops-ir-lifecycle-flink`, `aiops-ir-lifecycle-flink-taskmanager`
`aiops-ir-analytics-ai-datalayer-system-secret`	`AIOpsAnalyticsOrchestrator`	Credentials for AI datalayer	`aiops-ir-lifecycle-flink`, `aiops-ir-lifecycle-flink-taskmanager`
`aiops-topology-asm-credentials`	`ASM/aiops-topology`	Credentials for `https://aiops-topology-topology.aiops.svc:8080`	`aiops-ir-lifecycle-flink`, `aiops-ir-lifecycle-flink-taskmanager`
`aiops-topology-cassandra-auth-secret`	`ASM/aiops-topology`	Credentials for Cassandra	`aiops-ir-lifecycle-flink-taskmanager`, `aiops-ir-lifecycle-policy-registry-svc`
`cp4waiops-cartridge-es-auth`	`Installation`	Credentials for Elasticsearch	`aiops-ir-lifecycle-flink`, `aiops-ir-lifecycle-flink-taskmanager`
`iaf-system-elasticsearch-es-ss-cacert-kp`	`Certificate/iaf-system-elasticsearch-es-ss-ca`	CA certificate for Elasticsearch	`aiops-ir-lifecycle-flink`, `aiops-ir-lifecycle-flink-taskmanager`
`<instance>-es-admin-user-connection-secret`	`Installation`	Binding secret for Elasticsearch	`aiops-ir-lifecycle-datarouting`, `aiops-ir-lifecycle-policy-registry-svc`
`<instance>-kafka-secret`	`Installation`	Binding secret for Kafka	`aiops-ir-lifecycle-datarouting`, `aiops-ir-lifecycle-flink`, `aiops-ir-lifecycle-flink-taskmanager`, `aiops-ir-lifecycle-policy-registry-svc`
`zen-service-broker-secret`	`ibm-zen-operator`	Token for accessing Zen API	`aiops-ir-lifecycle-flink`, `aiops-ir-lifecycle-flink-taskmanager`

Where `<instance>` is the name of your IBM Cloud Pak for AIOps instance, for example `ibm-cp-aiops`.

Administering CouchDB data storage

Manage disk space used by runbook executions to avoid outages when the available disk space has been exhausted.

You can offload and delete historic runbook executions to reduce database footprint and increase database performance. Offloaded runbook executions are deleted from the production database and stored on a file system instead to preserve them for future audit purposes. They can be stored either in JSON or HTML format. A script for offloading runbook executions is included in the product and can be run by using the fully automated sample runbook **Example: Automatically reduce amount of data in runbook data stores**. Running this fully automated runbook has the negative side effect that the API key (which is contained in the parameters of the fully automated runbook) is stored and can be viewed within the runbook execution details. This description shows how you can avoid this limitation and still use the script that is available at the core of the fully automated runbook.

Runbook executions can be offloaded to store them somewhere else, in a different format (JSON or HTML), or to reduce the amount of data stored in the database. This is especially important if either the database is getting short of disk space, or a performance degradation that is caused by too much data is becoming noticeable. It is also a good practice to regularly offload runbook executions that exceed a certain age and are therefore no longer required on the production system.

When you offload and delete runbook execution records, the referenced action execution records are included in the offloading output and deleted as well. However, additional action execution records might remain for runbook executions that were already deleted separately, or were executed outside of a runbook context. You can use the following instructions on how to offload and delete those action execution records. Make sure to delete the action execution records only after the runbook execution records for the same time frame have been offloaded and deleted. Otherwise, the runbook execution records contain broken links to action execution records that have been deleted already.

Note:

- Once runbook executions are offloaded and deleted, they are no longer accessible from the runbook history page and are not reflected in the aggregated statistics of their runbook anymore.
- The size of the database does not shrink immediately after the runbook executions have been deleted, but only after the next CouchDB database compaction has been performed by CouchDB itself.

It is recommended to set up a scheduled task that offloads runbook execution data is regularly. For your planning you should consider the following aspects:

- What is the maximum size of the persistent volume that hosts the database where the runbook execution data is stored? For example, 20 GB.
- What is the average size of a single runbook execution in your environment? For simple runbooks, you can assume a value of about 15 KB per runbook execution.
- How many runbooks are executed per day?

Create an initial schedule for offloading the runbooks based on these values, allowing some buffer for days with higher load. Observe the behavior for some time, and tune the frequency of the scheduled task or the parameters of the runbook to match your needs.

The offloading script must run on a Linux system that meets the following requirements:

- Network access to the OpenShift or Linux® cluster where IBM Cloud Pak® for AIOps is installed.
- The `curl` tool must be installed.
- The `jq` tool must be installed. Starting with RHEL 8.0, `jq` can be installed by using the AppStream repository. On earlier RHEL versions, `jq` can be installed by following the instructions on https://stedolan.github.io/jq/.
- Some free disk space must exist to store copies of the offloaded runbook executions. See the FOLDER parameter for more information.
- The `base64` encoding tool must be installed.

The offloading script requires several environment variables to be set:

- API: The Runbook API URL as described in Automation (runbook) API. For example, set the value of the API parameter to `https://cpd-cp4waiops.apps.mycomp.com/aiops/api/story-manager/rba/v1`.
- APIKEY_NAME: The user ID that you used to log in when creating an API key. As described in Accessing APIs, section *Get access token by using your API key*, step *Generate your platform API key*.
- APIKEY_PW: The API key password that you created when following the instructions in Accessing APIs, section *Get access token by using your API key*, step *Generate your platform API key*.
- FOLDER: The fully qualified path name of an existing directory on the Linux system that will be used for storing the offloaded runbooks.
- MODE: Used to specify a format for the offloaded runbook executions. You can choose between `html` and `json`.
- DELETE: Whether to delete the offloaded runbook executions from the runbook database after successful offload. Recommended to be set to `false` while performing early tests, but set to `true` once you are ready to delete documents and reduce the size of the database.
- OLDER_THAN_MONTHS: A number of months to specify the timeframe for offloading runbook executions. Only runbook executions older than the specified number of months are offloaded. If you set this value to `0` then all runbook executions that were created before today will be offloaded. You can set this parameter to `-1` and instead use the parameters AFTER and BEFORE if you need to have more granular control on the time period.
- AFTER: Only offload execution records that are started after this timestamp. You can specify a date like `2022-03-31` or a timestamp in the moment.js format, for example, `2022-03-31T12:59:59`. If you want to use this parameter, the parameter OLDER_THAN_MONTHS must be set to the empty string.
- BEFORE: Only offload execution records that are started before this timestamp. You can specify a date like `2022-04-04` or a timestamp in the moment.js format, for example, `2022-04-04T12:59:59`. If you want to use this parameter, the parameter OLDER_THAN_MONTHS must be set to the empty string.

1. One-time set up to create the script

1. Create an API key as described in Accessing APIs, section *Get access token by using your API key*, step *Generate your platform API key*.
2. If it does not exist already, create the SSH integration as described in Creating SSH integrations. You do not need to define a jumpserver and you do not need to copy the SSH key to any target system for the purpose of this procedure.
3. Load the sample runbooks as described in Load and reset example runbooks.
4. Switch to the **Actions** tab.
5. Select the action **Example: Reduce amount of data in runbook data stores** and click **Edit**.
6. Copy the content of the **Script** field to the clipboard.
7. Log in to the Linux system and change to the directory where you want the script to be stored.
8. Create a file called "offloadRunbookExecutions.sh" and paste the content of the clipboard into this file.
9. Make the file "offloadRunbookExecutions.sh" executable, for example, run the following command:

   ```
   chmod a+x offloadRunbookExecutions.sh
   ```

10. Create a file called "offload.config" and paste the following content into this file:

    ```
    export API="https://cpd-cp4waiops.apps.mycomp.com/aiops/api/story-manager/rba/v1"
    export APIKEY_NAME="<api_key_user_id>"
    export APIKEY_PW="<api_key_password>"
    export FOLDER=$(pwd)
    export MODE=json
    export DELETE=false
    export OLDER_THAN_MONTHS=-1
    export AFTER="2022-03-01"
    export BEFORE="2022-04-13T12:00:00"
    #export CURL_OPTIONS="-k"
    ```

11. Adjust the values of the environment variables as needed (see the variable descriptions listed earlier) and save the file.

2. Test the script

1. Run the following command to load the environment variables into the current shell environment:

   ```
   . ./offload.config
   ```

2. Run the script:

   ```
   ./offloadRunbookExecutions.sh
   ```

3. If the action fails and you see the error `curl: \(60\) SSL certificate problem: self-signed certificate in certificate chain`, you can either replace the IBM Netcool Operations Insight server certificate with a trusted certificate, or enable the environment variable `CURL\_OPTIONS="-k"` to ignore the certificate errors.
4. If the action fails and you see the following error:

   ```
   ARBD0420E: Could not fetch list of runbook instances:
   curl: (22) The requested URL returned error: 500 Internal Server Error
   ```

   Then reduce the length of the time interval for offloading execution records (for example, by increasing the value of OLDER_THAN_MONTHS) and try again.
5. Review the results that were created by the script. If needed adjust the environment variables and repeat the procedure until you are fine with the results.

3. Schedule the script

1. Convert this script command to the format that is expected by your scheduling tool, and set up a schedule to regularly offload the runbook data.
2. Make sure that the DELETE environment variable is set to true once you are finished with the testing of the script and the schedule, and when you are ready to delete the offloaded runbook execution instances from the database.

Offloading and deleting remaining action execution records

Once you have deleted the runbook execution records for a particular time frame you should also check if there are remaining action execution records (also called action instances) for that time frame, and delete them as well.

Step 1: Prepare and verify the API URL and the credentials Adjust the values of the environment variables as needed (see the variable descriptions listed earlier) and run the following commands:

```
export API="https://cpd-cp4waiops.apps.mycomp.com/aiops/api/story-manager/rba/v1"
export APIKEY_NAME="<api_key_user_id>"
export APIKEY_PW="<api_key_password>"
export CURL_AUTHENTICATION=$(echo -n "${APIKEY_NAME}:${APIKEY_PW}" | base64 -w 0 | echo "Authorization: ZenApiKey $(cat
-)")
export CURL_OPTIONS="-ks"
curl "${CURL_OPTIONS}" -H "${CURL_AUTHENTICATION}" "${API}/status"
```

The command should run without any authentication errors, and the output should look similar to this (for the purpose of this task, you can ignore the value of the "connection" property, in particular any potential "errors" and "warnings" within the respective properties):

```
{"rbs":0,"as":0,"db":0,"connection":{"AWX":0,"IMPACT":2,"SCRIPT":0,"GITHUB":2},"errors":[],"warnings":[]}
```

Step 2: Get the remaining action execution records Adjust the "to" time stamp in the following command and run it to retrieve a list of the records that were created up to that date, and store their IDs in an environment variable:

```
curl "${CURL_OPTIONS}" -H "${CURL_AUTHENTICATION}" "${API}/actioninstances?from=2010-01-01&to=2022-12-01" >
actionInstances.json
export ACTION_INST_IDS=`cat actionInstances.json | jq '.[]' | jq -r '.[]._actionInstanceId'`
```

The records in `actionInstances.json` do not contain the actual output of each action. If you want to preserve the output for each of the actions, run the following command:

```
for actionInstId in ${ACTION_INST_IDS}; do echo "Getting details for ${actionInstId}"; curlOutput=$(curl -X GET
"${CURL_OPTIONS}" -H "${CURL_AUTHENTICATION}" "${API}/actioninstances/${actionInstId}" > ${actionInstId}.json); if (test
${#curlOutput} -gt 0); then echo ${curlOutput}; fi; done
```

Store the each of these action execution record files as appropriate.

Step 3: Delete the remaining action execution records Run the following command to delete the records:

```
for actionInstId in ${ACTION_INST_IDS}; do echo "Deleting ${actionInstId}"; curlOutput=$(curl -X DELETE "${CURL_OPTIONS}"
-H "${CURL_AUTHENTICATION}" "${API}/actioninstances/${actionInstId}"); if (test ${#curlOutput} -gt 0); then echo
${curlOutput}; fi; done
```

Restricting the number of alerts returned by the data layer to the Alert Viewer

By default, the data layer returns all the alerts that match the filter in the Alert Viewer. If this has the potential to cause performance issues on the backend, you can restrict the number of alerts to be returned by the data layer.

Note: When increasing this limit an increase to CPU and memory resources may also be required.

Edit the `aiops-ui-configurations` ConfigMap with your required values. You can perform this task from the command line or the console UI.

Command line

1. Run the command:

   ```
   oc edit cm aiops-ui-configurations
   ```

2. Add the following line under the **data** section:

   ```
   IR_UI_MAX_ALERT_FETCH_LIMIT: "x"
   ```

   Where "x" equals the number of alerts that are returned by the data layer.

3. Save your changes to the ConfigMap file.

Red Hat OpenShift Container console

1. Log in to the Red Hat OpenShift Container console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads > Config Maps**.
3. Search for `aiops-ui-configurations`.
4. Click **... > Edit Config Map**.

5. Add the following line under the **data** section:

   ```
   IR_UI_MAX_ALERT_FETCH_LIMIT: "x"
   ```

   Where "x" equals the number of alerts that are returned by the data layer.

6. Click **Submit** and wait for the `aiops-akora-ui pod` to restart.

Example

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  IR_UI_MAX_ALERT_FETCH_LIMIT: "5"
  cacertVersion: "41802278"
  prerequisiteCheck: Complete
  productVersion: 4.5.0
  zenVersion: 5.1.1
kind: ConfigMap
metadata:
  annotations:
    consoleui.aiops.ibm.com/controller-workload: ibm-watson-aiops-ui-operator-controller-manager-5cbd547bcbz7nbk-
1708622477
  creationTimestamp: "2024-02-20T00:41:57Z"
  labels:
    app.kubernetes.io/component: aiops-aiops-ui-configurations
    app.kubernetes.io/instance: aiops-aiops-ui-configurations-4.5.0
    app.kubernetes.io/managed-by: ibm-watson-aiops-ui-operator
    app.kubernetes.io/name: aiops-aiops-ui-configurations
    app.kubernetes.io/specs: aiops
    release: aiops
  name: aiops-ui-configurations
  namespace: cp4waiops
  ownerReferences:
  - apiVersion: consoleui.aiops.ibm.com/v1
    blockOwnerDeletion: true
    controller: true
    kind: BaseUI
    name: baseui-instance
    uid: 0d85b930-3f93-458f-8426-cf631545de7d
  resourceVersion: "46263896"
  uid: 91f04ab8-61a8-4cc1-8d5c-361a30def34d
```

A message on the Alert Viewer indicates when the list has been configured to limit the number of alerts displayed.

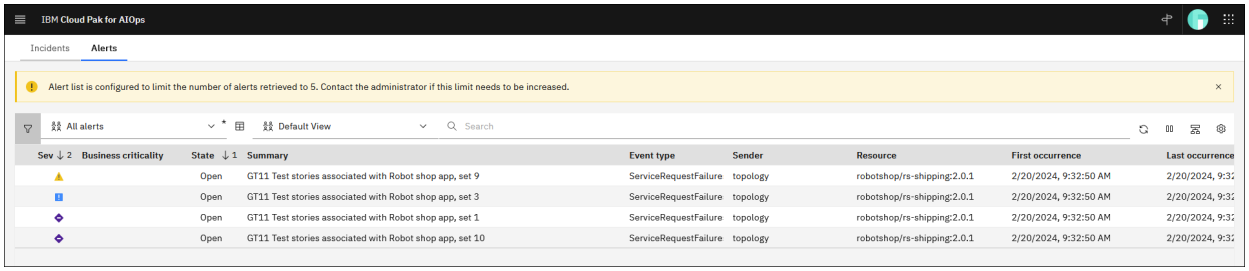


Figure. Alert Viewer

Disabling additional alert count queries from the Alert Viewer

You can disable alert count queries in the Alert Viewer in order to reduce the load on Postgres. The `IR_UI_ALERT_MAX_ROWS_ENABLED` environment variable is set to `true` by default. If disabled, the **Total matched** and **Total active** alert counts are not shown on the modal that is launched from the information icon at the bottom of the table.

Edit the `aiops-ui-configurations` configmap with your required value. You can perform this task from the command line or the console UI.

Command line

1. Run the command:

   ```
   oc edit cm aiops-ui-configurations
   ```

2. Add the following lines under **apiVersion: v1**:

   ```
   data:
     IR_UI_ALERT_MAX_ROWS_ENABLED: "false"
   ```

   Example (after adding):

```
apiVersion: v1
data:
   IR_UI_ALERT_MAX_ROWS_ENABLED: "false"
kind: ConfigMap
metadata:
```

3. Save your changes to the configmap file.

Red Hat OpenShift Container console

1. Log in to the Red Hat OpenShift Container console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads > Config Maps**.
3. Search for **aiops-ui-configurations**.
4. Click **... > Edit Config Map**.
5. Add the following line under **apiVersion: v1**:

```
data:
   IR_UI_ALERT_MAX_ROWS_ENABLED: "false"
```

6. Click **Submit** and wait for the **aiops-akora-ui pod** to restart.

The alert count information in the Alert Viewer is now displayed without the **Total matched** and **Total active** alert counts.

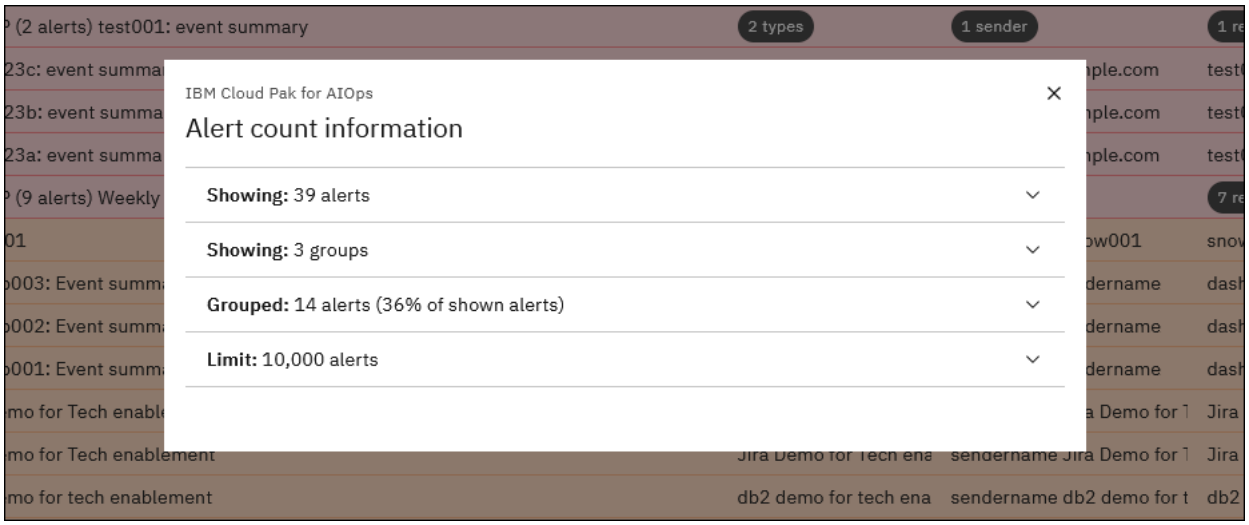


Figure. Alert count

Updating usage data collection preferences

To assist in the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The collection of usage data is enabled or disabled when you install or upgrade to IBM Cloud Pak for AIOps 4.8.0, but can be changed at any time.

This usage data is collected by the **cp4waiops-metricsprocessor** pod, and is sent to and stored in IBM controlled GDPR-compliant systems. The collection of usage data is enabled by default, but can be disabled. For transparency, the **cp4waiops-metricsprocessor** pod's logs will contain all of the information that is collected. The usage data that is collected is all numeric, or is about the deployment type and platform. It does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks executed since install
- Current number of integrations of each type (For example, ServiceNow, Instana, Falcon Logscale)
- Secure tunnel enablement: whether *connection* (which controls whether you can create a secure tunnel) is enabled in the *Installation* custom resource
- Deployment type: base deployment or extended deployment
- Deployment platform: Red Hat® OpenShift® Container Platform or Linux®

Disabling the collection of usage data

1. Set environment variables.

```
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>
export PROJECT_CP4AIOPS=<the project where IBM Cloud Pak for AIOps is deployed, usually cp4aiops>
```

2. Disable usage data collection.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME}
--from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT} --from-
literal=enableCollection=false
```

3. Find the name of the metricsprocessor pod by running the following command:

```
oc get pods -n ${PROJECT_CP4AIOPS} | grep cp4waiops-metricsprocessor
```

Example output:

```
cp4waiops-metricsprocessor-f4dc776d-7pwlm                         1/1     Running     0                19h
```

4. Restart the metricsprocessor pod by running the following command:

```
oc delete pod <metrics_processor_pod> -n ${PROJECT_CP4AIOPS}
```

Where **<metrics_processor_pod>** is the pod returned by the previous step.

Enabling the collection of usage data

1. Set environment variables.

```
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>
```

2. Configure the usage data with your customer details.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-literal=customerName=${CUSTOMER_NAME}
--from-literal=customerICN=${CUSTOMER_ICN} --from-literal=environment=${CUSTOMER_ENVIRONMENT}
```

3. If you have a firewall enabled, ensure that outbound traffic to https://api.segment.io is allowed.

4. Find the name of the metricsprocessor pod by running the following command:

```
oc get pods -n ${PROJECT_CP4AIOPS} | grep cp4waiops-metricsprocessor
```

Example output:

```
cp4waiops-metricsprocessor-f4dc776d-7pwlm                         1/1     Running     0                19h
```

5. Restart the metricsprocessor pod by running the following command:

```
oc delete pod <metricsprocessor_pod> -n ${PROJECT_CP4AIOPS}
```

Where **<metricsprocessor_pod>** is the pod returned by the previous step.

Clearing out topology data without reinstalling

You can clear all topology data without having to reinstall your system.

Before you begin

Ensure you have all the required administrator or management permissions to perform administration tasks.

Also ensure that you have the information specific to your deployment to hand, such as your specific release name prefix and namespace; (you will be setting these variables in step three, after stopping the observer jobs and backing up any required configuration settings).

About this task

To clear all topology data without having to reinstall your system, you perform the following sequence of steps:

1. Stop all observer jobs from running to ensure no new data is received until the old data has been removed.
2. Back up any configurations if required.
3. Set the environment variables manually. This will help with the pre- and post-validation of the topology and inventory databases.
4. Pre-validate the topology and inventory databases to confirm that the data to be removed is present.
5. Run the script to clean the topology and inventory databases.
6. Post-validate to confirm that the data has been removed from the topology and inventory databases.
7. Scale up the topology pods to restart the system and any configured observer jobs.

To clear topology data

1. Stop all observer jobs.

   a. Expand **Define**, click **Integrations**, click the **Manage observer jobs** tab, then click **Configure, schedule, and manage observer jobs**. The Observer jobs page is displayed.

   b. Ensure that no observer jobs are running or are scheduled to run.

2. Back up IBM Cloud Pak for AIOps configurations if required. For more information, see Backing up and restoring UI topology configuration data in the IBM Netcool Operations Insight documentation.

3. Set the required environment variables. These will be specific to your deployment.

   a. Set the release and namespace as required (as in the following examples):

```
export release=aiops
export NAMESPACE=cp4aiops
```

   b. Set the following variables:

```
export POSTGRES_PASS=$(oc get secret $release-topology-postgres-user -o jsonpath='{.data.password}' | base64 -d)
export POSTGRES_USER=$(oc get secret $release-topology-postgres-user -o jsonpath='{.data.username}' | base64 -d)
export INVENTORY_DB=$(oc get asmformation $release-topology -o jsonpath='{.spec.helmValues.global.postgres.dbname}')

export INVENTORY_SCHEMA=$(oc get asmformation $release-topology -o
jsonpath='{.spec.helmValues.global.postgres.schema}')
export CASSANDRA_PASS=$(oc get secret $release-topology-cassandra-auth-secret -o jsonpath='{.data.password}' |
base64 -d)
export CASSANDRA_USER=$(oc get secret $release-topology-cassandra-auth-secret -o jsonpath='{.data.username}' |
base64 -d)
export installation=$(oc get installation.orchestrator.aiops.ibm.com -o jsonpath="{.items[0].metadata.name}")
export POSTGRES_POD=$(oc get clusters.postgresql.k8s.enterprisedb.io $installation-edb-postgres --namespace
$NAMESPACE -o jsonpath="{.status['currentPrimary']}")
```

4. Pre-validate the topology and inventory databases to confirm the presence of the data to be cleared.

   a. Use the following command to verify that the topology database has the janusgraph keyspace:

```
oc exec $release-topology-cassandra-0 -- bash -c "echo 'SELECT * FROM system_schema.keyspaces;exit' |cqlsh --ssl -u
'$CASSANDRA_USER' -p '$CASSANDRA_PASS'"
```

   The following example output indicates that the janusgraph keyspace is present in the topology (Cassandra) database:

```
| keyspace_name | durable_writes | replication
|
| ------------- | -------------- | ------------------------------------------------------------------------------------
---- |
|         aiops |           True | {'class': 'org.apache.cassandra.locator.NetworkTopologyStrategy', 'datacenter1':
'1'} |
|        x_in_y |           True | {'class': 'org.apache.cassandra.locator.NetworkTopologyStrategy', 'datacenter1':
'1'} |
|    janusgraph |           True |   {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor':
'1'} |
|   system_auth |           True |   {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor':
'1'} |
```

   b. Use the following command to verify that the inventory database contains data (under types, edges and other tables).

```
oc exec -it $POSTGRES_POD -- /bin/bash -c 'export PGPASSWORD='$POSTGRES_PASS'; psql --host localhost --username
'$POSTGRES_USER' --dbname '$INVENTORY_DB' --command "select * from types"'
```

   The following example output shows that data is present in the inventory (postgres) database:

```
|  id |               type | description |
| --- | ------------------ | ----------- |
|   1 | undergroundStation |             |
|   2 |        railStation |             |
|   5 | interchangeStation |             |
|  31 |      completeGroup |             |
|  32 |                TFL |             |
| 550 |  waiopsApplication |             |
| 552 |             server |             |
| 553 |                 vm |             |
| 554 |           database |             |
```

5. Clear out the topology and inventory databases.

   Important: Make a note of the current scaling of the following core services that access the topology and search databases before you configure and run the script:

```
$release-topology-inventory
$release-topology-topology
$release-topology-layout
$release-topology-merge
$release-topology-status
```

   The following command provides the current scaling of core services:

```
oc get pods --field-selector=status.phase=Running --no-headers=true --output=custom-
columns="CNAME:.metadata.ownerReferences[0].name" | grep $release-topology | uniq --count
```

   Remember that the `$release` prefix for the topology services will be specific to your scenario, for example `aiops-`.

   a. Scale down the core services which access the storage.

```
oc scale deploy $release-topology-inventory --replicas=0
oc scale deploy $release-topology-topology --replicas=0
oc scale deploy $release-topology-layout --replicas=0
```

```
oc scale deploy $release-topology-merge --replicas=0
oc scale deploy $release-topology-status --replicas=0
```

b. Drop the topology janusgraph keyspace.

```
oc exec $release-topology-cassandra-0 -- bash -c "echo 'DROP KEYSPACE janusgraph;exit' |cqlsh --ssl -u
'$CASSANDRA_USER' -p '$CASSANDRA_PASS'"
```

c. Finally, clear out the inventory data

```
oc exec -it $POSTGRES_POD -- /bin/bash -c 'export PGPASSWORD='$POSTGRES_PASS'; psql --host localhost --username
'$POSTGRES_USER' --dbname '$INVENTORY_DB' --command "CREATE FUNCTION pg_temp.clear_inventory(inventory_schema text)
RETURNS void AS \
\$\$ \
    DECLARE \
        statements CURSOR FOR \
        SELECT schemaname, tablename \
        FROM pg_tables \
        WHERE schemaname = FORMAT('\''%s'\'',inventory_schema) \
        AND NOT tablename IN ('\''flyway_schema_history'\','\''spatial_ref_sys'\'' ); \
    BEGIN \
        FOR stmt IN statements LOOP \
            EXECUTE '\''TRUNCATE TABLE '\'' || string_agg(format('\''%I.%I'\'', stmt.schemaname, stmt.tablename),
'\'', '\'') || '\'' CASCADE;'\''; \
        END LOOP; \
    END; \
\$\$ \
LANGUAGE plpgsql;" --command "SELECT pg_temp.clear_inventory('\'''$INVENTORY_SCHEMA''\'');"'
```

6. Post-validate to confirm that the data has been removed from the topology and inventory databases.

   a. Use the following command to verify that the topology database **no longer** has the `janusgraph` keyspace:

   ```
   oc exec $release-topology-cassandra-0 -- bash -c "echo 'SELECT * FROM system_schema.keyspaces;exit' |cqlsh --ssl -u
   '$CASSANDRA_USER' -p '$CASSANDRA_PASS'"
   ```

   The following example output indicates that the janusgraph keyspace is **no longer** present in the topology (Cassandra) database:

   ```
   | keyspace_name | durable_writes | replication
   |
   | ------------- | -------------- | ------------------------------------------------------------------------------------
   ---- |
   |         aiops |           True | {'class': 'org.apache.cassandra.locator.NetworkTopologyStrategy', 'datacenter1':
   '1'} |
   |          x_in_y |           True | {'class': 'org.apache.cassandra.locator.NetworkTopologyStrategy', 'datacenter1':
   '1'} |
   |   system_auth |           True |   {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor':
   '1'} |
   ```

   b. Use the following command to verify that the inventory database contains **no** data.

   ```
   oc exec -it $POSTGRES_POD -- /bin/bash -c 'export PGPASSWORD='$POSTGRES_PASS'; psql --host localhost --username
   '$POSTGRES_USER' --dbname aiops_topology --command "select * from types"'
   ```

   The following example output shows that **no** data is present in the inventory (postgres) database:

   ```
   | id | type | description |
   | -- | ---- | ----------- |

   (0 rows)
   ```

7. Scale up the pods to the same state as before you cleared out the data.

   Note: The number of replicas can be found using the command mentioned in the **Important** note in Step 5.

   ```
   oc scale deploy $release-topology-inventory --replicas=<number_of_replicas>
   oc scale deploy $release-topology-topology --replicas=<number_of_replicas>
   oc scale deploy $release-topology-layout --replicas=<number_of_replicas>
   oc scale deploy $release-topology-merge --replicas=<number_of_replicas>
   oc scale deploy $release-topology-status --replicas=<number_of_replicas>
   ```

Setting the idle session timeout

You can adjust the idle session timeout for IBM Cloud Pak for AIOps in accordance with your security and compliance requirements. If a user leaves their session idle in a web browser for the specified length of time, the user is automatically logged out of the IBM Cloud Pak for AIOps console.

Before you begin

If you have an installation of IBM Cloud Pak for AIOps on Red Hat® OpenShift®, then you must have one of the following roles:

- Red Hat OpenShift cluster administrator.
- Red Hat OpenShift project administrator on the project (namespace) where IBM Cloud Pak for AIOps is installed.

About this task

By default, IBM Cloud Pak for AIOps logs users out of the Cloud Pak for AIOps console after 12 hours. You can edit the **product-configmap** to adjust the following session timeout settings:

- The length of time until a user's session expires (TOKEN_EXPIRY_TIME). When the user leaves their session idle for the specified length of time, the user is automatically logged out of the web client. The default is 12 hours.

  If you set **TOKEN_EXPIRY_TIME: "1"**, a user's session will expire after 1 hour of inactivity. If you set **TOKEN_EXPIRY_TIME: "0.5"**, a user's session will expire after 30 minutes of inactivity. You are recommended to set the value between 0.1 and 1.

- The length of time that a user has to refresh their session (TOKEN_REFRESH_PERIOD). The default is 12 hours. You are recommended to set the value between 1 and 24.

  If you set **TOKEN_REFRESH_PERIOD: "1"** and the user's session does not expire, the user's session is automatically refreshed during this 60 minute period. The session is extended based on the value that is set for the **TOKEN_EXPIRY_TIME** parameter. However, after the token refresh period passes, the user must log back into the web client when their current session expires.

  If you don't want to allow users to extend their sessions, set the value of the **TOKEN_REFRESH_PERIOD** parameter to a value less than the value of the **TOKEN_EXPIRY_TIME** parameter.

For example, the following configuration shows potential values for these settings:

```
TOKEN_EXPIRY_TIME: "0.5"
TOKEN_REFRESH_PERIOD: "2"
```

If a user starts work at 8 AM and logs in to the console, the user must be active in the web session within 30 minutes for their token to be refreshed. If the user stops using the Cloud Pak for AIOps console at 8:10 and attempts to use the console again until 8:41, the user must re-authenticate to the console because their session expired.

However, if the user remains active in their session and their token refreshes at 9:59 AM, their session lasts until 10:29 AM. When the session expires at 10:29, the user must re-authenticate to the console because the token refresh period expired.

Procedure

1. Change to the project where IBM Cloud Pak for AIOps is deployed:

   If you have an installation of IBM Cloud Pak for AIOps on Red Hat OpenShift, log in to your cluster and then run the following command:

   ```
   oc project <project>
   ```

   Where **<project>** is the namespace (project) that IBM Cloud Pak for AIOps is deployed in.

   If you have an installation of IBM Cloud Pak for AIOps on Linux, log in to a control plane node and then run the following command:

   ```
   oc project aiops
   ```

2. Run the following command to edit the **product-configmap** file:

   ```
   oc edit configmap product-configmap
   ```

3. Add an entry for the **TOKEN_EXPIRY_TIME** parameter to the data section of the **product-configmap** file. For example:

   ```
   data:
     ...
     TOKEN_EXPIRY_TIME: "1"
     ...
   ```

4. Add an entry for the **TOKEN_REFRESH_PERIOD** parameter to the data section of the **product-configmap** file. For example:

   ```
   data:
     ...
     TOKEN_REFRESH_PERIOD: "1"
     ...
   ```

5. Save your changes to the **product-configmap** file.

6. Restart the **usermgmt** pods for the changes to take effect. To restart the pods, run the following command:

   ```
   oc delete pod -l component=usermgmt
   ```

Configuring probable cause

The probable cause capability identifies the alert with the greatest probability of being the cause of the alert group, by using a combination of text classification and analysis of the topological information within the alerts.

You can configure how the system classifies alerts and performs probable cause scoring by changing the default values in the API.

Changing default settings of probable cause

Edit MIME Customization APIs in Swagger to change settings.

- Add keywords

- Update keywords
- Set severity
- Update severity
- Other

Next steps

Access Mime Customization APIs with your Platform UI authentication token or API keys. Include the Platform UI access token in the authorization header of your API calls.

For more information on the available API and related Swagger documentation, see APIs.

Nonproduction features (Technology preview)

IBM Cloud Pak® for AIOps can include components and features for nonproduction use only as technology previews. As licensee of IBM Cloud Pak for AIOps, you can use these components and features only as part of your internal development and test environment for internal nonproduction use. Components and features that are available for nonproduction use are not to be relied on in a production environment.

Features available for nonproduction use:

- Visualizing analytics with IBM Cognos Analytics

  IBM Cloud Pak for AIOps supports an integration with IBM Cognos® Analytics. This integration provides you with additional tools to create custom dashboards and reports to visualize and analyze your IBM Cloud Pak for AIOps data. When you set up this integration, you can open IBM Cognos Analytics from the Cloud Pak for AIOps console and begin using IBM Cognos Analytics dashboards to build visualizations of your IBM Cloud Pak for AIOps data, such as alert and topology data.

- Incident (Story) prioritization from persistence analysis and golden signals extraction

  Incident prioritization uses the golden signal policy and the alert suppression policy to filter out incidents when the observed alerts contain a golden signal and are not suppressed. The policy suppresses alerts that do not need attention.

- Multi-zone high availability and disaster recovery (HADR)

Reference

View reference information about IBM Cloud Pak® for AIOps.

- Infrastructure Automation

Operator cluster service versions (CSVs)

Learn about the operator cluster service versions (CSVs) that are installed for IBM Cloud Pak® for AIOps.

The following table lists the CSVs that are installed for IBM Cloud Pak for AIOps.

Table 1. Operators

NAME	DISPLAY	VERSION
aimanager-operator.v4.8.0	IBM AIOps AI Manager	4.8.0
aiopsedge-operator.v4.8.0	IBM AIOps Edge	4.8.0
asm-operator.v4.8.0	IBM Netcool Agile Service Manager	4.8.0
cloud-native-postgresql.v1.22.7	EDB Postgres for Kubernetes	1.22.7
elasticsearch-operator.v5.8.15	OpenShift Elasticsearch Operator	5.8.15
ibm-aiops-ir-ai.v4.8.0	IBM Watson AIOps Issue Resolution AI & Analytics	4.8.0
ibm-aiops-ir-core.v4.8.0	IBM Watson AIOps Issue Resolution Core	4.8.0
ibm-aiops-ir-lifecycle.v4.8.0	IBM Cloud Pak for Watson AIOps Lifecycle	4.8.0
ibm-aiops-orchestrator.v4.8.0	IBM Cloud Pak for AIOps	4.8.0
ibm-appconnect.v12.3.0	IBM App Connect	12.3.0
ibm-cert-manager-operator.v4.2.10	IBM Cert Manager	4.2.10
ibm-common-service-operator.v4.6.8	IBM Cloud Pak foundational services	4.6.8
ibm-commonui-operator.v4.4.7	Ibm Common UI	4.4.7
ibm-elasticsearch-operator.v1.1.2346	IBM OpenContent Elasticsearch	1.1.2346
ibm-events-operator.v5.0.1	IBM Events Operator	5.0.1
ibm-iam-operator.v4.5.7	IBM IM Operator	4.5.7
ibm-opencontent-flink.v2.0.6	IBM OpenContent Flink Operator	2.0.6

NAME	DISPLAY	VERSION
ibm-redis-cp.v1.2.2	ibm-redis-cp-controller	1.2.2
ibm-secure-tunnel-operator.v4.8.0	IBM Secure Tunnel	4.8.0
ibm-watson-aiops-ui-operator.v4.8.0	IBM AIOps UI	4.8.0
ibm-zen-operator.v5.1.10	IBM Zen Service	5.1.10
operand-deployment-lifecycle-manager.v4.3.7	Operand Deployment Lifecycle Manager	4.3.7

Note: Deployments of IBM Cloud Pak for AIOps on Linux do not have IBM App Connect, IBM Elastic, or IBM Secure Tunnel.

You can view a list of CSVs and subscriptions by running the following commands:

```
oc get csv
oc get sub
```

Permissions (IBM Cloud Pak for AIOps)

The following content shows the permissions and cluster permissions that are required by the IBM Cloud Pak for AIOps operators.

```
aimanager-operator:
  clusterPermissions:
  - rules:
    - apiGroups:
      - scheduling.k8s.io
      resources:
      - priorityclasses
      verbs:
      - create
      - delete
      - list
      - watch
      - get
      - patch
      - update
    serviceAccountName: aimanager-operator-controller-manager
  permissions:
  - rules:
    - apiGroups:
      - ''
      resources:
      - configmaps
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - patch
      - delete
    - apiGroups:
      - coordination.k8s.io
      resources:
      - leases
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - patch
      - delete
    - apiGroups:
      - ''
      resources:
      - events
      verbs:
      - create
      - patch
    - apiGroups:
      - ''
      - policy
      - template.openshift.io
      resources:
      - pods
      - pods/exec
      - pods/log
      - services
      - services/finalizers
      - endpoints
      - persistentvolumeclaims
      - events
      - configmaps
      - secrets
      - serviceaccounts
      - serviceaccount
      - poddisruptionbudgets
      - processedtemplates
```

```
    verbs:
    - '*'
  - apiGroups:
    - coordination.k8s.io
    resources:
    - leases
    verbs:
    - get
    - list
    - watch
    - create
    - update
    - patch
    - delete
  - apiGroups:
    - monitoring.coreos.com
    resources:
    - servicemonitors
    verbs:
    - get
    - create
  - apiGroups:
    - rbac.authorization.k8s.io
    resources:
    - roles
    - rolebindings
    - clusterroles
    - clusterrolebindings
    verbs:
    - get
    - list
    - patch
    - update
    - delete
    - create
    - watch
  - apiGroups:
    - batch
    resources:
    - jobs
    - cronjobs
    verbs:
    - '*'
  - apiGroups:
    - apps
    resources:
    - deployments
    - deployments/finalizers
    - daemonsets
    - replicasets
    - statefulsets
    verbs:
    - '*'
  - apiGroups:
    - security.openshift.io
    resourceNames:
    - restricted
    resources:
    - securitycontextconstraints
    verbs:
    - use
  - apiGroups:
    - security.openshift.io
    resources:
    - securitycontextconstraints
    verbs:
    - create
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - monitoring.coreos.com
    resources:
    - servicemonitors
    verbs:
    - get
    - create
  - apiGroups:
    - apps
    resourceNames:
    - aimanager-operator
    resources:
    - deployments/finalizers
    verbs:
    - update
  - apiGroups:
    - ''
    resources:
    - pods
    verbs:
```

```
    - get
  - apiGroups:
    - apps
    resources:
    - replicasets
    - deployments
    verbs:
    - get
  - apiGroups:
    - cert-manager.io
    resources:
    - issuers
    - certificates
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - eventprocessing.automation.ibm.com
    resources:
    - eventprocessors
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ibmevents.ibm.com
    resources:
    - kafkas
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - '*'
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ai-manager.watson-aiops.ibm.com
    resources:
    - '*'
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - helm.operator-sdk
    resources:
    - IbmMinio
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - apiextensions.k8s.io
    resources:
    - '*'
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    - batch
```

```
    - extensions
    - apps
    - policy
    - rbac.authorization.k8s.io
    - autoscaling
    - route.openshift.io
    - authorization.openshift.io
    - networking.k8s.io
    - metrics.k8s.io
    resources:
    - pods
    - pods/log
    - poddisruptionbudgets
    - secrets
    - jobs
    - configmaps
    - deployments
    - daemonsets
    - statefulsets
    - replicasets
    - services
    - services/finalizers
    - persistentvolumeclaims
    - cronjobs
    - pods/exec
    - pods/portforward
    - serviceaccounts
    - namespaces
    - roles
    - rolebindings
    - horizontalpodautoscalers
    - routes
    - routes/custom-host
    - ingresses
    - endpoints
    - cronjob
    - networkpolicies
    - events
    - jobs/status
    - pods/status
    - resourcequotas
    - resourcequotas/status
    - nodes
    verbs:
    - apply
    - create
    - get
    - delete
    - watch
    - update
    - edit
    - exec
    - list
    - patch
    - deletecollection
  - apiGroups:
    - cpd.ibm.com
    resources:
    - cpdinstalls
    - cpdinstalls/spec
    - cpdinstalls/status
    verbs:
    - apply
    - create
    - delete
    - edit
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - build.openshift.io
    resources:
    - buildconfigs
    - buildconfigs/instantiate
    - buildconfigs/instantiatebinary
    - buildconfigs/webhooks
    - buildlogs
    - builds
    - builds/clone
    - builds/details
    - builds/log
    verbs:
    - create
    - delete
    - list
    - watch
    - get
    - patch
    - update
  - apiGroups:
```

```
    - image.openshift.io
    resources:
    - imagestreams
    - imagestreams/layers
    - imagestreams/secrets
    - imagestreams/status
    - imagestreamimages
    - imagestreamimports
    - imagestreammappings
    - imagestreamtags
    verbs:
    - create
    - delete
    - list
    - watch
    - get
    - patch
    - update
  - apiGroups:
    - apps
    resourceNames:
    - cpd-meta-operator
    resources:
    - deployments/finalizers
    verbs:
    - update
  - apiGroups:
    - ''
    resources:
    - pods
    verbs:
    - get
  - apiGroups:
    - apps
    resources:
    - replicasets
    - deployments
    verbs:
    - get
  - apiGroups:
    - metaoperator.cpd.ibm.com
    resources:
    - '*'
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - certmanager.k8s.io
    resources:
    - issuers
    - issuers/status
    - issuers/finalizers
    - certificates
    - certificates/status
    - certificates/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - operator.ibm.com
    resources:
    - operandrequests
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - zen.cpd.ibm.com
    resources:
    - '*'
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
```

```
    - apiGroups:
      - admissionregistration.k8s.io
      resources:
      - validatingwebhookconfigurations
      - mutatingwebhookconfigurations
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - flink.ibm.com
      resources:
      - flinkdeployments
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    serviceAccountName: aimanager-operator-controller-manager
aiopsedge-operator:
  clusterPermissions:
  - rules:
    - apiGroups:
      - ''
      resourceNames:
      - console-config
      resources:
      - configmaps
      verbs:
      - get
      - watch
    - apiGroups:
      - apiextensions.k8s.io
      resources:
      - customresourcedefinitions
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - config.openshift.io
      resources:
      - ingresses
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - connectors.aiops.ibm.com
      resources:
      - aiopskafkatopics
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - connectors.aiops.ibm.com
      resources:
      - aiopskafkatopics/status
      verbs:
      - get
      - patch
      - update
    - apiGroups:
      - connectors.aiops.ibm.com
      resources:
      - bundlemanifests
      - bundlemanifests/finalizers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - connectors.aiops.ibm.com
      resources:
      - bundlemanifests/status
      verbs:
```

```
    - get
    - patch
    - update
  - apiGroups:
    - operator.openshift.io
    resources:
    - dnses
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - storage.k8s.io
    resources:
    - storageclasses
    verbs:
    - get
    - list
    - watch
  serviceAccountName: manager
permissions:
- rules:
  - apiGroups:
    - ''
    resources:
    - configmaps
    verbs:
    - get
    - list
    - watch
    - create
    - update
    - patch
    - delete
  - apiGroups:
    - ''
    resources:
    - configmaps/status
    verbs:
    - get
    - update
    - patch
  - apiGroups:
    - ''
    resources:
    - events
    verbs:
    - create
  - apiGroups:
    - ''
    resources:
    - '*'
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - configmaps
    - endpoints
    - secrets
    - serviceaccounts
    - services
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - configmaps/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - configmaps/finalizers
```

```
    - secrets/finalizers
    - serviceaccounts/finalizers
    - services/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - events
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - pods
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - ''
    resources:
    - secrets
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - ''
    resources:
    - services
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - ''
    resources:
    - services/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - apidiscovery.ibm.com
    resources:
    - apis
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - appconnect.ibm.com
    resources:
    - configurations
    - dashboards
    - integrationruntimes
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - apps
    resources:
    - '*'
    verbs:
    - create
    - delete
```

```
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - apps
      resources:
      - deployments
      - replicasets
      - statefulsets
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - apps
      resources:
      - deployments/finalizers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - asm.ibm.com
      resources:
      - asmformations
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - autoscaling
      resources:
      - horizontalpodautoscalers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - build.openshift.io
      resources:
      - buildconfigs
      verbs:
      - delete
      - get
      - list
      - watch
    - apiGroups:
      - camel.apache.org
      resources:
      - camelcatalogs
      - integrationkits
      - integrationplatforms
      - integrations
      verbs:
      - delete
      - get
      - list
      - watch
    - apiGroups:
      - camel.apache.org
      resources:
      - integrations
      - kameletbindings
      - kamelets
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - cert-manager.io
      resources:
      - certificates
      verbs:
      - create
```

```
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - cert-manager.io
    resources:
    - certificates
    - issuers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - cert-manager.io
    resources:
    - issuers
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - aiopsedges
    - aiopsedges/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - aiopsedges/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - connectorcomponents
    - connectorcomponents/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - connectorcomponents
    - connectorconfigurations
    - connectorconfigurations/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - connectorcomponents/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - connectorconfigurations/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - connectors.aiops.ibm.com
```

```
    resources:
    - connectorschemas
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - gitapps
    - gitapps/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - gitapps/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - microedgeconfigurations
    - microedgeconfigurations/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - coordination.k8s.io
    resources:
    - leases
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ibmevents.ibm.com
    resources:
    - kafkas
    - kafkatopics
    - kafkausers
    - mirrormakers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - image.openshift.io
    resources:
    - imagestreams
    verbs:
    - delete
    - get
    - list
    - watch
  - apiGroups:
    - monitoring.coreos.com
    resources:
    - podmonitors
    - servicemonitors
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
```

```
      - watch
    - apiGroups:
      - network.openshift.io
      resources:
      - egressnetworkpolicies
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - networking.k8s.io
      resources:
      - ingresses
      - networkpolicies
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - openliberty.io
      resources:
      - openlibertyapplications
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - operator.ibm.com
      resources:
      - operandconfigs
      - operandregistries
      - operandrequests
      verbs:
      - create
      - delete
      - deletecollection
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - operators.coreos.com
      resources:
      - operatorgroups
      - subscriptions
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - rbac.authorization.k8s.io
      resources:
      - rolebindings
      - roles
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - route.openshift.io
      resources:
      - routes
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - route.openshift.io
      resources:
      - routes/custom-host
      verbs:
```

```
      - create
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - vault.aiops.ibm.com
      resources:
      - vaultaccesses
      - vaultdeploys
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    serviceAccountName: manager
asm-operator:
  permissions:
  - rules:
    - apiGroups:
      - policy
      resources:
      - poddisruptionbudgets
      verbs:
      - get
      - list
      - patch
      - update
      - delete
      - create
      - watch
    - apiGroups:
      - rbac.authorization.k8s.io
      resources:
      - roles
      - rolebindings
      - clusterroles
      - clusterrolebindings
      verbs:
      - get
      - list
      - patch
      - update
      - delete
      - create
      - watch
    - apiGroups:
      - autoscaling
      resources:
      - horizontalpodautoscalers
      verbs:
      - get
      - list
      - patch
      - update
      - delete
      - create
      - watch
    - apiGroups:
      - ''
      resources:
      - pods
      - services
      - services/finalizers
      - endpoints
      - persistentvolumeclaims
      - events
      - configmaps
      - secrets
      - role
      - rolebinding
      verbs:
      - get
      - list
      - patch
      - update
      - delete
      - create
      - watch
    - apiGroups:
      - ''
      - extensions
      resources:
      - deployments
      - configmaps
      - ingresses
      - services
      - serviceaccounts
```

```
    - persistentvolumeclaims
    verbs:
    - '*'
  - apiGroups:
    - ''
    resources:
    - namespaces
    verbs:
    - get
  - apiGroups:
    - apps
    resources:
    - deployments
    - deployments/finalizers
    - daemonsets
    - replicasets
    - statefulsets
    verbs:
    - '*'
  - apiGroups:
    - batch
    resources:
    - jobs
    - cronjobs
    verbs:
    - '*'
  - apiGroups:
    - route.openshift.io
    resources:
    - routes
    - routes/custom-host
    verbs:
    - '*'
  - apiGroups:
    - asm.ibm.com
    resources:
    - asms
    - asms/finalizers
    - Asms
    - Asm/status
    - Asms
    verbs:
    - '*'
  - apiGroups:
    - asm.ibm.com
    resources:
    - '*'
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - monitoring.coreos.com
    resources:
    - servicemonitors
    verbs:
    - get
    - create
  - apiGroups:
    - networking.k8s.io
    resources:
    - networkpolicies
    - ingresses
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - configmaps/status
    - endpoints
    verbs:
    - get
    - update
    - patch
  - apiGroups:
    - asm.ibm.com
    resources:
    - appdiscoes
    verbs:
    - create
    - delete
    - get
```

```
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - asm.ibm.com
      resources:
      - appdiscoes/status
      verbs:
      - get
      - patch
      - update
    - apiGroups:
      - base.automation.ibm.com
      resources:
      - cartridgerequirements
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - zen.cpd.ibm.com
      resources:
      - zenservices
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - core.ir.aiops.ibm.com
      resources:
      - issueresolutioncores
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - cert-manager.io
      resources:
      - issuers
      - certificates
      verbs:
      - get
      - list
      - watch
      - create
      - patch
      - update
      - delete
    serviceAccountName: asm-operator
ibm-aiops-ir-ai:
  permissions:
  - rules:
    - apiGroups:
      - coordination.k8s.io
      resources:
      - leases
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - patch
      - delete
    - apiGroups:
      - ''
      resources:
      - configmaps
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - patch
      - delete
    - apiGroups:
      - ''
      resources:
      - configmaps/status
      verbs:
      - get
      - update
      - patch
    - apiGroups:
      - ''
      resources:
      - events
      verbs:
      - create
      - patch
```

```
  - apiGroups:
    - ai-manager.watson-aiops.ibm.com
    resources:
    - algorithms
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ai.ir.aiops.ibm.com
    resources:
    - aiopsanalyticsorchestrators
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ai.ir.aiops.ibm.com
    resources:
    - aiopsanalyticsorchestrators/finalizers
    verbs:
    - update
  - apiGroups:
    - ai.ir.aiops.ibm.com
    resources:
    - aiopsanalyticsorchestrators/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - apps
    resources:
    - deployments
    - statefulsets
    verbs:
    - create
    - delete
    - get
    - list
    - update
    - watch
  - apiGroups:
    - batch
    resources:
    - jobs
    verbs:
    - create
    - delete
    - get
    - list
    - update
    - watch
  - apiGroups:
    - cert-manager.io
    resources:
    - certificates
    verbs:
    - create
    - delete
    - get
    - list
    - update
  - apiGroups:
    - ''
    resources:
    - configmaps
    verbs:
    - create
    - delete
    - get
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - persistentvolumeclaims
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
```

```
      - watch
    - apiGroups:
      - ''
      resources:
      - secrets
      - services
      verbs:
      - create
      - delete
      - get
      - list
      - update
      - watch
    - apiGroups:
      - ''
      resources:
      - serviceaccounts
      verbs:
      - create
      - delete
      - get
      - update
    - apiGroups:
      - networking.k8s.io
      resources:
      - networkpolicies
      verbs:
      - create
      - delete
      - get
      - update
    serviceAccountName: ibm-ir-ai-operator-controller-management-sa
ibm-aiops-ir-core:
  permissions:
  - rules:
    - apiGroups:
      - ''
      - coordination.k8s.io
      resources:
      - configmaps
      - leases
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - patch
      - delete
    - apiGroups:
      - ''
      resources:
      - events
      verbs:
      - create
      - patch
    - apiGroups:
      - apps
      resources:
      - deployments
      - statefulsets
      verbs:
      - create
      - delete
      - get
      - list
      - update
      - watch
    - apiGroups:
      - batch
      resources:
      - cronjobs
      - jobs
      verbs:
      - create
      - delete
      - get
      - list
      - update
      - watch
    - apiGroups:
      - cert-manager.io
      resources:
      - certificates
      verbs:
      - create
      - delete
      - get
      - list
      - update
    - apiGroups:
      - ''
```

```
      resources:
      - configmaps
      - persistentvolumeclaims
      - secrets
      - services
      verbs:
      - create
      - delete
      - get
      - list
      - update
      - watch
    - apiGroups:
      - ''
      resources:
      - pods
      verbs:
      - list
      - watch
    - apiGroups:
      - ''
      resources:
      - serviceaccounts
      verbs:
      - create
      - delete
      - get
      - list
      - update
    - apiGroups:
      - core.ir.aiops.ibm.com
      resources:
      - issueresolutioncores
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - core.ir.aiops.ibm.com
      resources:
      - issueresolutioncores/finalizers
      verbs:
      - update
    - apiGroups:
      - core.ir.aiops.ibm.com
      resources:
      - issueresolutioncores/status
      verbs:
      - get
      - patch
      - update
    - apiGroups:
      - networking.k8s.io
      resources:
      - networkpolicies
      verbs:
      - create
      - delete
      - get
      - update
    - apiGroups:
      - route.openshift.io
      resources:
      - routes
      verbs:
      - create
      - delete
      - get
      - update
    serviceAccountName: ir-core-operator-controller-manager
ibm-aiops-ir-lifecycle:
  permissions:
  - rules:
    - apiGroups:
      - ''
      resources:
      - configmaps
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - patch
      - delete
    - apiGroups:
      - coordination.k8s.io
      resources:
```

```
    - leases
    verbs:
    - get
    - list
    - watch
    - create
    - update
    - patch
    - delete
  - apiGroups:
    - ''
    resources:
    - events
    verbs:
    - create
    - patch
  - apiGroups:
    - apps
    resources:
    - deployments
    - statefulsets
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - batch
    resources:
    - cronjobs
    - jobs
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - cert-manager.io
    resources:
    - certificates
    - issuers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - configmaps
    - persistentvolumeclaims
    - secrets
    - serviceaccounts
    - services
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - pods
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - flink.automation.ibm.com
    resources:
    - flinkclusters
    verbs:
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - flink.ibm.com
```

```
      resources:
      - flinkdeployments
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - ibmevents.ibm.com
      resources:
      - kafkausers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - lifecycle.ir.aiops.ibm.com
      resources:
      - lifecycleservices
      - lifecycletriggers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - lifecycle.ir.aiops.ibm.com
      resources:
      - lifecycleservices/finalizers
      - lifecycletriggers/finalizers
      verbs:
      - update
    - apiGroups:
      - lifecycle.ir.aiops.ibm.com
      resources:
      - lifecycleservices/status
      - lifecycletriggers/status
      verbs:
      - get
      - patch
      - update
    - apiGroups:
      - monitoring.coreos.com
      resources:
      - servicemonitors
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - networking.k8s.io
      resources:
      - networkpolicies
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - route.openshift.io
      resources:
      - routes
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    serviceAccountName: ir-lifecycle-operator-controller-manager
ibm-aiops-orchestrator:
  clusterPermissions:
  - rules:
    - apiGroups:
```

```
    - operators.coreos.com
    resources:
    - catalogsources
    verbs:
    - get
  serviceAccountName: ibm-aiops-orchestrator-controller-manager
permissions:
- rules:
  - apiGroups:
    - ''
    resources:
    - configmaps
    verbs:
    - get
    - list
    - watch
    - create
    - update
    - patch
    - delete
  - apiGroups:
    - coordination.k8s.io
    resources:
    - leases
    verbs:
    - get
    - list
    - watch
    - create
    - update
    - patch
    - delete
  - apiGroups:
    - ''
    resources:
    - events
    verbs:
    - create
    - patch
  - apiGroups:
    - ai-manager.watson-aiops.ibm.com
    resources:
    - aimanagers
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - ai.ir.aiops.ibm.com
    resources:
    - aiopsanalyticsorchestrators
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - apps
    resources:
    - deployments
    - statefulsets
    verbs:
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - asm.ibm.com
    resources:
    - asms
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - base.automation.ibm.com
    resources:
    - automationbases
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - base.automation.ibm.com
    resources:
    - cartridgerequirements
    verbs:
    - create
```

```
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - base.automation.ibm.com
    resources:
    - cartridgesrequirements/finalizers
    verbs:
    - update
  - apiGroups:
    - batch
    resources:
    - cronjobs
    - jobs
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - cert-manager.io
    resources:
    - certificates
    - issuers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - connectors.aiops.ibm.com
    resources:
    - aiopsedges
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - aiopsuis
    - baseuis
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - ''
    resources:
    - configmaps
    - persistentvolumeclaims
    - secrets
    - serviceaccounts
    - services
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - pods
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - core.automation.ibm.com
    resources:
    - automationuiconfigs
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
```

```
    - core.automation.ibm.com
    resources:
    - cartridges
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - core.automation.ibm.com
    resources:
    - cartridges/finalizers
    verbs:
    - update
  - apiGroups:
    - core.ir.aiops.ibm.com
    resources:
    - issueresolutioncores
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - elastic.automation.ibm.com
    resources:
    - elasticsearches
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - elasticsearch.opencontent.ibm.com
    resources:
    - elasticsearchclusters
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - eventprocessing.automation.ibm.com
    resources:
    - eventprocessors
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - flink.automation.ibm.com
    resources:
    - flinkclusters
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - flink.ibm.com
    resources:
    - flinkdeployments
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ibmevents.ibm.com
    resources:
    - kafkas
    - kafkausers
    verbs:
    - create
```

```
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - lifecycle.ir.aiops.ibm.com
    resources:
    - lifecycleservices
    - lifecycletriggers
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - modeltrain.ibm.com
    resources:
    - modeltrains
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - monitoring.coreos.com
    resources:
    - servicemonitors
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - networking.k8s.io
    resources:
    - ingresses
    - networkpolicies
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - operator.ibm.com
    resources:
    - commonservices
    - operandbindinfos
    - operandconfigs
    - operandregistries
    - operandrequests
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - operators.coreos.com
    resources:
    - clusterserviceversions
    verbs:
    - delete
    - get
    - list
    - watch
  - apiGroups:
    - operators.coreos.com
    resources:
    - operatorgroups
    verbs:
    - create
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - operators.coreos.com
    resources:
    - subscriptions
    verbs:
    - create
    - get
    - list
    - patch
```

```
    - update
    - watch
  - apiGroups:
    - orchestrator.aiops.ibm.com
    resources:
    - customsizes
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - orchestrator.aiops.ibm.com
    resources:
    - customsizes/finalizers
    verbs:
    - update
  - apiGroups:
    - orchestrator.aiops.ibm.com
    resources:
    - customsizes/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - orchestrator.aiops.ibm.com
    resources:
    - installations
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - orchestrator.aiops.ibm.com
    resources:
    - installations/finalizers
    verbs:
    - update
  - apiGroups:
    - orchestrator.aiops.ibm.com
    resources:
    - installations/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - packages.operators.coreos.com
    resources:
    - packagemanifests
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - postgres.aiops.ibm.com
    resources:
    - postgresdbs
    - postgreservices
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - postgresql.k8s.enterprisedb.io
    resources:
    - clusters
    verbs:
    - get
    - list
    - watch
  - apiGroups:
    - rbac.authorization.k8s.io
    resources:
    - rolebindings
    - roles
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
```

```
    - apiGroups:
      - redis.ibm.com
      resources:
      - rediscps
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - shim.bedrock.ibm.com
      resources:
      - kafkaclaims
      verbs:
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - shim.bedrock.ibm.com
      resources:
      - kafkaclaims/finalizers
      verbs:
      - update
    - apiGroups:
      - sretooling.management.ibm.com
      resources:
      - tunnels
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - zen.cpd.ibm.com
      resources:
      - zenextensions
      - zenservices
      verbs:
      - get
      - list
      - watch
    serviceAccountName: ibm-aiops-orchestrator-controller-manager
ibm-secure-tunnel-operator:
  clusterPermissions:
  - rules:
    - apiGroups:
      - ''
      resources:
      - pods
      verbs:
      - get
      - list
      - watch
      - update
    - apiGroups:
      - ''
      resources:
      - configmaps
      - endpoints
      - events
      - secrets
      - serviceaccounts
      - services
      - services/finalizers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - apiextensions.k8s.io
      resources:
      - customresourcedefinitions
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - apps
      resources:
```

```
    - deployments
    - replicasets
    - statefulsets
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - authentication.k8s.io
    resources:
    - tokenreviews
    verbs:
    - create
    - patch
  - apiGroups:
    - authorization.k8s.io
    resources:
    - subjectaccessreviews
    verbs:
    - create
    - patch
  - apiGroups:
    - extensions
    resources:
    - ingresses
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - monitoring.coreos.com
    resources:
    - servicemonitors
    verbs:
    - create
    - get
  - apiGroups:
    - networking.k8s.io
    resources:
    - ingresses
    - networkpolicies
    - ingressclasses
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - oauth.openshift.io
    resources:
    - oauthclients
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - operator.openshift.io
    resources:
    - dnses
    verbs:
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - rbac.authorization.k8s.io
    resources:
    - clusterrolebindings
    - clusterroles
    - rolebindings
    - roles
    verbs:
    - create
    - delete
    - get
    - list
```

```
    - patch
    - update
    - watch
  - apiGroups:
    - route.openshift.io
    resources:
    - routes
    - routes/custom-host
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - securetunnel.management.ibm.com
    resources:
    - applicationmappings
    - applicationmappings/finalizers
    - applicationmappings/status
    - applications/finalizers
    - templates
    - templates/finalizers
    - templates/status
    - tunnelconnections
    - tunnelconnections/finalizers
    - tunnelconnections/status
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - sretooling.management.ibm.com
    resources:
    - tunnels
    - tunnels/finalizers
    - tunnels/status
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - tunnel.management.ibm.com
    resources:
    - applications
    - applications/finalizers
    - applications/status
    - templates
    - templates/finalizers
    - templates/status
    - tunnelconnections
    - tunnelconnections/finalizers
    - tunnelconnections/status
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - cert-manager.io
    resources:
    - certificates
    - issuers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - zen.cpd.ibm.com
    resources:
    - zenextensions
    - zenservices
    verbs:
    - create
    - delete
```

```
      - get
      - list
      - patch
      - update
      - watch
    serviceAccountName: ibm-secure-tunnel-operator
ibm-watson-aiops-ui-operator:
  permissions:
  - rules:
    - apiGroups:
      - ''
      resources:
      - configmaps
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - patch
      - delete
    - apiGroups:
      - ''
      resources:
      - configmaps/status
      verbs:
      - get
      - update
      - patch
    - apiGroups:
      - ''
      resources:
      - events
      verbs:
      - create
      - patch
    - apiGroups:
      - coordination.k8s.io
      resources:
      - leases
      verbs:
      - get
      - create
      - update
    - apiGroups:
      - ''
      resources:
      - configmaps
      - pods
      verbs:
      - create
      - delete
      - deletecollection
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - ''
      resources:
      - configmaps
      - secrets
      - services
      - services/finalizers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - ''
      resources:
      - configmaps
      - serviceaccounts
      - services
      - services/finalizers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - ''
      resources:
      - configmaps
```

```
      - services
      - services/finalizers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - apps
      resources:
      - deployments
      - deployments/finalizers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - asm.ibm.com
      resources:
      - asms
      verbs:
      - get
      - list
    - apiGroups:
      - batch
      resources:
      - jobs
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - cert-manager.io
      resources:
      - certificates
      verbs:
      - create
      - delete
      - get
      - list
      - update
      - watch
    - apiGroups:
      - connectors.aiops.ibm.com
      resources:
      - connectorschemas
      verbs:
      - get
      - list
    - apiGroups:
      - consoleui.aiops.ibm.com
      resources:
      - aimodeluis
      - aimodeluis/finalizers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - consoleui.aiops.ibm.com
      resources:
      - aimodeluis/status
      verbs:
      - get
      - patch
      - update
    - apiGroups:
      - consoleui.aiops.ibm.com
      resources:
      - aiopsuiextensions
      - aiopsuiextensions/finalizers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
```

```
    - update
    - watch
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - aiopsuiextensions/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - aiopsuis
    - aiopsuis/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - aiopsuis/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - baseuis
    - baseuis/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - baseuis/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - connectoruis
    - connectoruis/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - connectoruis/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - insightsuis
    - insightsuis/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - consoleui.aiops.ibm.com
    resources:
    - insightsuis/status
    verbs:
    - get
```

```
      - patch
      - update
    - apiGroups:
      - ''
      resources:
      - configmaps
      verbs:
      - create
      - delete
      - get
      - update
    - apiGroups:
      - networking.k8s.io
      resources:
      - networkpolicies
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - rbac.authorization.k8s.io
      resources:
      - rolebindings
      - roles
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - route.openshift.io
      resources:
      - routes
      verbs:
      - get
    - apiGroups:
      - zen.cpd.ibm.com
      resources:
      - zenextensions
      - zenextensions/status
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - zen.cpd.ibm.com
      resources:
      - zenservices
      verbs:
      - get
      - list
      - watch
    serviceAccountName: ibm-watson-aiops-ui-operator-manager-role
```

Permissions (Infrastructure Automation)

The following content shows the permissions and cluster permissions that are required by the Infrastructure Automation operators.

```
ibm-infrastructure-automation-operator:
  clusterPermissions:
  - rules:
    - apiGroups:
      - ''
      resources:
      - configmaps
      verbs:
      - create
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - ''
      resources:
      - pods
      verbs:
```

```
    - delete
    - deletecollection
    - get
    - list
  - apiGroups:
    - operator.ibm.com
    resources:
    - namespacescopes
    verbs:
    - delete
    - deletecollection
    - get
    - list
  - apiGroups:
    - operators.coreos.com
    resources:
    - catalogsources
    verbs:
    - get
  - apiGroups:
    - operators.coreos.com
    resources:
    - subscriptions
    verbs:
    - create
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - packages.operators.coreos.com
    resources:
    - packagemanifests
    verbs:
    - get
    - list
    - watch
  serviceAccountName: ibm-infrastructure-automation-operator-controller-manager
permissions:
- rules:
  - apiGroups:
    - ''
    resources:
    - configmaps
    - persistentvolumeclaims
    verbs:
    - get
    - list
    - watch
    - create
    - update
    - patch
    - delete
  - apiGroups:
    - coordination.k8s.io
    resources:
    - leases
    verbs:
    - get
    - list
    - watch
    - create
    - update
    - patch
    - delete
  - apiGroups:
    - ''
    resources:
    - events
    verbs:
    - create
    - patch
  - apiGroups:
    - ''
    resources:
    - serviceaccounts
    - services
    - services/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - secrets
    verbs:
```

```
    - create
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - aiops.ibm.com
    resources:
    - iaconfigs
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - aiops.ibm.com
    resources:
    - iaconfigs/finalizers
    verbs:
    - update
  - apiGroups:
    - aiops.ibm.com
    resources:
    - iaconfigs/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - apps
    resources:
    - deployments
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - base.automation.ibm.com
    resources:
    - automationbases
    - cartridgerequirements
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - core.automation.ibm.com
    resources:
    - automationuiconfigs
    - cartridges
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - operator.ibm.com
    resources:
    - commonservices
    - operandbindinfos
    - operandconfigs
    - operandregistries
    - operandrequests
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - operators.coreos.com
    resources:
    - catalogsources
    - clusterserviceversions
    - subscriptions
```

```
      - operatorgroups
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - zen.cpd.ibm.com
      resources:
      - zenservices
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - cert-manager.io
      resources:
      - issuers
      - certificates
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - extensions
      resources:
      - networkpolicies
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - networking.k8s.io
      resources:
      - networkpolicies
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    serviceAccountName: ibm-infrastructure-automation-operator-controller-manager
ibm-management-cam-install:
  clusterPermissions:
  - rules:
    - apiGroups:
      - coordination.k8s.io
      resources:
      - leases
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - patch
      - delete
    - apiGroups:
      - apiextensions.k8s.io
      resources:
      - customresourcedefinitions
      verbs:
      - '*'
    - apiGroups:
      - rbac.authorization.k8s.io
      resources:
      - clusterrolebindings
      - clusterroles
      verbs:
      - '*'
    - apiGroups:
      - certmanager.k8s.io
      resources:
      - clusterissuers
      verbs:
```

```
      - use
    - apiGroups:
      - certmanager.k8s.io
      resources:
      - issuers
      verbs:
      - use
      - get
      - create
      - delete
      - patch
      - list
    - apiGroups:
      - ibmcpcs.ibm.com
      resources:
      - secretshares
      verbs:
      - use
      - get
      - create
      - delete
      - patch
    - apiGroups:
      - policy
      resources:
      - podsecuritypolicies
      verbs:
      - '*'
    - apiGroups:
      - extensions
      resourceNames:
      - cam-services-psp
      resources:
      - podsecuritypolicies
      verbs:
      - '*'
    - apiGroups:
      - security.openshift.io
      resources:
      - securitycontextconstraints
      verbs:
      - get
      - delete
      - create
      - watch
      - list
    - apiGroups:
      - route.openshift.io
      resources:
      - routes
      - routes/custom-host
      verbs:
      - '*'
    - apiGroups:
      - foundation.ibm.com
      resources:
      - navconfigurations
      verbs:
      - '*'
    - apiGroups:
      - operator.ibm.com
      resources:
      - operandrequests
      verbs:
      - '*'
    serviceAccountName: cam-install-operator-controller-manager
  permissions:
  - rules:
    - apiGroups:
      - ''
      resources:
      - services
      - services/finalizers
      - endpoints
      - persistentvolumeclaims
      - events
      - configmaps
      - secrets
      - serviceaccounts
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - apps
      resources:
      - deployments
      - daemonsets
```

```
      - replicasets
      - statefulsets
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - monitoring.coreos.com
      resources:
      - servicemonitors
      verbs:
      - get
      - create
    - apiGroups:
      - apps
      resourceNames:
      - cam-install-operator-controller-manager
      resources:
      - deployments/finalizers
      verbs:
      - update
    - apiGroups:
      - ''
      resources:
      - pods
      verbs:
      - get
    - apiGroups:
      - apps
      resources:
      - replicasets
      - deployments
      verbs:
      - get
    - apiGroups:
      - networking.k8s.io
      resources:
      - networkpolicies
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - rbac.authorization.k8s.io
      resources:
      - roles
      - rolebindings
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - cam.management.ibm.com
      resources:
      - manageservices
      - manageservices/finalizers
      - manageservices/status
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - certmanager.k8s.io
      - cert-manager.io
      resources:
      - certificates
      - issuers
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
```

```
      - networking.k8s.io
      resources:
      - NetworkPolicy
      - ingresses
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - batch
      resources:
      - jobs
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - oidc.security.ibm.com
      resources:
      - clients
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - zen.cpd.ibm.com
      resources:
      - zenextensions
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    serviceAccountName: cam-install-operator-controller-manager
ibm-management-im-install:
  clusterPermissions:
  - rules:
    - apiGroups:
      - foundation.ibm.com
      resources:
      - multicluster-hub-nav
      - navconfigurations
      verbs:
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - infra.management.ibm.com
      resources:
      - connections
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - rbac.authorization.k8s.io
      resourceNames:
      - ibm-infra-management-install-operator-connection
      - ibm-infra-management-install-operator-navigation
      resources:
      - clusterrole
      verbs:
      - get
      - list
      - patch
      - update
    serviceAccountName: ibm-infra-management-install-operator
  permissions:
  - rules:
    - apiGroups:
      - ''
```

```
    resources:
    - configmaps
    - events
    - persistentvolumeclaims
    - pods
    - pods/finalizers
    - secrets
    - serviceaccounts
    - services
    - services/finalizers
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ''
    resources:
    - pods/log
    verbs:
    - get
  - apiGroups:
    - apps
    resources:
    - deployments
    - deployments/scale
    - replicasets
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - apps
    resourceNames:
    - ibm-infra-management-application
    - ibm-infra-management-install-operator
    resources:
    - deployments/finalizers
    verbs:
    - update
  - apiGroups:
    - cert-manager.io
    resources:
    - certificates
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - coordination.k8s.io
    resources:
    - leases
    verbs:
    - create
    - delete
    - get
    - list
    - update
  - apiGroups:
    - extensions
    resources:
    - deployments
    - deployments/scale
    - networkpolicies
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - ibmevents.ibm.com
    resources:
    - kafkas
    - kafkatopics
    - kafkausers
    verbs:
    - create
    - delete
```

```
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - infra.management.ibm.com
    resources:
    - iminstalls
    verbs:
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - infra.management.ibm.com
    resources:
    - iminstalls/finalizers
    verbs:
    - update
  - apiGroups:
    - infra.management.ibm.com
    resources:
    - iminstalls/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - manageiq.org
    resources:
    - manageiqs
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - manageiq.org
    resources:
    - manageiqs/finalizers
    verbs:
    - update
  - apiGroups:
    - manageiq.org
    resources:
    - manageiqs/status
    verbs:
    - get
    - patch
    - update
  - apiGroups:
    - monitoring.coreos.com
    resources:
    - servicemonitors
    verbs:
    - create
    - get
  - apiGroups:
    - networking.k8s.io
    resources:
    - ingresses
    - networkpolicies
    verbs:
    - create
    - delete
    - get
    - list
    - patch
    - update
    - watch
  - apiGroups:
    - oidc.security.ibm.com
    resources:
    - clients
    verbs:
    - create
    - delete
    - get
    - patch
    - update
  - apiGroups:
    - operator.ibm.com
    resources:
    - operandrequests
    verbs:
    - create
    - delete
```

```
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - rbac.authorization.k8s.io
      resources:
      - rolebindings
      - roles
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    - apiGroups:
      - route.openshift.io
      resources:
      - routes
      - routes/custom-host
      verbs:
      - create
      - delete
      - get
      - list
      - patch
      - update
      - watch
    serviceAccountName: ibm-infra-management-install-operator
```

Alert and Event schemas

See the following tables to understand Cloud Pak for AIOps alert and event properties.

Cloud Pak for AIOps Alert schema

An alert is a mutable record of something that has happened over a certain period of time. It indicates an anomalous condition.

Cloud Pak for AIOps alert property	Description	Example
id	A system-generated unique id for the alert.	1a2a6787-59ad-4acd-bd0d-46c1ddfd8e06
state	The current state of this alert.	Open
eventCount	The count of events that have contributed to this alert.	2
acknowledged	Flag indicating alert acknowledgement.	false
team	Team with responsibility for the alert.	Db2 admin group
owner	Individual with responsibility for the alert.	user123
deduplicationKey	The key that uniquely identifies the anomalous condition that this alert represents.	{hostname=users-db.myapp.example.com, name=users-db.myapp.example.com, service=users-db.myapp.example.com, type=Service}-High query timeout count:++:Database query timeout-

Cloud Pak for AIOps alert property	Description	Example
signature	An id which uniquely identifies an anomalous condition on a resource. This may be used to identify historic alerts which refer to the same condition on the same resource.	{hostname=users-db.myapp.example.com, name=users-db.myapp.example.com, service=users-db.myapp.example.com, type=Service}-High query timeout count:++:Database query timeout-
firstOccurrenceTime	The time at which the anomalous condition was first detected, in millisecond granularity. Format: yyyy-mm-ddThh:mm:ss.sssZ (ISO8601)	2022-06-14 3:14:10.862
lastOccurrenceTime	The time at which the anomalous condition was last detected, in millisecond granularity. Format: yyyy-mm-ddThh:mm:ss.sssZ (ISO8601)	2022-06-14 3:14:10.862
lastStateChangeTime	The time in millisecond granularity for which the alert was last updated.	2022-06-14 3:14:10.862
summary	A human-readable description of the event.	Fan failure detected on server1.example.com.
langId	ISO 639-3 encoding if not specifed assumption is eng.	eng
severity	Indicates the alert severity level, which indicates how the perceived capability of the managed object has been affected.	5 - Major
sender	Identifies the resource which sent this alert. This may be the resource that is the subject of the event, or it may be some external system which is monitoring the subject.	{"service":"DemoMon","name":"hyper.example.com","type":"Netcool/OMNIbus"}
resource	Identifies the resource that is the subject of this alert. This is made up of multiple fields that describe the resource. There are a set of standard fields which should be used where applicable, but additional fields may be set. **Tip:** If (when querying the topology service APIs for resource data) the optional parameter `_include_status_severity` is set to `true`, the `_hasStatus` field will reflect the maximum and most severe severity of all of the status records related to the resource, for example `"_hasStatus": "critical"`.	{"service":"hyper.example.com","name":"hyper.example.com","hostname":"hyper.example.com", "type":"Service"}
expirySeconds	The number of seconds before the event should automatically expire. A value of zero indicates that there is no expiration.	0
links	An optional array of links to external systems which provide addition information or control over the event, or its subject. This may include a management console for the subject resource, or a page containing further event information from the source monitoring system. Each link should be in the form of a fully qualified URL.	[{"url":"`https://fan-controller.example.com/?fanId=1234`","linkType":"webpage","name":"management-console", "description":"Fan controller management console"}]
details	Additional properties that describe the alert. **Note:** You can include custom fields in the `details` object. Their values must be of type string.	{"additionalProp1":"string","additionalProp2":"string", "additionalProp3":"string"}
insights	An array of insights associated with the alert with one or more other entities or groupings.	insights.topology, insights.runbook, insights.scopeGroup

Cloud Pak for AIOps alert property	Description	Example
relatedIncidentIds	The set of incident id's that this alert is a member of either as a trigger or related context.	["related-incident-2","related-incident-1","related-incident-3"]

Cloud Pak for AIOps Event schema

An event is an immutable record of something happening at a certain time point. It may or may not indicate an anomalous condition.

Cloud Pak for AIOps event property	Description	Example
id	A system-generated unique id for the event.	1a2a6787-59ad-4acd-bd0d-46c1ddfd8e06
occurrenceTime	The time at which this event occurred.	6/15/2022, 3:14:10 PM
summary	A human-readable description of the event.	Fan failure detected on server1.example.com.
severity	Indicates the event severity level, which indicates how the perceived capability of the managed object has been affected.	5 - Major
sender	Identifies the resource which sent this event. This may be the resource that is the subject of the event, or it may be some external system which is monitoring the subject.	{"service":"DemoMon","name": "hyper.example.com","type":"Netcool/OMNIbus"}
resource	Identifies the resource that is the subject of this event. This is made up of multiple fields that describe the resource. There are a set of standard fields which should be used where applicable, but additional fields may be set. **Tip:** If (when querying the topology service APIs for resource data) the optional parameter `_include_status_severity` is set to `true`, the `_hasStatus` field will reflect the maximum and most severe severity of all of the status records related to the resource, for example `"_hasStatus": "critical"`.	{"service":"hyper.example.com","name": "hyper.example.com","hostname":"hyper.example.com", "type":"Service"} . For additional fields in property, see Resource property.
expirySeconds	The number of seconds before the event should automatically expire. A value of zero indicates that there is no expiration.	0
links	An optional array of links to external systems which provide addition information or control over the event, or its subject. This may include a management console for the subject resource, or a page containing further event information from the source monitoring system. Each link should be in the form of a fully qualified URL.	[{"url":"`https://fan-controller.example.com/?fanId=1234`","linkType": "webpage","name":"management-console", "description":"Fan controller management console"}] . For additional fields in property, see Links property.
details	Additional properties that describe the event. **Note:** You can include custom fields in the `details` object. Their values must be of type string.	{"additionalProp1":"string","additionalProp2":"string", "additionalProp3":"string" }
insights	An array of insights associated with the event with one or more other entities or groupings.	insights.topology, insights.runbook, insights.scopeGroup. . For additional fields in property, see Insights property.
type	Values in an object that indicate the type of an event, such as a problem event or a resolution event, and the classification of an event.	{ "classification":":++:CEASelfMonitoring", "eventType":"problem"}. For additional fields in property, see EventType properties.

Alerts and Events schema fields

On this page, find fields that can be included in an Event or Alert schema in IBM Cloud Pak for AIOps for these properties:

- EventType
- Insights
- Links
- Resource

Note: format type used is Open API v3 definition.

EventType property schema

```
EventType:
      description: >
        The classification of the event, i.e. a description of what happened.
      type: object
      properties:
        eventType:
          description: >
            String description of whether the event is a problem
            or resolution event.
          type: string
          enum:
            - problem
            - resolution
          default: problem
        classification:
          description: >
            Description of the type of the event. E.g. Utilization, System
            status, Threshold breach
          type: string
          example: Utilization
        condition:
          description: >
            The condition/status or threshold causing the event. E.g. Down, 95%,
            Unavailable
          type: string
          example: "> 30 seconds"
```

Insights property schema

```
Insight:
      description: >
        Represents an insight reference to one or more other entities provided
        by a specified source that is related to associated element.
      type: object
      properties:
        id:
          description: >
            The unique identifier for this specific relationship, used to
            identify other related entities. Entities with the same insight
            id will participate in the same insight (i.e. this may act as a
            correlation key).
          type: string
          example: 98aa5f81-7da7-4063-bd66-dccaf59097d1
        type:
          description: The type of insight being described
          type: string
          example: ibm.com/causal-alert-relationship
        source:
          description: The source of the insight information
          type: string
          example: ibm.com/causal-correlation/temporal
        details:
          description: >
            Additional details about this insight. The schema of this object
            depends on the insight type.
          type: object
          $ref: "#/components/schemas/InsightDetails"

    AssociatedChange:
      type: object
      properties:
        link:
          type: string
          description: The URL to the associated change in source system.

    Conversation:
      type: object
      properties:
        link:
          type: string
          description: The URL to the conversation in source system.

    Grouping:
      type: object
      properties:
        policyId:
          type: string
          description: The id of the policy that creates the grouping.
        groupDetails:
          type: object
```

```
          description: >
            Details or links on group and/or to explain the reasons for the
            grouping.
    Runbook:
      type: object
      additionalProperties: false
      required:
        - policyId
        - runbookId
        - isAutomated
      properties:
        policyId:
          type: string
          description: The id of the policy that connected this insight.
        runbookId:
          type: string
          format: uuid
          description: The ID of the runbook
        runbookVersion:
          type: integer
          minimum: 1
          description: Optional. Defines whether a fixed version of a runbook should be linked. If, not specified latest
version is used.
        order:
          type: integer
          minimum: 1
          description: Optional. Order for runbook actions. Defines in which order runbooks should be executed.
        isAutomated:
          type: boolean
          description: Defines whether or not the connected Runbook will be automatically executed, once the policy
matches.
        mappings:
          type: array
          description: In order to execute a runbook all required runbook parameters need to be filled out. If the
incident wants to define the values based on the policy or on its own attributes a suitable mapping has to be created.
This is required if automatically triggered execution is desired.
          items:
            type: object
            additionalProperties: false
            properties:
              mappingType:
                type: string
                description: Choose how a parameter is mapped. 'parameter' means using a field/property from the
Incident. 'fixed' means the same value will always be used. 'useDefault' will use the default from the runbook. 'runtime'
leave entry up to the operator.
                enum:
                  - parameter
                  - fixed
                  - useDefault
                  - runtime
              runbookParameterName:
                type: string
                description: The name of the runbook parameter as valued is mapped to.
              parameterValue:
                type: string
                description: The value put into the runbook for the parameter. Depending on the 'mappingType' this value
is optional (useDefault), represents the name of an incident parameter (parameter) or is the value to be put in (fixed).

    Seasonality:
      type: object
      properties:
        policyId:
          type: string
          description: >
            The id of the policy that relates to the seasonality insight.
        seasonalDetails:
          type: object
          description: >
            Details or links on seasonality and/or to explain the reasons for
            the grouping.
    SimiliarIncident:
      type: object
      properties:
        incidentid:
          type: string
          description: id of incident that it related to
        link:
          type: string
          description: The URL to the incident in source system.
        distance:
          description: this should be a double but open API does not seem to like it  :-(
          type: integer

    SimilarIncidents:
      description: >
        Represents a list of similar historical tickets, based on query
        submitted to the service. It also includes a list of recommended
        actions that are extracted from resolutions, which can be highlighted
        with help of their offset information.
      type: object
      properties:
        similar_incidents:
```

```
        description: The list of similar incients for the query.
        type: array
        items:
          $ref: "./similar_incidents.schema.yaml#/components/schemas/SimilarIncidentItem"
      recommended_actions:
        description: The list of extracted actions from resolutions in related incidents.
        type: array
        items:
          $ref: "./similar_incidents.schema.yaml#/components/schemas/RecommendedActionItem"

  AlertResolutions:
    description: Represents a list of resolution links for current alert.
    type: object
    properties:
      shortDescription:
        type: string
        description: Short description of the log anomaly alert.
      description:
        type: string
        description: Description about the current log anomaly alert.
      category:
        type: string
        description: Detected error category.
      subCategory:
        type: string
        description: Detected error sub category.
      resolutionLinks:
        type: array
        description: A list of resolution url links for current alert.
        items:
          type: object
          properties:
            url:
              type: string
              description: URL link for the resolution.
            title:
              type: string
              description: Title for the resolution.
            content:
              type: string
              description: Content excerpt for the resolution.

  AssociatedAlertTemplates:
    description: >
      Represents the error template info for explainable purpose.
    type: object
    properties:
      error_template_list:
        type: array
        items:
          type: string
        description: A list of error template
      error_count_vector:
        type: array
        items:
          type: integer
        description: Count vector corresponding to error template list.
      significant_message_code:
        type: string
        description: Message code for the significant error template.
      type_window:
        type: string
        description: Indicates the window type, e.g., "liberty_twas"

  TopologicalGroup:
    description: >
      Signifies that one of the resources that this entity pertains to is a member
      of a resource group. The id of the insight is the id of the resource group
      within the topology store.
    type: object
    properties:
      name:
        type: string
        description: Human readable name of the resource group
      tags:
        description: A list of any tags assigned to the resource group
        type: array
        items:
          type: string
      entityTypes:
        description: >
          An array of group types that pertain to this group, ex. AiopsApplication which
          signifies this resource group represents an application.
        type: array
        items:
          type: string

  TopologicalResource:
    description: >
      Signifies that this entity pertains to a specific resource within the topology.
      The id of this insight will be set to the unique id of the resource within the topology.
    type: object
```

```
      properties:
        name:
          type: string
          description: Human readable name of the resource
        compositeOfIds:
          type: string
          description: >
            If this resource is a composite resource, contains a comma
            separated list of id's of composed resources.
        compositeId:
          type: string
          description: id of the compositeId

    ProbableCauseRanking:
      type: object
      description: the ranking of alerts in the group
      properties:
        id:
          type: string
          description: in the alert it is the correlationId or the eventId if in the incident
        rank:
          type: integer
          description: the rank of the alert

    IncidentTopology:
      type: object
      description: >
        details of the group id where incident topology is stored.  The incident topology is
        generated by probable cause component
      properties:
        id:
          type: string
          description: asm group id
        time:
          type: integer
          description: time in ms when the last resource in the asm group was added

    Union:
      type: object
      description: >
        Represents a union of insights of a particular type across entities. The
        union is uniquely identified by the id field of the insight.
      properties:
        subsumedUnions:
          type: array
          description: >
            An array of ids of previous union insights which were subsumed by
            this one, due to a new entity occurring that joined them.
          items:
            type: string
            description: The id of a subsumed union insight

    ProposedIncident:
      type: object
      description: >
        Represent a suggested incident to create for this alert.
      properties:
        policyId:
          type: string
          description: The id of the policy that proposes this incident.
        delay:
          type: integer
          description: Suggested hold-off delay in seconds before creating the incident.

    ChatOpsIntegrators:
      type: object
      description: >
        An array of configured integrators for a given incident to hold integrator specific details.
      properties:
        id:
          type: string
          description: Unique id for this integrator
        channel_name:
          type: string
          description: Chatops channel name
        name:
          type: string
          description: kind of chat ops i.e. slack, teams etc
        app_state:
          type: string
          description: Custom state field
        permalink:
          type: string
          description: Permalink for Integrators

    ITSMIntegrators:
      type: object
      description: >
        An array of configured ITSMs for a given incident to hold integrator specific details.
      properties:
        id:
          type: string
```

```
          description: Unique id for this platform
        ticket_num:
          type: string
          description: Ticket number associated with the ITSM system
        name:
          type: string
          description: kind of ITSM i.e. ServiceNow, etc
        app_state:
          type: string
          description: Custom state field
        permalink:
          type: string
          description: Permalink for ITSM platform
        type:
          type: string
          description: type of ITSM Integrators

    ActionHistory:
      description: >
        Represents the actions this entity has passed through.
      type: object
      properties:
        policyId:
          type: string
          description: The id of the policy that triggered the action
        actionId:
          type: string
          description: The id of the action the entity has passed through

    InsightDetails:
      anyOf:
        - $ref: "#/components/schemas/AssociatedChange"
        - $ref: "#/components/schemas/Conversation"
        - $ref: "#/components/schemas/Grouping"
        - $ref: "#/components/schemas/Runbook"
        - $ref: "#/components/schemas/Seasonality"
        - $ref: "#/components/schemas/SimiliarIncident"
        - $ref: "#/components/schemas/SimilarIncidents"
        - $ref: "#/components/schemas/AlertResolutions"
        - $ref: "#/components/schemas/AssociatedAlertTemplates"
        - $ref: "#/components/schemas/TopologicalGroup"
        - $ref: "#/components/schemas/TopologicalResource"
        - $ref: "#/components/schemas/ProbableCauseRanking"
        - $ref: "#/components/schemas/IncidentTopology"
        - $ref: "#/components/schemas/Union"
        - $ref: "#/components/schemas/ChatOpsIntegrators"
        - $ref: "#/components/schemas/ITSMIntegrators"
        - $ref: "#/components/schemas/ActionHistory"
```

Links property schema

```
Link:
      description: >
        Represents a link to an external system which has further information
        or control over an event.
      type: object
      properties:
        linkType:
          description: >
            Describes what type of link this is, giving a hint as to what may
            consume it.
          type: string
          example: webpage
        name:
          description: >
            An identifier for this link that can be referred to programmatically
          type: string
          example: management-console
        description:
          description: Descriptive text which defines what the link leads to
          type: string
          example: Fan controller management console
        url:
          description: >
            A fully qualified URL that can be used to locate the target of the
            link
          type: string
          example: https://fan-controller.example.com/?fanId=1234
```

Resource property schema

```
Resource:
      description: >
        Identifies a resource within the environment. This is made up of
        multiple fields that describe the resource. There are a set of standard
        fields which should be used where applicable, but additional fields may
        be set.
      type: object
      additionalProperties:
        description: Additional event-specific resource locators
        $ref: '#/components/schemas/StringOrInteger'
      properties:
```

```
        type:
          description: The type of resource
          type: string
          example: host
        name:
          description: The name of the resource
          type: string
          example: myhost.example.com
        sourceId:
          type: string
          description: The id the resource is known by in the source system
        hostname:
          description: The hostname of the resource
          type: string
          example: myhost.example.com
        ipAddress:
          description: The IP address of the resource
          type: string
          example: 9.123.123.1
        service:
          description: The name of the service that the resource is serving
          type: string
          example: mobile-app
        port:
          description: The port that is the subject of this event
          $ref: '#/components/schemas/StringOrInteger'
          example: "80"
        interface:
          description: The interface that is the subject of this event
          type: string
          example: eth0
        application:
          type: string
          description: The application that is the subject of this event
        controller:
          type: string
          description: The controller that is the subject of this event
        component:
          type: string
          description: The component that is the subject of this event
        cluster:
          type: string
          description: The cluster that is the subject of this event
        location:
          description: The location of the resource
          type: string
          example: Dallas 10
        accessScope:
          type: string
          description: The project or namespace the resource is part of
          example: default
        connectionId:
          type: string
          description: >-
            Configuration unique identifier of the integration this event
            came from.
          example: 'f5aa7fa9-92eb-4bec-942c-37eb3e3e9601'
```

Resource schemas

For more information about IBM Cloud Pak® for AIOps resource properties, see the following table.

Table. Resource schemas

Cloud Pak for AIOps resource property	Description	Type	Example
type	The type of the resource	string	host
name	The name of the resource	string	myhost.example.com
sourceId	The ID that the resource is known by in the source system	string	`7045b61d-e59c-4270-b525-545d55ecd06e`
hostname	The hostname of the resource	string	myhost.example.com
ipAddress	The IP address of the resource	string	9.123.123.1
service	The name of the service that the resource is serving	string	mobile-app
port	The port that is the subject of this event	string or integer	`ssh` or `22`
interface	The interface that is the subject of this event	string	eth0
application	The application that is the subject of this event	string	mobile-app-1
controller	The controller that is the subject of this event	string	controller1
component	The component that is the subject of this event	string	database
cluster	The cluster that is the subject of this event	string	staging-cluster
location	The location of the resource	string	Dallas 10
accessScope	The project or namespace that the resource is part of	string	default

Cloud Pak for AIOps resource property	Description	Type	Example
connectionId	The unique identifier of the integration that this event came from	string	`f5aa7fa9-92eb-4bec-942c-37eb3e3e9601`
scopeId	The scope identifier value for this event from the originating system	string	myhost.example.com:location01

You can also define more resource properties for resources that are not included in the table. Added properties with valid JSON are stored and available, but special knowledge or handling of them is not included.

EventType schemas

EventType schemas indicate whether an event is a problem or a resolution and how it is classified. For more information about IBM Cloud Pak® for AIOps EventType properties, see the following table.

Cloud Pak for AIOps resource property	Description	Type	Example
eventType	Describes whether the event is a problem or resolution event. The only possible values that can be set are `problem` or `resolution`.	string	`problem`, `resolution`
classification	Describes the type of the event. This property is used to determine the golden signals for an alert.	string	`system status`, `threshold breach`, `utilization`
condition	Provides context to the type of the event and states the condition, status, or threshold that caused the event.	string	`> 30 seconds`, `95%`, `Down`, `Unavailable`

Normalized Mapping Rules

To help you map event data from incoming data sources review the following event schemas and samples.

- Kafka integration normalized event schema
  - Normalized PagerDuty events sample
  - Normalized IBM Netcool Operations Insight events sample
- PagerDuty normalized mapping rules
  - Sample PagerDuty event
- IBM Netcool Operations Insight normalized mapping rules
  - Sample IBM Netcool Operations Insight event

Kafka integration normalized event schema

```
{
"$schema": "http://json-schema.org/draft-04/schema#",
"title": "Kafka Integration normalized event schema",
"description": "Normalized event for None mapping use case",
"type": "object",
"properties": {
   "sender": {
   "description": "Identifies the resource which sent this event. This may be the resource that is the subject of the
event, or it may be some external system which is monitoring the subject.",
   "type": "object",
   "properties": {
      "service": {
         "description": "The identifier of the service",
         "type": "string"
      },
      "name": {
         "description": "The descriptive name of the service",
         "type": "string"
      },
      "type": {
         "description": "Data source name. For examples, PagerDuty, Netcool/OMNIbus",
         "type": "string"
      }
   },
   "required": [
      "service",
      "name",
      "type"
   ]
   },
   "resource": {
   "description": "Identifies the resource that is the subject of this event. This is made up of multiple fields that
describe the resource. There are a set of standard fields which should be used where applicable, but additional fields
may be set.",
   "type": "object",
   "properties": {
      "name": {
         "description": "The name of the resource. For example, myhost.example.com",
```

```
         "type": "string"
      },
      "hostname": {
         "description": "The hostname of the resource.",
         "type": "string"
      },
      "type": {
         "description": "The type of resource. For example, host",
         "type": "string"
      },
      "ipaddress": {
         "description": "The IP address of the resource.",
         "type": "string"
      },
      "location": {
         "description": "The location of the resource. For example, Dallas 10",
         "type": "string"
      }
   },
   "required": [
      "name",
      "hostname",
      "type",
      "ipaddress",
      "location"
   ]
   },
   "type": {
   "description": "Indicates whether the source alert is a problem or a resolution along with its classification.",
   "type": "object",
   "properties": {
      "classification": {
         "description": "Description of the type of the event. E.g. Utilization, System status, Threshold breach",
         "type": "string"
      },
      "eventType": {
         "description": "String description of whether the event is 'problem' or 'resolution'. Note: Only these two
values are permitted.",
         "type": "string"
      }
   },
   "required": [
      "classification",
      "eventType"
   ]
   },
   "severity": {
   "description": "Indicates the event severity level, which indicates how the perceived capability of the managed object
has been affected.",
   "type": "integer",
   "minimum": 0,
   "maximum": 6
   },
   "summary": {
   "description": "A human-readable description of the event.",
   "type": "string"
   },
   "occurrenceTime": {
   "description": "The time at which this event occurred. (ISO format: yyyy-mm-ddThh:mm:ss.SSSZ)",
   "type": "string",
   "pattern": "\\d{4}-[01]\\d-[0-3]\\dT[0-2]\\d:[0-5]\\d:[0-5]\\d(?:\\.\\d+)Z"
   },
   "expirySeconds": {
   "description": "The number of seconds before the event should automatically expire. A value of zero indicates that
there is no expiration.",
   "type": "integer",
   "minimum": 0
   }
},
"required": [
   "sender",
   "resource",
   "type",
   "severity",
   "summary",
   "occurrenceTime",
   "expirySeconds"
]
}
}
```

Normalized PagerDuty events sample

```
{
   "sender":{
      "service": "PEPF123",
      "name": "WAS_Sysdig",
      "type": "PagerDuty"
   },
   "resource":{
```

```
      "name":"Hostdown",
      "hostname":"alerts.sample.com",
      "type":"host",
      "ipaddress":"10.123.123.123",
      "location":"sample.mail.ibm.com"
   },
   "type":{
      "classification":"Pods ready less than desired is Triggered on kubernetes.cluster.name = RHOpenShift and
kubernetes.daemonSet.name = sysdig-agent",
      "eventType":"problem"
   },
   "severity":5,
   "summary":"Pods ready less than desired is Triggered on kubernetes.cluster.name = RHOpenShift and
kubernetes.daemonSet.name = sysdig-agent",
   "occurrenceTime":"2021-10-07T07:55:46.000Z",
   "expirySeconds":0
}
```

Normalized IBM Netcool Operations Insight events sample

```
{
   "sender":{
      "service":"CEA SelfMonitoring",
      "name":"CEA SelfMonitoring @AGG_P",
      "type":"Netcool/OMNIbus"
   },
   "resource":{
      "name":"AGG_P",
      "hostname":"samplehost.ibm.com",
      "type":"host",
      "ipaddress":"10.1.96.24",
      "location":"Tokyo"
   },
   "type":{
      "classification":":++:CEASelfMonitoring",
      "eventType":"problem"
   },
   "severity":4,
   "summary":"CEAAnalytics SelfMonitoring Alert",
   "occurrenceTime":"2021-10-05T14:50:38.000Z",
   "expirySeconds":0
}
```

PagerDuty normalized mapping rules

Note: Rules take precedence, that is the mapped value is taken from the highest rules that fulfils the condition.

Table. PagerDuty Normalized Mapping Rules

Resultant Normalized Alert			Mapping rules
{			
	"sender": {		
		"service": "P3ZB24W",	{{**service.id**}}
		"name": "PagerDuty - [SI] Beta - Website - Degraded",	"PagerDuty -" + {{**service.summary**}}
		"type": "PagerDuty"	"PagerDuty"
	},		
	"resource": {		
		"name": "Beta URL Down",	Rule 1: {{**body.cef_details.source_component**}}
			Rule 2: {{**incident.summary**}}
		"hostname": "alerts.newrelic.com",	Rule 1: hostname of {{**body.cef_details.details.incident_url**}}
			Rule 2: hostname of {{**incident.self**}}
		"type": "host",	Rule 1: "host" if the "**hostname**" field (see above) is hostname.
			Rule 2: "ip" if the "**hostname**" field (see above) is IPv4.
			Rule 3: Otherwise, empty.
		"ipaddress": "",	Rule 1: Same value as the "**hostname**" field if the value is IPv4
			Rule 2: Otherwise, empty.
		"location": "Beta URL GUI"	Rule 1: {{**body.cef_details.source_origin**}}
			Rule 2: {{**incident.summary**}}
	},		

Resultant Normalized Alert			Mapping rules
	"type": {		
		"classification": "Beta URL Down (Beta URL GUI violated Beta URL Down)",	{{**summary**}}
		"eventType": "resolution"	Rule 1: "resolved" if {{**status**}} = "resolved"
			Rule 2: Otherwise "problem".
	},		
	"severity": 5,		{{**severity**}} -> IBM Cloud Pak for AIOps severity level info -> 1
			low -> 2
			warning -> 3
			error -> 4
			critical -> 5
			Otherwise, 1
	"summary": "Beta URL Down (Beta URL GUI violated Beta URL Down)",		{{**summary**}}
	"occurrenceTime": "2021-07-11T03:39:53.000Z",		{{**created_at**}} converted to yyyy-mm-ddT:hh:mm:ss.000Z
	"expirySeconds": 0		0
}			

Sample PagerDuty event used in the mapping


```
{
  "id": "Q2BSJKUORBWAWT",
  "type": "alert",
  "summary": "Beta URL Down (Beta URL GUI violated Beta URL Down)",
  "self": "https://api.pagerduty.com/alerts/Q2BSJKUORBWAWT",
  "html_url": "https://ibm-sandbox.pagerduty.com/alerts/Q2BSJKUORBWAWT",
  "created_at": "2021-07-10T20:39:53-07:00",
  "status": "resolved",
  "resolved_at": "2021-07-10T21:41:18-07:00",
  "alert_key": "799c00fbcf461bf553f9b4f30bcc1d9908db947c",
  "suppressed": false,
  "service": {
    "id": "P3ZB24W",
    "type": "service_reference",
    "summary": "[SI] Beta - Website - Degraded",
    "self": "https://api.pagerduty.com/services/P3ZB24W",
    "html_url": "https://ibm-sandbox.pagerduty.com/service-directory/P3ZB24W"
  },
  "severity": "critical",
  "incident": {
    "id": "P3HUMBH",
    "type": "incident_reference",
    "summary": "[#42113] Beta URL Down (Beta URL GUI violated Beta URL Down)",
    "self": "https://api.pagerduty.com/incidents/P3HUMBH",
    "html_url": "https://ibm-sandbox.pagerduty.com/incidents/P3HUMBH"
  },
  "first_trigger_log_entry": {
    "id": "R4Z5TVA4KYEVQL6DIFD9SHYPOY",
    "type": "trigger_log_entry_reference",
    "summary": "Triggered through the API",
    "self": "https://api.pagerduty.com/log_entries/R4Z5TVA4KYEVQL6DIFD9SHYPOY",
    "html_url": "https://ibm-sandbox.pagerduty.com/alerts/Q2BSJKUORBWAWT/log_entries/R4Z5TVA4KYEVQL6DIFD9SHYPOY"
  },
  "body": {
    "contexts": [],
    "details": {
      "VIOLATION DESCRIPTION": null,
      "account_id": 2848358,
      "account_name": "Storage Insights",
      "closed_violations_count": {
        "critical": 0,
        "warning": 0
      },
      "condition_family_id": 15905725,
      "condition_id": 114530789,
      "condition_name": "Beta URL Down",
      "current_state": "open",
      "details": "2 or more locations failed on 'Beta URL GUI'",
      "duration": 65,
      "event_type": "INCIDENT",
      "incident_acknowledge_url": "https://alerts.newrelic.com/accounts/2848358/incidents/313256636/acknowledge",
      "incident_id": 313256636,
      "incident_url": "https://alerts.newrelic.com/accounts/2848358/incidents/313256636",
      "metadata": {
        "entity.name": "Beta URL GUI",
        "entity.type": "MONITOR",
        "location_statuses": [
          {
            "location": "AWS_EU_CENTRAL_1",
```

```
            "locationLabel": "Frankfurt, DE",
            "result": "FAILED",
            "timestamp": 1625974724220
          },
          {
            "location": "AWS_US_WEST_1",
            "locationLabel": "San Francisco, CA, USA",
            "result": "FAILED",
            "timestamp": 1625974281070
          },
          {
            "location": "AWS_SA_EAST_1",
            "locationLabel": "São Paulo, BR",
            "result": "FAILED",
            "timestamp": 1625974154020
          },
          {
            "location": "AWS_AP_SOUTH_1",
            "locationLabel": "Mumbai, IN",
            "result": "FAILED",
            "timestamp": 1625974214555
          }
        ]
      },
      "open_violations_count": {
        "critical": 1,
        "warning": 0
      },
      "owner": "",
      "policy_name": "Beta URL Down",
      "policy_url": "https://alerts.newrelic.com/accounts/2848358/policies/1029635",
      "severity": "CRITICAL",
      "targets": [
        {
          "id": "03e19e74-cf92-4ea5-a0db-2e6d968fc9f9",
          "labels": {
            "account": "Storage Insights",
            "accountId": "2848358",
            "guid": "Mjg0ODM1OHxTWU5USHxNT05JVE9SfDAzZTE5ZTc0LWNmOTItNGVhNS1hMGRiLTJlNmQ5NjhmYzlmOQ",
            "monitorStatus": "Enabled",
            "monitorType": "Simple Browser",
            "period": "10",
            "publicLocation": "São Paulo, BR",
            "trustedAccountId": "1592783"
          },
          "link": "https://synthetics.newrelic.com/accounts/2848358/monitors/03e19e74-cf92-4ea5-a0db-2e6d968fc9f9",
          "name": "Beta URL GUI",
          "product": "SYNTHETICS",
          "type": "Monitor"
        }
      ],
      "timestamp": 1625974792836,
      "timestamp_utc_string": "2021-07-11, 03:39 UTC",
      "version": "1.0",
      "violation_callback_url": "https://synthetics.newrelic.com/accounts/2848358/monitors/03e19e74-cf92-4ea5-a0db-
2e6d968fc9f9"
    },
    "cef_details": {
      "client": "New Relic",
      "client_url": "https://alerts.newrelic.com/accounts/2848358/incidents/313256636",
      "creation_time": "2021-07-11T03:39:53.673Z",
      "dedup_key": "799c00fbcf461bf553f9b4f30bcc1d9908db947c",
      "description": "Beta URL Down (Beta URL GUI violated Beta URL Down)",
      "details": {
        "VIOLATION DESCRIPTION": null,
        "account_id": 2848358,
        "account_name": "Storage Insights",
        "closed_violations_count": {
          "critical": 0,
          "warning": 0
        },
        "condition_family_id": 15905725,
        "condition_id": 114530789,
        "condition_name": "Beta URL Down",
        "current_state": "open",
        "details": "2 or more locations failed on 'Beta URL GUI'",
        "duration": 65,
        "event_type": "INCIDENT",
        "incident_acknowledge_url": "https://alerts.newrelic.com/accounts/2848358/incidents/313256636/acknowledge",
        "incident_id": 313256636,
        "incident_url": "https://alerts.newrelic.com/accounts/2848358/incidents/313256636",
        "metadata": {
          "entity.name": "Beta URL GUI",
          "entity.type": "MONITOR",
          "location_statuses": [
            {
              "location": "AWS_EU_CENTRAL_1",
              "locationLabel": "Frankfurt, DE",
              "result": "FAILED",
              "timestamp": 1625974724220
            },
            {
```

```
              "location": "AWS_US_WEST_1",
              "locationLabel": "San Francisco, CA, USA",
              "result": "FAILED",
              "timestamp": 1625974281070
            },
            {
              "location": "AWS_SA_EAST_1",
              "locationLabel": "São Paulo, BR",
              "result": "FAILED",
              "timestamp": 1625974154020
            },
            {
              "location": "AWS_AP_SOUTH_1",
              "locationLabel": "Mumbai, IN",
              "result": "FAILED",
              "timestamp": 1625974214555
            }
          ]
        },
        "open_violations_count": {
          "critical": 1,
          "warning": 0
        },
        "owner": "",
        "policy_name": "Beta URL Down",
        "policy_url": "https://alerts.newrelic.com/accounts/2848358/policies/1029635",
        "severity": "CRITICAL",
        "targets": [
          {
            "id": "03e19e74-cf92-4ea5-a0db-2e6d968fc9f9",
            "labels": {
              "account": "Storage Insights",
              "accountId": "2848358",
              "guid": "Mjg0ODM1OHxTWU5USHxNT05JVE9SfDAzZTE5ZTc0LWNmOTItNGVhNS1hMGRiLTJlNmQ5NjhmYzlmOQ",
              "monitorStatus": "Enabled",
              "monitorType": "Simple Browser",
              "period": "10",
              "publicLocation": "São Paulo, BR",
              "trustedAccountId": "1592783"
            },
            "link": "https://synthetics.newrelic.com/accounts/2848358/monitors/03e19e74-cf92-4ea5-a0db-2e6d968fc9f9",
            "name": "Beta URL GUI",
            "product": "SYNTHETICS",
            "type": "Monitor"
          }
        ],
        "timestamp": 1625974792836,
        "timestamp_utc_string": "2021-07-11, 03:39 UTC",
        "version": "1.0",
        "violation_callback_url": "https://synthetics.newrelic.com/accounts/2848358/monitors/03e19e74-cf92-4ea5-a0db-
2e6d968fc9f9"
      },
      "message": "Beta URL Down (Beta URL GUI violated Beta URL Down)",
      "mutations": [],
      "reporter_component": "08fe44fc923247aa9b931d186559a105",
      "severity": "critical",
      "source_component": "Beta URL Down",
      "source_origin": "Beta URL GUI",
      "version": "1.0"
    },
    "type": "alert_body"
  },
  "integration": {
    "id": "PL68E48",
    "type": "generic_events_api_inbound_integration_reference",
    "summary": "New Relic",
    "self": "https://api.pagerduty.com/services/P3ZB24W/integrations/PL68E48",
    "html_url": "https://ibm-sandbox.pagerduty.com/services/P3ZB24W/integrations/PL68E48"
  },
  "privilege": null
}
```


IBM Netcool Operations Insight normalized mapping rules

Note: Rules take precedence, that is the mapped value is taken from the highest rules that fulfils the condition.

Table. IBM Netcool Operations Insight normalized mapping rules

Resultant Normalized Alert			Mapping rules
{			
	"sender": {		
		"service": "Ping Probe",	{{**Agent**}}
		"name": "Ping Probe",	{{**Manager**}}
		"type": "Netcool/OMNIbus"	"Netcool/OMNIbus"
	},		

Resultant Normalized Alert			Mapping rules
	"resource": {		
		"name": "RNIB-MPLS-FTTC013",	Rule 1: Take {{**Node**}} if it is not empty
			Rule 2: Take {{**NodeAlias**}} if it is not empty
			Rule 3: Rule 3: Otherwise "unknown"
		"hostname": "RNIB-MPLS-FTTC013",	Rule 1: {{**Node**}} if not empty and is valid hostname
			Rule 2: {{**NodeAlias**}} if not empty and is valid hostname
		"type": "host",	Rule 1: "host" if either {{**Node**}} or {{**NodeAlias**}} is not empty,
			Rule 2: Otherwise, "unknown".
		"ipaddress"10.44.113.88",	Rule 1: {{**Node**}} if not empty and is valid IPv4
			Rule 2: {{**NodeAlias**}} if not empty and is valid IPv4
			Rule 3: Otherwise, empty
		"location": "Paper St"	{{**Location**}}
	},		
	"type": {		
		"classification": "probeping-icmptimeout:++:Ping Status",	{{**EventId**}} ":++:" {{**AlertGroup**}}
		"eventType": "problem"	Rule 1: "resolution" if {{**Severity**}} is 0.
			Rule 2: "resolution" if {{**Type**}} is one of {2,4}
			Rule 3: Otherwise, "problem"
	},		
	"severity": 3,		Rule 1: {{**Severity**}} + 1 if {{**Severity**}} falls within 2 to 5.
			Rule 2: 1 if {{**Severity**}} is 1
			Rule 3: 2 if {{**Severity**}} is > 0
	"summary": "RNIB-MPLS-FTTC013 is not reachable",		{{**Summary**}}
	"occurrenceTime": "2021-02-23T09:56:46.000Z",		Output format: yyyy-mm-ddThh:mm:ss.SSSZ
			Rule 1: If {{**NetcoolEventAction**}} is "INSERT", then {{**FirstOccurrence**}} - yyyy-mm-ddThh:mm:ss {{**FirstOccurrenceUSec**}} - if available then .SSSZ, otherwise .000Z
			Rule 2: If {{**NetcoolEventAction**}} is "UPDATE", then {{**LastOccurrence**}} - yyyy-mm-ddThh:mm:ss {{**LastOccurrenceUSec**}} - if available then .SSSZ, otherwise .000Z
			Note: {{**FirstOccurrence**}} and {{**LastOccurrence**}} can be in Unix epoch or datetime string.
	"expirySeconds": 0		Rule 1: Take {{**ExpireTime**}} if the field exists.
			Rule 2: Otherwise, 0.
}			

Sample IBM Netcool Operations Insight event used in the mapping


```
  {
    "Identifier": "RNIB-MPLS-FTTC013 Ping Status:10.44.113.88 Ping Status 1 Ping Probe Ping Probe RNIB-MPLS-FTTC013
unreachable",
    "SERIAL": 25230712,
    "Node": "RNIB-MPLS-FTTC013",
    "NodeAlias": "10.44.113.88",
    "Manager": "Ping Probe",
    "Agent": "Ping Probe",
    "AlertGroup": "Ping Status",
    "AlertKey": "Ping Status:10.44.113.88",
    "Severity": 3,
    "Summary": "RNIB-MPLS-FTTC013 is not reachable",
    "StateChange": "1614074206",
    "FirstOccurrence": "1614074206",
    "LastOccurrence": "1614074206",
    "LASTMODIFIED": "1614074206",
    "Poll": 0,
    "Type": 1,
    "Tally": 1,
    "Class": 100,
    "Grade": 0,
    "Location": " Paper St",
    "OwnerUID": 65534,
    "OwnerGID": 0,
    "Acknowledged": 0,
    "Flash": 0,
    "EventId": "probeping-icmptimeout",
    "ExpireTime": 0,
    "ProcessReq": 0,
```

```
    "SuppressEscl": 0,
    "Customer": "RNIB",
    "Service": "",
    "PhysicalSlot": 0,
    "PhysicalPort": 0,
    "PhysicalCard": "",
    "TaskList": 0,
    "NmosSerial": "",
    "NmosObjInst": 0,
    "NmosCauseType": 0,
    "LocalNodeAlias": "10.44.113.88",
    "LocalPriObj": "",
    "LocalSecObj": "",
    "LocalRootObj": "",
    "RemoteNodeAlias": "",
    "RemotePriObj": "",
    "RemoteSecObj": "",
    "RemoteRootObj": "",
    "X733EventType": 0,
    "X733ProbableCause": 0,
    "X733SpecificProb": "",
    "X733CorrNotif": "",
    "ServerName": " AGG_P",
    "ServerSerial": 25230712,
    "DELETEDAT": "2015-01-01T01:06:22",
    "ACKOWNERUID": 0,
    "ALERTCATEGORY": "0",
    "CIRCUIT": "BBEU07581255",
    "DESCRIPTION": "A device is not responding to ICMP ping monitoring",
    "EVENTCATEGORY": "1",
    "EVENTOWNERGID": 0,
    "EVENTSUBCATEGORYONE": "1",
    "EVENTSUBCATEGORYTWO": "1",
    "ITMDISPLAYITEM": "",
    "ITMSITFORMULA": "",
    "ITMSITTYPE": "S",
    "LINEID": "",
    "PARTYID": "0",
    "TICKETID": 0,
    "TIMEESCL": "1970-01-01T01:00:00",
    "TIMEFIRSTACK": "1970-01-01T01:00:00",
    "TIMEJOURN": "1970-01-01T01:00:00",
    "TIMELASTACK": "1970-01-01T01:00:00",
    "TIMETICKET": "1970-01-01T01:00:00",
    "CUSTOMERTIER": "0",
    "CUSTOMERSCORE": "0",
    "EVENTPRIORITY": "93",
    "EVENTORIGIN": "1",
    "TIMECLEAR": "2015-01-01T00:05:15",
    "CIRCUITTYPE": "ADSL",
    "SNDCCLASSNAME": "cmdb_ci_ip_router",
    "SNDCMESSINASHORTNAME": "RNIBA",
    "SNDCSYSID": "de16ced88712e004055fe5b8c9434d54",
    "MANUFACTURER": "Cisco",
    "CUSTOMERALIASNAME": "Paper St - ADSL [CIRCUITID:MKVL435293]",
    "SNDCMESSINAOAID": "1191313",
    "SNDCACCESSROUTER": "",
    "SNDCACCESSROUTERPORT": "",
    "SNDCBEARER": "",
    "SNDCINSTALLNUMBER": "01509269339 ",
    "SNDCDSLCARETYPE": "Enhanced",
    "TICKETREF": "",
    "TELCO": "ACME Telecom",
    "SERVICEAFFECTING": "0",
    "CUSTOMERREFERENCE": "UK-RNIBA",
    "TIMEEVENTOWNERGID": "1970-01-01T01:00:00",
    "TICKETREFLAST": "",
    "SUPPORTSEGMENT": "2",
    "NETWORKINTERFACE": "",
    "URL": "",
    "CORRFLAG": "0",
    "PARENTCHILD": "0",
    "PARENTSERIAL": 0,
    "NEWSUMMCOUNT": -1,
    "CMSURI": "",
    "ENTITYTYPE": "Circuit_ADSL",
    "NetcoolEventAction": "insert"
  }
```


IBM Cloud Pak for AIOps troubleshooting

Learn how to isolate and resolve problems with IBM Cloud Pak® for AIOps.

Verify that your issues are not related to operating system requirements, such as disk, memory, and CPU capacities. For more information, see Planning.

Additionally, you can see whether the problem is documented in Known issues and limitations.

To potentially avoid problems, run routine health checks to analyze your system for potential issues. For more information about health check options for IBM Cloud Pak for AIOps, see Health checks and monitoring.

If you need direct help from IBM Support, see the Support.

Troubleshooting topics

- Install and upgrade
- Clusters and nodes
- Troubleshooting datastores
- Integrations
- AI model management
- ChatOps
- Topology
- Troubleshooting Identity Management (IM) issues
- Troubleshooting certificates
- Secure Tunnel
- Alert management
- Runbook Automation
- Infrastructure Automation

Troubleshooting installation and upgrade

Learn how to isolate and resolve problems with installing and upgrading IBM Cloud Pak for AIOps.

Known issues and limitations

In addition to the common problems and solutions that are listed, review the installation related known issues for potential issues that you might encounter during installation. For more information, see Known issues and limitations.

Troubleshooting tips

- General troubleshooting tips (Console)
- General troubleshooting tips (CLI)
- Troubleshooting storage
- Recovering IBM Cloud Pak for AIOps after a cluster restart

Note: If you have an aborted or stalled installation, uninstall any installed or partially installed components and restart the installation. For more information about uninstalling, see the following topics:

- Uninstalling IBM Cloud Pak for AIOps on Linux
- Uninstalling IBM Cloud Pak for AIOps on OpenShift

Installation issues

- Installation fails to proceed when creating the IBM Cloud Pak for AIOps custom resource
- The AI manager operator does not progress past `Reconciling`, and the Cassandra setup pod is stuck
- "Application is not available" message is displayed intermittently
- Installation on Linux fails with network connectivity errors when using `aiopsctl` to install on vSphere VMs
- Installation on Linux: deployment is unstable after cluster restart
- Installation on Linux: unable to log in to IBM Cloud Pak for AIOps

Upgrade issues

- After upgrade, policies do not process alerts due to problem with the `aiops-ir-lifecycle-create-policies-job`
- After upgrade, a metric integration goes into a failed state
- After upgrade, alerts are not visible in the UI
- Navigation menu items disappear after an upgrade
- Database migration is stuck
- Elasticsearch CA renewal rollout becomes stuck and causes the *Policies* page to not display policies

Issues common to installation and upgrade

- Offline install or upgrade throws 'invalid character' error
- Offline install or upgrade stuck because the 'oc ibm-pak generate mirror-manifests' command fails with 'no space left on device'

General troubleshooting

General troubleshooting tips (Console)

1. From the Red Hat® OpenShift® console, go to **Workloads** > **Pods**.
2. Select the project that IBM Cloud Pak for AIOps is installed in from the drop-down menu to view the status of all the pods in this project.
3. If you want to see the status and details for one of the pods, click the pod that you want to investigate.
4. If you want to see the logs for the pod, click **Logs**.

General troubleshooting tips (CLI)

- To find out the status of all the pods in a project, run the following command:

  ```
  oc get pods -n <project>
  ```

  Where **<project>** is the project (namespace) where IBM Cloud Pak for AIOps is deployed.

- To see the status of a particular pod, run the following command:

  ```
  oc get pods | grep <pod_name>
  ```

  Where **<pod_name>** is the pod that you want to see status for.

- To see a pod's logs, run the following command:

  ```
  oc logs -f <pod_name> -n <project>
  ```

  Where

  - **<pod_name>** is the pod that you want to see status for.
  - **<project>** is the project (namespace) where IBM Cloud Pak for AIOps is deployed.

Troubleshooting storage

Storage that is associated with persistent volumes claims (PVCs) reaches capacity. This issue can occur as storage requests and resources fluctuate over time.

Solution: If storage fills up, increase the size of the PVC. For more information about increasing the size of a PVC, see Storage class requirements.

Recovering IBM Cloud Pak for AIOps after a cluster restart

After a Red Hat® OpenShift® Container Platform cluster restart, your IBM Cloud Pak for AIOps deployment has not come up successfully.

Use the following steps to help you to identify any problems.

1. Run the following command and then check that all of the OpenShift nodes come back up, and that they all report a *STATUS* of **Ready**.

   ```
   oc get nodes
   ```

   If all of the OpenShift nodes have not come back up, then see the *Updating clusters* and *Troubleshooting* sections in the OpenShift documentation for guidance.

2. Run the following command and then check that the IBM Cloud Pak for AIOps pods all have a *STATUS* OF **Running** or **Completed**.

   ```
   oc get pods -n <project>
   ```

   Where **<project>** is the project (namespace) where IBM Cloud Pak for AIOps is deployed.

3. Check the logs of any pods that did not report a *STATUS* of **Running** or **Completed** in steps 2 and 3 to see whether you can identify a cause.

   ```
   oc logs -f <failed_pod_name> -n <project>
   ```

   Where

   - **<project>** is the project (namespace) where IBM Cloud Pak for AIOps is deployed.
   - **<failed_pod_name>** is a failed pod name, as returned in steps 2 and 3.

Solution: Restart any failed pods with the following command:

```
oc delete pod <failed_pod_name> -n <project>
```

Where

- **<project>** is the project (namespace) where IBM Cloud Pak for AIOps is deployed.
- **<failed_pod_name>** is the pod name as returned in steps 2 and 3.

Troubleshooting installation

Installation not proceeding after creating the IBM Cloud Pak for AIOps custom resource

When you are installing, the installation is not proceeding as expected after you create an instance of the IBM Cloud Pak for AIOps custom resource. For instance, after you create the instance, the installation process appears to be stalled, however no pods in the namespace and cluster display show errors.

This stalling of the installation process can be due to an issue with the Operator Lifecycle Manager (OLM), which causes the expected ClusterServiceVersion (CSV) for some operator subscriptions to not be found. For example, installPlan instances for the **aimanager-operator**, **connector-utilities-operator**, and other operators can be found, but without a reference to the required ClusterServiceVersion (CSV) YAML. While the operators can be created within your defined namespace, if the ClusterServiceVersion (CSV) cannot be found, the operators cannot be managed. If this issue occurs, the overall installation cannot proceed correctly.

Solution: Delete any operator subscription that does not have an associated ClusterServiceVersion (CSV) YAML and has an unresolved subscription. The installation process should re-create the associated operator and generate a new installPlan. If a new installPlan is not generated, repeat the process to delete the operator subscription and try again.

The AI manager operator does not progress past `Reconciling`, and the Cassandra setup pod is stuck

When installing, the AI Manager operator remains in **Phase: Reconciling**. The **aiops-ir-analytics-cassandra-setup** pod is in Crashloopbackoff state, and its log has the following error:

```
ERROR [2023-06-13 09:57:44,010] com.ibm.itsm.analytics.metric.service.cassandra.SessionManager: Cassandra not available.
```

Solution: Use the following steps to reset the Cassandra admin password, and delete the **aiops-ir-analytics-cassandra-setup** and **aiops-ir-core-archiving-setup** jobs.

1. Run the following command to switch to your IBM Cloud Pak for AIOps project.

   ```
   oc project <project>
   ```

   Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deploying in.

2. Reset the Cassandra admin password.

   ```
   oc rsh aiops-topology-cassandra-0 bash -c 'python3 -u /opt/ibm/change-superuser.pyc --initial --user admin --password $(cat /opt/ibm/cassandra_auth/password) --replication-factor $CASSANDRA_AUTH_REPLICATION_FACTOR'
   ```

3. Delete the **aiops-ir-analytics-cassandra-setup** job, if it does not own any pods.

   1. Check if there are any pods belonging to the **aiops-ir-analytics-cassandra-setup** job.

      ```
      oc get pod -l app.kubernetes.io/managed-by=aiops-analytics-operator,app.kubernetes.io/component=cassandra-setup
      ```

   2. If no pods are returned, then delete the **aiops-ir-analytics-cassandra-setup** job.

      ```
      oc delete job -l app.kubernetes.io/managed-by=aiops-analytics-operator,app.kubernetes.io/component=cassandra-setup
      ```

4. Delete the **aiops-ir-core-archiving-setup** job, if it does not own any pods.

   1. Check if there are any pods belonging to the **aiops-ir-core-archiving-setup** job.

      ```
      oc get pod -l app.kubernetes.io/managed-by=ir-core-operator,app.kubernetes.io/component=archiving-setup
      ```

   2. If no pods are returned, then delete the **aiops-ir-core-archiving-setup** job.

      ```
      oc delete job -l app.kubernetes.io/managed-by=ir-core-operator,app.kubernetes.io/component=archiving-setup
      ```

"Application is not available" message is displayed intermittently.

This is caused by a defect in some versions of Red Hat OpenShift. For more information, see OpenShift 4 - Intermittent 503 from all routes or timeouts when calling to backends directly when **allow-from-openshift-ingress** network policy is applied in the Red Hat OpenShift documentation.

Solution: Upgrade to a supported version of Red Hat OpenShift. For more information, see Supported Red Hat OpenShift versions.

Installation on Linux fails with network connectivity errors when using `aiopsctl` to install on vSphere VMs

The **aiopsctl** script installs the Kubernetes distribution **k3s**. When **k3s** is installed on a VMWare vSphere virtual machine (VM), routing across ClusterIPs does not work. The creation of the subscriptions for **Cert Manager** and the **Licensing Service** fail with errors similar to the following because they cannot reach the catalog:

```
- message: 'failed to populate resolver cache from source ibm-aiops-catalog/olm:
      failed to list bundles: rpc error: code = Unavailable desc = connection error:
      desc = "transport: Error while dialing: dial tcp: lookup ibm-aiops-catalog.olm.svc
      on 10.43.0.10:53: read udp 10.42.1.2:41727->10.43.0.10:53: i/o timeout"'
```

This problem is caused by a known kernel issue with Red Hat® Enterprise Linux® on vSphere. To fix the problem, run the following command on each of your nodes:

```
ethtool -K flannel.1 tx-checksum-ip-generic off
```

For more information, see k3s support

Installation on Linux: deployment is unstable after cluster restart

After a cluster restart, one of IBM Cloud Pak for AIOps's components, **zookeeper**, cannot start due to issues around DNS resolution causing it to misidentify its assigned cluster IP address.

Running the following command shows that **iaf-system-kafka-0** and some other pods failed:

```
kubectl get pods
```

Running the following command shows errors in the **zookeeper** logs:

```
kubectl logs -l app.kubernetes.io/name=zookeeper --tail -1
```

Example errors:

```
2024-06-17 20:57:26,674 ERROR Couldn't bind to iaf-system-zookeeper-0.iaf-system-zookeeper-
nodes.aiops.svc/10.42.1.53:2888 (org.apache.zookeeper.server.quorum.Leader) [QuorumPeer[myid=1](plain=127.0.0.1:12181)
(secure=[0:0:0:0:0:0:0:0]:2181)]
java.net.BindException: Cannot assign requested address
java.net.SocketException: Unresolved address
```

Solution: Run the following command to restart the `zookeeper` pods, and then wait a few minutes for the system to recover:

```
kubectl delete pods -l app.kubernetes.io/name=zookeeper
```

Installation on Linux: unable to log in to IBM Cloud Pak for AIOps

If you have a deployment of IBM Cloud Pak for AIOps on Linux and your Domain Name System (DNS) does not provide resolution for the `cp-console` hostname on all of your nodes, then you may not be able to log in to the Cloud Pak for AIOps console.

Run the following command from the control plane node to view the `platform-identity-provider` pod logs:

```
oc logs deployment/platform-identity-provider -n aiops
```

Example output where `cp-console` is not resolved:

```
{"name":"platform-identity-provider","hostname":"platform-identity-provider-79855f988-zqp2z","pid":1,"level":50,"err":
{"message":"getaddrinfo ENOTFOUND cp-console-aiops.example.com","name":"Error","stack":"Error: getaddrinfo ENOTFOUND cp-
console-aiops.example.com\n    at GetAddrInfoReqWrap.onlookup [as oncomplete]
(node:dns:107:26)","code":"ENOTFOUND"},"msg":"getaddrinfo ENOTFOUND cp-console-aiops.example.com","time":"2025-03-
24T20:10:31.696Z","v":0}
```

Solution: Use one of the following options to resolve this issue:

Option 1: Configure DNS records to resolve the hosts for accessing the Cloud Pak for AIOps console, and ensure that the DNS is used by all of the nodes in the cluster. For more information, see DNS requirements.

Option 2: Configure CoreDNS to provide DNS resolution within the cluster. Use the following steps:

1. Run the following command to create a ConfigMap with a custom configuration for the CoreDNS server:

   ```
   oc apply -f - <<EOF
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: coredns-custom
     namespace: kube-system
   data:
     default.server: |
       cp-console-aiops.<example.com> {
           hosts {
                 <load_balancer_ip> cp-console-aiops.<example.com>
                 fallthrough
           }
       }
   EOF
   ```

   Where:

   - `<example.com>` is your hostname
   - `<load_balancer_ip>` is the IP address of your load balancer
2. Run the following command to apply the configuration:

   ```
   oc -n kube-system rollout restart deployment coredns
   ```

Troubleshooting upgrade

After upgrade, policies do not process alerts due to problem with the `aiops-ir-lifecycle-create-policies-job`

This issue can occur if there is a problem connecting to Cassandra during upgrade. The `aiops-ir-lifecycle-create-policies-job` reports successful completion even it was unable to connect to Cassandra and run some of its policy-related queries. After upgrade policies do not process alerts, and error messages similar to the following are present in the `aiops-ir-lifecycle-create-policies-job's` log:

```
"name":"installdefaultpolicies","hostname":"aiops-ir-lifecycle-create-policies-job-c8k7t","pid":42,"level":40,"err":
{"message":"All host(s) tried for query failed.

"name":"installdefaultpolicies","hostname":"aiops-ir-lifecycle-create-policies-job-
c8k7t","pid":42,"level":50,"msg":"Failed to update policy with content
```

Solution: Restart the `aiops-ir-lifecycle-create-policies-job` by running the following command:

```
oc delete job aiops-ir-lifecycle-create-policies-job
```

After upgrade, a metric integration goes into a failed state

After upgrade a metric integration pod goes into a failed state waiting for a missing secret volume, because the secret `aiopsedge-postgres-edb-secret` does not yet exist. The upgrade process should create this secret after a few hours, or you can use the following solution to create it immediately and restore

functionality.

Solution: Run the following command to update the **aiopsedge** custom resource with the postgres access secret if it is not set.

```
# Set environment variables
NAMESPACE=<project>
AIOPS_INSTALL_NAME=$(oc get install -n $NAMESPACE -o jsonpath='{.items[*].metadata.name}')
AIOPSEDGE_NAME=$(oc get aiopsedge -n $NAMESPACE -o jsonpath='{.items[*].metadata.name}')
EDB_POSTGRESS_SECRET=$AIOPS_INSTALL_NAME-edb-secret

# Check if spec.postgres.accessSecret is set
CURRENT_SECRET=$(oc get aiopsedge $AIOPSEDGE_NAME -n $NAMESPACE -o jsonpath='{.spec.postgres.accessSecret}')

# Patch the aiopsedge custom resource
if [[ "${CURRENT_SECRET}" != "${EDB_POSTGRESS_SECRET}" ]]; then
    oc patch aiopsedge $AIOPSEDGE_NAME -n $NAMESPACE --type='json' -p="[{'op': 'replace', 'path':
'/spec/postgres/accessSecret', 'value': '$EDB_POSTGRESS_SECRET'}]"
else
    echo "Postgres access secret name already set to $CURRENT_SECRET in $AIOPSEDGE_NAME resource."
fi

# Restart the Edge operator manager pod to force a reconcile:
oc scale deployment/aiopsedge-operator-controller-manager -n $NAMESPACE --replicas=0
oc scale deployment/aiopsedge-operator-controller-manager -n $NAMESPACE --replicas=2
```

Where **<project>** is the project that your IBM Cloud Pak for AIOps installation is deployed in, usually **cp4aiops**.

After upgrade, alerts are not visible in the UI

1. Run the following command to view the LifecycleTrigger job.

   ```
   oc get LifecycleTrigger aiops -o yaml
   ```

2. Check the **status** section of the returned YAML.

   Example **status** section:

   ```
   status:
     conditions:
     - lastTransitionTime: "2024-04-15T10:12:44Z"
       message: Job '5e569b68014d9a669574096001ce51e6' in state 'RESTARTING'
       observedGeneration: 2
       reason: NotReady
       status: "False"
       type: LifecycleTriggerReady
     jobs:
     - jid: 5e569b68014d9a669574096001ce51e6
       startTime: "2024-04-13T20:22:17Z"
       state: RESTARTING
     savepoints:
     - path: s3://aiops-ir-lifecycle/savepoints/savepoint-d9c1a5-95d27a6574b4
       time: "2024-04-13T20:10:56Z"
     - lifecycleVersion: 4.5.1
       namespace: aiops
       path: s3://aiops-ir-lifecycle/savepoints/savepoint-d9c1a5-40e97c1c5bdd
       time: "2024-04-13T20:15:22Z"
     - lifecycleVersion: 4.5.1
       namespace: aiops
       path: s3://aiops-ir-lifecycle/savepoints/savepoint-d9c1a5-40e97c1c5bdd-upgraded
       time: "2024-04-13T20:21:53Z"
     - lifecycleVersion: 4.5.1
       namespace: aiops
       path: s3://aiops-ir-lifecycle/savepoints/savepoint-5e569b-6d60d2f35ca8
       time: "2024-04-13T20:23:20Z"
   ```

   Check to see if:

   - the **conditions** section has **Job <...> in state 'RESTARTING'**.
   - the **savepoints** section has an entry with a path that ends with **-upgraded**, which follows an entry with the same path without this suffix.
   - the two entries that are described above are followed by at least one more entry. This check is important.

Solution: If all of the above checks are true, then you can remediate this problem with the following steps.

1. Note the value of **jid** in the status block (**5e569b68014d9a669574096001ce51e6** in the example above) and export it as an environment variable.

   ```
   export JOB_ID=<jid>
   ```

2. Run the following command to cancel the LifecycleTrigger job that is restarting.

   ```
   oc patch LifecycleTrigger aiops --type json --patch '[{"op":"add","path":"/spec/cancelJobs","value":
   ["'${JOB_ID}'"]}]'
   ```

3. Check the LifecycleTrigger YAML again.

   ```
   oc get LifecycleTrigger aiops -o yaml
   ```

   After about a minute the **status.jobs** section should have two entries:

- The first entry has the old job ID and the state will change to CANCELLED.
- the second entry has a new job ID and the state should eventually settle at RUNNING.

Navigation menu items disappear after an upgrade

You might notice an issue where navigation menu items in IBM Cloud Pak for AIOps console disappear when you upgrade .

To determine if you are encountering this issue, run the following command:

```
kubectl logs -l component=zen-core-api
```

The output can resemble the following example:

```
time="2024-07-18 13:46:00" level=error msg=GetNavExtensions func=zen-core-
api/source/apis/internal_extensions/nav.GetNavItemsWrapper file="/go/src/zen-core-
api/source/apis/internal_extensions/nav/get_nav_items.go:124" exception="circular dependency or dangling node found in
aiops-nav-menuitem-alerts_page,aiops-connectors-nav,aiops-nav-menuitem-metric_search,aiops-insights-nav,aiops-nav-
menuitem-incidents_page,aiops-aimodel-nav,aiops-nav-menuitem-resource_management,aiops-nav-menuitem-automations"
```

Solution: If you see the preceding error, then use the following step to resolve the menu items disappear issue:

1. Delete the **aiops-baseui-frontdoor-ext** resource:

   ```
   oc delete zenextension aiops-baseui-frontdoor-ext
   ```

   The output can resemble the following example:

   ```
   zenextension.zen.cpd.ibm.com "aiops-baseui-frontdoor-ext" deleted
   ```

Database migration is stuck

The migration of the database from Elasticsearch to OpenSearch does not complete because OpenSearch is not ready.

1. Change to your IBM Cloud Pak for AIOps project, and export an environment variable for it.

   ```
   export PROJECT_CP4AIOPS=<project>
   oc project ${PROJECT_CP4AIOPS}
   ```

   Where **<project>** is the namespace (project) that IBM Cloud Pak for AIOps is deployed in.

2. Run the following command to check the status of the database migration, and verify whether the problem is with OpenSearch.

   ```
   oc describe installation -n ${PROJECT_CP4AIOPS}
   ```

   Example output for stalled migration due to problem with OpenSearch:

   ```
   Conditions:
       Last Transition Time:            2024-10-29T16:58:55Z
       Message:                         Elasticsearch migration has not started. Opensearch is not ready after 10
   minutes.
       Observed Generation:             2
       Reason:                          ForegroundMigrationFailed
       Status:                          False
       Type:                            MigratingElastic
   ```

   If your output is different from the example output, then exit this troubleshooting topic and contact IBM Support.

3. Run the following command to check the status of the OpenSearch pods:

   ```
   oc get pods -n ${PROJECT_CP4AIOPS} -l app.kubernetes.io/managed-by=ibm-elasticsearch | grep Pending
   ```

   Example output where two OpenSearch replicas are stuck:

   ```
   aiops-ibm-elasticsearch-es-server-all-0                           0/2     Pending     0          3h29m
   aiops-ibm-elasticsearch-es-server-all-2                           0/2     Pending     0          3h29m
   ```

   Note down the number of pods that are returned, and their names. If no pods are returned, then exit this troubleshooting topic and contact IBM Support.

4. Run the following command to confirm whether the OpenSearch pods are stuck due to resource constraints.

   ```
   oc describe pod -n ${PROJECT_CP4AIOPS} <Opensearch pod name>
   ```

   Where **<Opensearch pod name>** is the name of an OpenSearch replica returned by the previous step.

   Example output:

   ```
   # oc describe pod -n cp4aiops aiops-ibm-elasticsearch-es-server-all-0

   Events:
   Type     Reason            Age                From               Message
   ----     ------            ----               ----               -------
   Warning  FailedScheduling  33m (x9 over 66m)  default-scheduler  0/9 nodes are available: 3 Insufficient memory, 3
   node(s) had untolerated taint {node-role.kubernetes.io/master: }, 6 Insufficient memory. preemption: 0/9 nodes are
   available: 3 Preemption is not helpful for scheduling, 6 No preemption victims found for incoming pod
   ```

   If output is shown that is similar to the Warning in the example output, then use the following steps to resolve the problem.

Solution: To deploy, OpenSearch needs a large chunk of contiguous memory on a single node. Even if your cluster meets the resource requirements, there might not be a single node in the cluster that is large enough to schedule the OpenSearch pod. Use the following steps to evict pods from the nodes that have the most

available resources.

This script requires cluster-admin permissions, and might cause a service outage while pods are rescheduled. If you have concerns with these requirements, contact IBM Support so that they can assist.

1. Set an environment variable for the number of OpenSearch pods that need to be scheduled.

   ```
   export NUM_SELECT=<Number of pending OpenSearch pods>
   ```

   Where **<<Number of pending Opensearch pods>** is the number of pods that you previously noted down as **Pending**.

2. Run the following commands:

   ```
   while IFS= read -r TARGET_NODE; do
       for pod in $(kubectl get pods -o go-template="{{range .items}}{{\$x := .metadata.name}}{{if eq .spec.nodeName
   \"${TARGET_NODE}\"}}{{range .metadata.ownerReferences}}{{if eq .kind \"ReplicaSet\"}}{{\$x}} {{end}}{{end}}{{end}}
   {{end}}"); do
           REPLICAS=$(kubectl get replicaset $(kubectl get pod "${pod}" -o go-template='{{range
   .metadata.ownerReferences}}{{.name}}{{end}}') -o go-template='{{.spec.replicas}}')
           if [ "${REPLICAS}" -gt 1 ]; then
            kubectl delete pod "${pod}" --dry-run=client
           fi
       done
   done < <(
       for node in $(kubectl get node -l 'node-role.kubernetes.io/worker,node-role.kubernetes.io/infra!=' -o go-
   template='{{range .items}}{{$noSchedule := false}}{{range .spec.taints}}{{if or (eq .effect "NoSchedule") (eq
   .effect "PreferNoSchedule")}}{{$noSchedule = true}}{{end}}{{end}}{{if not $noSchedule}}{{printf "%s " .metadata.name
   }}{{end}}{{end}}'); do
           echo "${node} $(kubectl describe node ${node} | grep 'Allocated resources' -A 5 | awk '/^ *memory/ {print
   $3}')"
       done | sort -r -k2 | awk '{print $1}' | tail -"${NUM_SELECT}"
   )
   ```

The database migration is now able to proceed.

Elasticsearch CA renewal rollout becomes stuck and causes the *Policies* page to not display policies

During upgrade, some Elasticsearch replicas enter a crashing state when the Certificate authority (CA) renewal rollout begins. This is caused by trust issues between Elasticsearch pods that have different versions of the CA.

1. Run the following command to check your Elasticsearch cluster for this state:

   ```
   export AIOPS_PROJECT=<project>
   oc get pod -l ibm-es-server=aiops-ibm-elasticsearch-es-server -n "${AIOPS_PROJECT}"
   ```

   Where **<project>** is the project (namespace) where IBM Cloud Pak for AIOps is deployed, usually **cp4aiops**.

   Example output where the replica **aiops-ibm-elasticsearch-es-server-all-2** is crashing:

   ```
   aiops-ibm-elasticsearch-es-server-all-0   2/2   Running            0               12h   10.42.12.9    worker-
   1.acme.com  <none>  <none>
   aiops-ibm-elasticsearch-es-server-all-1   2/2   Running            0               12h   10.42.5.35    worker-
   2.acme.com  <none>  <none>
   aiops-ibm-elasticsearch-es-server-all-2   1/2   CrashLoopBackOff   59 (2m4s ago)   10h   10.42.11.36   worker-
   3.acme.com  <none>  <none>
   ```

   If one or more replicas have a state of **CrashLoopBackOff** then proceed to the next steps to confirm the cause.

2. Run the following command to check whether the crashing pod's logs indicate that the crashing pod failed to join the Elasticsearch cluster:

   ```
   oc logs <crashing_pod> -n "${AIOPS_PROJECT}" -c elasticsearch | grep "cluster-manager not discovered"
   ```

   Where **<crashing_pod>** is the name of the pod that crashed. For example, **aiops-ibm-elasticsearch-es-server-all-2**

   Example output where the crashing pod failed to join the Elasticsearch cluster:

   ```
   [2025-01-02T23:18:14,045][WARN ][o.o.c.c.ClusterFormationFailureHelper] [aiops-ibm-elasticsearch-es-server-all-2]
   cluster-manager not discovered or elected yet, an election requires at least 2 nodes with ids from
   [ZtUu55QzRBq7u7LIMgvTEA, 8H_HGLquRZSy3Qqw1TtgeA, ot0uVCmTRr-l8sa6M4oaDg], have discovered [{aiops-ibm-elasticsearch-
   es-server-all-2}{ot0uVCmTRr-l8sa6M4oaDg}{j3V12hDBSAK9LhjNEbOGgg}{127.0.0.1}{127.0.0.1:9702}{dim}
   {shard_indexing_pressure_enabled=true}] which is not a quorum; discovery will continue using [127.0.0.1:9700,
   127.0.0.1:9701] from hosts providers and [{aiops-ibm-elasticsearch-es-server-all-2}{ot0uVCmTRr-l8sa6M4oaDg}
   {j3V12hDBSAK9LhjNEbOGgg}{127.0.0.1}{127.0.0.1:9702}{dim}{shard_indexing_pressure_enabled=true}] from last-known
   cluster state; node term 16, last-accepted version 33 in term 16
   ```

   If your crashing pod's logs have output similar to the preceding example output, then proceed to the next step.

3. Run the following command to check whether the Elasticsearch HAProxy logs indicate SSL handshake failures:

   ```
   oc logs <crashing_pod> -n "${AIOPS_PROJECT}" -c haproxy | grep "SSL handshake failure"
   ```

   Where **<crashing_pod>** is the name of the pod that crashed. For example, **aiops-ibm-elasticsearch-es-server-all-2**

   Example output:

   ```
   127.0.0.1:53756 [20/Dec/2024:18:42:28.873] ft_elastic_transport_local_all_2 bk_elastic_transport_all_2/<NOSRV>
   -1/-1/0 0 SC 3/1/0/0/0 0/0
   10.42.4.57:51244 [20/Dec/2024:18:42:28.994] ft_elastic_http/1: SSL handshake failure
   ```

```
10.42.5.35:39470 [20/Dec/2024:18:42:29.124] ft_elastic_transport/1: SSL handshake failure
10.42.4.57:51250 [20/Dec/2024:18:42:29.201] ft_elastic_http/1: SSL handshake failure
```

If your logs have output similar to the preceding example output, then use the following solution to rectify this issue.

Solution: Run the following command to restart all three Elasticsearch pods at the same time, to help ensure that they all mount the new CA certificate.

```
oc delete pod -l ibm-es-server=aiops-ibm-elasticsearch-es-server -n "${AIOPS_PROJECT}"
```

Example output:

```
pod "aiops-ibm-elasticsearch-es-server-all-0" deleted
pod "aiops-ibm-elasticsearch-es-server-all-1" deleted
pod "aiops-ibm-elasticsearch-es-server-all-2" deleted
```

Your elasticsearch replicas will restart and mount the new CA certificate.

Issues that are common to installation and upgrade

Offline install or upgrade throws 'invalid character' error

When doing an offline install or upgrade, running the **`oc ibm-pak generate mirror-manifests <..>`** command throws an error similar to the following:

```
Error: failed to load the catalog FBC at cp.stg.icr.io/cp/ <...> invalid character '<' in string escape code
```

Solution: You must have **`IBM Catalog Management Plug-in for IBM Cloud Pak`** (**`ibm-pak-plugin`**) v1.10 or higher installed. Run the following commands to ensure that **`ibm-pak-plugin`** is at the required level.

1. Check which version of **`ibm-pak-plugin`** you have installed.

   Run the following command on your bastion host, portable compute device, or connected compute device if you are using a portable storage device.

   ```
   oc ibm-pak --version
   ```

   Example output:

   ```
   oc ibm-pak --version
   v1.11.0
   ```

2. If the **`ibm-pak-plugin`** version is lower than v1.10.0, then you must download and install the most recent version.

   Follow the steps for your installation approach:

   - Bastion host: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.
   - Portable device: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.

Offline install or upgrade stuck because the 'oc ibm-pak generate mirror-manifests' command fails with 'no space left on device'

The **`oc ibm-pak generate mirror-manifests $CASE_NAME $TARGET_REGISTRY --version $CASE_VERSION`** command fails with a message similar to the following in **`$IBMPAK_HOME/.ibm-pak/logs/oc-ibm_pak.log`**:

```
write /tmp/render-unpack-4002583241/var/lib/rpm/Packages: no space left on device
```

Solution: The default temporary directory does not have enough space to run the **`ibm-pak`** tool. You must set the **`TMPDIR`** environment variable to a different directory with more space before running the **`oc ibm-pak generate mirror-manifests`** command.

```
TMPDIR=<new_temp_dir> oc ibm-pak generate mirror-manifests ${CASE_NAME} ${TARGET_REGISTRY} --version ${CASE_VERSION}
```

Where **`<new_temp_dir>`** is the path for a directory with more space.

Troubleshooting clusters and nodes

Learn how to isolate and resolve problems that involve cluster and node failures.

- Temporal grouping policies not created after training historic events
- Illegal reflective access operation in Jobmanager pod log
- Kafka pods fail to restart
- Cloud Pak for AIOps console not accessible after shutdown of backup cluster
- Log anomaly detection Kafka intermittent Issue, as Log anomaly detector abruptly stops making predictions
- EDB Postgres cluster failure
- Postgres EDB database crashes
- CouchDB pod reports a permission error when draining a node
- Kafka broker pods are unhealthy because Kafka PVCs are full
- aimanager is not in Ready state after cluster restart, integrations are missing from the console
- Deployment on Linux: unable to access Cloud Pak for AIOps console

- Deployment on Linux: API server down after cluster restart

Temporal grouping policies not created after training historic events

Groups of related alerts that occur together (temporal groups) are not correlated automatically by the system. This issue can occur when temporal grouping policies are not created automatically due to communication issues between the aiops-ir-core-archiving pod and some Kafka topics.

Solution: Complete the following steps:

1. Run the following command, while logged in as an admin user in the project (namespace) where IBM Cloud Pak® for AIOps is running:

   ```
   kubectl logs $(kubectl get pods|grep aiops-ir-core-archiving|grep -v setup|awk '{print $1}') | grep "lib-rdkafka
   status"| tail -1
   ```

   Example command output:

   ```
   {
   "name": "client.kafka",
   "hostname": "aiops-ir-core-archiving-5dfd8fdcc5-l52g8",
   "pid": 20,
   "level": 30,
   "brokerStates": {
      "0": "UP",
      "-1": "UP"
   },
   "partitionStates": {
      "cp4waiops-cartridge.irdatalayer.replay.alerts.0": "UP",
      "cp4waiops-cartridge.irdatalayer.replay.alerts.1": "UP",
      "cp4waiops-cartridge.irdatalayer.replay.alerts.2": "UP",
      "cp4waiops-cartridge.irdatalayer.replay.alerts.3": "UP",
      "cp4waiops-cartridge.irdatalayer.replay.alerts.4": "UP",
      "cp4waiops-cartridge.irdatalayer.replay.alerts.5": "UP"
   },
   "msg": "lib-rdkafka status",
   "time": "2021-10-19T11:00:49.852Z",
   "v": 0
   }
   ```

2. If any of the broker or partition states are not up, restart the aiops-ir-core-archiving pod with the following command:

   ```
   kubectl delete pod $(kubectl get pods|grep aiops-ir-core-archiving|grep -v setup|awk '{print $1}')
   ```

Illegal reflective access operation in Jobmanager pod log

Reports of illegal reflective access operation.

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.flink.shaded.akka.org.jboss.netty.util.internal.ByteBufferUtil
(file:/opt/flink/lib/flink-dist_2.11-1.13.2.jar) to method java.nio.DirectByteBuffer.cleaner()
WARNING: Please consider reporting this to the maintainers of
org.apache.flink.shaded.akka.org.jboss.netty.util.internal.ByteBufferUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
```

This warning is reported when the Flink library is run by Java 11. For more information, see https://issues.apache.org/jira/browse/FLINK-17524

Solution: No action required. This warning message can be ignored and does not impact functionality.

Kafka pods fail to restart

Kafka pods fail to start when a worker node goes down, as in the following example:

```
evtmanager-kafka-0                                    1/2     Running            55          15h
evtmanager-kafka-1                                    2/2     Running            0           16h
evtmanager-kafka-2                                    1/2     Running            0           15h
evtmanager-kafka-3                                    1/2     Running            0           15h
evtmanager-kafka-4                                    1/2     Running            0           15h
evtmanager-kafka-5                                    2/2     Running            0           15h
evtmanager-zookeeper-0                                1/1     Running            0           16h
evtmanager-zookeeper-1                                1/1     Running            0           15h
evtmanager-zookeeper-2                                1/1     Running            0           15h
```

If the worker node where the Kafka pods are located goes down, the Kafka pods might fail to restart on another node.

Solution: Restart all zookeeper and Kafka pods:

```
oc delete pod {evtmanager-zookeeper-0} {evtmanager-zookeeper-1} {evtmanager-zookeeper-n} {evtmanager-kafka-0}
{evtmanager-kafka-1} {evtmanager-kafka-n}
```

Cloud Pak for AIOps console not accessible after shutdown of backup cluster

Unable to access the IBM Cloud Pak for AIOps console on the backup cluster after a systematic shut down and start up of the backup cluster. The following error message is displayed:

```
"Document was encrypted with unknown key '<namespace>-<release name>-<timestamp>'"
```

Solution: Find and then delete the required databases through the API.

1. First, run the following command to retrieve the hostname:

   ```
   oc get route couchdb-georedundancy
   ```

   For this scenario, use the hostname `couchdb-georedundancy-geonoi.apps.geo01primary.myibm.com`.

2. Run the two `SECRET_NAME` commands to retrieve the `username` and `password` that are required to retrieve the database names.

   For Username:

   ```
   SECRET_NAME=$(oc get secret | grep couchdb-secret | awk '{ print $1 }'); kubectl get secret ${SECRET_NAME} -o json |
   grep "username" | cut -d : -f2 | cut -d '"' -f2 | base64 -d
   username
   ```

   For Password:

   ```
   SECRET_NAME=$(oc get secret | grep couchdb-secret | awk '{ print $1 }'); kubectl get secret ${SECRET_NAME} -o json |
   grep "password" | cut -d : -f2 | cut -d '"' -f2 | base64 -d
   password
   ```

3. Use the `hostname`, `username`, and `password` obtained in the previous steps to run the command to retrieve the database names:

   ```
   curl -sk -u "username:password" https://couchdb-georedundancy-geonoi.apps.geo01primary.myibm.com/_all_dbs | jq "."
   ```

   Example output:

   ```
   [
   "_global_changes",
   "_replicator",
   "_users",
   "collabopsuser",
   "emailrecipients",
   "genericproperties",
   "icp-63666439356237652d336263372d343030362d613461382d6137336137396337313235-rba-as",
   "icp-63666439356237652d336263372d343030362d613461382d6137336137396337313235-rba-rbs",
   "integration",
   "noi-drdb",
   "noi-osregdb",
   "normalizercfd95b7e-3bc7-4006-a4a8-a73a79c71255",
   "osb-map",
   "otc_omaas_broker",
   "rba-pdoc",
   "schedule",
   "tenant",
   "trainingjob"
   ]
   ```

   The relevant database name here is **collabopsuser**.

4. Run the following command to delete the database:

   ```
   curl -sk -u "root:netcool" -X DELETE "https://couchdb-georedundancy-
   geonoi.apps.geo01primary.myibm.com/${DATABASE_NAME}"
   ```

   Where `DATABASE_NAME` = `collabopsuser`

5. Restart the `cem-users` pod to recreate the database:

   ```
   oc get pods |grep cem-users
   <cem_users_pod_name>

   oc delete pod <cem_users_pod_name>
   ```

6. You can confirm the database is created by running the curl command from step 3:

   ```
   curl -sk -u "username:password" https://couchdb-georedundancy-geonoi.apps.geo01primary.myibm.com/_all_dbs | jq "."
   ```

LAD Kafka intermittent Issue, as Log anomaly detector abruptly stops making predictions

The log anomaly detector abruptly stops making predictions.

On checking the anomaly pod logs, you see error messages related to Kafka, and due to these intermittent issues, log anomaly stops consuming data from the windowed logs topic, even when there are continuous windows generated from the data preprocessing component.

An example of such an error in the logs is as follows:

```
[2022-03-19 14:28:34,233] [logprophet.logs.watson_aiops] [ERROR] An error has occurred while processing message in batch
mode. Error: 'NoneType' object is not iterable
[2022-03-19 14:28:34,234] [app.anomaly_detector.service_controller] [ERROR] Consumer failed with unhandled exception :
'NoneType' object is not iterable
```

Solution: There is a two-step process to resume the flow:

1. Clear the windowed logs topic content. This step is required to clear any old data that is not yet consumed by the log anomaly detector. If you do not clear this data, when LAD is ready to make predictions, it will try to generate events for old data showing a lag. So, you need to disable the data flow, and clear data from windowed logs.

- Disable data flow from data integration
- Clear the windowed logs topic using following command

  ```
  oc edit kt cp4waiops-cartridge-windowed-logs-1000-1000
  ```

- Set `retention.ms` to `1`.
- Wait for a few minutes, and once the data is cleared and no more windows are generated, reset the 'retention.ms' to the original value
- Enable the data flow in the data integration

2. Once the data is cleared using the above steps, restart the anomaly pod logs with the following command:

   ```
   oc get po | grep anomaly
   oc delete pod <anomaly-pod>
   ```

EDB Postgres cluster failure

Run the following command on your cluster:

```
oc get clusters.postgresql.k8s.enterprisedb.io <installation_name>-edb-postgres -n <namespace>
```

Where:

- `<installation_name>` is the name of your IBM Cloud Pak for AIOps instance, for example `ibm-cp-aiops`.
- `<namespace>` is the namespace that IBM Cloud Pak for AIOps is deployed in.

If the cluster *STATUS* does not show `Cluster in healthy state` and the *READY* count is 0, then the EDB Postgres cluster has failed.

EDB Postgres is a component that is used by IBM Cloud Pak for AIOps. A defect in EDB Postgres 1.18.6 and lower can cause the EDB Postgres cluster to fail. If the pod that hosts the primary instance of EDB Postgres is deleted and there is not a healthy EDB Postgres replica instance available to promote, then the EDB Postgres operator may not reschedule the pod hosting the primary instance.

Solution: If you encounter this problem, then you must contact IBM Support who can help you to determine and bring up the primary EDB Postgres node.

Postgres EDB database crashes

After a failure of one of the Postgres EDB replicas, a new replica is unable to come up, and the failed replica remains.

Solution: Delete the failed replica and its persistent volume claim (PVC) if the replica failure is due to a synchronization problem.

1. Run the following command to find all Postgres instances that are not running:

   ```
   export PROJECT_CP4AIOPS=<project>
   oc get pod -n $PROJECT_CP4AIOPS -l "k8s.enterprisedb.io/podRole=instance" | grep -vE "Running"
   ```

   Where `<project>` is the project (namespace) where IBM Cloud Pak for AIOps is deployed, usually `cp4aiops`.

   Example output where `zen-metastore-edb-1` has failed:

   ```
   oc get pod -n cpaiops -l "k8s.enterprisedb.io/podRole=instance" | grep -vE "Running"

   NAMESPACE NAME                  READY   STATUS             RESTARTS        AGE
   aiops     zen-metastore-edb-1   0/1     CrashLoopBackOff   904 (36s ago)   7d1h
   ```

2. Examine the logs for the failed replica to see whether the non-primary replicas are not synchronized with the WAL logs in the primary replica.

   ```
   oc logs -n "${PROJECT_CP4AIOPS}" <failed_replica>
   ```

   Where `<failed_replica>` is the failed replica returned in step 1, for example `zen-metastore-edb-1`.

   Look for entries similar to the following examples:

   ```
   pg_rewind: servers diverged at WAL location 5/6606A4E0 on timeline 1
   pg_rewind: error: could not open file \"/var/lib/postgresql/data/pgdata/pg_wal/000000010000000500000068\": No such
   file or directory
   pg_rewind: fatal: could not read WAL record at 5/68000028
   ```

   ```
   {"level":"info","ts":1677595436.8679705,"logger":"postgres","msg":"record","logging_pod":"ibm-cp-aiops-edb-postgres-
   1","record":{"log_time":"2023-02-28 14:43:56.867
   UTC","user_name":"streaming_replica","process_id":"90988","connection_from":"10.254.24.22:32852","session_id":"63fe1
   32c.1636c","session_line_num":"1","command_tag":"idle","session_start_time":"2023-02-28 14:43:56
   UTC","virtual_transaction_id":"13/0","transaction_id":"0","error_severity":"ERROR","sql_state_code":"58P01","message
   ":"requested WAL segment 000000030000000200000A0 has already been removed","query":"START_REPLICATION 2/A0000000
   TIMELINE 2","application_name":"ibm-cp-aiops-edb-postgres-2","backend_type":"walsender"}}
   ```

3. Check that the failed replica is not the primary replica.

   Run the following command, and check that the failed replica identified in step 1 is not listed in the **PRIMARY** column for any of the Postgres clusters.

   ```
   oc get clusters.postgresql.k8s.enterprisedb.io -n $PROJECT_CP4AIOPS
   ```

Example output:

```
oc get clusters.postgresql.k8s.enterprisedb.io -n cp4aiops
NAME                        AGE   INSTANCES   READY   STATUS                     PRIMARY
common-service-db           23h   1           1       Cluster in healthy state   common-service-db-1
ibm-cp-aiops-edb-postgres   23h   1           1       Cluster in healthy state   ibm-cp-aiops-edb-postgres-1
zen-metastore-edb           23h   2           2       Cluster in healthy state   zen-metastore-edb-2
```

4. Delete the failed replica and its PVC.

   Run the following command if the failed replica is not the primary and the logs for the failed replica showed errors similar to the example output in step 2. Otherwise, contact IBM Support.

   ```
   oc delete pod,pvc zen-metastore-edb-1 -n "${PROJECT_CP4AIOPS}"
   ```

   Where **<failed_replica>** is the failed replica returned in step 1, for example **zen-metastore-edb-1**.

CouchDB pod reports a permission error when draining a node

The **CouchDB** pod's log reports a permission error similar to the following example:

```
[error] 2022-12-02T01:24:20.726024Z couchdb@c-example-couchdbcluster-m-0.c-example-couchdbcluster-m <0.13913.7> --------
Could not open file /data/db/shards/00000000-7fffffff/_users.1669801403.couch: permission denied
```

Note: After a restore into a new namespace, similar symptoms can be seen. For more information about the solution for this scenario, see Restore fails with CouchDB pod in CrashLoopBackOff.

The way that Kubernetes mounts persistent volumes can cause this problem.

Solution: Delete the **CouchDB** pods that are not ready. When the replacement pods are created, the persistent volumes are mounted correctly.

1. Find the names of your **couchdb** pods.

   ```
   oc get pods -l app.kubernetes.io/name=IssueResolutionCoreCouchDB -n <namespace>
   ```

   Where **<namespace>** is the namespace that IBM Cloud Pak for AIOps is deployed in.

2. Delete all the **couchdb** pods that the previous step showed as not ready.

   ```
   oc delete pod <couchdb-pod> -n <namespace>
   ```

   Where:

   - **<namespace>** is the namespace that IBM Cloud Pak for AIOps is deployed in.
   - **<couchdb-pod>** is a couchdb pod that is not ready.

Kafka broker pods are unhealthy because Kafka PVCs are full

When the Kafka broker pods are unhealthy due to insufficient storage, the Events operator is unable to expand the PVCs.

Solution: Check whether the Kafka broker pods have storage available, and expand the storage available to Kafka if they do not.

1. Check the state of the Kafka broker pods:

   ```
   export AIOPS_NAMESPACE=<project>
   oc get pod -n "${AIOPS_NAMESPACE}" -l app.kubernetes.io/name=kafka
   ```

   Where **<project>** is the namespace (project) that IBM Cloud Pak for AIOps is deployed.

   Example output:

   ```
   iaf-system-kafka-0                                                1/1     Running                 0
   5d21h
   iaf-system-kafka-1                                                0/1     CreateContainerError    0
   95m
   iaf-system-kafka-2                                                0/1     CreateContainerError    1 (2d18h ago)
   5d21h
   ```

2. If the broker pods are in a *CreateContainerError* state, check the logs for errors:

   ```
   oc logs -n "${AIOPS_NAMESPACE}" iaf-system-kafka-1
   ```

   Example output:

   ```
   Warning  Failed          5m40s (x13 over 89m)  kubelet  Error: relabel failed /var/lib/kubelet/pods/37e6519a-b372-
   429d-b5f5-9aaf5b12fd37/volumes/kubernetes.io~csi/pvc-ff53e966-fee1-41e2-9448-b0eeac2e49c6/mount: lsetxattr
   /var/lib/kubelet/pods/37e6519a-b372-429d-b5f5-9aaf5b12fd37/volumes/kubernetes.io~csi/pvc-ff53e966-fee1-41e2-9448-
   b0eeac2e49c6/mount/kafka-log1/.lock: no space left on device
   ```

   Do not proceed with the rest of these troubleshooting steps if the error is not related to disk space.

3. Adjust the Kafka storage size in the CustomResource:

```
oc patch kafka/iaf-system -n "${AIOPS_NAMESPACE}" --type merge -p '{"spec":{"kafka":{"storage":{"size":"
<size>Gi"}}}}'
```

Where **<size>** is the increased storage size that you require for the Kafka PVCs.

4. Increase the Kafka PVCs:

```
for p in $(oc get pvc -l app.kubernetes.io/name=kafka -n "${AIOPS_NAMESPACE}" -o
jsonpath='{.items[*].metadata.name}'); do oc patch pvc $p -n "${AIOPS_NAMESPACE}" --type merge -p '{"spec":
{"resources":{"requests":{"storage":"<size>Gi"}}}}' ; done
```

Where **<size>** is the same value that was used in the previous command.

Note : The time required for this command to run depends on the number of PVCs and the amount of data that is stored in each one.

aimanager is not in Ready state after cluster restart, integrations are missing from the console

After a cluster restart, **aimanager** does not have a ready state when queried with **oc get aimanager -o yaml**, and some previously configured integrations such as log integrations, ChatOps, runbooks, kafka, or PagerDuty are missing from the console. This problem can be caused by a timing issue where the **aimanager** pod attempts to start before the **Postgres** database is ready.

Solution

1. Run the following command to check if the **aimanager-aio-controller** pod is running.

```
oc get pods | grep aimanager-aio-controller
```

Example output for a pod that is running correctly:

```
aimanager-aio-controller-67d6857cbb-n9skp      1/1     Running     0               7d3h
```

2. If the **aimanager-aio-controller** is not running, then check the pod's logs for errors.

```
oc logs <aimanager-aio-controller-pod-name>
```

Where **<aimanager-aio-controller-pod-name>** is the name of the **aimanager-aio-controller-pod-name** from step 1.

Example error output:

```
2024-12-03 04:51:14 ERROR ServletListener:801 - Failed to get edb connection from SQL query
java.sql.SQLException: Cannot create PoolableConnectionFactory (Connection to ibm-cp-aiops-edb-postgres-rw:5432
refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections.)
    at org.apache.commons.dbcp2.BasicDataSource.createPoolableConnectionFactory(BasicDataSource.java:653) ~[commons-
dbcp2-2.9.0.jar:2.9.0]
```

If you have an error similar to the preceding example error output, then note down the name of the **aimanager-aio-controller** pod and use the following steps to resolve the problem.

3. Check that the **Postgres** database is up and running, and do not continue until it is.

```
oc get pods | grep edb-postgres
```

Example output if PostgreSQL is running:

```
aiops-installation-edb-postgres-1      1/1     Running     0               6d2h
```

4. Run the following command to restart the **aimanager-aio-controller** pod.

```
oc delete pod <aimanager-aio-controller-pod-name>
```

Where **<aimanager-aio-controller-pod-name>** is the name of the **aimanager-aio-controller-pod-name** from step 1.

Deployment on Linux: unable to access Cloud Pak for AIOps console

The pods in the IBM Cloud Pak for AIOps cluster are in a *Running* state, but the user interface is not accessible.

Solution: When you installed IBM Cloud Pak for AIOps, you used an existing load balancer or configured a new one of your choice. For more information, see Load balancing. The load balancer is the entry point for accessing deployments of IBM Cloud Pak for AIOps on Linux. Problems with your load balancer can prevent access to the Cloud Pak for AIOps console. Check the load balancer status and logs for problems, and restart the load balancer if it is not running.

Deployment on Linux: API server is down after a cluster restart

After restarting the Linux cluster, all requests to the Kubernetes API server such as **oc get pod** and **aiopsctl status** fail, as in the following example output:

```
$ oc get pod -n aiops
E0903 08:55:39.586644    1287 memcache.go:265] couldn't get current server API group list: the server is currently unable
to handle the request
E0903 08:55:39.589138    1287 memcache.go:265] couldn't get current server API group list: the server is currently unable
to handle the request
E0903 08:55:39.592133    1287 memcache.go:265] couldn't get current server API group list: the server is currently unable
```

```
to handle the request
E0903 08:55:39.594610    1287 memcache.go:265] couldn't get current server API group list: the server is currently unable
to handle the request
E0903 08:55:39.597172    1287 memcache.go:265] couldn't get current server API group list: the server is currently unable
to handle the request
Error from server (ServiceUnavailable): the server is currently unable to handle the request

$ aiopsctl status
o- [03 Sep 24 08:58 PDT] Getting cluster status
[ERROR] Failed to get cluster status apiserver not ready
```

Solution: Run the following steps to confirm that the problem is due to the Kubernetes API server being down, and to rectify this.

1. Check the control plane nodes.
    1. On each control plane node, run the following command to assess the platform state. Affected deployments have a state of *activating*.

        ```
        systemctl status k3s.service
        ```

        Example output for a deployment affected by this issue:

        ```
        $ systemctl status k3s.service

        k3s.service - Lightweight Kubernetes    Loaded: loaded (/etc/systemd/system/k3s.service; enabled; vendor
        preset: enabled)    Active: activating (start) since Mon 2024-09-09 05:10:30 PDT; 6s ago
        ```

    2. On each control plane node, run the following command to check the platform logs:

        ```
        journalctl -xeu k3s.service
        ```

        Example output for a deployment affected by this issue:

        ```
        Sep 09 05:11:28 control-plane-2.acme.com k3s[50701]: {"level":"warn","ts":"2024-09-09T05:11:28.735466-
        0700","caller":"rafthttp/http.go:413","msg":"failed to find remote peer in cluster","local-member-
        id":"8c10b42dfa5ffc54","rem>
        Sep 09 05:11:28 control-plane-2.acme.com k3s[50701]: {"level":"warn","ts":"2024-09-09T05:11:28.828015-
        0700","caller":"rafthttp/http.go:413","msg":"failed to find remote peer in cluster","local-member-
        id":"8c10b42dfa5ffc54","rem>
        Sep 09 05:11:28 control-plane-2.acme.com k3s[50701]: {"level":"warn","ts":"2024-09-09T05:11:28.834347-
        0700","caller":"rafthttp/http.go:145","msg":"failed to process Raft message","local-member-
        id":"8c10b42dfa5ffc54","error":"ra>
        ```

    If your deployment is affected, continue with the following steps.

2. Stop **k3s** on the primary control plane node:

    ```
    systemctl stop k3s
    ```

    Note: The primary control plane is the first control plane attached to the cluster. This is the value of CONTROL_PLANE_NODE in your aiops_var.sh environment variables file. The non-primary control plane nodes are the nodes in the ADDITIONAL_CONTROL_PLANE_NODES array in your aiops_var.sh environment variables file.

3. Run a cluster reset on the primary control plane node to soft reset **etcd**.

    ```
    k3s server --cluster-reset
    ```

4. Stop k3s on the non-primary control plane nodes.

    ```
    systemctl stop k3s
    ```

5. Back up the **etcd** data directory on each non-primary control plane node.

    ```
    cp -r <data_directory> <backup_directory>
    ```

    Where

    - **<data_directory>** is: **/var/lib/rancher/k3s/server/db** if you are not using a custom platform directory, or **PLATFORM_STORAGE_PATH** in aiops_var.sh if you are using a custom platform directory
    - **<backup_directory>** is a directory where you want to store the backup.
6. Delete the **etcd** data directory on each non-primary control plane node.

    Important: Do not delete the platform data directory on the primary control plane node.

    ```
    rm -rf <data_directory>
    ```

    Where **<data_directory>** is **/var/lib/rancher/k3s/server/db** if you are not using a custom platform directory, or **PLATFORM_STORAGE_PATH** in aiops_var.sh if you are using a custom platform directory.

7. Run the following command on the primary control plane node, and then on each of the non-primary control plane nodes:

    ```
    systemctl start k3s
    ```

Troubleshooting datastores

Review frequently encountered issues with the datastores that are used by IBM Cloud Pak® for AIOps.

Elasticsearch

- Elasticsearch CA renewal rollout becomes stuck and causes the *Policies* page to not display policies

Elasticsearch CA renewal rollout becomes stuck and causes the *Policies* page to not display policies

Elasticsearch replicas enter a crashing state when the Certificate authority (CA) renewal rollout begins. This is caused by trust issues between Elasticsearch pods that have different versions of the CA.

1. Run the following command to check your Elasticsearch cluster for this state:

   ```
   export AIOPS_PROJECT=<project>
   oc get pod -l ibm-es-server=aiops-ibm-elasticsearch-es-server -n "${AIOPS_PROJECT}"
   ```

   Where **<project>** is the project (namespace) where IBM Cloud Pak for AIOps is deployed, usually **cp4aiops**.

   Example output where the replica **aiops-ibm-elasticsearch-es-server-all-2** is crashing:

   ```
   aiops-ibm-elasticsearch-es-server-all-0   2/2   Running            0               12h   10.42.12.9    worker-
   1.acme.com  <none>  <none>
   aiops-ibm-elasticsearch-es-server-all-1   2/2   Running            0               12h   10.42.5.35    worker-
   2.acme.com  <none>  <none>
   aiops-ibm-elasticsearch-es-server-all-2   1/2   CrashLoopBackOff   59 (2m4s ago)   10h   10.42.11.36   worker-
   3.acme.com  <none>  <none>
   ```

   If one or more replicas have a state of **CrashLoopBackOff** then proceed to the next steps to confirm the cause.

2. Run the following command to check whether the crashing pod's logs indicate that the crashing pod failed to join the Elasticsearch cluster:

   ```
   oc logs <crashing_pod> -n "${AIOPS_PROJECT}" -c elasticsearch | grep "cluster-manager not discovered"
   ```

   Where **<crashing_pod>** is the name of the pod that crashed. For example, **aiops-ibm-elasticsearch-es-server-all-2**

   Example output where the crashing pod failed to join the Elasticsearch cluster:

   ```
   [2025-01-02T23:18:14,045][WARN ][o.o.c.c.ClusterFormationFailureHelper] [aiops-ibm-elasticsearch-es-server-all-2]
   cluster-manager not discovered or elected yet, an election requires at least 2 nodes with ids from
   [ZtUu55QzRBq7u7LIMgvTEA, 8H_HGLquRZSy3Qqw1TtgeA, ot0uVCmTRr-l8sa6M4oaDg], have discovered [{aiops-ibm-elasticsearch-
   es-server-all-2}{ot0uVCmTRr-l8sa6M4oaDg}{j3V12hDBSAK9LhjNEbOGgg}{127.0.0.1}{127.0.0.1:9702}{dim}
   {shard_indexing_pressure_enabled=true}] which is not a quorum; discovery will continue using [127.0.0.1:9700,
   127.0.0.1:9701] from hosts providers and [{aiops-ibm-elasticsearch-es-server-all-2}{ot0uVCmTRr-l8sa6M4oaDg}
   {j3V12hDBSAK9LhjNEbOGgg}{127.0.0.1}{127.0.0.1:9702}{dim}{shard_indexing_pressure_enabled=true}] from last-known
   cluster state; node term 16, last-accepted version 33 in term 16
   ```

   If your crashing pod's logs have output similar to the preceding example output, then proceed to the next step.

3. Run the following command to check whether the Elasticsearch HAProxy logs indicate SSL handshake failures:

   ```
   oc logs <crashing_pod> -n "${AIOPS_PROJECT}" -c haproxy | grep "SSL handshake failure"
   ```

   Where **<crashing_pod>** is the name of the pod that crashed. For example, **aiops-ibm-elasticsearch-es-server-all-2**

   Example output:

   ```
   127.0.0.1:53756 [20/Dec/2024:18:42:28.873] ft_elastic_transport_local_all_2 bk_elastic_transport_all_2/<NOSRV>
   -1/-1/0 0 SC 3/1/0/0/0 0/0
   10.42.4.57:51244 [20/Dec/2024:18:42:28.994] ft_elastic_http/1: SSL handshake failure
   10.42.5.35:39470 [20/Dec/2024:18:42:29.124] ft_elastic_transport/1: SSL handshake failure
   10.42.4.57:51250 [20/Dec/2024:18:42:29.201] ft_elastic_http/1: SSL handshake failure
   ```

   If your logs have output similar to the preceding example output, then use the following solution to rectify this issue.

Solution: Run the following command to restart all three Elasticsearch pods at the same time, to help ensure that they all mount the new CA certificate.

```
oc delete pod -l ibm-es-server=aiops-ibm-elasticsearch-es-server -n "${AIOPS_PROJECT}"
```

Example output:

```
pod "aiops-ibm-elasticsearch-es-server-all-0" deleted
pod "aiops-ibm-elasticsearch-es-server-all-1" deleted
pod "aiops-ibm-elasticsearch-es-server-all-2" deleted
```

Your elasticsearch replicas will restart and mount the new CA certificate.

Troubleshooting integrations

Review frequently encountered issues with creating and managing outgoing and incoming integrations.

- Multiple integrations
- Integrations page
- Metrics integrations
- Instana
- Kafka
- Netcool
- ServiceNow
- Splunk
- IBM Tivoli Netcool/Impact

Multiple integrations

- Integrations do not initialize

Integrations do not initialize

If an integration gets stuck in a state that is not running, there may be an issue with the secrets being created.

To check this log in to openshift and check the pods. For example, the netcool integration has the following issue:

```
NAME                                                              READY   STATUS              RESTARTS   AGE
netcool-conn-cfe182e0-d7eb-42f6-9677-fb86b180bd0f-8bd546cbvv5km   0/1     ContainerCreating   0          27m
```

If you describe the pod **`oc describe pod netcool-conn-cfe182e0-d7eb-42f6-9677-fb86b180bd0f-8bd546cbvv5km`** and you see a mount error in the events log such as the following:

```
34s (x20 over 25m)         Warning   FailedMount                       Pod/netcool-conn-cfe182e0-d7eb-42f6-9677-
fb86b180bd0f-8bd546cbvv5km             MountVolume.SetUp failed for volume "grpc-bridge-service-binding" : secret
"connector-cfe182e0-d7eb-42f6-9677-fb86b180bd0f" not found
```

you can conclude that the secrets failed to create. The way to fix this is to use the following command:

```
oc get pods | grep connector-manager
```

Then delete the connector-manager pods and restart it. On restarting it, the secrets are re-created for all integrations.

Integrations page

On viewing the Integrations page, you might encounter the following issues:

- Deleting an integration does not remove the PVC for the integration
- Error displays when you activate Kafka, Humio, Splunk, Mezmo, ELK, PagerDuty, ServiceNow integrations
- Integrations on the Integrations page show a red 'Unknown' status
- Integrations UI pod in CrashLoopBackOff for the Integrations page
- Integrations page becomes unresponsive after some time or shows an error
- Incorrect integration status on the Integrations page
- The integration keeps showing *Error* or *Retrying* and integration pod is stuck in *Pending*
- Ansible Automation Controller integration not running due to prohibited egress

Deleting an integration does not remove the PVC for the integration

For StatefulSet type integrations, when the integration is deleted the Persistent Volume Claim (PVC) resource for the integration is not automatically deleted. You must manually delete the resource with the Red Hat OpenShift **`oc`** CLI or with the Red Hat OpenShift UI.

This behavior occurs because the StatefulSet PVC is created by specifying the **`volumeClaimTemplate`**, but the created PVC is not automatically deleted by Red Hat OpenShift when the StatefulSet resource is deleted.

To delete a PVC resource with the Red Hat OpenShift **`oc`** CLI, complete the following steps:

1. Get the list of PVC resources for your current integrations:

   ```
   oc get pvc | grep 'conn'
   ```

2. Select the PVC resource that you want to delete and record the name.

3. Delete the PVC resource:

   ```
   oc delete pvc <pvc-name>
   ```

   Where **`<pvc-name>`** is the name of the PVC resource that you want to delete.

Error displays when you activate Kafka, Humio, Splunk, Mezmo, ELK, PagerDuty, ServiceNow integrations

When you activate a Kafka, Falcon LogScale, Splunk, Mezmo, ELK, PagerDuty, ServiceNow integration in the console, an error can occur when the required Flink task slots to run the integration are not available. When this error occurs, the following message is displayed:

```
Failure: Not enough task slots to activate the connection.
```

This error occurs as there is a finite number of Flink task slots to use for creating integrations to remote resources. The error message displays the number of slots that are needed to support the current active integration set, along with the total number of slots available.

Solution: To activate the integration, you must either reduce the number of slots that are needed by this (or another) integration, or allocate more slots. To decrease the number of slots needed by a integration, consider reducing the "base parallelism" or "logs per second" value for the integration. This reduced value decreases the rate at which data flows through the integration, making more slots available to other integrations. To allocate more slots, see Increasing data streaming capacity.

Integrations on the Integrations page show a red 'Unknown' status

All integrations are showing red 'Unknown' integration status and data is not flowing after creating a new integration, such as a Mezmo integration. For the new integration if you enable 'Data collection', you can see that the data is not flowing, and the integration shows the same red 'Unknown' status in the Data collection status field instead of the expected 'Running' status. The common cause for this is unstable Kafka in the background.

Solution: If this issue occurs, restart the bridge and operator pod.

Integrations UI pod in CrashLoopBackOff for the Integrations page

You can encounter an intermittent issue that causes the Integrations page to fail and become unavailable for viewing.

Solution: If the page does not automatically recover, you can increase the timeout setting for the **`aimanager-aio-controller`** deployment to help prevent the issue. To increase this setting, edit the **`aimanager-aio-controller`** deployment (**`liveliness`** and **`healthcheck`** probe) and set it to a longer timeout of say 15 seconds, up from the default 5 seconds. This change can stop the **`Connections-ui`** pod from going into CrashLoopBackOff.

1. Check the **`aimanager-aio-controller`**

   ```
   oc get pods | grep aio-controller
   aimanager-aio-controller-6d9f68d5f5-8dkk9                         1/1     Running     0               27h
   ```

2. Edit the **`aimanager-aio-controller`**'s deployment

   ```
   oc edit deployment aimanager-aio-controller
   ```

3. Edit these 2 variables (**`timeoutSeconds`**) for both the **`livenessProbe`** and the **`readinessProbe`** to 15 seconds:

   ```
   livenessProbe
   - timeoutSeconds
   readinessProbe
   - timeoutSeconds
   ```

Integrations page becomes unresponsive after some time or shows an error

If you are using IBM Cloud Pak for AIOps, you might encounter an unresponsive Integrations page around 24 hours after you install your cluster. The integrations page might take a long time to load and then display no integrations despite configuring them earlier. Another symptom of this problem is that you might fail to create an integration and see the following error:

```
Failed submitting creating request
```

Solution: To ensure the Integrations page works normally, delete the **`aimanager-aio-controller`** pod:

```
for n in $(oc get pods |grep aimanager-aio-controller |awk '{print $1}')
do
oc delete pod $n
done
```

Within one to two minutes of running the script, the pod will automatically restart and the Integrations page will work as usual.

Incorrect integration status on the Integrations page

You might encounter an incorrect integration status for all gRPC integrations (Instana, Netcool, AppDynamics, Dynatrace, AWS CloudWatch, Splunk, Zabbix, and New Relic) on the Integrations page. The integration status might appear as *Unknown* when it must be *Running*. The incorrect status might be the result of the pod connector-bridge being in CrashLoopBackOff state and restarting multiple times due to an unstable Kafka integration.

Solution: If this issue occurs, check the Kafka and zookeeper pods for any restarts. Kafka integration becomes unstable and restarts when the PVC used for Kafka is full. Review the current capacity of the PVC and increase the capacity as needed to resolve the issue. For more information about adjusting the size of the PVC, see Increasing Kafka PVC.

The integration keeps showing *Error* or *Retrying* and integration pod is stuck in *Pending*

On the Integrations UI page, an integration keeps showing *Error* or *Retrying* and the integration pod is stuck in *Pending*. This issue occurs because the Persistent Volume Claim (PVC) is requesting a volume from a storage provider that does not support the **`ReadWriteOnce`** (RWO) access mode.

Solution: You can use either one of the following methods to unblock the stuck integration pod:

- Change the default storage class

Or

- Re-create the Persistent Volume Claim with a different storage class

Method 1: Change the default storage class. Review the default storage class in the cluster and change to a storage class that supports **`ReadWriteOnce`** (RWO) access mode, such as file storage types.

Note: This method should resolve integration PVC issues but since it changes the global default storage class type in the cluster, it might also affect other workloads that rely on the default storage class settings when requesting a volume.

1. Run the following command to determine which is the current default storage class.

   ```
   oc get storageclass
   ```

2. Note the name of the default storage class **(default)**.

3. Change the default storage class to one that supports the **ReadWriteOnce** (RWO) access mode like file storages. The following command sets the **storageclass.kubernetes.io/is-default-class** annotation to **true** on the new storage class and sets the annotation to **false** on the old storage class. (Change the **<new-default-storageclass-name>** and **<old-default-storageclass-name>** with the actual storage class names in your cluster.)

```
oc patch storageclass <old-default-storageclass-name> -p '{"metadata": {"annotations":
{"storageclass.kubernetes.io/is-default-class":"false"}}}'
oc patch storageclass <new-default-storageclass-name> -p '{"metadata": {"annotations":
{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

4. Verify that the default storage class is updated by running

```
oc get storageclass
```

5. Log in to the Cloud Pak for AIOps console and navigate to the **Integrations** page.

6. Delete the existing integration, then reinstall it.

7. Verify that the integration enters a *Running* state.

Method 2: Re-create the Persistent Volume Claim with a different storage class. Delete the existing PVC and re-create a new one with the same name that uses a storage class that supports the **ReadWriteOnce** (RWO) access mode.

Note: This method updates only a single integration instance. Repeat this step for all integrations unable to request a persistent volume, due to the incorrect storage class type.

1. Get the PVC name of the integration. You can list all PVCs by using **oc get pvc**. Look for the one that is still in *Pending* state with your integration name.

2. Delete the existing PVC and re-create a new one with the same name. Review the following sample command as reference. Update the **<connector-pvc-name>** with the PVC name from the previous step and set **<storage-class-name>** with the storage class that supports the required **ReadWriteMan** (RWX) access mode.

```
# Set the PVC name
CONNECTOR_PVC_NAME=<connector-pvc-name>

# Set new storage class
STORAGE_CLASS=<new-storage-class>

CONNECTOR_APP=$(oc get pvc $CONNECTOR_PVC_NAME -o jsonpath='{.metadata.labels.app}')
CONNECTOR_ID=$(oc get pvc $CONNECTOR_PVC_NAME -o jsonpath='{.metadata.labels.instance}')
PVC_SIZE=$(oc get pvc $CONNECTOR_PVC_NAME -o jsonpath='{.spec.resources.requests.storage}')

oc delete pvc $CONNECTOR_PVC_NAME && \
cat << EOF | tee >(oc apply -f -) | cat
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
    name: $CONNECTOR_PVC_NAME
    labels:
        app: $CONNECTOR_APP
        app.kubernetes.io/instance: aiopsedge
        app.kubernetes.io/managed-by: aiopsedge-operator
        app.kubernetes.io/name: ibm-aiops-edge
        app.kubernetes.io/part-of: ibm-aiops-edge
        instance: $CONNECTOR_ID

spec:
    accessModes:
    - ReadWriteOnce
    resources:
        requests:
            storage: $PVC_SIZE
    storageClassName: $STORAGE_CLASS
    volumeMode: Filesystem
EOF
```

For example,

```
# Set the PVC name
CONNECTOR_PVC_NAME=netcool-vct-netcool-conn-325fba02-440f-4135-bd00-8560bcf5eda1-0

# Set new storage class
STORAGE_CLASS=rook-cephfs

CONNECTOR_APP=$(oc get pvc CONNECTOR_PVC_NAME -o jsonpath='{.metadata.labels.app}')
CONNECTOR_ID=$(oc get pvc CONNECTOR_PVC_NAME -o jsonpath='{.metadata.labels.instance}')

oc delete pvc $CONNECTOR_PVC_NAME && \
cat << EOF | tee >(oc apply -f -) | cat
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
    name: $CONNECTOR_PVC_NAME
    labels:
        app: $CONNECTOR_APP
        app.kubernetes.io/instance: aiopsedge
        app.kubernetes.io/managed-by: aiopsedge-operator
        app.kubernetes.io/name: ibm-aiops-edge
```

```
        app.kubernetes.io/part-of: ibm-aiops-edge
        instance: $CONNECTOR_ID

spec:
    accessModes:
    - ReadWriteOnce
    resources:
        requests:
            storage: 50Mi
    storageClassName: $STORAGE_CLASS
    volumeMode: Filesystem
EOF
```

3. Verify that the integration pod transitions to **Running** state.
4. If necessary, repeat the steps above on any other integration that is producing the same problem.

Metrics integrations

If you are creating, or have, a metrics integration, you might encounter the following issue:

Metrics connection is unable to pull further historical data when historical mode is completed

When a metrics integration (for example Dynatrace, Appdynamics, Zabbix, New Relic, Splunk, or AWS) has completed its historical data pulling, historical mode can never be restarted again even if the connection configuration is edited or updated from the UI to specify a different start date, end date or a different set of metrics.

Solution: If you need to pull historical data again for another set of metrics or different dates, you can create a new connection.

Instana integrations

If you are creating, or have, an Instana integration, you might encounter the following issues:

- integration is unable to update the metric list in the UI
- IBM Cloud Pak for AIOps cannot close some alerts when an Instana integration exists
- After a repair of Instana, alerts in the Instana application do not associate with the corresponding IBM Cloud Pak for AIOps application
- Due to an out-of-memory error the Instana integration pod keeps restarting when retrieving data from Instana

integration is unable to update the metric list in the UI

The Instana integration UI page displays a list of metrics from Instana that are updated when the Instana integration pod runs. Some Instana hosts can have many technology plug-ins installed, which collect many built-in and custom metrics. These metrics can cause the Instana integration to exceed the allowed message limit when the integration sends an update to the UI. If this limit is exceeded, the Instana integration pod logs the following message:

```
0000004d Connector     I   UI configuration patch exceeds the safe message limit, skip sending configuration patch. The
metrics list in the UI could not be updated with the actual list of technologies and metrics. Please configure more
filter patterns to reduce the number of metrics to query.
```

The integration logs each technology plug-in and the number of metrics that are included or filtered.

Solution: Apply additional filter patterns in the Instana configmap for the integration to exclude when querying the metric list from Instana.

1. Get the Instana pod logs to determine which technology plug-in contains many metrics

   ```
   oc logs $(oc get pod | grep 'instana-conn' | awk '{print $1}') | grep -e "Plugin \'.*\' \(.*\)"
   ```

   The following example log indicates that a **aceMessageFlow** technology plug-in that contains a large list of metrics, and that is a candidate to filter.

   Tip: Look for any technology plug-ins that contain more than 100 metrics.

   ```
   0000004d InstanaInstan I Plugin 'ACE Message Flow' (aceMessageFlow) metrics: 2,666 included, 0 filtered.
   ```

2. Determine the available metrics names by technology from the Instana API or from your Instana administrator. To use the Instana API, you can use a command similar to the following sample command, which queries **aceMessageFlow** metrics:

   ```
   INSTANA_ENDPOINT=<your Instana host>
   INSTANA_TOKEN=<your Instana API token>
   curl --request GET -s  --url ${INSTANA_ENDPOINT}/api/infrastructure-monitoring/catalog/metrics/aceMessageFlow \
   --header "authorization: apiToken ${INSTANA_TOKEN}" \
   | jq 'group_by(.pluginId)[]|  {(.[0].pluginId): [.[] | .metricId] | sort}'
   ```

3. Create a pattern to filter metrics from the output of the previous command. For example, **flowNodes\\..*** to filter all metrics with **flowNodes.** prefix.
4. Get the Instana integration configmap name to add the filter pattern from the previous step.

   Note: If multiple Instana integrations exist, select the configmap with the UID that matches the pod name.

   ```
   oc get configmap | grep instana-connector
   ```

5. Extract the **exclude-filter-patterns** from the configmap. This command saves the content of the configmap into the local file:

   ```
   oc extract configmap/ibm-grpc-instana-connector-d59eea9f-501d-4bd1-b3a3-6f98b49b0930 --keys=exclude-filter-patterns
   --to=.
   ```

6. Edit the local **exclude-filter-patterns** file to add the metric pattern in addition to the default ones that are defined in the **filters** array.

   Note: To escape a slash **** character, you must use double ****. For example:

```
{"filters":["fs\\./dev/.*","fs_mount\\./.*","metrics\\.counters\\..*\\..*\\..*","(.*\\.)?
metrics\\.gauges\\..*\\..*\\..*","metrics\\.meters\\..*","(.*\\.)?
metrics\\.timers\\..*","prometheus\\.metrics.*","flowNodes\\..*"]}
```

7. Apply the change to the configmap:

```
oc set data configmap/ibm-grpc-instana-connector-d59eea9f-501d-4bd1-b3a3-6f98b49b0930 --from-file=exclude-filter-
patterns
```

8. Verify that the new filter pattern is loaded by the pod, and that the number of metrics that are filtered for the technology plug-in is changed.

```
oc logs $(oc get pod | grep 'instana-conn' | awk '{print $1}') | grep 'Loaded filter patterns from ConfigMap'
```

9. Verify that the Instana integration can send an update to the UI successfully:

```
oc logs $(oc get pod | grep 'instana-conn' | awk '{print $1}') | grep -e 'Status message .* write with configuration
patch .* successful.'
```

10. In the *Edit Instana integration* UI page, the list of technologies and metrics list is now updated.

IBM Cloud Pak for AIOps cannot close some alerts when an Instana integration exists

When an integration with Instana exists, IBM Cloud Pak for AIOps might not be able to close some alerts. Instead, you might need to check the status for the associated event within the Instana UI and clear the alert manually.

Solution: To find the alert URL for clearing the alert, complete the following actions:

1. From the IBM Cloud Pak for AIOps console, open the *Incidents and alerts* tool.
2. Select the alert in the table to display the Alert details side panel for that alert. The Information section is open by default in the side panel.
3. Copy the URL from links and Paste the URL into your browser.
4. If you are an Instana authorized user, you see the associated event in the Instana UI.

If the event is not active and has "end-time" for the event in the Instana UI, clear the alert:

1. Click *Alert* to see the alert details.
2. Click *Clear Alert* to clear the alert.

After a repair of Instana, alerts in the Instana application do not associate with the corresponding IBM Cloud Pak for AIOps application

When you restart Instana post repair, alerts in the Instana application do not map with the corresponding application in IBM Cloud Pak for AIOps. The reason is that Instana creates new identifiers for the resources on its application on restarting, without updating the topology in IBM Cloud Pak for AIOps. As a result, the IDs of the resources do not match leading to the disassociation of alerts.

Solution: To map the alerts, you must complete the following steps:

1. Clean up the application in Instana.
2. Re-create the application after a few minutes.

To clean up and recreate applications in Instana, see Getting started with Instana.

Due to an out-of-memory error the Instana integration pod keeps restarting when retrieving data from Instana

When the Instana integration receives a large amount of data from one or more Instana APIs while the integration is observing many resources, and there is insufficient memory to process this large amount of data, an out of memory error can occur.

Solution: If this error occurs, you can apply custom memory settings to allocate more memory to the Instana integration pod by increasing the memory limit setting. This allocation can only be done post-installation as the integration must exist before applying the custom setting. To apply custom resource settings, follow the steps to patch the Instana integration:

Prerequisite: Ensure you have the **yq** and **jq** packages installed on your system before running the commands below. To get installation instructions for these packages, refer to the documentation or website for **yq** and **jq**.

1. From a command line, change to your IBM Cloud Pak for AIOps project (namespace):

```
namespace=<aiops-namespace>
oc project $namespace
```

2. Get the display name for the Instana integration you would like to increase resources for from the command line:

```
oc get connectorconfiguration
```

3. Set the required variables:

```
connectionname=<instana-connection-name>
connconfig=$(oc get connectorconfiguration --no-headers | grep $connectionname | awk '{print $1}')
connconfiguid=$(oc get connectorconfiguration $connconfig -o jsonpath='{.metadata.uid}')
gitappname=$(oc get gitapp -l connectors.aiops.ibm.com/connection-id==$connconfiguid --no-headers | awk '{print
$1}')
```

Note: Update the **connectionname** variable with your Instana display name from the command above.

4. Run this command to create a variable with a patch specification for the Instana integration. This will apply only to one Instana integration at a time. This command should not have an output. Copy this command directly from the documentation and do not paste it into a text editor because this can affect the

format.

```
jsonpatch=$(yq -o json <<EOF
    patch: |-
      apiVersion: apps/v1
      kind: Deployment
      metadata:
        name: ibm-grpc-instana-connector
      spec:
        template:
          spec:
            containers:
            - name: ibm-grpc-instana-connector
              resources:
                requests:
                  memory: 1536Mi
                  ephemeral-storage: 2Gi
                limits:
                  memory: 8Gi
                  ephemeral-storage: 4Gi
    target:
      group: apps
      kind: Deployment
      name: ibm-grpc-instana-connector
      version: v1
EOF
)
```

5. Run the following command to apply the custom resource setting that you created in the previous step:

```
oc get gitapp $gitappname  -n $namespace -o json | jq ".spec.components[0].kustomization.patches += [$jsonpatch]" |
oc apply -f -
```

Note: If you see an error message when running this command that states there was a configuration conflict, try running the command a second time.

6. It takes a moment for the changes to show in the integration pod.

Note: If the memory limit is not updated for the Instana integration pod after a few minutes, run **`oc get gitapp $gitappname`** and check to see if there are any errors. If there is an error stating that the YAML is invalid, run the command below to revert the patch and go through all steps again. The cause of this error is that the format of the commands may have been altered by a text editor and should be copied directly from the documentation and run directly in the terminal.

Revert the custom resource settings to default resource settings:

To revert from custom to default resource settings, run the following command:

```
oc patch gitapp $gitappname -n $namespace --type='json' -p="[{'op': 'remove', 'path':
'/spec/components/0/kustomization/patches'}]"
```

Kafka integrations

If you are creating, or have, a Kafka integration, you might encounter the following issues:

- Kafka events not displayed in IBM Cloud Pak for AIOps
- After copying log data into a Kafka cluster, verify that the data is imported into Elasticsearch

Kafka events not displayed in IBM Cloud Pak for AIOps

You cannot see Kafka events in IBM Cloud Pak for AIOps after successfully creating a Kafka integration. This failure occurs when the Kafka key or message violates the format requirements. The **`MismatchedInputException`** error in the task manager pod log indicates this formatting error.

Solution: To ensure that the Kafka events display in IBM Cloud Pak for AIOps, check whether the following conditions are met:

1. The key in Kafka payload is null.

   Note: The key is determined by Kafka producer. Most producer tools use null object as key unless the producer is programmed to assign value.

2. The message in Kafka payload is a JSON message that satisfies the following requirements:

   - Exists in the format of the selected data source.
   - Contains sufficient data for the mapping.

After copying log data into a Kafka cluster, verify that the data is imported into Elasticsearch

To publish log data, you can copy it into a Kafka cluster. For more information, see Copying log data into the Kafka cluster. Then, you can check whether the log data is imported into Elasticsearch.

Solution: Verify that the log data is imported into Elasticsearch by getting shell access to the **`ai-platform-api-server`** pod. Run the following commands from the namespace where IBM Cloud Pak for AIOps is installed:

```
oc exec -it $(oc get pods | grep ai-platform-api-server | awk '{print $1}') -- /bin/bash
```

Example:

```
curl -X GET -u $ES_USERNAME:$ES_PASSWORD $ES_URL/_cat/indices -k | sort
```

Example output:

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  4710  100  4710    0     0  28373      0 --:--:-- --:--:-- --:--:-- 28373
green  open .security                          L0Gou7AXSC6PyuZ8-skGRQ 1 0    1 0   4.6kb   4.6kb
green  open aiops-searchservice_v10            Yml s6nL3S-KP4vC4ROdRRw 5 0    0 0     1kb     1kb
yellow open 1000-1000-20200803-logtrain        QImZoOd7QiuM3mfxUIdFAA 8 2   24 0  34.5kb  34.5kb
yellow open 1000-1000-20200804-logtrain        54m5F8K6Q3iIbSZ4KvMlYQ 8 2  286 0  80.8kb  80.8kb
yellow open 1000-1000-20200805-logtrain        I_GhHFz9Th62KV_7wGwyMA 8 2  137 0  61.1kb  61.1kb
yellow open 1000-1000-20200806-logtrain        7HZcyCaLRa2VoOwJrH7Ftg 8 2  292 0  76.2kb  76.2kb
yellow open 1000-1000-20200807-logtrain        LzUFhGcGSs-Rp_uJtw17sw 8 2  686 0 126.1kb 126.1kb
yellow open 1000-1000-20200808-logtrain        HR7n9NIATnm9MOjgnq2K_Q 8 2  268 0    77kb    77kb
yellow open 1000-1000-20200809-logtrain        NSU1X31OTq2coxdo-_I1rQ 8 2   48 0  48.7kb  48.7kb
yellow open 1000-1000-20200810-logtrain        2M54aDMXRyOp_06XnInTEg 8 2  885 0 140.3kb 140.3kb
yellow open 1000-1000-20200811-logtrain        pQjyF1pQTX6GT70WY7T7ZQ 8 2  797 0 132.2kb 132.2kb
yellow open 1000-1000-20200812-logtrain        U0YasxsqTpuricpyyfQ62Q 8 2 1294 0 201.6kb 201.6kb
yellow open 1000-1000-20200813-logtrain        pZA79rSSTYeTrsRuT8K44g 8 2 1155 0 178.3kb 178.3kb
yellow open 1000-1000-20200814-logtrain        eu5qqE31TcGtz5yjVKxlnQ 8 2  720 0   134kb   134kb
yellow open 1000-1000-20200815-logtrain        rKHrLUooSP-lx6r_ettalQ 8 2   42 0  46.2kb  46.2kb
yellow open 1000-1000-20200816-logtrain        0bRKhqgdQviTBc0FnJ4pvg 8 2   58 0  49.9kb  49.9kb
yellow open 1000-1000-20200817-logtrain        PMrlHzjZQymb5hXiZanhOw 8 2  287 0  99.3kb  99.3kb
yellow open 1000-1000-20200818-logtrain        ItLXqZIxQWy7pirLCPcfKw 8 2  483 0 103.7kb 103.7kb
yellow open 1000-1000-20200819-logtrain        W2ti9RlVTRiW3ZpynEZ8wQ 8 2  879 0 141.8kb 141.8kb
yellow open 1000-1000-20200820-logtrain        Zsz7Vd0rQpSuVwbUi4AfQw 8 2  488 0  98.7kb  98.7kb
yellow open 1000-1000-20200821-logtrain        VVcsZ1HEQJi08WrjX6CJ_Q 8 2  268 0  43.4kb  43.4kb
yellow open algorithmregistry                  wLjGMGL4SI-weK666yr8pA 1 1    8 0  26.2kb  26.2kb
yellow open cp4waiops-cartridge-poli.....      6dQclBWSQA6FR3yRurpxVQ 1 1   17 0  66.8kb  66.8kb
yellow open dataset                            TA3TB-MNTkScyxDNZxuEcw 1 1    0 0    208b    208b
yellow open postchecktrainingdetails           G8lHdh9WQ8OxIHhOAwBUnw 1 1    0 0    208b    208b
yellow open prechecktrainingdetails            WA00PRwuTgWLsQNvcBcoug 1 1    0 0    208b    208b
yellow open trainingdefinition                 dC2UuOLiR_qGiax-8tEfEg 1 1    0 0    208b    208b
yellow open trainingrun                        fRqCLpQ1TCi_kt4zzNvwDA 1 1    0 0    208b    208b
yellow open trainingsrunning                   gzyQzqP7SJuy9dbUZI1dVA 1 1    0 0    208b    208b
yellow open trainingstatus                     AiLkAg0yRQKzzIpbsk0CCw 1 1    0 0    208b    208b
```

Netcool integrations

If you are creating, or have, a Netcool integration, you might encounter the following issues:

- Runbook automation pod "rba-rbs" crashes during an alert burst, for example, while connecting to Netcool
- ObjectServer alerts missing in IBM Cloud Pak for AIOps due to `No object found with id` error in the data layer
- Setting values for mandatory fields for events sent to the embedded ObjectServer
- Events from IBM Tivoli Netcool/OMNIbus are not updated in Alert Viewer
- Test connection fails in the Netcool connector UI configuration page
- Fresh alerts are missing in Alert Viewer

Runbook automation pod "rba-rbs" crashes during an alert burst, for example, while connecting to Netcool

The "rba-rbs" pod crashes and gets restarted automatically, and most of the fully automated runbooks that are associated with the new alerts are not run. This restart behavior might be observed multiple times until all the events from the event burst have been ingested, and the system reaches its regular state of operation again.

Solution: During an event burst like the one that occurs when connecting to a Netcool server, all enabled policies are evaluated, and for the matching events, the associated fully automated runbooks are started. If many matching events exist, these events might result in a high load of concurrent runbook invocations that the runbook service rba-rbs is not able to handle. The system recovers automatically. However, some runbooks will not be run during the burst.

ObjectServer alerts missing in IBM Cloud Pak for AIOps due to a `No object found with id` error in the data layer

If a new alert is rejected by the data layer, subsequent update alerts may be rejected by the data layer with the error: **`No object found with id`** being found in the **`cp4waiops-cartridge.irdatalayer.errors`** Kafka topic.

Solution: The **`AIOpsAlertId`** column must be cleared so that when its updates reach the connector, the data will be perceived as a fresh alert by IBM Cloud Pak for AIOps.

To do this, use the following steps:

1. Disable the connector's data collection (this will trigger the gateway to shut down).
2. Go to the connector pod, remove the gateway cache file and SAF file under the following directory:

   `/bindings/netcool-connector/omnibus/var/G_CONNECTOR`

3. In the ObjectServer:
   1. Rectify the data columns that had resulted in wrongly mapped values.
   2. Clear the **`AIOpsAlertId`** column of the alert.
4. Enable the connector's data collection.

When processing the next update event to that alert, the connector will see an empty **`AIOpsAlertId`** and will generate **`"type": "update"`** in the northbound CloudEvent payload.

Setting values for mandatory fields for events sent to the embedded ObjectServer

If you are manually inserting events into the embedded ObjectServer using **`nco_sql`**, you must set values for the following fields so that the events can be processed correctly:

- **`Node`**
- **`AlertKey`**

- `AlertGroup`
- `Manager`

Note: This is only a concern if you are creating an event manually using an insert statement. All IBM Netcool Operations Insight probes will set all required columns.

Events from IBM Tivoli Netcool/OMNIbus are not updated in Alert Viewer

You might notice an issue where events from IBM Tivoli Netcool/OMNIbus reach Cloud Pak for AIOps, but the updates are not reflected in the Alert Viewer. The reason for this issue is that the Netcool connector might have restarted. The connector restart causes existing alerts in Cloud Pak for AIOps to dissociate from the origin alerts in the ObjectServer.

Solution: Upgrade Cloud Pak for AIOps to version 4.5.1 or newer. After the upgrade, the new alerts are not impacted by the Netcool connector restart.

Test connection fails in the Netcool connector UI configuration page

You might notice that the test connection fails in the Netcool connector UI configuration page.

Solution: Use the following steps to diagnose and resolve the connection failure.

1. Set the IBM Netcool Operations Insight ObjectServer log message level to `debug`. This can be done by updating the `MessageLevel` property in the `$OMNIHOME/etc/{ObjectServer_name}.props` file on the Netcool host. `ObjectServer_name` is the name of IBM Netcool Operations Insight ObjectServer.
2. Click **test connection** in the Netcool UI. Check if the IBM Netcool Operations Insight ObjectServer login attempt is received.
   - If no log in attempt is seen in the IBM Netcool Operations Insight ObjectServer log, check whether the IBM Netcool Operations Insight ObjectServer host IP that is resolved on Cloud Pak for AIOps matches the IBM Netcool Operations Insight ObjectServer host IP.
   - If a log in attempt is seen but rejected by the OS authentication, check the Transport Layer Security (TLS) certificate, especially the Common Name (CN) value, which must match the fully qualified domain name (FDQN) of the server.
3. If the Connection manager is down or not responsive to the request to connect to the IBM Netcool Operations Insight ObjectServer, then restart the Connection manager pod.
4. For an on-premises IBM Netcool Operations Insight ObjectServer integration, you might notice these issues:
   - If the CN value in the TLS certificate does not match the host server FQDN, create a TLS certificate that contains the host FQDN.
   - If a firewall restriction exists on the IBM Netcool Operations Insight ObjectServer port, then update the firewall rules to unblock the port.

Fresh alerts are missing in Alert Viewer

You might notice an issue where the most recent alerts are not showing in Alert Viewer. You need to inspect the data in different stages of the pipeline to diagnose the problem.

When an alert is missing, you can scan the north-bound Kafka topics for the alert message. The following example shows the direction of the north-bound data flow:

Netcool connector -> `cp4waiops-cartridge.lifecycle.input.alerts` [topic] -> `cp4waiops-cartridge.irdatalayer.requests.alerts` [topic] -> Alert Viewer

Where:

- `[topic]` refers to a Kafka topic.
- `cp4waiops-cartridge.lifecycle.input.alerts` is a Kafka topic where all the new alerts are collected.
- `cp4waiops-cartridge.irdatalayer.requests.alerts` is a Kafka topic. `cp4waiops-cartridge.irdatalayer.requests.alerts` is one of the outcomes by the lifecycle.

Solution: Use the following steps to diagnose and resolve the missing alerts issue.

Note: Download and extract the `Ext_IAFKafkaTools.tar.gz` file located in Download Ext_IAFKafkaTools.tar.gz. After you extract the `Ext_IAFKafkaTools.tar.gz` file, follow the steps in the `README.txt` file.

1. Check whether an alert message appears in the `cp4waiops-cartridge.irdatalayer.errors` topic. Look for the alert message and run the following command.

   ```
   consumer.sh | grep <pattern>
   ```

   Inspect the cause of the error in the message.

2. Check whether the alert message appears in the `cp4waiops-cartridge.lifecycle.input.alerts` topic.

   The following example shows an alert message with `type: update`.

   ```
   {"tenantid":"cfd95b7e-3bc7-4006-a4a8-a73a79c71255","requestid":"b6079c67-8b2c-4a84-ae1d-
   9e50df05bfa1","requestTime":"2024-04-16T03:23:37.443484418Z","type":"update","entityType":"alert","entity":
   {"insights":[{"details":{"lastProcessedEventOccurrenceTime":"2024-04-16T03:23:33Z"},"id":"event-occurrence","type":"
   [aiops.ibm.com/insight-type/deduplication-details](https://aiops.ibm.com/insight-type/deduplication-
   details)"}],"deduplicationKey":"{hostname=null, name=null}-UserLogoutSessionEvent-
   ","eventCount":84341,"state":"open","id":"9e96637d-15cf-4618-bdf7-ef97e6016128","type":
   {"classification":"UserLogoutSessionEvent","eventType":"problem"},"lastOccurrenceTime":"2024-04-16T03:23:33.000Z"}}
   ```

3. If a new alert message appears in `cp4waiops-cartridge.lifecycle.input.alerts` and fulfills the positive criteria of `type: create`, check whether the message is delivered to `cp4waiops-cartridge.irdatalayer.requests.alerts`.

If the message is not found in the **requests.alerts** topic, the message is either blocked or rejected by the lifecycle process or not yet consumed from the **input.alerts** topic.

4. If the message passes through **cp4waiops-cartridge.irdatalayer.requests.alerts**, and is not reflected in the Alert Viewer, check the data layer process.
5. Restart the pod of the processes that are not working based on the error in the message. The pods that might need a restart depending on the error can be **Grpc server**, **Life cycle processor**, and **Kafka broker**.
6. If Kafka topics storage limitation is an issue, update the Kafka storage size.

ServiceNow integrations

Incidents aren't created

Incidents might not be created for various reasons.

Solution: Complete the following checks to help ensure that incidents can be created.

1. Validate the ServiceNow URL and credentials. Go to the ServiceNow integration page, select the integration, and select **Test connection**. If the test fails, check that the URL and credentials are correct.
2. Make sure that the versions of Cloud Pak for AIOps and the IBM ServiceNow marketplace applications are compatible. For more information about which version of Cloud Pak for AIOps is in use, see Prerequisites in Creating a ServiceNow integration.
3. Make sure that a policy is configured to use the existing ServiceNow connection. For more information about creating policies, see Promote alerts to an incident.
4. Make sure that the ServiceNow integration user has the correct permissions. For more information, see Managing roles.
5. Incident tables can have business rules that prevent incident creation. Check business rules and change them so that they don't affect incident tables.

Splunk integrations

If you are creating, or have, a Splunk integration, you might encounter the following issues while creating or editing the integration:

- All fields in the Splunk form reset if edited while loading metrics
- Splunk integrations that are created before upgrade fail to update after upgrade

All fields in the Splunk form reset if edited while loading metrics

If you edit the fields in the Splunk form while the metrics are loading in the integration console, then you cannot save the changes made to the form.

Solution: To avoid this issue, do not modify the fields in the Splunk form while metrics are loading. If you do so, then redo the changes after the metrics are loaded to save the edits.

Splunk integrations that are created before upgrade fail to update after upgrade

On upgrading from 3.5.0, you might not be able to enable or disable log or metric data flows on Splunk integrations that were created before upgrade. This issue can occur intermittently.

Solution: When this issue occurs, you can refresh the console and try again after some time. Alternatively, you can create a new integration and later delete the Splunk integration that was defined before upgrade.

IBM Tivoli Netcool/Impact integrations

If you are creating, or have, a IBM Tivoli Netcool/Impact integration, you might encounter the following issues:

- IBM Tivoli Netcool/Impact integration status shows "Retrying"
- IBM Tivoli Netcool/Impact stops sending data to widgets in a JazzSM dashboard
- AIOps data source in IBM Tivoli Netcool/Impact is missing required headers

IBM Tivoli Netcool/Impact integration status shows "Retrying"

After Creating IBM Tivoli Netcool/Impact integrations, you might see a **Connection Orchestration** status of "Done", but the **IBM Tivoli Netcool/Impact integration** shows an error with a status of "Retrying".

The reason for this error might be a locked IBM Tivoli Netcool/Impact file that is preventing the creation of the IBM Cloud Pak for AIOps data model in IBM Tivoli Netcool/Impact. If this is the issue, the **impactserver-errors** logs in IBM Tivoli Netcool/Impact will display an error similar to the following:

```
30 Jun 2023 03:27:21,137 ERROR [DataModelUIResource] File etc/NCIP_datasourcelist is locked by user: root.
```

Solution: Locate and delete a file called **<servername>_versioncontrol.locks** in IBM Tivoli Netcool/Impact and restart the IBM Tivoli Netcool/Impact server and GUI.

1. Log in to the IBM Tivoli Netcool/Impact host and navigate to the **$IMPACT_HOME/etc** directory. Locate a file called **<servername>_versioncontrol.locks**. For example, **NCIP_versioncontrol.locks**.
2. Delete the file.
3. Restart the Impact server and GUI:

```
$IMPACT_HOME/bin/stopImpactServer.sh
$IMPACT_HOME/bin/stopGUIServer.sh
```

```
$IMPACT_HOME/bin/startImpactServer.sh
$IMPACT_HOME/bin/startGUIServer.sh
```

The status of the **IBM Tivoli Netcool/Impact integration** should switch to **Running** in Cloud Pak for AIOps.

IBM Tivoli Netcool/Impact stops sending data to widgets in a JazzSM dashboard

After Creating IBM Tivoli Netcool/Impact integrations, the IBM Tivoli Netcool/Impact server stops sending data to widgets in a JazzSM dashboard. The widget will report an **ATKRST132E** or **404** error code.

Solution: Re-create the parameter files for the AIOPS_HandleAction and AIOPS_ExecJS Impact policies.

For more information, refer to the troubleshooting article Netcool/Impact datasets for DASH stop working after deploying the AIOps to Impact integration.

AIOps data source in IBM Tivoli Netcool/Impact is missing required headers

When an integration is made to IBM Tivoli Netcool/Impact, a RESTful data source is created in Netcool/Impact during the initialization stage.

However, the data source may be missing one or more fields including required headers. Attempting to edit or delete such a data source will fail with an error message:

```
Error : Version Control System failed. Check system error logs for further details
```

Solution:

To remedy this, you can add the required connection details and header values using the Netcool/Impact GUI.

1. Log into the Netcool/Impact GUI. Open the **Data Model** tab and make a note of the data source name. For example, **AIOps-impact-e92b7669-8809-4cc2-ab3c-5cbdf0db1534**
2. Open an SSH terminal to the primary Netcool/Impact server.
3. Change to the **etc** directory of the application:

   ```
   cd /opt/IBM/tivoli/impact/etc
   ```

4. Locate the data source file with the .ds extension. For example, **NCI_AIOps-impact-e92b7669-8809-4cc2-ab3c-5cbdf0db1534.ds**

   Note: If the file does not exist, create an empty file with the name of the missing data source.

5. Use the **svn add** command to manually add the file to version control:

   ```
   /opt/IBM/tivoli/impact/platform/linux/svn/bin/svn add NCI_AIOps-impact-e92b7669-8809-4cc2-ab3c-5cbdf0db1534.ds
   ```

6. Use the **svn commit** command to check in the file:

   ```
   /opt/IBM/tivoli/impact/platform/linux/svn/bin/svn commit -m "manual commit" NCI_AIOps-impact-e92b7669-8809-4cc2-ab3c-5cbdf0db1534.ds
   ```

7. Log in to the Impact GUI **Data Model** tab. Open the AIOPs data source.
8. Enter the following values in the connection details fields:

Field Name	Value
Host Name	**<HOSTNAME>**
Resource Path	/aiops
Port	443
Use HTTTPS	Enabled

   Replace **<HOSTNAME>** with the Cloud Pak for AIOps hostname. For example, **cpd-aiops.apps.clustername.cp.ibm.com**.

9. In **Request Headers**, add the following headers:

Header	Value
x-tenant-id	**cfd95b7e-3bc7-4006-a4a8-a73a79c71255**
Content-Type	**application/json;charset=utf-8**
Authorization	**ZenApiKey <KEY>**

   In the **Authorization** field, replace **<KEY>** with the API key. Follow the procedure from Configuring a connection in IBM Tivoli Netcool/Impact for IBM Cloud Pak for AIOps to retrieve the key.

   Note: In Netcool/Impact 7.1.0.32, you can hide the values by declaring the request headers as a protected header instead.

10. In the **Proxy Settings** tab, set the **Proxy Port** to **8080** and select **No Authentication** as the **Authentication Method**.

    Note: The **Password** field is ignored.

11. Save the data source. Click **Test Connection** to confirm the connection works.

Ansible Automation Controller integration not running due to prohibited egress

After configuring an Ansible Automation Controller integration, the integration reports *Not running*, and no Ansible templates are shown in the *Automation Actions* table.

Use the following steps to check whether the failure is due to prohibited egress:

1. Run the following commands to view the RBA Automation Service pod logs, and check whether responses are being received from Ansible:

```
export AIOPS_NAMESPACE=<AIOps installation namespace>
oc logs -l app.kubernetes.io/component=rba-as -n ${AIOPS_NAMESPACE}
```

Example output if responses are not being received from Ansible:

```
Error: awxp/requestAwx.requestAwx: Request to Ansible failed... Request was not responded after 5 seconds
```

2. Run the following commands to attempt to connect to the Ansible instance from the RBA Automation Service pod:

```
export AIOPS_NAMESPACE=<AIOps installation namespace>
oc exec $(oc get pod -l app.kubernetes.io/component=rba-as -n ${AIOPS_NAMESPACE} -o
jsonpath='{.items[0].metadata.name}') -c rba-as -n ${AIOPS_NAMESPACE} -it -- /bin/bash

# An interactive shell inside the container will open
curl -k -vvv <Your Ansible host>
```

Solution: If responses are not received from Ansible and you cannot connect to Ansible, then create a NetworkPolicy that allows egress from the RBA service pods. Run the following commands:

```
export AIOPS_NAMESPACE=<AIOps installation namespace>

cat << EOF | oc apply -f -
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: aiops-rba-rbs-egress
  namespace: ${AIOPS_NAMESPACE}
spec:
  egress:
  - {}
  podSelector:
    matchLabels:
      app.kubernetes.io/component: rba-rbs
  policyTypes:
  - Egress
EOF

cat << EOF | oc apply -f -
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: aiops-rba-as-egress
  namespace: ${AIOPS_NAMESPACE}
spec:
  egress:
  - {}
  podSelector:
    matchLabels:
      app.kubernetes.io/component: rba-as
  policyTypes:
  - Egress
EOF
```

Troubleshooting Secure Tunnel

Learn how to isolate and resolve problems with Secure Tunnel.

- Secure Tunnel connector is not running after restart with Podman
- fuser hangs when Secure Tunnel install script is launched
- When creating a Secure Tunnel connection, the status remains in pending for a long time
- Slack is unable to verify the secure tunnel URL
- Secure Tunnel connector pod in crash loop state after installation on OpenShift Container Platform cluster

Secure Tunnel connector is not running after a restart with Podman

This issue can occur when you install the Secure Tunnel connector to a host machine on which Podman is installed. When the host machine is rebooted and the Secure Tunnel connector is checked by using the **`podman ps -a`** command, the Secure Tunnel connector container does not display **`running`** status.

If this issue occurs, the **`podman-restart`** service must be activated by using the **`systemctl`** command:

```
systemctl start podman-restart
systemctl enable podman-restart
```

After entering the command, check **`podman-restart`** worked by using the following command:

```
systemctl status podman-restart
```

If the Connector is still not running, try restarting the host machine.

fuser hangs when Secure Tunnel install script is launched

After launching the Secure Tunnel install script, fuser hangs and you see something that resembles this

```
Checking whether docker or podman is installed ...
docker found.
Starting docker engine...
fuser found.
```

This issue can occur when fuser is having trouble finding a specific available port and so the execution stalls.

Solution: This issue rarely happens but should you encounter it edit the `install-connector.sh` and add a timeout in the fuser command.

Change `PortCheck=$(fuser $HEALTHY_PORT/tcp 2>/dev/null)` to `PortCheck=$(timeout 2 fuser $HEALTHY_PORT/tcp 2>/dev/null)`. This change can resolve the issue.

When creating a Secure Tunnel connection, the status remains in pending for a long time

When you use the **Custom** method to create a Secure Tunnel connection, the **Status** of the connection is stuck in **Pending** for a long time. This also occurs when you create a secure tunnel connection that uses a shortcut (for example, ChatOps, VMware or OpenStack).

This issue can occur when the `tls secret` was not created by the CertManager but the tunnel connection is waiting for it.

Solution: In most cases, re-creating the connection resolves the problem. If that does not work, you need to run the following command to restart the **IBM Cert Manager** and re-create the certificate CR to resolve this problem. Replace the namespace used here, `cp4aiops`, with your own namespace.

1. Find the namespace that IBM Cert Manager is deployed in.

   ```
   oc get pod -A | grep cert-manager-controller | awk '{print $1}'
   ```

2. Export environment variables:

   ```
   export CERT_MGR_NAMESPACE=<cert_namespace>
   export CONNECTION_NAME=<conn_name>
   ```

   Where:

   - `<cert_namespace>` is the namespace that IBM Cert Manager is deployed in.
   - `<conn_name>` is the name of the tunnel connection

3. Run the following commandL:

   ```
   for NAME in $(oc -n ${CERT_MGR_NAMESPACE} get pod --no-headers | grep cert-manager | awk '{print $1}')
   do
       oc -n ${CERT_MGR_NAMESPACE} delete pod ${NAME}
   done
   CONNECTION_ID=`oc -n cp4aiops get TunnelConnection ${CONNECTION_NAME} -o=jsonpath={.status.id}`
   oc -n cp4aiops delete certificates.certmanager.k8s.io sre-tunnel-${CONNECTION_ID}-server-cert
   oc -n cp4aiops delete certificates.certmanager.k8s.io sre-tunnel-${CONNECTION_ID}-connector-cert
   oc -n cp4aiops delete certificates.certmanager.k8s.io sre-tunnel-${CONNECTION_ID}-api-cert
   ```

Slack is unable to verify the secure tunnel URL

You can encounter an issue with Slack being unable to verify the secure tunnel URL after you set up a ChatOps secure tunnel connection. This issue can occur even when the Secure Tunnel UI shows that the installation was successful.

This issue can occur if you enabled the Egress Network policy with the `enableEgressNetworkPolicy: true` when you install the Secure Tunnel.

You can use the following command to check whether the egress network policy is enabled. If the egress network policy is enabled the output should be `true`.

```
oc -n <the namespace of the aiops> get Tunnel sre-tunnel -o=jsonpath='{.spec.enableEgressNetworkPolicy}'
```

Solution: Use the following YAML code to help you create an egress network policy in your IBM Cloud Pak for AIOps project (namespace) to allow Secure Tunnel access to the `ibm-nginx` gateway.

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: chatopstunnel-egress-ibm-nginx
  namespace: <the namespace of the aiops>
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: sre-tunnel
      app.kubernetes.io/managed-by: tunnel
      app.kubernetes.io/name: sre-tunnel-chatopstunnel.app
      tunnel_connection_name: chatopstunnel
  egress:
    - to:
        - podSelector:
            matchLabels:
              app.kubernetes.io/component: ibm-nginx
              icpdsupport/app: gateway
              component: ibm-nginx
  policyTypes:
    - Egress
```

Secure Tunnel connector pod in crash loop state after installation on OpenShift Container Platform cluster

You might notice an issue where the Secure Tunnel connector pods stay in a crash loop state. The pod logs might show proxy related errors.

This issue can occur when the Secure Tunnel connector is installed on a cluster within an air-gapped (offline) environment that does not have outbound network access.

Solution: For the Secure Tunnel connector that is installed within the air-gapped (offline) environment, create a network connection with the Secure Tunnel server by using an HTTP proxy. You can create the connection by completing the following steps:

1. From a command line, go to the **tunnel-connector-install-scripts** directory.
2. Open the **tunnel-connector.yaml** file by using the following command:

   ```
   vi tunnel-connector.yaml
   ```

   When you open the YAML file, it can resemble the following example YAML file:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     labels:
       app.kubernetes.io/name: sre-tunnel-df012501eede4b02.secret
       app.kubernetes.io/managed-by: tunnel
       app.kubernetes.io/instance: sre-tunnel
       tunnel-network-name: woprod-test
       tunnel_connection_name: woprod-test
       tunnel_connection_id: "df012501eede4b02"
     name: sre-tunnel-df012501eede4b02-config
   data:
     config_connector.yaml: |
       <base64 encoded data>
   type: Opaque
   ```

3. Decode the **<base64 encoded data>** of the YAML file by using the following command:

   ```
   echo <base64 encoded data> | base64 -d
   ```

   Where **<base64 encoded data>** is the encoded data from the YAML file.

4. Update the **websocket_http_proxy** and **websocket_https_proxy** properties that are found in the decoded data.

   ```
   websocket_http_proxy: <http proxy endpoint>
   websocket_https_proxy: <https proxy endpoint>
   ```

   Where

   - **<http proxy endpoint>** is the HTTP proxy endpoint. This proxy is used by the tunnel connector to create a nonencrypted network web socket connection with the tunnel server. For example, the value can be **proxy.example.ibm.com:3128**.
   - **<https proxy endpoint>** is the HTTPS proxy endpoint. This proxy is used by the tunnel connector to create an encrypted network web socket connection with the tunnel server. For example, the value can be **proxysecure.example.ibm.com:3128**.

5. Encode the data by using the following command.

   ```
   echo '<decoded data>' | base64
   ```

   Where **<decoded data>** is the decoded data with the two modified properties.

6. Add the Base64 encoded data to the **<base64 encoded data>**.

   ```
   cat <<EOF | oc -n <namespace> apply -f -
   apiVersion: v1
   kind: Secret
   metadata:
     labels:
       app.kubernetes.io/name: sre-tunnel-df012501eede4b02.secret
       app.kubernetes.io/managed-by: tunnel
       app.kubernetes.io/instance: sre-tunnel
       tunnel-network-name: woprod-test
       tunnel_connection_name: woprod-test
       tunnel_connection_id: "df012501eede4b02"
     name: sre-tunnel-df012501eede4b02-config
   data:
     config_connector.yaml: |
       <base64 encoded data>

   type: Opaque
   EOF
   ```

   Where

   - **<base64 encoded data>** is the Base64 encoded data.
   - **<namespace>** is the namespace where Secure Tunnel connector is installed.

7. Restart the Secure Tunnel connector pods by using the following command:

   ```
   oc delete pod <secure-tunnel-connector-pod> -n <namespace>
   ```

   Where

   - **<secure-tunnel-connector-pod>** is the name of the Secure Tunnel connector pod.
   - **<namespace>** is the namespace where Secure Tunnel connector is installed.

Troubleshooting AI model management

Learn how to isolate and resolve problems with AI model training.

- Larger data sets are taking a long time to train
- Log anomaly detection training is taking longer than expected
- Network issues with Elasticsearch
- Network issues with Spark
- Training encounters insufficient resources
- Training reports failure but process completes
- Log anomaly detection model deployment fails with error undefined
- Ingested raw logs are not available
- Statistical baseline reference can require manual reset
- How to stop a precheck job
- Evaluation of historical metric data gives inconsistent results
- Model training considers less historical data in a data quality check
- Deployed model status in a metric anomaly or temporal grouping training algorithm is changed to pending
- Model version deployment reports failure despite being successfully deployed
- The Alert suppression XinY policies tile does not appear in AI model management
- Related events training can fail when when you are processing a high alert count
- Ticketing integration missing from list of integrations
- Metric anomaly detection training fails after an extended time period

Larger data sets are taking a long time to train

If your data sets are large, it can take a long time for AI training to complete. For example, when you run log anomaly training a large log data set might take a full day to train.

By default, two training pods are assigned to handle training jobs. This number of pods is sufficient to complete training within acceptable time periods if your data sets are not too large. However, larger data sets can take a considerable amount of time to train based on two training pods that are running in parallel.

Solution: To speed up training, you must increase the number of training pods that are dedicated to training. The default value is two pods, regardless of the type of training definition that you are running.

A limit applies to the number of training pods that can be dedicated to training. Before you increase the number of training pods that are dedicated to training, it is important for you to be aware of these limits.

Each additional training pod requires approximately the following extra resources:

- 2.5 CPU cores
- 2 - 16 GBytes of memory, depending on the AI type that is trained.

Complete the following steps:

1. Log in to your Red Hat® OpenShift® cluster, or SSH into a control plane node or the loadbalancer VM on your Linux® deployment. Then swtich to the namespace where IBM Cloud Pak for AIOps is installed.
2. Find the training resource quota by running the following command:

   ```
   oc get resourcequota | grep training
   ```

3. Edit the resource quota by running `oc edit resourcequota <name of resource quota found in step 3>`. Also, edit the `spec.hard.pods` field with the number of pods you want to dedicate to training simultaneously. Ensure that you have the needed resources as described in the Before you begin section.
4. Save the edit by typing `:wq` or `:x`. After a few minutes, the changes are automatically applied to your cluster, and you can start training.

Log anomaly detection training is taking longer than expected

Log anomaly detection training is taking longer than expected. Currently, log training times are on the order of 860 log records per second. This number is significantly slower than the 25,000 log records per second that is expected in enterprise environments. Log training time is proportional to the volume of log data passed through the training system.

This issue can occur as the training rate is constrained by the current system architecture.

Solution: This issue has no simple resolution. Depending on the data you are training, the following actions might have some impact on training times:

- Increase the number of training pods that are dedicated to training, as described in the **Solution** section of Larger data sets are taking a long time to train.
- Decrease the sampling size used for the training.

Network issues with Elasticsearch

The IBM Cloud Pak for AIOps IBM Cloud Pak for AIOps console is generating lots of network errors.

This issue occurs when Elasticsearch is in an unhealthy state. It occurs most likely due to issues with Elasticsearch connectivity.

Solution: Try to get more information on potential Elasticsearch issues by examining the AI Platform pod logs. Check to see whether any error logs associated with Elasticsearch connectivity exists.

1. Log in to your Red Hat® OpenShift® cluster, or SSH into a control plane node or the loadbalancer VM on your Linux® deployment. Then swtich to the namespace where IBM Cloud Pak for AIOps is installed.
2. Display the logs and look for error logs associated with Elasticsearch connectivity.

```
oc logs $(oc get pods | grep ai-platform-api-server| awk '{print $1}')
```

Network issues with Spark

The AI platform has problems either launching training jobs, or reporting ongoing training status for the temporal grouping AI algorithm. As a result, you encounter errors on the Manage AI algorithms page.

This issue can occur as training jobs for the temporal grouping AI algorithm are orchestrated by the internal component Spark. When there are issues with launching training jobs or training status is not being displayed, it indicates that Spark is in an unhealthy state.

Solution: Try to get more information on potential Spark issues by examining the AI Platform pod logs.

Proceed as follows:

1. Log in to your Red Hat® OpenShift® cluster, or SSH into a control plane node or the loadbalancer VM on your Linux® deployment. Then swtich to the namespace where IBM Cloud Pak for AIOps is installed.
2. Display the logs and look for error logs associated with Spark.

```
oc logs $(oc get pods | grep ai-platform-api-server| awk '{print $1}')
```

Training encounters insufficient resources

Training an AI algorithm returns an insufficient resources exception. The reason for this exception is that internal components within the training process were unable to locate a node with sufficient CPU or memory requirements to satisfy the training run.

Solution: The solution is to reduce the number of CPU and memory requests. Usually memory is the bottleneck so if you are unsure of what to do, a good option is to initially reduce memory by half.

1. Find the custom resource of the algorithm for which you want to alter the default resource requirements by running the following command:

```
oc get algorithms
```

2. On running the command, custom resources containing all the algorithm specifications are displayed as follows:

```
oc get algorithms

NAME                AGE
change-risk         46h
log-anomaly         46h
similar-incidents   46h
```

3. Edit the specific algorithm by running the following command:

```
oc edit algorithm <name>
```

4. Decode the *spec.manifestBase64* field. Once decoded, you see a *resources* field that is defined as follows:

```
resources:
    requests:
        cpu: 2
        memory: 4Gi
    limits:
        cpu: 2
        memory: 4Gi
```

Note: You can edit requests and limits (make sure that **limits=requests** for training algorithms), re-encode to base64, and then replace the *manifestBase64* field in the CR with this newly encoded manifest.

5. Save the changes, and proceed with your training.

Training reports failure but process completes

An AI algorithm training run has completed, but the IBM Cloud Pak for AIOps console shows that the training has failed. No concrete model was generated during the training run.

The reason for this failure might be insufficient data or a misconfiguration in the range of the data set. Failure does not mean that the training failed; it just means that no value was obtained from the training run and consequently no model was produced.

Solution: Examine the data sources for the training, examine the training configuration, make any necessary changes, and launch the training again.

Log anomaly detection model deployment fails with error undefined

When you are training the log anomaly detection algorithm with historical data and you attempt to deploy the AI model, you might encounter an issue where the model deployment fails with an `Error: undefined` error in the UI.

Solution: As a workaround when this issue occurs, restart all **cp4waiops-eventprocessor** pods, which can cause the model deployment to complete successfully. To restart the pods, run the following command, which deletes the current instance of the pods and creates and starts a new instance of the pods.

```
oc delete po cp4waiops-eventprocessor-eve-29ee-ep-jobmanager-0 cp4waiops-eventprocessor-eve-29ee-ep-jobmanager-1
cp4waiops-eventprocessor-eve-29ee-ep-taskmanager-0 cp4waiops-eventprocessor-eve-29ee-ep-taskmanager-1
```

Note: If you installed IBM Cloud Pak for AIOps with a larger production deployment sizing, you need to specify any additional **cp4waiops-eventprocessor** pods that exist when you run the preceding command.

Ingested raw logs are not available immediately

Ingested raw logs are not available immediately for "Log anomaly detection - natural language" model training after the import is complete. Raw logs are not available for review in the "Preview log" section of the ChatOps Alert Viewer.

This issue can occur as additional time is required to store normalized logs in Elasticsearch after they are ingested. The amount of time can be reduced by allocating more Flink tasks to the Flink job that stores the normalized logs.

Solution: To resolve this issue, complete the following steps:

1. The environment variables that control the number of Flink tasks that are used to move logs to Elasticsearch are located in the Kubernetes deployment for aimanager-aio-controller. Edit this deployment: oc edit deployment aimanager-aio-controller
2. Find the environment variables by searching for "LOGS_JOB2_BASE_PARALLELISM". Change the value of LOGS_JOB2_BASE_PARALLELISM and LOGS_JOB2_NUM_PARTITIONS from their default values of 2 to something higher. Consider doubling the value to see whether it has the wanted effect.
3. Save the yaml and wait for the pods to restart. You can verify that the pods have restarted by monitoring the output of: oc get pods | grep aimanager-aio-controller
4. Stop the data flow for all logs integrations. It is important that there are no active logs integrations, as this is how the controller knows that it is OK to stop the Flink job, which copies the logs to Elasticsearch.
5. Restart the data flow for all logs integrations.

The Flink job restarts by using a higher number of tasks. It copies the logs to Elasticsearch more quickly.

Be careful not to allocate a higher parallelism than you have Flink task slots available. The number of tasks that you need can be calculated by adding the following values:

- sum of base_parallelism from all defined integrations, as defined in the Integration UI
- value of LOGS_JOB3_BASE_PARALLELISM multiplied by the number of logs type defined integrations (for example, Mezmo, Falcon Logscale, Splunk, and so on)
- value of LOGS_JOB2_BASE_PARALLELISM
- value of LOGS_JOB4_BASE_PARALLELISM

If you need additional task manager slots, you can add a task manager by Increasing data streaming capacity.

Statistical baseline reference can require manual reset

There is a chance that infrequently occurring anomalies are learned as normal behavior and they might not be detected anymore. For instance, if interval for the infrequently occurring anomalies do not exceed the predefined threshold of 0.15, the interval is not tagged as biased and the reference values are updated and the anomaly is learned as normal behavior. If this occurs, you need to reset the learning by resetting the reference of the statistical baseline with the following steps:

1. Log in to your Red Hat® OpenShift® cluster, or SSH into a control plane node or the loadbalancer VM on your Linux® deployment. Then swtich to the namespace where IBM Cloud Pak for AIOps is installed.

   ```
   export PROJECT_CP4AIOPS=<project>
   ```

   Where **<project>** is the namespace (project) that IBM Cloud Pak for AIOps is deployed in.

2. Reset the statistical baseline reference:

   ```
   export ELASTIC_SET="$(oc get statefulset -l app.kubernetes.io/managed-by=ibm-elasticsearch -o
   jsonpath='{.items[0].metadata.name}')"
   oc exec "statefulset/${ELASTIC_SET}" -- curl -XDELETE -k --fail "http://127.0.0.1:19200/1000-1000-reference_oob"
   ```

How to stop a precheck job

The workaround is to stop the precheck job manually using Openshift CLI commands, delete the training definition, recreate the training definition, and relaunch the training configuration with the correct details.

There are two methods to do this. The first is recommended as it limits mistakes and there is less risk of making training off sync with data in the Elasticsearch database, but the models are deleted if you use this method, whereas with the second method that is listed they will not, so it's preferable to use the second method if you want to conserve the models in question.

Method 1: Stop the precheck job and delete the training definition to start afresh

1. Retrieve the precheck job in the IBM Cloud Pak for AIOps namespace, or a different namespace that it is installed under:

   ```
   oc get jobs | grep precheck
   ```

2. Delete the precheck job in order to stop the precheck run:

   ```
   oc delete job <job-name>
   ```

3. Delete the training definition using the delete button on the console.

4. Re-create the training definition with the correct configurations required for the training algorithm, for example date range or scheduling.
5. Run the training as usual.

Method 2: Stop precheck job and clean up Elasticsearch data

1. Retrieve the precheck job in the IBM Cloud Pak for AIOps namespace, or a different namespace that it is installed under:

   ```
   oc get jobs | grep precheck
   ```

2. Delete the precheck job in order to stop the precheck run:

   ```
   oc delete job <job-name>
   ```

3. Retrieve the AI Platform pod name in the IBM Cloud Pak for AIOps namespace or different namespace that it is installed under:

   ```
   oc get pod | grep api-server
   ```

4. Enter inside the pod:

   ```
   oc rsh <pod-name> # Example: oc rsh aimanager-aio-ai-platform-api-server-55d5b59bd4-jk7nz
   ```

5. Verify the latest precheck run and retrieve the '_id' required for cleanup:

   ```
   curl -k -u <username>:<password> -X POST $ES_URL/precheckrun/_search?pretty=true" -H 'Content-Type:
   application/json' -d '{"query": {"match_all": {}},"size": 1,"sort": [{"timestamp": {"order": "desc"}}]}'
   ```

   Example:

   ```
   "_id":
   "log_anomaly_detection_configurationv3"
   ```

6. Use the '_id' and version specified in the ID to delete the precheck respective run entries in elasticsearch:

   ```
   curl -k -u <username>:<password> -X POST $ES_URL/precheckrun/_delete_by_query" -H 'Content-Type: application/json' -
   d '{"query": {"term": {"_id": <ID retrieved from previous step>}}}'
   ```

   Example:

   ```
   curl -k -u <username>:<password> -X POST $ES_URL/precheckrun/_delete_by_query" -H 'Content-Type: application/json' -
   d '{"query": {"term": {"_id": "log_anomaly_detection_configurationv3"}}}'
   ```

   ```
   curl -k -u <username>:<password> -X POST $ES_URL/precheckstatus/_delete_by_query" -H 'Content-Type:
   application/json' -d '{"query": {"term": {"precheckRunVersion": <version retrieved from previous step>}}}'
   ```

   Example:

   ```
   curl -k -u <username>:<password> -X POST $ES_URL/precheckstatus/_delete_by_query" -H 'Content-Type:
   application/json' -d '{"query": {"term": {"precheckRunVersion": "v3"}}}'
   ```

7. Update the training definition document to decrement the precheck run version:

   ```
   curl -k -u <username>:<password> -X POST $ES_URL/trainingdefinition/_update/log_anomaly_detection_configuration" -H
   'Content-Type: application/json' -d '{"doc": {"precheckRunVersion": <decremented version>}}'
   ```

   Example:

   ```
   curl -k -u <username>:<password> -X POST $ES_URL/trainingdefinition/_update/log_anomaly_detection_configuration" -H
   'Content-Type: application/json' -d '{"doc": {"precheckRunVersion": "v2"}}'
   ```

8. Make your changes to the training configuration, for example to the date range or scheduling.
9. Run the training as usual.

Evaluation of historical metric data gives inconsistent results

Sending historical metric data for training – such as in a product demonstration – you see inconsistent outcomes in the IBM Cloud Pak for AIOps console metric anomaly results.

At inference (metric evaluation) time, it takes a short amount of time to load analytics models into memory. This delay might lead to some metric data being left out of an evaluation.

Solution: Send in one datapoint, for each individual metric, to allow the model to be loaded. Then, send remaining data after a short interval, 30 - 60 seconds later.

Model training considers less historical data in a data quality check

Less data is included in a data quality check if you select **Training** first instead of running a precheck and training separately.

If you select **Training** to run both a data quality check and training sequentially, you might notice this problem. This situation occurs only when some integrations are moved into live data after the training completes, and the training dates are changed to check historical data from another integration source.

Solution: Run the precheck and training separately. Start the data quality precheck, and after the precheck is complete, run the training after.

Deployed model status in a metric anomaly or temporal grouping training algorithm is changed to pending

A temporal grouping or a metric anomaly detection algorithm is in a pending state although the model was previously trained and deployed with success.

Solution: To resolve this issue, complete the following steps:

1. Run the `oc -n <namespace> rsh <aimanager-aio-ai-platform-api-server-*>` command.
2. Delete all versions of the `metric_anomaly_detection_configuration` or `temporal_grouping_configuration` training algorithms in the `trainingrun` index. Search the index by running the following command:

   ```
   ./elastic.sh -X GET -E trainingrun/_search\?pretty=true&size=1000
   ```

3. Identify all the documents that are related to your `metric_anomaly_detection_configuration` or `temporal_grouping_configuration` training algorithms. The `_id` field of the documents is `metric_anomaly_detection_configurationv<version>` or `temporal_grouping_configurationv<version>`, where `<version>` is the version of the training run.
4. Delete the training run documents:

   ```
   ./elastic.sh -X DELETE -E trainingrun/_doc/metric_anomaly_detection_configurationv1
   ./elastic.sh -X DELETE -E trainingrun/_doc/metric_anomaly_detection_configurationv2
   [..]
   ```

   or

   ```
   ./elastic.sh -X DELETE -E trainingrun/_doc/temporal_grouping_configurationv1
   ./elastic.sh -X DELETE -E trainingrun/_doc/temporal_grouping_configurationv2
   [..]
   ```

5. Delete the training definition to clean the UI:

   ```
   ./elastic.sh -X DELETE -E trainingdefinition/_doc/metric_anomaly_detection_configuration
   ```

   or

   ```
   ./elastic.sh -X DELETE -E trainingdefinition/_doc/temporal_grouping_configuration
   ```

6. Now, you can train a new model.

Model version deployment reports failure despite being successfully deployed

When you deploy LAD model version manually, you might encounter an error message on your console that reports deployment failure. The error message reads that the failure occurs due to the version being already deployed.

This error arises due to the successful deployment of the model version in the background despite getting timed out.

Solution: Restart the data flow for all the integrations. To do so, first disable and then re-enable the data flow for all integrations.

The Alert suppression XinY policies tile does not appear in AI Model Management

In the IBM Cloud Pak for AIOps console, click **AI model management** and find the *Alert suppression XinY policies* tile. If this tile does not appear, an administrator or developer user needs to restart the `aimanager-aio-ai-platform-api-server` pod.

To restart the target pod, delete the pod so that a new pod automatically starts up:

1. Connect to your cluster with the `oc login` command.
2. Restart the `aimanager-aio-ai-platform-api-server` pod.

   ```
   oc delete pod/"$(oc get pod|grep aimanager-aio-ai-platform-api-server|awk '{print$1; exit}')"
   ```

   Successfully running this command produces output similar to the following example:

   ```
   pod "aimanager-aio-ai-platform-api-server-55c5d975c4-6qpwk" deleted
   ```

Related events training can fail when when you are processing a high alert count

When you are running related events training on a data set that includes a high number of alerts (approximately 11.5M or more alerts), the training can fail. When this failure occurs the Spark log includes thrown exceptions.

The spark log exceptions can resemble the following output:

```
24/02/23 10:16:01 ERROR TaskSetManager: Task 0 in stage 3215.0 failed 4 times; aborting job
24/02/23 10:16:01 INFO TaskSchedulerImpl: Removed TaskSet 3215.0, whose tasks have all completed, from pool
24/02/23 10:16:01 INFO DAGScheduler: Job 326 failed: $anonfun$withThreadLocalCaptured$1 at <unknown>:0, took 0.996638 s
24/02/23 10:16:01 INFO TaskSchedulerImpl: Cancelling stage 3215
24/02/23 10:16:01 INFO TaskSchedulerImpl: Killing all running tasks in stage 3215: Stage cancelled: Job aborted due to
stage failure: Task 0 in stage 3215.0 failed 4 times, most recent failure: Lost task 0.3 in stage 3215.0 (TID 2975)
(10.254.75.83 executor 0): org.apache.spark.SparkException: Checkpoint block rdd_680_0 not found! Either the executor
that originally checkpointed this partition is no longer alive, or the original RDD is
unpersisted. If this problem persists, you may consider using `rdd.checkpoint()`
instead, which is slower than local checkpointing but more fault-tolerant.
        at org.apache.spark.errors.SparkCoreErrors$.checkpointRDDBlockIdNotFoundError(SparkCoreErrors.scala:102)
        at org.apache.spark.rdd.LocalCheckpointRDD.compute(LocalCheckpointRDD.scala:61)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:364)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:328)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:362)
```

```
        at org.apache.spark.rdd.RDD.$anonfun$getOrCompute$1(RDD.scala:378)
        at org.apache.spark.storage.BlockManager.$anonfun$getOrElseUpdate$1(BlockManager.scala:1372)
        at org.apache.spark.storage.BlockManager.$anonfun$doPutIterator$1(BlockManager.scala:1601)
        at org.apache.spark.storage.BlockManager.org$apache$spark$storage$BlockManager$$doPut(BlockManager.scala:1528)
        at org.apache.spark.storage.BlockManager.doPutIterator(BlockManager.scala:1592)
        at org.apache.spark.storage.BlockManager.getOrElseUpdate(BlockManager.scala:1389)
        at org.apache.spark.storage.BlockManager.getOrElseUpdateRDDBlock(BlockManager.scala:1343)
        at org.apache.spark.rdd.RDD.getOrCompute(RDD.scala:376)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:326)
        at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:52)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:364)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:328)
        at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:52)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:364)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:328)
```

Solution: As a workaround if this failure occurs, increase the spark worker memory limit and run the training again.

The spark worker memory limit can be modified by using the following command:

```
oc edit sts aiops-ir-analytics-spark-worker
```

The field that needs modification is the **memory** field as shown in the following example output:

```
resources:
  limits:
    memory: 8000Mi
```

Ticketing integration missing from list of integrations

If a ticketing integration is missing from the list of integrations in a training module such as Similar tickets or Change risk, you should update that integration. To do this, edit the integration and either update the existing description or add a new one, then save your changes. This action will trigger a resynchronization of the connector schema for that integration. After a few minutes, the updated integration should appear in the trainer **Integrations** table.

Metric anomaly detection training fails after an extended time period

Previously deployed metric anomaly detection training models stop running successfully.

An error similar to the following example appears in the Cassandra logs:

```
ERROR [ReadStage-2] 2024-10-22 11:53:22,943 StorageProxy.java:2175 - Scanned over 100001 tombstones during query 'SELECT
value FROM tararam.dt_metric_value WHERE token(t_uid, mr_id) >= token('cfd95b7e-3bc7-4006-a4a8-a73a79c71255', 28f522d0-
5c6a-3d8a-b867-8a449f3e5fa9) AND token(t_uid, mr_id) <= 5156857202419367684 LIMIT 1000 ALLOW FILTERING' (last scanned row
token was 4404220741919580179 and partion key was ((cfd95b7e-3bc7-4006-a4a8-a73a79c71255, bab5050a-5780-3e92-b8b4-
8baa00cbfe42), 2024-09-27T18:25:00.000Z)); query aborted
```

Solution: Clear the already expired metric data and then restore the expiration interval to its original value.

1. Connect to the Cassandra database:

   ```
   Login the CQL for cassandra:
          export CASSANDRA_PASSWORD=$(oc get secret aiops-topology-cassandra-auth-secret --template={{.data.password}}
   | base64 --decode);echo $CASSANDRA_PASSWORD
          oc exec -ti aiops-topology-cassandra-0 -- /opt/ibm/cassandra/bin/cqlsh -u admin -p <PASSWORD FROM PREVOUS> --
   ssl
   ```

2. Set the `gc_grace_seconds` period on the metric data table to 10 minutes:

   ```
   cqlsh> ALTER TABLE tararam.dt_metric_value WITH gc_grace_seconds=600;
   ```

3. Wait at least 10 minutes for the data to expire.
4. Then, set the table back to its original value:

   ```
   cqlsh> ALTER TABLE tararam.dt_metric_value WITH gc_grace_seconds=864000;
   ```

Troubleshooting alert management

Learn how to isolate and resolve problems with alert management.

- No alert or incident created when an event is sent
- No alert ranking with large installations
- Closed alerts are not being removed from Alert Viewer
- No alerts shown in Alert Viewer and "Unknown error" message is displayed
- Alert list ignores saved view sort order
- "Unknown error" message in Alert Viewer when you click the alert view without selecting the ID and Summary columns

No alert or incident created when an event is sent

Symptom A: An event is sent, but no alert is received by IBM Cloud Pak® for AIOps and no incident is created.

Running `waiops-mustgather.sh -V flink-jobs` using the must-gather tool, returns `"RESTARTING"` or `"FAILED"` under the entry `[FLINK = aiops-ir-lifecycle]`. For example:

```
[FLINK = cp4aiops]
{
  "jobs": []
}

[FLINK = aiops-ir-lifecycle]
{
  "jobs": [
    {
      "id": "4e442c7880239bede340e953e6b920b3",
      "status": "RESTARTING"
    }
  ]
}
```

This issue might be caused by a volume of event data that is too large for the given installation size and configured policy set. This can cause the policy engine to run out of available memory.

Solution: To resolve this issue, complete the following steps:

1. Evaluate the sizing of the installation compared to the current event load. If there is a miss-match, change the installed sizing, or the configured event sources.
2. Ensure that incident creation policies are only configured to raise incidents for alerts which require immediate resolution. Incidents are more resource intensive to process than alerts, so creating incidents for all alerts will require a larger sized installation.
3. After re-configuring the system, the policy engine may still be in a failure state as it tries to re-process the already-inserted events. It is possible to reset the policy engine so that it only processes new events using the must-gather tool:

   ```
   waiops-mustgather.sh -DR -C clear-lifecycle-state.sh
   ```

   NOTE: Resetting the policy engine will cause all policy state to be lost. This may result in events creating new alerts and incidents rather than being deduplicated into existing ones. It will also mean that any events inserted in the time between the failure and running this command will not be processed by the system.
4. Restart the `flink` pods:
   1. Search for the pods that match the string `aiops-ir-lifecycle-flink`
      This should return four pods.
   2. Right-click and **Delete** the pods.
      The pods should restart immediately.

Symptom B: An event is sent, but no alert is received by IBM Cloud Pak® for AIOps and no incident is created. The log of the lifecycle task manager pod contains the following error:

```
WARNING: Exception thrown during asynchronous load
java.util.concurrent.CompletionException: io.grpc.StatusRuntimeException: UNAVAILABLE: ssl exception
```

The issue might be a certificate mismatch between the lifecycle operator and the policy GRPC server, causing the connection to fail with SSL exceptions.

Solution: To resolve this issue, complete the following steps:

1. Run the following command to delete the certificate:

   ```
   oc delete secret aiops-ir-lifecycle-policy-grpc-clt-svc aiops-ir-lifecycle-policy-grpc-svc
   ```
2. Get the lifecycle-operator pod name:

   ```
   oc get pods |grep ir-lifecycle-operator-controller-manager
   ```
3. Restart the lifecycle operator:

   ```
   oc delete pod ir-lifecycle-operator-controller-manager-<lifecycle-operator-pod-name>
   ```

No alert ranking with large installations

On larger production scale installations of IBM Cloud Pak® for AIOps, no ranking of alerts is displayed in the Alert Viewer.

Solution: To resolve this issue, complete the following steps:

1. Run the following command to find the `classifier` and `probablecause` pods:

   ```
   oc get pods | egrep -i "classifier|probablecause"
   ```

   For example:

   ```
   oc get pods | egrep -i "classifier|probablecause"

   aiops-ir-analytics-classifier-6477fd7788-4rzwx                    1/1     Running     0
   aiops-ir-analytics-classifier-6477fd7788-8jd55                    1/1     Running     0
   aiops-ir-analytics-probablecause-5b89d56c44-b752h                 1/1     Running     1
   aiops-ir-analytics-probablecause-5b89d56c44-kpqcb                 1/1     Running     1
   ```
2. Restart the pods:

```
oc delete pods aiops-ir-analytics-classifier-6477fd7788-4rzwx aiops-ir-analytics-classifier-6477fd7788-8jd55 aiops-ir-analytics-probablecause-5b89d56c44-b752h aiops-ir-analytics-probablecause-5b89d56c44-kpqcb
```

Closed alerts are not being removed from Alert Viewer

You might notice an issue where closed alerts are not being removed from the Alert Viewer.

If this issue occurs, you might see an error that is related to timeouts for requests to Postgres. This error can be similar to the following example:

```
[2023-12-12T17:30:30.022Z] ERROR: datalayer.postgres/14 on aiops-ir-core-ncodl-jobmgr-7f4c64dcdd-d7xjz: Error detected on streaming query
    Error: Query timed out
        at Timeout._onTimeout (/app/lib/helpers/dbservice/lib/helpers/dbservice/postgres.js:178:20)
        at listOnTimeout (node:internal/timers:559:17)
        at processTimers (node:internal/timers:502:7)
[2023-12-12T17:30:30.022Z] ERROR: datalayer.postgres/14 on aiops-ir-core-ncodl-jobmgr-7f4c64dcdd-d7xjz: Error detected on client connection. Ending query stream (reqid=13778660-9914-11ee-844d-e573891ebc4b)
    Error: Query timed out
        at Timeout._onTimeout (/app/lib/helpers/dbservice/lib/helpers/dbservice/postgres.js:178:20)
        at listOnTimeout (node:internal/timers:559:17)
        at processTimers (node:internal/timers:502:7)
[2023-12-12T17:30:30.024Z] ERROR: datalayer.postgres/14 on aiops-ir-core-ncodl-jobmgr-7f4c64dcdd-d7xjz: Error detected on streaming query cursor
    Error: Connection terminated
        at Connection.<anonymous> (/app/node_modules/pg/lib/client.js:132:36)
        at Object.onceWrapper (node:events:627:28)
        at Connection.emit (node:events:525:35)
        at Socket.<anonymous> (/app/node_modules/pg/lib/connection.js:63:12)
        at Socket.emit (node:events:525:35)
        at TCP.<anonymous> (node:net:301:12)
```

This issue might be caused by the failure of the datalayer job manager. Since the job is not running, there is no automated process for clearing the closed alerts.

Solution: Use the following steps to resolve the closed alerts issue:

1. Determine the name of the datalayer job manager pod.

   ```
   oc get pod -l "app.kubernetes.io/component=ncodl-jobmgr"
   ```

   The output from the preceding command can look like the following example::

   ```
   [root@6c025b05c0a7 /]# oc get pod -l "app.kubernetes.io/component=ncodl-jobmgr"
   NAME                                        READY   STATUS    RESTARTS   AGE
   aiops-ir-core-ncodl-jobmgr-85458b4979-gqgj8   1/1     Running   0          5h11m
   ```

2. Use the pod name to describe the job manager pod. If the pod is in a crash loop or not scheduled status, use the following command to see the failure:

   ```
   oc describe pod <pod name>
   ```

   Where **<pod name>** is the name of the datalayer job manager pod.

   If the failure reason from describe command is not clear or the pod appears to be running fine then check the pod logs for potential run failure conditions. Use the following command to check the logs:

   ```
   oc logs <pod name>
   ```

   Where **<pod name>** is the name of the datalayer job manager pod.

   You can, for example, search for "Alert delete job" in the logs. "Alert delete job" must be running. The logs for delete job are shown in the following example.

   ```
   [2023-12-12T17:38:00.002Z]  INFO: datalayer.jobmgr/14 on aiops-ir-core-ncodl-jobmgr-7f4c64dcdd-d7xjz: Running: Alert delete job [2023-12-12T17:38:30.120Z]  INFO: datalayer.jobmgr/14 on aiops-ir-core-ncodl-jobmgr-7f4c64dcdd-d7xjz:
   Complete: Alert delete job
    metrics: {
      "fetched": 0,
      "deleted": 0,
      "failed": 0
    }
    --
    psql: [
      {
        "name": "deletes",
        "mean": 0,
        "min": 0,
        "max": 0
      },
      {
        "name": "relatedDeletes",
        "mean": 0,
        "min": 0,
        "max": 0
      }
    ]
   ```

3. Use the following command to check whether the postgres instance is **Running**:

   ```
   oc get pods | grep postgres
   ```

The postgres pod name might be similar to **ibm-cp-watson-aiops-edb-postgres-***. The asterisk (*) represents multiple pods that might be running for postgres.

4. If the Postgres instance is running fine, then it is possible that the instance might be busy and overloaded. If you see the preceding error such as "timeouts on request Postgres" or if the Postgres instance is running fine, in both cases use the following script to manually purge and recover the system.

```
#!/bin/bash

EDBPOD=ibm-cp-watson-aiops-edb-postgres-1
EBDSECRET=aiops-ir-core-postgresql

# Get the credentials from required sec
USERNAME=$(oc get secret ${EBDSECRET} -o=jsonpath='{.data.username}' | base64 -d)
PASSWORD=$(oc get secret ${EBDSECRET} -o=jsonpath='{.data.password}' | base64 -d)
DBNAME=$(oc get secret ${EBDSECRET} -o=jsonpath='{.data.dbname}' | base64 -d)
SCHEMA=$(oc get secret ${EBDSECRET} -o=jsonpath='{.data.schema}' | base64 -d)

# RSH into EDB container.
oc rsh ${EDBPOD} psql -h localhost -p 5432 -U ${USERNAME} "dbname=${DBNAME} sslmode=require" -- <<EOF
${PASSWORD}

DELETE FROM aiops_irb.alerts WHERE state=2;
DELETE FROM aiops_irb.insights insight WHERE NOT EXISTS(SELECT 1 FROM aiops_irb.alerts alert WHERE insight.parentid
= alert.alertid) AND NOT EXISTS(SELECT 1 FROM aiops_irb.stories story WHERE insight.parentid = story.storyid);
DELETE FROM aiops_irb.links link WHERE NOT EXISTS(SELECT 1 FROM aiops_irb.alerts alert WHERE link.parentid =
alert.alertid) AND NOT EXISTS(SELECT 1 FROM aiops_irb.stories story WHERE link.parentid = story.storyid);
DELETE FROM aiops_irb.timeline timeline WHERE NOT EXISTS(SELECT 1 FROM aiops_irb.alerts alert WHERE
timeline.parentid = alert.alertid) AND NOT EXISTS(SELECT 1 FROM aiops_irb.stories story WHERE timeline.parentid =
story.storyid);
DELETE FROM relationships relation WHERE NOT EXISTS(SELECT 1 FROM aiops_irb.alerts alert WHERE relation.childid =
alert.alertid) AND NOT EXISTS(SELECT 1 FROM aiops_irb.stories story WHERE relation.parentid = story.storyid);
DELETE FROM aiops_irb.alerts_details detail WHERE NOT EXISTS(SELECT 1 FROM aiops_irb.alerts alert WHERE
detail.identifier = alert.identifier);
EOF
```

Where **EDBPOD** is the Postgres pod name that you obtained previously.

The script deletes all closed alerts from the Alert Viewer. The job manager will also be able to remove the future closed alerts automatically.

A sample output from successful execution of the script:

```
Defaulted container "postgres" out of: postgres, bootstrap-controller (init)
Password for user aiops_irb:
DELETE 0
DELETE 0
DELETE 0
DELETE 0
DELETE 0
DELETE 0
```

The number after the **DELETE** keyword is the number of rows that are deleted.

"Unknown error" message in Alert Viewer and no alerts displayed

Symptom: No alerts are shown in the Cloud Pak for AIOps Alert Viewer and "Unknown error" message is displayed on the UI.

Cause: The problem might be that the number of alerts exceeds what the Alert Viewer can process. A production deployment of Cloud Pak for AIOps can process a standing alert count of up to 200,000 alerts, with a chosen subset displayed in the Alert Viewer. The precise number of alerts that can be shown depends on different permutations of variables that affect the performance of the Alert Viewer. For example, standing alert count, filtered alert count, and view used (such as simple columns with alert data, or complex columns with insights and custom user enriched columns).

Solution: At first (during deployment or creation of a new view), try excluding columns with insights or columns based on custom properties. You can then add these columns sequentially to better gauge how many alerts a view can handle. For more information, see Creating views.

You can also restrict the number of alerts that are returned by the data layer to the Alert Viewer. This can improve performance issues on the backend. For more information, see Restricting the number of alerts returned by the data layer to the Alert Viewer.

Alert list ignores saved view sort order

Symptom: The alerts table sort order you wanted to apply to it does not save as expected.

Cause: It might be related to a similar issue, which deals with editing the alert view and how it then does not show the correct configuration when launched for the first time. This is due to having to fetch users and groups before being able to show the right view configuration. This is being fixed now also, so should no longer be an issue in future releases.

Solution: If the original alert table sort order didn't save as expected, you have to apply the sort order again after the alerts table is loaded.

"Unknown error" message in Alert Viewer when you click the alert view without selecting the ID and Summary columns

Symptom: You may get the unknown error when you click the alert view without having selected the ID and SUMMARY columns.

Cause: This problem is because the view requires the ID and SUMMARY columns to be part of the view. If these columns are not included in the view then it will be unusable and the only way to delete that view will be by using the API.

Solution: Include the ID and SUMMARY columns in the alert view. If you do not and cannot then delete the unusable view that results, you will have to delete the view using the API.

Troubleshooting Runbook Automation

Learn how to isolate and resolve problems with Runbook Automation.

- SSH action output shows "Permission denied" error
- SSH automation action output for PowerShell shows garbled multi-byte characters
- Exported runbook includes password from referenced action in clear text (if any)
- Runbook approval request not visible

SSH action output shows "Permission denied" error

Symptom: When attempting to run an SSH action script on some Linux target system, an error such as `bash: /tmp/tmp.gXO9WCs7Zy: Permission denied` is displayed in the script output area.

Cause: In order to run an SSH script on a Linux target system, Runbook Automation writes a temporary script file to the `/tmp` directory of the target system, and executes it from there. So the user on the target system that is configured to run the SSH script remotely does not only need to have write access to the `/tmp` directory , but also the `/tmp` directory itself must not be mounted with the `noexec` property.

Solution: Mount the `/tmp` directory without the `noexec` property, and verify that scripts can be run from within `/tmp`. Then retry running the SSH action.

If using the `/tmp` directory with write and execute access is prohibited in your organization then you can configure `mktemp` to use a different directory, for example, by generally setting the target system user's `TMPDIR` environment variable to an alternative directory path. See Creating an SSH integration for more details on the requirements for target systems of SSH connections.

SSH automation action output for PowerShell shows garbled multi-byte characters

Symptom: When running a PowerShell script on a Windows target system that is expected to print multi-byte or other non-ASCII characters, these characters are not properly displayed. Instead, these characters are replaced with question marks.

Cause: Runbook Automation expects the script output to be returned in UTF-8 format, but PowerShell is not configured to use UTF-8 as the default console output encoding.

Solution: Add an instruction to your PowerShell script code to set the console output encoding to UTF-8. For example, start your script with the following command:

```
$OutputEncoding = [Console]::OutputEncoding = [Text.UTF8Encoding]::UTF8
```

See the Microsoft documentation for more details on character encoding in PowerShell.

Exported runbook includes password from referenced action in clear text (if any)

Symptom: If an HTTP action has parameters for basic authentication configured, and the action is referenced by a runbook, then exporting the runbook using the RBA API includes the password in clear text.

Cause: The exported runbook is intended to be self-contained, so when imported to the same or a different deployment, the runbook is usable again (which includes having the password available). However, there is no password encryption/decryption mechanism in place for this scenario.

Solution:

- Avoid specifying passwords within actions.
- Store the exported runbook file in a place that only persons with a need-to-know for the password(s) have access to.
- Encrypt the exported file manually.

Runbook approval request not visible

Symptom: When the approval flow is enabled, and a runbook has been previously published, then a new request for approval for an updated draft of the runbook is not visible in the runbook library table. The expected approval status is "Approval pending". Instead, the runbook continues to be marked as "Approved".

Cause: The runbook library table displays the data from the last published version of the runbook, or from the draft runbook if it has never been published before. Therefore, approval requests for new draft runbooks are visible, but not for runbooks that have been previously published.

Solution: As author of the change, notify the approver which runbook needs to be reviewed and approved. As approver, open the runbook in "edit" or "preview" mode to see the latest changes. Directly communicate with the author if changes are needed. Once the runbook is ready to be published, disable the approval flow temporarily, and publish the runbook.

Troubleshooting ChatOps

Learn how to isolate and resolve problems with ChatOps IBM Cloud Pak® for AIOps.

- Slack buttons are unresponsive
- Log preview dialog or View alerts dialog hangs intermittently in an incident in Slack or Microsoft Teams
- Alert and incidents do not appear in Slack
- Slack ChatOps integration is not working within an air-gapped (offline) environment
- Microsoft Teams ChatOps integration shows no response in the console
- During a ChatOps secure tunnel creation an 'installation failed' message displays
- Show more button does not work in Slack after an upgrade

Slack buttons are unresponsive

Unresponsive buttons might be caused by a number of issues. If you encounter this issue, try completing the following steps to find the cause of the problem:

1. **Ensure two-way communication is established:**
    1. **Test two-way communication:** Incoming incidents are proof that one-way communication from IBM Cloud Pak for AIOps to Slack is successful. To check two-way communication, tag the IBM Cloud Pak for AIOps bot in the Slack channel. It responds with "Hey there I'm up and running." If it doesn't, IBM Cloud Pak for AIOps might not be receiving any communication from Slack.
    2. **Verify endpoints:** If the IBM Cloud Pak for AIOps bot did not respond, it might be that Slack is trying to communicate with the wrong endpoints, or the endpoints are not publicly accessible. To make sure that the endpoints are correct, open up the slack api for the app and navigate to the app manifest. The `event_subscriptions` and `interactivity` urls should match the ones that are produced by the IBM Cloud Pak for AIOps slack configuration and should be verified by Slack. If they do match but Slack has not verified them, that means Slack cannot access the endpoints likely because they are private. Slack needs some kind of proxy to access the private endpoints. A secure tunnel can be used in scenarios like this. For more information, see Secure Tunnel.
2. **Check communication speed:** If there are no problems with the steps that are outlined above, then it is possible the communication between Slack and IBM Cloud Pak for AIOps is taking too long. If a button is clicked, Slack waits 3 seconds to receive a response. If it doesn't get one, the communication is rejected and the button click fails. This is a rare occurrence, but can happen if the network communication is too slow, or if something else is preventing communication from happening in a timely manner. Try reconnecting and reconfiguring the Slack integration. For more information, see Creating a Slack integration. Then, verify that the issue is resolved.
3. **Look for errors in the ChatOps pods:** If 2-way comms are successful, and network communication seems to be fine, try logging in to the cluster to look at the logs. Look at the `aimanager-aio-chatops-slack-integrator` and `aimanager-aio-chatops-orchestrator` pods for any unusual errors and then contact the L2 support team.

Log preview modal or View alerts modal hangs intermittently in an incident in Slack or Microsoft Teams

This issue might happen when the Log Anomaly detection is in a high load, and takes longer time than normal to respond. When it occurs, you might encounter the following :

- When you select **View alerts** in an incident in Slack or Microsoft Teams, it hangs with the `Loading ...` message.

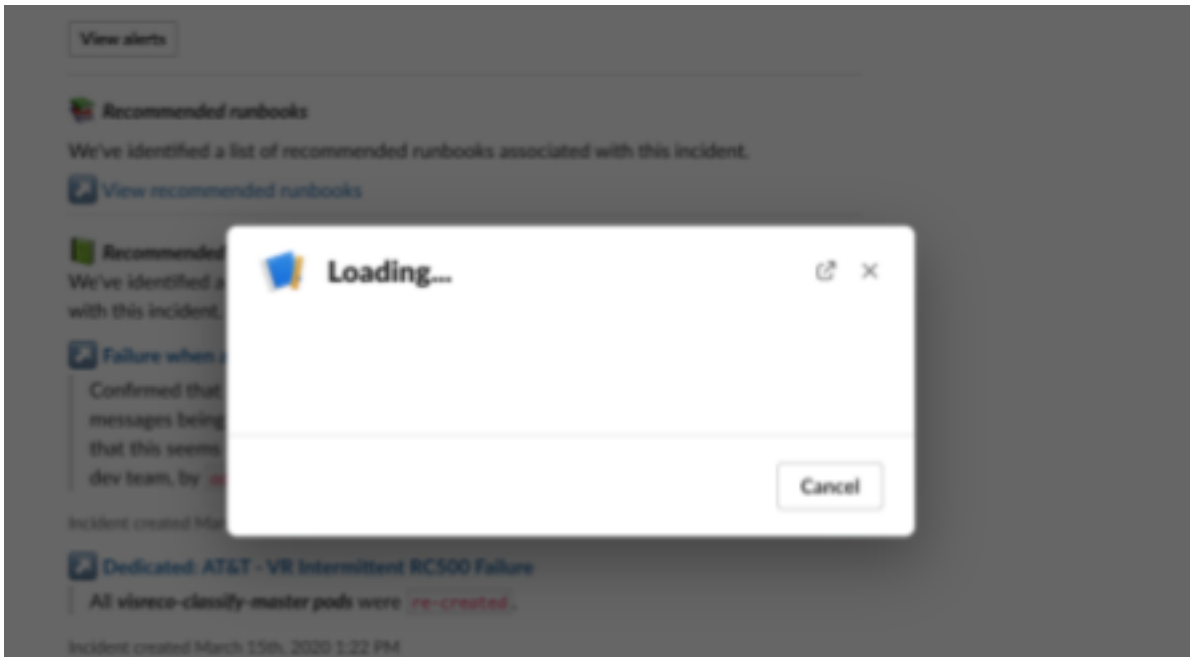


Figure. View alerts

- The **Log preview** dialog in Slack or Microsoft Teams hangs with the `Loading log data..` message.

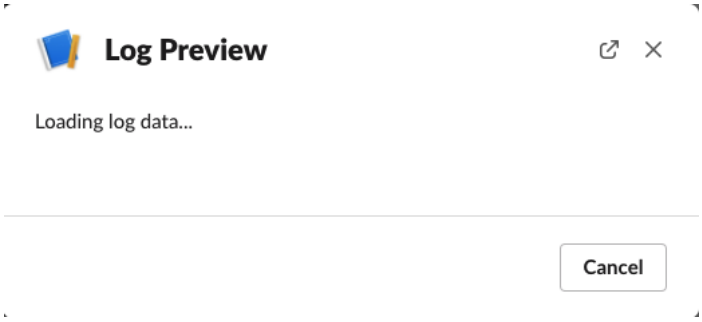


Figure. Log preview

- Checking the log results in something similar to this:

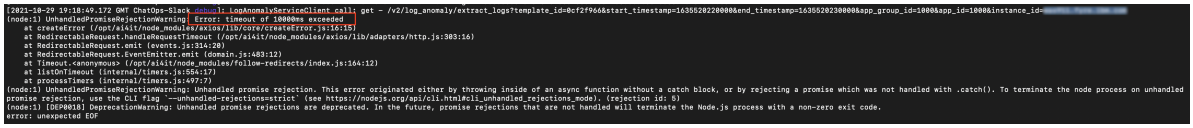


Figure. View log

Solution: close the **View alerts** dialog or the **Log preview** dialog in Slack or Microsoft Teams, and retry the operation.

Alerts and incidents not appearing in Slack

Alerts and incidents are showing up on IBM Cloud Pak for AIOps console but not in Slack ChatOps.

A number of causes can be considered when alerts are not showing up in Slack. Try one or more of the following to help resolve the problem.

1. Make sure that alerts and incidents are being created from events: alerts and incidents can be seen in the *Incidents and alerts* page in the IBM Cloud Pak for AIOps console. If events are flowing but no alerts or incidents are being created, review Troubleshooting alert management.
2. Double check channel IDs: if alerts and incidents are visible in the IBM Cloud Pak for AIOps console, make sure the channel IDs are correctly pasted in the ChatOps data integration form. Ensure the channel ID rather than the channel name is being used and make sure that no extra blank spaces are in the form field. Instructions for retrieving channel IDs can be found in the Configuring Slack applications for integration page.
3. Ensure that the Slack app is a channel member: to post incident messages, the IBM Cloud Pak for AIOps Slack app must be a channel member of the channel IDs being used. To add an application to a channel, tag the bot with the "@" symbol followed by the bot name.
4. Look for any visible errors in the **`chatops-slack-integrator`** logs. If the IDs are correct and the app is a channel member, look for error messages in the pod and contact the L2 support team.

Note: If the Slack app is not properly configured when an incident is initially created, that incident is not posted to Slack. Only when the app is successfully configured do new incidents, and updates to those incidents, get posted.

Slack ChatOps integration is not working within an air-gapped (offline) environment

If you create a Slack integration within an air-gapped (offline) environment, the integration might not work as expected. When you first create the Slack integration, the test of the integration (by clicking **Test connection to Slack.com.**) must be successful. If the test was successful, but the Slack integration is still not working, restart the Slack integrator pod to retrieve the latest integration information.

To restart the pod, complete the following steps:

1. From a command line, use the OpenShift CLI **`oc login`** command to log in to your cluster.
2. Switch to the project (namespace) where IBM Cloud Pak for AIOps is installed:

   `oc project <project>`

   Where **`<project>`** is the project (namespace) where IBM Cloud Pak for AIOps is installed.
3. Retrieve Slack integrator pod name:

   `oc get pods | grep slack`

   Record the name of the Slack integrator pod.
4. Delete the pod to cause the pod to restart:

   `oc delete pod <slack integrator pod>`

   Where **`<slack integrator pod>`** is the name of the pod name.

Microsoft Teams ChatOps integration shows no response in the console

If you have a Microsoft Teams ChatOps integration, you can encounter an issue the Microsoft Teams console shows no response through the integration. This issue can occur when you install and configure a Secure Tunnel for IBM Cloud Pak for AIOps on an on-premises Red Hat OpenShift Container Platform and configure a corresponding Secure Tunnel connecter for Microsoft Teams on a virtual machine (VM) that is accessible from a public network.

If you encounter this issue, complete the following steps:

1. Verify that the Secure Tunnel is working.
    1. Log in to the IBM Cloud Pak for AIOps console.
    2. Go to **Administration** > **Secure Tunnel**.
    3. Verify that the Secure Tunnel connection is configured and that the status is `Ready`.
    4. Click into the **Application mappings** column of your Secure Tunnel connection. Verify the URLs list in the **Application mapping address** column.

    Notes:

    - If the status is not `ready`, then check whether the Secure Tunnel Connector is installed and running. For more information about setting up the connector, see Install the Secure Tunnel Connector.
    - If your Secure Tunnel Connector is running, but you do not see the status as `Ready` for the connection, check the log messages from the Secure Tunnel Connector container.
        - If the log message indicates TLS errors, you might not have the correct Secure Tunnel Connector package. Verify that the installed package is correct. If necessary, download and install the correct package. For more information about installing the connector, see Install the Secure Tunnel Connector.
        - If the log message indicates a connecting error, check whether you can access the virtual machine (VM) where the Secure Tunnel Connector is installed from the Secure Tunnel pod.
2. Verify that the network between the Secure Tunnel and the VM on which Secure Tunnel Connector is installed is accessible.
    1. Log in to the Red Hat OpenShift Container Platform console where IBM Cloud Pak for AIOps is installed.
    2. Go to **Workloads** > **Pods**.
    3. Switch to the project (namespace) for IBM Cloud Pak for AIOps.
    4. Filter the pods to find the secure tunnel pod. For instance filter by `sre-tunnel-network`.
    5. Click listed pod. Then, click **Terminal**.
    6. Access the IP address of your VM by running a command similar to the following `curl` command on port `50443`:

        ```
        curl -k https://your.vm.external.host.com:50443
        ```

        - If you get any SSL error from the command, the error indicates that the network connection is working.
        - If you a message tht indicates `Connection refused` from the command, the message indicates that the network connection is not reachable. Contact your network administrator to fix the network connection.
3. Verify that the configuration of the Microsoft Azure Bot for the ChatOps integration is correct.
    1. Log in to the Microsoft Azure portal, and click **Bot services**. Then, click your Azure bot.
    2. Click **Configuration** under **Settings**.
    3. Check the `messaging endpoint` setting. Replace the `request URL` value with the `application mapping URL` from the Secure Tunnel connection. For more information about the settings for the connection, see Creating a Microsoft Teams connection.
    4. Attempt to send the message `@<bot_name> /waiops_welcome` in Microsoft Teams. If the connection still does not work, other issues might exist. contact your network administrator for assistance with investigating and fixing the network connection.
4. Verify that the network between the Microsoft Azure Bot and the VM on which the Secure Tunnel connector is installed is accessible:
    1. Log in to the terminal of the VM on which you installed the Secure Tunnel connector.
    2. Find the network interface (NIC) that is accessible from public network. You can use a command like `ip address` to find this interface. For example, if your network interface is `eth1`, listen to the port `12443` on that network interface by using the command `tcpdump -i eth1 port 12443`.
    3. Test the access to the `application mapping URL` of your Secure Tunnel connection from your device or from another device outside of your on-premise netowork. For example, run the following command, where `your.vm.external.host.com` is your VM host.

        ```
        curl -k -X POST
        https://your.vm.external.host.com:12443/aiops/aimanager/instances/1683613041902750/teams/api/messages
        ```

    4. Verify that there are outputs from the `tcpdump` command while you run the `curl` command.

        If you do not find any output from the `tcpdump` command, you might have firewalls or ACL restrictions that are blocking inbound traffic to the VM. Contact your network administrator to grant inbound traffic on port `12443` for the VM and retest the command.

During a ChatOps secure tunnel creation an 'installation failed' message displays

When you create a ChatOps integration and it fails, wait for a few minutes to see if the installation retries, and if it does not simply create the integration again.

Show more button does not work in Slack after an upgrade

You might notice an issue where the **Show more** button does not work in Slack after an IBM Cloud Pak for AIOps upgrade.

The reason for this issue is that the TLS certificate CA for ChatOps is updated through the upgrade, and the tunnel `ApplicationMapping` is still using the previous TLS certificate CA.

If you encounter this issue, use the following steps to restart the pod `sre-tunnel-controller-a-`. This restart causes the Secure tunnel controller pod to update the TLS certificate CA:

1. Get the name of the Secure tunnel controller pod:

```
oc get pod | grep sre-tunnel-controller-a-
```

2. Restart the pod **sre-tunnel-controller-a-***:

```
oc delete pod <Secure tunnel controller pod>
```

Where **Secure tunnel controller pod** is the name you extracted in the preceding step.

Troubleshooting topology

Learn how to isolate and resolve problems with the Topology viewer and topology service.

To help you with troubleshooting topology issues, review the topology service logs. For more information, see Viewing the topology logs.

Common issues:

- Chrome extension display issues on Incident Topology page
- Error connecting topology services to Cassandra database
- Observer schedule toggle becomes disabled after saving the job
- File Observer fails due to hidden characters
- Kubernetes Observer job fails to restart after OOM
- Halt a Cassandra node for maintenance
- Topology observer job failure due to prohibited egress
- Topology viewer user interface crashes after the update manager is displayed

Chrome extension display issues on Incident Topology page

When displaying the details for a listed event in the Resolution Hub by using the Chrome browser, the resizing of the page might cause screen errors. This is caused by the MEGA Secure Cloud Storage and Chat Chrome browser extension.

Solution: Remove the MEGA Secure Cloud Storage and Chat extension.

Error connecting topology services to Cassandra database

During startup, the topology service can attempt to connect to the Cassandra database before it is fully started, which causes an error, followed by a warning message similar to the following message:

```
WARN   [2022-02-28 11:01:36,748] [main] c.i.i.t.g.ConnectionManager -  failed to connect to storage back end sleeping before retrying.
```

This issue can occur when the topology service has started before the Cassandra database is fully ready.

Solution: You do not need to complete any workaround. The topology service tries to connect to the Cassandra database repeatedly. When the Cassandra database is up and running, the service can successfully connect.

Observer schedule toggle becomes disabled after saving the job

After editing and saving an observer job that includes scheduled jobs, the **Schedule request** toggle becomes disabled. However, the job remains scheduled and continues to run in the background.

This issue can occur when the **Schedule request** toggle fails to read the existing job schedule and erroneously reverts to its default setting of **Off**.

Solution: The job scheduling remains active and no workaround is required. To confirm the existing observer job schedule:

1. Log in to IBM Cloud Pak for AIOps console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Integrations**.
3. On the *Integrations* page, click **Add integration**.
4. On the *Add integration** page, click **Topology** in the *Category* list that is next to the list of all integrations.
5. Click the **configure, schedule, and manage other observer jobs** link in the description for the topology integration.
6. Select the observer.
7. Confirm that the job schedule remains active.

File Observer fails due to hidden characters

An error can occur when your File Observer input file contains hidden characters that interfere with the processing of the content.

```
The following lines numbers had problems, check the logs for details: 1
```

This issue can occur when a file appears compliant in content and format, but is in a UTF-8 Unicode (with BOM) file format (instead of the regular UTF-8 file format).

Solution: Change the file format. For example, you can create a new file from the source file by using the following command:

```
sed '1s/^\xEF\xBB\xBF//' < topology.txt > new.txt
```

Kubernetes Observer job fails to restart after OOM

Kubernetes Observer jobs with very large payloads can encounter an OOM (out-of-memory) error, after which they may fail to restart. The observer appears offline, but a health check fails to flag any errors.

Solution: Restart the observer if it appears as offline in the UI.

Halt a Cassandra node for maintenance

Sometimes you need to manually maintain the Cassandra server nodes. For example, you might need to halt one node for long enough to allow you to complete the required maintenance work or debug a problem.

The following steps suspend (halt) one Cassandra node while the other nodes still serve queries, which allows you to perform maintenance tasks without shutting down your production environment.

Note: To reduce the impact of pausing a Cassandra node on the performance of your remaining system resources, keep such maintenance periods to a minimum.

Halt the node

1. Edit the bootstrap configMap.

```
oc edit configmaps aiops-topology-cassandra-bootstrap-config
```

For example, if you want to halt node 0, you can set the following columns.

```
hostname_filter: aiops-topology-cassandra-0
running_mode: maintenance
```

2. Save and exit.

3. To restart the target node pod, you delete the pod so that the statefulSet will restart the pod automatically. Then it should be halted.

```
oc delete pod aiops-topology-cassandra-0
```

Debug the Cassandra server

After the node is restarted and halted, you should see that its state stays as **`0/1 Running`**. When the pod is in this mode, the Cassandra server is not automatically started in it. The **`liveness`** probe and **`readiness`** probe will not be activated for many days in this pod, allowing you to perform any maintenance that is required.

There are two methods to start the Cassandra server for debug purposes.

- You can go into the pod container to manually start the server.

  ```
  oc exec -it aiops-topology-cassandra-0 -- bash
  bash-4.4$ /opt/ibm/cassandra/bin/cassandra -fR > /tmp/server.log 2>&1 &
  ```

  The server process will be started in the background and the log messages can be found in file **`/tmp/server.log`**.

- You can reuse the entry point script.

  ```
  oc exec -it aiops-topology-cassandra-0 -- bash
  bash-4.4$ export RUNNING_MODE=normal
  bash-4.4$ /opt/ibm/start-cassandra.sh > /tmp/server.log 2>&1 &
  ```

Example maintenance task: Remove corrupt commit logs that are causing Cassandra to not start.

```
$ oc exec -it aiops-topology-cassandra-0 -- bash
bash-4.4$ rm -rf /opt/ibm/cassandra/data/commitlog/CommitLog-6-1682019553495.log
```

After performing the required maintenance, you can stop the manually started Cassandra server.

```
oc exec -it aiops-topology-cassandra-0 -- bash
bash-4.4$ nodetool stopdaemon
```

Note: You can also perform other actions, like fixing volumes or transforming the data files. Every time you restart the pod, it will be halted.

Return the node to normal

After you complete your manual work, you can edit the configMap and restart the pod to return it back to normal.

```
oc edit configmaps aiops-topology-cassandra-bootstrap-config
```

Set the running mode to normal.

```
hostname_filter: aiops-topology-cassandra-0
running_mode: normal
```

Restart normal operations

Finally, you can delete the pod again so it will revert to the normal node and continue to work in the cluster.

Topology observer job failure

When you attempt to run a topology observer, the job fails because no response is returned for the request to the corresponding external service.

Solution: Run the following commands to create a NetworkPolicy that allows egress from the topology observer pods:

```
export AIOPS_NAMESPACE=<AIOps installation namespace>

cat << EOF | oc apply -f -
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: aiops-observer-egress
  namespace: ${AIOPS_NAMESPACE}
spec:
  egress:
  - {}
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: aiops-topology
  policyTypes:
  - Egress
EOF
```

Topology viewer user interface crashes after the update manager is displayed

The issue is that you cannot use the update manager feature in topology viewer.

The solution is to change your topology viewer user preferences to auto render changes on refresh, which prevents the update manager from appearing.

Viewing the topology logs

You can see all topology and observer logs by using the **oc logs** command.

About this task

All observers are deployed in pods. You can identify logs either by their names or their labels.

To increase or decrease the log levels for a specific observer to help investigate issues, you first use the **oc get deployments** command to display a list of all deployments. Then, use the **oc edit deployment** command to change the logging levels to, in this example, 'debug'. A log level of 'debug' logs all request or response operations to provide detailed diagnostic data.

Viewing logs

1. To list all pods and the labels that are applied to the pods, run the **oc get pod --show-labels** command.

   ```
   oc get pod --show-labels
   ```

   Sample output:

   ```
   deploy-test-ciscoaci-observer-8755fcc94-7cp7h      1/1       Running   0          2d        app=ciscoaci-
   observer,chart=ciscoaci-observer,heritage=Tiller,pod-template-hash=431197750,release=deploy
   deploy-test-file-observer-76c5869d8b-tm7xj         1/1       Running   0          2d        app=file-
   observer,chart=file-observer,heritage=Tiller,pod-template-hash=3271425846,release=deploy-test
   deploy-test-rest-observer-789cd8699d-zb8vx         1/1       Running   0          2d        app=rest-
   observer,chart=rest-observer,heritage=Tiller,pod-template-hash=3457842558,release=deploy-test
   deploy-test-vmvcenter-observer-675bd88f5c-8kgnk    1/1       Running   0          2d        app=vmvcenter-
   observer,chart=vmvcenter-observer,heritage=Tiller,pod-template-hash=2316844917,release=deploy-test
   ```

2. Display the logs by using either the pod names or labels. When you run the **oc** logs command, you must also specify the project (namespace).

   1. View logs for a specific pod:

      ```
      oc logs asm-layout-88bd88bdb-htjfp --namespace=netcool
      ```

   2. View logs by using the label:

      ```
      oc logs -l app=layout --namespace=netcool
      ```

Changing log levels to debug

Run the following command to change the debug level to **DEBUG**:

```
oc exec -it deploy-test-file-observer-76c5869d8b -- /opt/ibm/netcool/asm/bin/log-level.pyc --level DEBUG
```

Sample system output

```
Configured logging level for com.ibm to DEBUG
Response code: 200
```

Setting observer log levels

You can set the observer logging level to 'debug' (or a number of other levels) to troubleshoot in more detail.

About this task

To increase or decrease the log levels for a specific observer, you first obtain the observer's pod name, internal IP address, admin port and credentials. You then specify the required log level ('debug' in this example) before running the observer's load job and tailing the observer log.

Note: A log level of 'debug' logs all request or response operations to provide detailed diagnostic data.

The following task describes how to set the observer log levels to 'debug'. An OpenStack Observer example is provided.

Setting observer log levels to 'debug'

1. Obtain the observer pod name.

```
oc get pod | grep [observer name]
```

Example output:

```
[release name]-topology-[observer name]-observer-dd845b66c-vqs82                1/1     Running     0              106m
```

Note: The observer pod name is [release name]-topology-[observer name]-observer-dd845b66c-vqs82. The value of dd845b66c-vqs82 will vary depending on the pod name.

2. Use the observer pod name to obtain the observer's internal IP address.

```
oc describe pod [observer pod name] | grep IP
```

Example output:

```
IP: 10.254.21.131
IPs:
IP: 10.254.21.131
```

Note: Note down the IP address, which is the internal IP address.

3. Use the observer pod name to obtain the observer admin port.

```
oc describe pod [release-name]-topology-[observer name]-observer-dd845b66c-vqs82 | grep Ports
```

Example output:

```
Ports: 9082/TCP, 9083/TCP
Host Ports: 0/TCP, 0/TCP
```

Note: In this example 9083 is the admin port of the observer. (The port with the odd number as the last digit is always the admin port.)

4. Complete the following sub-steps.

   a. Obtain the base64 encoded string of `[username]:[password]`

   ```
   oc get secret | grep asm-credentials
   ```

   Example output:

   ```
   [release-name]-topology-asm-credentials                          Opaque                                2
   5d9h
   ```

   b. From the output above, run the following commands:

   ```
   oc get secret [release-name]-topology-asm-credentials --template={{.data.username}} | base64 -d
   ```

   ```
   oc get secret [release-name]-topology-asm-credentials --template={{.data.password}} | base64 -d
   ```

   Example commands and output:

   ```
   oc get secret aiops-topology-asm-credentials --template={{.data.username}} | base64 -d
   asm-netcool-user
   ```

   ```
   oc get secret aiops-topology-asm-credentials --template={{.data.password}} | base64 -d
   asmauto
   ```

   c. Encode to base64 format:

   ```
   echo "asm-netcool-user:asmauto" | base64
   ```

   Example output:

   ```
   YXNtLW5ldGNvb2wtdXNlcjphc21hdXRvCg==
   ```

5. Specify the DEBUG log level. Run the following command:

```
oc exec -it [observer pod name] -- curl -k -X POST --header 'Authorization: Basic [base64 encoded string of
username:password]' -d "logger=com.ibm.itsm.topology.observer.[observer name]&level=DEBUG" https://[internal IP address]:
[observer admin port]/tasks/log-level
```

- Replacing `[observer pod name]` and `[internal IP address]` in your environment. Note that `[observer name]` is the observer that is to be configured and [observer admin port] is the admin port of the observer.
- The `[base64 encoded string of username:password]` is the base64 encoded string of `[username]:[password]`.

OpenStack Observer example:

```
oc exec -it aiops-topology-openstack-observer-dd845b66c-vqs82 -- curl -k -X POST --header 'Authorization: Basic YXNtLW5ldGNvb2wtdXNlcjphc21hdXRvCg==' -d "logger=com.ibm.itsm.topology.observer.openstack&level=DEBUG" https://10.254.21.131:9083/tasks/log-level
```

The following prompt should be visible in the console:

```
Configured logging level for com.ibm.itsm.topology.observer.[observer name] to DEBUG
```

- In this example the package name `com.ibm.itsm.topology.observer.[observer name]` tells us that the observer package has been set to the DEBUG level.

6. Run the observer load job and tail the observer log.

```
oc logs -f [release name]-topology[-observer name]-observer-dd845b66c-vqs82
```

Note: If set successfully to DEBUG, the DEBUG messages are now recorded in the observer log.

Example log entry (for the OpenStack Observer):

```
DEBUG [2024-07-30 06:46:13,200] [pool-12-thread-1] c.i.i.t.o.o.OpenStackRESTSession - [iZx9pZr] - getOpenStackClientAccessOrToken: Http Logging filter is 'false'
```

Defining classifier log settings

When viewing incidents on the Incident page, the topology can be viewed on the Overview tab, but is not available on the Topology tab.

About this task

Top alerts are displayed on the Overview tab, while more detailed information is used to render the information on the Topology tab.

To display the topology for an incident on the Topology tab, it uses the classifier for event classification for probable cause calculation. If these are not being recorded due to the level of log settings, the information is unavailable and the topology is not displayed. You must set additional properties to control the classifier logging.

You add (or edit) the following properties for the classifier deployment in both AI Manager and Event Manager

Note: Only the LOGGING_LEVEL property exists with a default setting of **`INFO`**. You must add the other settings to the classifier deployment manually.

- `0` = no logging set
- `10` = `debug`
- `20` = `information`
- `30` = `warning`
- `40` = `error`
- `50` = `critical`

Changing classifier log settings

1. Change the settings for the following log levels.

   a. For kafka:

   ```
   `KAFKA_LOGGING_LEVEL` = ???
   ```

   b. For file and console for other mime utils file:

   ```
   `LOGGING_LEVEL` = ???
   ```

2. Change the settings to define rotation and backup settings for the log files.

   a. For file rotation:

   ```
   `DEFAULT_LOGGING_FILE_SIZE` = ???
   ```

   b. For the number of history backups:

   ```
   `DEFAULT_LOGGING_FILE_BK_COUNT` = ???
   ```

   c. For the output format to console:

   ```
   `%(asctime)s.%(msecs)03d %(levelname)s %(module)s - %(funcName)s: %(message)s`
   ```

IM issues

Troubleshoot Identity Management (IM) issues.

- Login

- LDAP
- Pod issues
- OIDC
- API calls
- MongoDB to EDB PostgreSQL database migration

- EDB PostgreSQL cluster is down and the pod is in `Completed` status
- Enable debugging for user authentication issues

Login

Review frequently encountered login issues.

- Internal server error while login Platform UI by using SAML with LDAP dependency
- Cannot log in to the console by using the admin username
- Docker login results in unencrypted password warning
- Login too slow or times out, or invalid username or password error
- Cannot log in to the console by using OpenShift authentication
- Cannot log in to the console after reinstallation of foundational services
- Default admin cannot log in due to LDAP server issues
- Username shows as Undefined Undefined after login to CloudPak dashboard through SAML login
- Cannot login Platform UI with CloudPak administrator username
- Gateway timeout error while you log in to the console
- CWWKS1406E: Unable to load the login page and an invalid client error is displayed

Internal server error while login Platform UI by using SAML with LDAP dependency

Login error appears when you try to access the Platform UI by using SAML that is registered with LDAP.

Symptoms

When you use LDAP dependency with SAML authentication and try to access the Platform UI by using SAML, the login error appears and displays a message, **500 internal server error**.

If you access the Common UI console, the following errors are displayed in the **common-ui** container logs:

```
[2022-03-04T11:58:45.375] [ERROR] [webui-nav] [platform-auth-service] SAML enabled, but unable to get idsource
registrations to determine if ldap is enabled for the saml config - default to true Error:
    at Request.<anonymous> (/opt/ibm/platform-header/lib/server/request.js:115:34)
    at Request._callback (/opt/ibm/platform-header/node_modules/lodash/lodash.js:10118:25)
    at Request.requestRetryReply [as reply] (/opt/ibm/platform-header/node_modules/requestretry/index.js:105:19)
    at Request.<anonymous> (/opt/ibm/platform-header/node_modules/requestretry/index.js:138:10)
    at Request.self.callback (/opt/ibm/platform-header/node_modules/request/request.js:185:22)
    at Request.emit (events.js:400:28)
    at Request.emit (domain.js:475:12)
    at Request.<anonymous> (/opt/ibm/platform-header/node_modules/request/request.js:1154:10)
    at Request.emit (events.js:400:28)
    at Request.emit (domain.js:475:12) {
  statusCode: 403,
  details: 'Unexpected response code 403 from request:\n' +
    'GET https://platform-identity-provider:4300/v2/auth/idsource/registration?protocol=saml HTTP/1.1\n' +
    'Accept: application/json\n' +
    'Authorization: Bearer ***\n' +
    'Accept: application/json\n' +
    '\n' +
    'HTTP/1.1 403\n' +
    'X-Dns-Prefetch-Control: off\n' +
    'X-Frame-Options: SAMEORIGIN\n' +
    'Strict-Transport-Security: max-age=15552000; includeSubDomains\n' +
    'X-Download-Options: noopen\n' +
    'X-Content-Type-Options: nosniff\n' +
    'X-Xss-Protection: 1; mode=block\n' +
    'Content-Type: application/json; charset=utf-8\n' +
    'Content-Length: 57\n' +
    'Etag: W/"39-Iup4jU4k5A0rUB2wIQCJV1dX4gw"\n' +
    'Date: Fri, 04 Mar 2022 11:58:45 GMT\n' +
    'Connection: close\n' +
    '\n' +
    '{"error":"Insufficient user permission for role :Viewer"}\n'
}
```

Causes

Service-id permission returns **Viewer** role access.

Resolving the problem

Upgrade your current foundational services version to 3.16.0 version or later.

Cannot authenticate Kubeadmin user as CloudPakAdministrator by default

The **`Kubeadmin`** user does not get **CloudPakAdministrator** role by default after login.

Symptoms

After you log in as a **`Kubeadmin`** user in the Administration panel, it assigns you the **Viewer** role.

Causes

Bootstrap process is not successful during the IM service installation.

Resolving the problem

Restart the **`platform-identity-management`** pod by using the following command:

```
oc -n <your-foundational-services-namespace> delete pod -l app=platform-identity-management
```

Cannot log in to the console by using the admin username

After you set up an LDAP connection, you cannot log in to IBM Cloud Pak® for AIOps cluster console by using the default cluster administrator **`admin`** username.

Causes

There might be a user by the name **`admin`** in the LDAP directory that you configured in your cluster.

Resolving the problem

To resolve the problem, see Default admin cannot log in due to LDAP server issues.

If you still want to use the admin user from the LDAP, you can change the user name in IBM Cloud Pak for AIOps cluster. To change the username in IBM Cloud Pak for AIOps cluster, see Changing the cluster administrator username.

Docker login results in unencrypted password warning

After you log in to your private image registry with the Docker login command, a warning is displayed that indicates that your password is stored unencrypted.

Symptoms

A warning message is displayed when you log in to your private image registry with the following command:

```
docker login <cluster_CA_domain>:8500
```

In this command, **`<cluster_CA_domain>`** is the certificate authority (CA) domain that was set during installation.

The command output includes the following warning:

```
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

Causes

By default, Docker stores the login password unencrypted within the **`/root/.docker/config.json`** Docker configuration file. This is the default Docker behavior.

Resolving the problem

You can store your user credentials in an external credentials store instead of within the Docker configuration file. Storing your credentials in a credentials store is more secure than storing the credentials in the Docker configuration file. For more information, see the Docker community credentials store documentation.

Login too slow or times out, or invalid username or password error

After you set up an LDAP connection, you cannot log in to IBM Cloud Pak® for AIOps cluster console.

Symptoms

There are two symptoms for the same cause.

- When you try to log in, the login process might take a long time, or might time out.
- You might see the following error:

  ```
  Invalid user name or password
  ```

Cause

The login failure is due to an LDAP error when Liberty looks up groups for the user. By default, Liberty searches which groups the user is a member of. It then searches which groups these groups are a member of. The message log shows the following error:

```
An FFDC Incident has been created: "com.ibm.wsspi.security.wim.exception.WIMSystemException: CWIML4520E: The LDAP
operation could not be completed. The LDAP naming exception
javax.naming.NameNotFoundException: [LDAP: error code 32 - 0000208D: NameErr: DSID-03152973, problem 2001 (NO_OBJECT),
data 0, best match of:
"
```

Resolving the problem

To resolve the issue, disable the **recursiveSearch** parameter in the LDAP **server.xml** file.

1. Log in to your boot node with the **oc login** command.
2. Edit the **platform-auth-idp** configmap.

   ```
   oc edit cm platform-auth-idp -n <your-foundational-services-namespace>
   ```

3. Change the **LDAP_RECURSIVE_SEARCH: "true"** parameter value to **LDAP_RECURSIVE_SEARCH: "false"**.
4. Save the changes.
5. Restart the **platform-auth-service** pods by deleting the pods.
   1. Get the **platform-auth-service** pod names.

      ```
      oc get pods -n <your-foundational-services-namespace> | grep platform-auth-service
      ```

   2. Delete the **platform-auth-service** pod.

      ```
      oc delete pods <pod-name> -n <your-foundational-services-namespace>
      ```

Cannot log in to the console by using OpenShift authentication

If you are using OpenShift authentication, you cannot log in to the console after you install IBM Cloud Pak foundational services on an OpenShift cluster that is in any cloud environment.

Symptom

You see a **400: Bad request - The request or response is invalid** error.

Cause

The rolebinding collection failed due to identity-provider:4300 connection time-out in the **oidclient-watcher** pod.

Resolving the problem

To resolve the issue, restart the **oidclient-watcher** pod.

1. Log in to your infrastructure node with the **oc login** command.
2. Get the **oidcclient-watcher** pod name.

   ```
   oc get pods -n <your-foundational-services-namespace> | grep oidcclient
   ```

3. Delete the **oidcclient-watcher** pod

   ```
   oc delete pod -n <your-foundational-services-namespace> <oidcclient-watcher-pod-name>
   ```

Wait for the pod to restart.

You can log in after the **oidclient-watcher** pod restarts.

Cannot log in to the console after reinstallation of foundational services

After you uninstall IBM Cloud Pak foundational services and reinstall them on the same cluster, you cannot log in to the console with the default admin credentials.

Symptoms

You might see the following error:

```
CWOAU0038E: The client could not be verified. Either the client ID: t19ph7yimexir00h517actss98zbrg0m or client secret is incorrect.
```

Cause

After you uninstall the foundational services, if all the resources are not properly cleaned, reinstallation on the same cluster fails.

The cause of the failure might be either or both of the following reasons:

- There is a mismatch of the **client_id** and **client_secret** values in the cluster, which causes the console login to fail.
- The IM jobs are not completed or are completed with errors.

Resolving the problem

To resolve the issue, correct the **client_id** and **client_secret** values and restart all IM jobs.

1. Log in to your boot node with the **oc login** command.
2. Check whether the **ibm-im-operator** pod is running.

   ```
   oc -n <your-foundational-services-namespace> get pods | grep ibm-iam-operator
   ```

3. Check whether all the auth pods are running without any errors. If you see errors, check the logs and resolve the issue.

   ```
   oc -n <your-foundational-services-namespace> get pods | grep auth
   ```

4. Check the **client_id** and **client_secret** values in the **platform-oidc-credentials** secret and in the **registration-json** configmap.

   - Check the values in the **platform-oidc-credentials** secret:

     ```
     oc -n <your-foundational-services-namespace> get secret platform-oidc-credentials -o "jsonpath=
     {.data.WLP_CLIENT_ID}" | base64 --decode
     oc -n <your-foundational-services-namespace> get secret platform-oidc-credentials -o "jsonpath=
     {.data.WLP_CLIENT_SECRET}" | base64 --decode
     ```

   - Check the values in the **registration-json** configmap:

     ```
     oc -n <your-foundational-services-namespace> get cm registration-json -o "jsonpath={.data['platform-oidc-
     registration\.json']}" > platform-oidc-registration.json
     cat platform-oidc-registration.json
     ```

5. If the values in the secret and configmap do not match, update the values in the **registration-json** configmap with the values from the **platform-oidc-credentials** secret. To edit the configmap, run the following command:

   ```
   oc -n <your-foundational-services-namespace> edit cm registration-json
   ```

   Replace the **client_id** and **client_secret** values, and save and exit the configmap.

6. Restart the following IM job.

   ```
   oc -n <your-foundational-services-namespace> delete job oidc-client-registration
   ```

7. Wait for some time and check the job status.

   ```
   oc -n <your-foundational-services-namespace> get jobs
   ```

   The status should show as **Completed**.

8. Restart the **common-web-ui** pods so that the **client_id** and **client_secret** get the correct values from the **platform-oidc-credentials** secret.

   1. Get the **common-web-ui** pod name.

      ```
      oc -n <your-foundational-services-namespace> get pods | grep common-web-ui
      ```

   2. Delete the pod.

      ```
      oc -n <your-foundational-services-namespace> delete <common-web-ui-pod-name>
      ```

9. Wait for sometime until the **common-web-ui** and **management-ingress** pods show the **Running** status. You can check the status by running the following command:

```
oc -n <your-foundational-services-namespace> get pods | grep -e common-web-ui
```

After the pods are running, you can log in to the console with the default cpadmin credentials.

Default admin cannot log in due to LDAP server issues

The default admin user can no longer log in to your product cluster console.

You might see the following error in the **platform-auth-service** container of the **platform-auth-service** pod.

```
[ERROR   ] CWIML4537E: The login operation could not be completed. The specified principal name admin is not found in the back-end repository.

com.ibm.wsspi.security.wim.exception.PasswordCheckFailedException: CWIML4537E: The login operation could not be completed. The specified principal name admin is not found in the back-end repository.
        at com.ibm.ws.security.wim.ProfileManager.loginImpl(ProfileManager.java:1820)
        at [internal classes]
```

```
## Causes

This issue can occur because of the following reasons:

- The LDAP parameters that you specified when you configured the LDAP connection might not be correct.
- LDAP server is inaccessible for some reason.
- LDAP configuration file is corrupted.
- A user who is named `admin` might exist in the LDAP directory. For this case, try this resolution, [Cannot log in to the management console by using the admin username](../troubleshoot/ldap_admin.html).

## Resolving the problem {: #resolve}

As a temporary measure, you can remove the LDAP configuration file that is in the `platform-auth-service` container of the `platform-auth-service` pod so that the default admin user can log in to the cluster.

The LDAP configuration is stored in the `/config/configDropins/defaults/ldap-<LDAP_connection_name>.xml` and `federated.xml` files in the `platform-auth-service` container. To remove the incorrect LDAP configuration, you must delete the files or move them to another location. You can then log in to the console and set up the LDAP connection again with the correct parameters.

Removing the LDAP configuration file in the `platform-auth-service` container has no impact on the LDAP connection configuration. The users, groups, roles, and other resources are retained.

Complete the following steps to remove the LDAP configuration files.

1. SSH to your boot node and log in to cluster with `oc login` command.

2. Get the `platform-auth-service` pod names.

   ```cmd {: codeblock}
   oc -n <your-foundational-services-namespace> get pods | grep platform-auth-service
```

3. Access the shell inside the **platform-auth-service** container.

   ```
   oc -n <your-foundational-services-namespace> exec -it platform-auth--service-xxxxxx-yyyy -c platform-auth-service -- /bin/bash
   ```

4. Make a note of the LDAP settings file name that is located in the **/config/configDropins/defaults/** directory. For example, the file name is **ldap-openLDAP.xml** if your LDAP connection name is **openLDAP**.

5. Move the LDAP settings file **ldap-<LDAP_connection_name>.xml** and the **federated.xml** files out of the **/config/configDropins/defaults/** directory. For example, you can move the files to the **/config/configDropins/** directory.

   ```
   mv /config/configDropins/defaults/ldap-<LDAP_connection_name>.xml /config/configDropins/
   mv /config/configDropins/defaults/federated.xml /config/configDropins/
   ```

   Note: Instead of moving the files, you can delete the LDAP settings file, and **federated.xml** files from the **/config/configDropins/defaults/** directory. However, when you move the files, ensure that you have a backup of the configuration files, which can be used for troubleshooting.
   You can also correct the settings in the files and reuse it to set up the LDAP connection. If you are reusing the files, you must move the corrected files back into the **/config/configDropins/defaults/** directory.

   Exit from the container.

6. Perform steps 3 - 5 for all **platform-auth-service** pods.

7. Wait for a minute and then try log in as the default admin user.

8. Set up your LDAP connection with proper configuration parameters. For more information about setting up an LDAP connection, see Configuring LDAP connection.

   Note: You must identify the root cause of the issue and take the necessary actions to resolve it. After you resolve the issue, you can restart the **platform-auth-service** pod and then log in to the console as the admin user.

   You can also enable debugging for Liberty and collect logs. For more information, see Enable debugging for user authentication issues.
   ```

Username shows `Undefined Undefined` after login to CloudPak dashboard using SAML login option

Symptoms

When you use SAML login as option to login to the Cloudpak console, you might notice that the username is showing **`Undefined Undefined`** in the CloudPak dashboard page. You can observe this issue if you are using the foundational services version 3.20 and 3.21.

Causes

Migration from the IdP V2 to V3.

Resolving the problem

As per the SAML V3 schema, **`given_name`** and **`family_name`** are mapped to the **`givenname`** and **`familyname`** of IdP. You can update the V3 registration call to resolve this issue.

Steps

Get the V3 registration data

Note:

- The SAML UID that is supported is **`defaultSP`**. For more information, see Getting SAML registration by UID.
- Replace **`base64-encoded-idp-xml`** with actual base64 encoded **`idp-metadata`** that you have received in the GET call.
- Get the **`cp-console-url`** by using the command, **`oc get routes`**.

To get the existing SAML registration:

API version
1.0.0
API URI components

Scheme

HTTPS

Host IP

Cluster Master Host

Port number

Cluster Master API Port

Path

idprovider/v3/auth/idsource/defaultSP

Command

GET

Command output format

application/json

```
curl -k -X GET 'https://cp-console.apps.mycluster.mydomain.com/idprovider/v3/auth/idsource/defaultSP' --header
"Authorization: Bearer $ACCESS_TOKEN"
```

Update the V3 registration data by adding first_name and last_name under token_attribute_mappings

1. Copy the response that you get by using the curl command in Get the V3 registration data.
2. Remove the **`uid`** entry from the response, **"uid":"defaultSP"**.
3. Add the following attributes, **`first_name`** and **`last_name`** under the **`token_attribute_mappings`**.

```
"token_attribute_mappings": {
    "sub": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name",
    "given_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname",
    "family_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname",
    "groups": "http://schemas.microsoft.com/ws/2008/06/identity/claims/groups",
    "email": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress",
    "first_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname",
    "last_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"
}
```

4. Update the IdP V3 registration by adding the **`token_attribute_mappings`** payload as described in Step 3.

Note:

- Replace **base64-encoded-idp-xml** with actual base64 encoded **idp-metadata** that you have received in the GET call.
- Get the **cp-console-url** by using the command, **oc get routes**.

For example:

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster Master Host

Port number

Cluster Master API Port

Path

idprovider/v3/auth/idsource

Command

PUT

Command output format

application/json

```
curl -k -X PUT 'https://cp-console.apps.mycluster.mydomain.com/idprovider/v3/auth/idsource/defaultSP' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $ACCESS_TOKEN"
--data-raw
'{
  "name": "w3id-sample-saml",
  "description": " this is plain saml testing",
  "protocol": "saml",
  "type": "default",
  "idp_config": {
    "token_attribute_mappings": {
    "sub": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name",
    "given_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname",
    "family_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname",
    "groups": "http://schemas.microsoft.com/ws/2008/06/identity/claims/groups",
    "email": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress",
    "first_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname",
    "last_name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"
    },
    "idp_metadata": "<base64-encoded-idp-xml>"
  },
  "jit": true
}'
```

Gateway timeout or blank page error while you log in to the console

Cannot log in to your IBM Cloud Pak foundational services or IBM Cloud Pak® console.

Symptoms

You might see a **504 Gateway timeout** error or blank page when you log in to the console.

Cause

IBM WebSphere Liberty has a limitation in their federated user registry mechanism where a problem in one registry affects all other registries in the cluster.

The foundational services Identity Management (IM) service uses Liberty federated registry. Hence, if you have multiple LDAP (Lightweight Directory Access Protocol) registries in your cluster, if any registry server is down, it impacts all other registries. You can't log in to your cluster. The issue might happen even if you have only one LDAP connection in your cluster.

Resolving the problem

If you find that all the IM pods are up and running, then follow these steps as a workaround to the issue:

1. Verify whether your LDAP registries are healthy and that you are able to reach the servers.

2. If there is a problem in any LDAP registry, disconnect the LDAP registry from your cluster by using either of the following ways:
    - If other identity providers are configured in your cluster, a user with admin access can log in to the console and delete the connection from the navigation menu > **Identity and access** > **Identity providers** page.
    - If you have the foundational services default `cpadmin` token, you can use the Deleting an LDAP connection by using IdP V3 API to remove the connection.
3. Restart the `platform-auth-service` pod.

```
oc get pod -n <your-foundational-services-namespace> | grep platform-auth
oc delete pod $platform-auth-pod
```

If the problem still persists, contact IBM Support.

LDAP

Review frequently encountered LDAP (Lightweight Directory Access Protocol) configuration issues.

- Enable debugging for user authentication issues
- Troubleshooting LDAP configuration
- Configuring LDAP over SSL
- Troubleshooting users and user groups search issues
- Timeout error while setting up an LDAP connection
- LDAP nested search for Microsoft Active Directory does not work correctly
- SAML with LDAP dependency using V2 API does not work correctly

Enable debugging for user authentication issues

Enable debugging for `platform-auth-service` Liberty to get trace logs for debugging user authentication issues.

Enable debug logs by using oc commands

Complete these steps to enable debug logging of platform-auth-service container pods by using the command-line interface:

1. Update the `platform-auth-idp` configmap. Then, save it.

```
oc edit cm platform-auth-idp -n <your-foundational-services-namespace>

LIBERTY_DEBUG_ENABLED: "false"   << true
LOG_LEVEL_AUTHSVC: info    << debug
LOG_LEVEL_IDMGMT: info     << debug
LOG_LEVEL_IDPROVIDER: info  << debug
```

   See the following notes:

    - `<your-foundational-service-namespace>` is the namespace where the platform-auth-service pods are deployed.
    - `LIBERTY_DEBUG_ENABLED` when set to `"true"`, enables the liberty container trace string to traceSpecification="com.ibm.ws.webcontainer*=all:com.ibm.wsspi.webcontainer*=all:HTTPChannel=all:GenericBNF=all:HTTPDispatcher=all".
    - `LOG_LEVEL` for the pods can be changed from `"info"` to `"debug"`
        - `LOG_LEVEL_AUTHSVC` is for the `platform-auth-service` pod
        - `LOG_LEVEL_IDMGMT` is for the `platform-identity-management` pod
        - `LOG_LEVEL_IDPROVIDER` is for the `platform-identity-provider` pod
2. Restart the `platform-auth-service`, `platform-identity-management`, and `platform-identity-provider` pod one at a time by deleting the pod. Wait for the pod to restart before you delete the next pod.
    1. Delete the pod.

```
oc delete pod -l component=<pod-name> -n <your-foundational-services-namespace>
```
```
   2. Check the pod status.
```
```cmd {: codeblock}
oc get pods -A | grep platform
```

3. Wait for all containers and pods to start.

```
watch "oc get pods -n <your-foundational-services-namespace> | grep platform"
```

Enable debug logs by using the Red Hat® OpenShift® Container Platform console

Complete these steps to enable debug logging of identity management pods by using the Red Hat® OpenShift® Container Platform console:

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Configuration** > **ConfigMaps.**
3. Search for `platform-auth-idp`, and select the configmap.
4. Click the YAML edit tab to open the configmap for editing.

5. Search for `LIBERTY_DEBUG` and update `LIBERTY_DEBUG_ENABLED: "false"` to `LIBERTY_DEBUG_ENABLED: "true"`.
6. Change the log levels of LOG_LEVEL_IDPROVIDER, LOG_LEVEL_AUTHSVC, LOG_LEVEL_IDMGMT`from`"info"`to`"debug"`.

```
LIBERTY_DEBUG_ENABLED: "false"     << true
LOG_LEVEL_AUTHSVC: info            << debug
LOG_LEVEL_IDMGMT: info             << debug
LOG_LEVEL_IDPROVIDER: info         << debug
```

7. Click **Save**.
8. From the navigation menu, click **Workloads** > **Pods**.
9. Delete the `platform-auth-service`, `platform-identity-management`, and `platform-identity-provider` pods to restart them. Delete the pods by clicking **...** > **Actions** > **Delete pod**.
10. Wait for some time. Then, check the status of the pods on the **Pods** pane. The status of all the pods must show as `4/4` under the **Ready** field name.

Collect identity management pod and liberty container logs, and foundational services configuration information

After you enable debugging, re-create the issue and collect debug logs. Then, share all the `platform-auth-service` pod logs with the support team.

The following `MG-IAM-CSv4.sh` script helps to collect the container and liberty logs, and foundational services configuration information.

Note: Before you run the script, define the `NAMESPACE` environment variable with the namespace where you deployed foundational services.

```
#!/bin/bash
################################################################################################
#Replace the NAMESPACE where platform-auth-service pods are running
#Collects description and container logs for all pods under the namespace
#Collect the liberty FFDC, logs, and default LDAP configuration from platform-auth-service container
#For the latest script Source  https://www.ibm.com/docs/en/cloud-paks/foundational-services/4.2?topic=ldap-enable-
debugging-user-authentication-issues
################################################################################################
# export LOGLIMIT="--tail=10000" # uncomment if you experience problems collecting large files
export NAMESPACE=<common-services-instance-Namesapce>
export AUTHMGDIR=AuthidpLogs-$(date '+%y%b%dT%H-%M-%S')
mkdir -p $AUTHMGDIR/$NAMESPACE

for pod in $(oc -n $NAMESPACE get pods -l component=platform-auth-service --no-headers -o custom-
columns=name:.metadata.name); do
  echo "===== $pod ====="
  LIBDIR="$AUTHMGDIR/$NAMESPACE/$pod-liberty"
  mkdir -p $LIBDIR
  echo "$pod collect liberty logs "
  oc -n $NAMESPACE cp $pod:/logs -c platform-auth-service $LIBDIR/logs
  echo "$pod collect liberty configuration  ====="
  oc -n $NAMESPACE cp $pod:/opt/ibm/wlp/usr/servers/defaultServer/ -c platform-auth-service $LIBDIR/defaultserver
done

echo gathering info from namespace $NAMESPACE
oc get all,secrets,cm,events -n $NAMESPACE -o wide &> $AUTHMGDIR/$NAMESPACE/all-list.txt
oc get pods -n $NAMESPACE | awk 'NR>1{print "oc -n $NAMESPACE describe pod "$1" > $AUTHMGDIR/$NAMESPACE/"$1"-describe.txt
&& echo described "$1"' | bash
oc get pods -n $NAMESPACE -o go-template='{{range $i := .items}}{{range $c := $i.spec.containers}}{{println
$i.metadata.name $c.name}}{{end}}{{end}}' > $AUTHMGDIR/$NAMESPACE/container-list.txt
awk '{print "oc -n $NAMESPACE logs "$1" -c "$2" $LOGLIMIT -p > $AUTHMGDIR/$NAMESPACE/"$1"_"$2"_previous.log && echo
gathered previous logs of "$1"_"$2"' $AUTHMGDIR/$NAMESPACE/container-list.txt | bash
awk '{print "oc -n $NAMESPACE logs "$1" -c "$2" $LOGLIMIT > $AUTHMGDIR/$NAMESPACE/"$1"_"$2".log && echo gathered logs of
"$1"_"$2"' $AUTHMGDIR/$NAMESPACE/container-list.txt | bash
echo;echo
echo Replace case number TS123456
tar czf CaseTS123456-$AUTHMGDIR.tgz $AUTHMGDIR/
```

Send the logs to the support team

Complete these tasks after you collect the logs:

- Compress the logs and send them to support for analysis.

  ```
  tar czf $AUTHMGDIR.tgz $AUTHMGDIR
  ```

Note: When debug logging is enabled, excess logs are generated, which might affect the container resources. After you collect the debug logs, revert the log level values by editing the settings in the `platform-auth-idp` configmap. For the previous log level values, see parameter values.

Troubleshooting LDAP configuration

Use the `ldapsearch` command line tool to troubleshoot your LDAP (Lightweight Directory Access Protocol) configuration.

Install ldapsearch

Install the **ldapsearch** program.

On Ubuntu, run the following command:

```
sudo apt-get install ldap-utils
```

On Red Hat Enterprise Linux (RHEL), run the following command:

```
sudo yum install openldap-clients
```

Test LDAP connection

To test your LDAP connection, run the following command:

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" -w "<LDAP_BINDPASSWORD>" -s sub
```

Following are the parameter descriptions:

- **<LDAP_URL>** is the URL of the LDAP server. For example, **ldap://<LDAP server domain name or IP address>:<port>** or **ldaps://<LDAP server domain name>:<port>**. The default port number is 389 for LDAP protocol and 636 for LDAP over Secure Sockets Layer (LDAPS) protocol.
- **<LDAP_BASEDN>** is the LDAP distinguished name (DN) of the search base. For example, **dc=abc,dc=com**.
- **<LDAP_BINDDN>** is the LDAP user who is allowed to search the base DN. For example, **cn=admin,dc=abc,dc=com**.
- **<LDAP_BINDPASSWORD>** is the password of the user who is mentioned in the bind DN.

Example commands

```
ldapsearch -x -H "ldap://<hostname or IP address>:389" -b "o=abc.com" -s sub
```

```
ldapsearch -x -H "ldap://<hostname or IP address>:389" -b "dc=abc,dc=com" -D "cn=admin,dc=abc,dc=com" -w "password" -s sub
```

Validate LDAP filters

Create a search string based on the LDAP filters to retrieve data from your LDAP server. If the search results show one or more LDAP entries, then the LDAP filter configuration is correct. If the search results do not show any entry, then the LDAP filter is not correct or is not compatible with your LDAP server type.

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" -w "<LDAP_BINDPASSWORD>" -s sub "<Search string>"
```

Following are the parameter descriptions:

- **<LDAP_URL>** is the URL of the LDAP server. For example, **ldap://<LDAP server domain name or IP address>:<port>** or **ldaps://<LDAP server domain name>:<port>**. The default port number is 389 for LDAP protocol and 636 for LDAPS protocol.
- **<LDAP_BASEDN>** is the LDAP DN of the search base. For example, **dc=abc,dc=com**.
- **<LDAP_BINDDN>** is the LDAP user who is allowed to search the base DN. For example, **cn=admin,dc=abc,dc=com**.
- **<LDAP_BINDPASSWORD>** is the password of the user who is mentioned in the bind DN.
- **<search string>** is the search string that is used to search your LDAP server.

IBM Tivoli Directory Server LDAP filters

Table 1. IBM Tivoli Directory Server LDAP filters

Attribute name	Default value
Group filter	(&(cn=%v)(objectclass=groupOfUniqueNames))
Group ID map	*:cn
Group Member ID map	groupOfUniqueNames:uniqueMember
User filter	(&(emailAddress=%v)(objectclass=person))
User ID map	*:emailAddress

- Example command to validate group filter

  ```
  ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
  -w "<LDAP_BINDPASSWORD>" -s sub "(&(cn=*)(objectclass=groupOfUniqueNames))"
  ```

- Example command to validate group ID map

  ```
  ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
  -w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=*)(cn=*))"
  ```

- Example command to validate group member ID map

  ```
  ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
  -w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=groupOfUniqueNames)(uniqueMember=*))"
  ```

- Example command to validate user filter

  ```
  ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
  -w "<LDAP_BINDPASSWORD>" -s sub "(&(emailAddress=*)(objectclass=person))"
  ```

- Example command to validate user ID map

  ```
  ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
  -w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=*)(emailAddress=*))"
  ```

Microsoft Active Directory LDAP filters

Table 2. Microsoft Active Directory LDAP filters

Attribute name	Default value
Group filter	(&(cn=%v)(objectcategory=group))
Group ID map	*:cn
Group Member ID map	memberOf:member
User filter	(&(sAMAccountName=%v)(objectcategory=user))
User ID map	user:sAMAccountName

- Example command to validate group filter

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(cn=*)(objectcategory=group))"
```

- Example command to validate group ID map

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectcategory=*)(cn=*))"
```

- Example command to validate group member ID map

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectcategory=*)(member=*))"
```

- Example command to validate user filter

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(sAMAccountName=*)(objectcategory=user))"
```

- Example command to validate user ID map

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectcategory=user)(sAMAccountName=*))"
```

Custom (OpenLDAP) server LDAP filters

Table 3. Custom (OpenLDAP) server LDAP filters

Attribute name	Default value
Group filter	(&(cn=%v)(objectclass=groupOfUniqueNames))
Group ID map	*:cn
Group Member ID map	groupOfUniqueNames:uniqueMember
User filter	(&(uid=%v)(objectclass=person))
User ID map	*:uid

- Example command to validate group filter

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(cn=*)(objectclass=groupOfUniqueNames))"
```

- Example command to validate group ID map

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=*)(cn=*))"
```

- Example command to validate group member ID map

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=groupOfUniqueNames)(uniqueMember=*))"
```

- Example command to validate user filter

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(uid=*)(objectclass=person))"
```

- Example command to validate user ID map

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" \
-w "<LDAP_BINDPASSWORD>" -s sub "(&(objectclass=*)(uid=*))"
```

Resolve common login issues

Unable to login to `cp-console` or `cpd` using LDAP authentication

The **`ClassCastException`** error is displayed if the **`ObjectClass`** or **`ObjectCategory`** attribute is not defined in the Liberty XML file. The following error is displayed in the logs of the **`auth-service`** pods:

```
Exception = java.lang.ClassCastException
Source = com.ibm.ws.security.oauth20.plugins.jose4j.OidcUserClaims
probeid = 178
Stack Dump = java.lang.ClassCastException: com.ibm.wsspi.security.wim.model.Entity incompatible with
com.ibm.wsspi.security.wim.model.PersonAccount
        at com.ibm.ws.security.oauth20.plugins.jose4j.OidcUserClaims.getUserinfoFromRegistryMap(OidcUserClaims.java:136)
        at com.ibm.ws.security.oauth20.plugins.jose4j.OidcUserClaims.getUserinfoFromRegistry(OidcUserClaims.java:177)
        at
com.ibm.ws.security.openidconnect.web.OidcEndpointServices.getUserinfoFromRegistry(OidcEndpointServices.java:1001)
```

```
        at com.ibm.ws.security.openidconnect.web.OidcEndpointServices.userinfo(OidcEndpointServices.java:922)
        at com.ibm.ws.security.openidconnect.web.OidcEndpointServices.handleOidcRequest(OidcEndpointServices.java:281)
        at com.ibm.ws.security.openidconnect.web.OidcEndpointServlet.handleRequest(OidcEndpointServlet.java:111)
        at com.ibm.ws.security.openidconnect.web.OidcEndpointServlet.doPost(OidcEndpointServlet.java:69)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:707)
```

Resolving the issue

IM supports the LDAP Entity type configuration for LDAP User and Group entities to define the **ObjectClass** or **ObjectCategory** attributes automatically in the Liberty XML file.

For the existing LDAP connection, you need to restart the **platform-auth-service** pods to enable the **LDAPEntityType** configuration.

1. Get the **platform-auth-service** pod name.

   ```
   oc get pods -n <your-foundational-services-namespace> | grep platform-auth-service
   ```

2. Delete the **platform-auth-service** pod.

   ```
   oc delete pod <platform-auth-service-pod-name> -n <your-foundational-services-namespace>
   ```

The following is the sample Liberty XML file with the LDAP configuration:

```
<server>
    <featureManager>
        <feature>ldapRegistry-3.0</feature>
    </featureManager>
    <ldapRegistry id="SunOneLDAP" realm="REALM" host="[dd103bgpvip1165.virginia.ms.com]
(https://dd103bgpvip1165.virginia.ms.com/)" port="389"
           ignoreCase="false" baseDN="o=Morgan Stanley" bindDN="msfwid=2053167,ou=prodids,o=Morgan Stanley"
           bindPassword="{xor}OQ9mHBIlFQVoaGk4NSgK" ldapType="Sun Java System Directory Server" recursiveSearch="true"
sslEnabled="false"
           certificateMapMode="CERTIFICATE_FILTER" certificateFilter="uid=${SubjectCN}">
         <ldapEntityType name="PersonAccount">
               <objectClass>msperson</objectClass>
         </ldapEntityType>
         <!-- For Groups -->
        <ldapEntityType name="Group">
              <objectClass>msaclgroup</objectClass>
       </ldapEntityType>
        <iplanetFilters
           userFilter="(&amp;(objectClass=msperson)(uid=%v))"
           groupFilter="(&amp;(cn=%v)(objectclass=msaclgroup)(mseonid=24507))"
           userIdMap="*:uid"
           groupIdMap="*:cn"
           groupMemberIdMap="groupofuniquenames:uniquemember" />
           <ldapCache>
             <attributesCache size="2000" timeout="1200s" enabled="true" sizeLimit="2000"/>
             <searchResultsCache size="2000" timeout="1200s" enabled="true" resultsSizeLimit="2000"/>
           </ldapCache>
    </ldapRegistry>
</server>
```

Unable to log in as LDAP user after you configure an LDAP connection

You might not be able to log in as an LDAP user even when the connection test and the LDAP configuration are successful.

You see an error that indicates an **invalid username or password**.

You might see the following error in the log files of the **platform-auth-service** container:

```
CWIML4537E: The login operation could not be completed. The specified principal name XXXXXX is not found in the back-end
repository.
```

To identify the error, enable trace for Liberty service to know the exact error in First failure data capture (FFDC) logs. For more information, see Enable debugging for user authentication issues.

Following are the possible causes. A resolution is provided for each cause.

Cause 1 - Invalid `username` value is used when you logged in

The user name value is not the same as the **userIdMap** filter attribute value.

Resolution for cause 1

Ensure that you enter the correct user name. The user name must be the same as the **userIdMap** filter attribute value. The user name is case-sensitive.

Consider the following example:

LDAP user details on an LDAP server:

```
dn: uid=testuser,ou=people,dc=abc,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
cn: TestUser
givenName: TestUser
sn: SN
```

```
uid: testuser
userPassword: testuser
mail: testuser@abc.com
```

LDAP configuration:

```
User filter: (&(uid=%v)(objectclass=person))
User ID map: *:uid
```

If you use **`*:uid`** as the **`userIdMap`** filter, then you must use the **`uid`** value **`testuser`** as the user name when you log in.

Cause 2 - LDAP login user or Bind DN user is disabled or locked at the LDAP server

- The LDAP login user is disabled or locked at the LDAP server due to many unsuccessful login attempts.
- The **Bind DN** user is disabled or locked at LDAP server due to many unsuccessful login attempts.

Resolution for cause 2

- Use the **`ldapsearch`** tool to test the LDAP connection.
- Enable or unlock the LDAP login user or **Bind DN** user at the LDAP server.

Cause 3 - Invalid Password value is used to log in

- The LDAP user password might be wrong.
- The **Bind DN password** might contain some unsupported special characters.

Resolution for cause 3

- Enter the correct password and try logging in.
- If the **Bind DN password** has special characters, change the password and do not use special characters in the changed password.

Cause 4 - LDAP configuration might not be synchronized across all the pods in a high availability (HA) cluster

The login request might be going to a pod in an HA cluster that does not have the LDAP configuration.

Resolution for cause 4

Restart the **`platform-auth-service`** pod. If the issue is still not resolved, restart the **`platform-auth-service`**, **`platform-identity-provider`**, and **`platform-identity-management`** pods.

Cause 5 - LDAP connection issues due to various reasons

- The LDAP server is not reachable.
- You used the LDAP server hostname in the LDAP server URL, and the hostname is not reachable. This issue might be because the correct DNS server entries were not added during cluster installation.
- If LDAPS is used to configure the LDAP connection, following might be the possible causes:
  - An IP address is used instead of the LDAP server hostname in the LDAP server URL.
  - The LDAP server certificate is not imported into your cluster.
  - The LDAP server certificate does not match with the LDAP server hostname (or certificate SANs (Subject Alternative Names)).

Resolution for cause 5

- Test the LDAP connection by using the **`ldapsearch`** tool.
- Add the LDAP server hostname in the **`/etc/hosts`** file either on the master node or in the **`platform-auth-service`** container.
- For configuring an LDAPS connection, use the LDAP server hostname in the LDAP server URL and the LDAP server certificate is imported into your cluster. Followed all the required steps for Configuring LDAP over SSL.
- Change the hostname, or get a new certificate that matches with the LDAP server hostname or any one of the certificate SANs.

Cause 6 - External database connection failed in the `platform-auth-service` container

You can see external database connection failure errors in the logs of the **`platform-auth-service`** container.

Resolution for cause 6

Restart the **`platform-auth-service`** pod that is configured with the external database.

Resolve common issues with search users or groups

Unable to search users or groups while you create a team.

Cause

You used an invalid search string.

Resolution

You must use the value of the **`cn`** attribute or the user or group attribute such as **`uid`** or **`emailaddress`** that you used in the LDAP configuration.

Consider the following example:

LDAP user details on an LDAP server

```
dn: uid=testuser,ou=people,dc=abc,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
cn: TestUser
givenName: TestUser
sn: SN
uid: testuser
userPassword: testuser
mail: testuser@abc.com
```

Following filters are used in the LDAP configuration:

- User filter: (&(uid=%v)(objectclass=person))
- User ID map: *:uid

The **userIdMap** that is used is ***:uid**.

The valid values that you can use to search for a user are as follows:

- TestUser (**cn** value)
- testuser (**uid** value)

Configuring LDAP over SSL

You can secure your Lightweight Directory Access Protocol (LDAP) connection by using SSL (Secure Sockets Layer).

If LDAP over SSL (LDAPS) is not automatically configured in your cluster, complete these steps to manually configure the LDAPS connection.

You must prepare your product cluster for connecting with your LDAPS directory.

Before you begin, you must import the public or private SSL certificate that you used for setting up your LDAPS directory.

Retrieving the SSL certificate

If you have the SSL certificate of your LDAP server, proceed with Encoding the SSL certificate.

If you do not have the SSL certificate of your LDAP server, use one of the following approaches to retrieve the SSL certificate:

Option 1: Using the `ldapsearch` command

Note: You need the **ldapsearch** program to run these commands. You can install it by running **apt install ldap-utils** on Ubuntu and **yum install openldap-clients** on Red Hat Enterprise Linux (RHEL).

1. Ensure that no SSL certificates are in the **/etc/openldap/cacerts** directory.
2. Run the following **ldapsearch** command to retrieve the certificate name:

   ```
   ldapsearch -H <LDAP server URL> -d 1  -b <searchbase> -D "" -s base "(<filter>)"
   ```

   Where,

   - **LDAP server URL** is your LDAP directory domain name, and port. Format: `ldaps://<LDAP server domain name or IP address>:<port>`.
   - **-d** is the debugging level.
   - **-b** is the search base.
   - **-D** is the bind DN. This parameter is optional.
   - **-s** is the scope of search.
   - **filter** is the LDAP filter. Default filter: `(objectClass=*)`.

   Sample command:

   ```
   ldapsearch -H ldaps://corp.example.com:636 -d 1  -b o=example.com -D "" -s base "(objectclass=*)"
   ```

   Sample output:

   ```
   ldap_url_parse_ext(ldaps://corp.example.com:636)
   ldap_create
   ldap_url_parse_ext(ldaps://corp.example.com:636/??base)
   ldap_sasl_bind
   ldap_send_initial_request
   ldap_new_connection 1 1 0
   ldap_int_open_connection
   ldap_connect_to_host: TCP corp.example.com:636
   ldap_new_socket: 3
   ldap_prepare_socket: 3
   ldap_connect_to_host: Trying 9.17.186.253:636
   ldap_pvt_connect: fd: 3 tm: -1 async: 0
   attempting to connect:
   connect success
   TLS: certdb config: configDir='/etc/openldap' tokenDescription='ldap(0)' certPrefix='cacerts' keyPrefix='cacerts'
   flags=readOnly
   TLS: cannot open certdb '/etc/openldap', error -8018:Unknown PKCS #11 error.
   ```

```
TLS: could not get info about the CA certificate directory /etc/openldap/cacerts - error -5950:File not found.
TLS: certificate [CN=DigiCert Global Root G2,OU=www.digicert.com,O=DigiCert Inc,C=US] is not valid - error
-8172:Peer's certificate issuer has been marked as not trusted by the user..
TLS: error: connect - force handshake failure: errno 2 - moznss error -8172
TLS: can't connect: TLS error -8172:Peer's certificate issuer has been marked as not trusted by the user..
ldap_err2string
ldap_sasl_bind(SIMPLE): Can't contact LDAP server (-1)
```

The certificate information is in the `TLS: certificate [CN=DigiCert Global Root G2,OU=www.digicert.com,O=DigiCert Inc,C=US] is not valid - error -8172:Peer's certificate issuer has been marked as not trusted by the user..` section of the output.

3. Locate the certificate in your web browser, export it to a file in a PEM format, and save the PEM file with a `.crt` extension.

   Note: If you have multiple certificates, export each certificate to a file in a PEM format and save the PEM file with a `.crt` extension.

4. Copy the `.crt` file to the infrastructure node of your cluster.

Option 2: Using the `openssl` tool

1. Run the following command:

   ```
   openssl s_client -connect <ldap-server-host>:636 -showcerts > ldap_server_certs.pem
   ```

2. Edit the `ldap_server_certs.pem` file:

   1. Remove the unwanted lines.
   2. Retain the content between `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` lines.

   Note: Ensure that final `ldap_server_certs.pem` certificate file is a valid PEM format and no empty lines are there in-between.

Preparing for a single LDAPS connection

If you are configuring a single LDAPS connection, complete the steps in the Encoding the SSL certificate section.

Encoding the SSL certificate

Complete the following steps to encode the certificate in base64:

1. Log on to the infrastructure node of your cluster.

2. Convert the certificate into PEM format.

   ```
   openssl x509 -in <cert>.cer -outform PEM -out <convertedCert>.pem
   ```

   If your LDAP server uses chain certificates (root CA and intermediate certificates), convert each certificate into PEM format. Then, combine them into one file. Use the following command to combine the converted certificates.

   ```
   cat <convertedFirstCert.pem> <convertedSecondCert.pem> ..<converted_n_Cert.pem> > <convertedCombinedCert>.pem
   ```

3. Encode your certificate in base64.

   ```
   cat <LDAPS SSL certificate name>.pem | base64 -w 0
   ```

   The output resembles the following code:

   ```
   LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdDRENDQS9DZ
   0F3SUJBZ0lRS3k1dTZ0bDFObXdVaW03Ym8zeU1CekFOQmdrcWhraUc5
   ...
   lDOHg0OU9oZ1E9Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0KDQo=
   ```

Next, proceed with Preparing your cluster.

Preparing for multiple LDAPS connections

If you are configuring multiple LDAPS connections, first check if you already have a certificate in the "data" > "certificate" section of `platform-auth-ldaps-ca-cert`. If you have a certificate, then complete the steps in the following sections:

- Retrieve the current certificate
- Encode the SSL certificate

If no certificate exists, then complete the steps in Encoding the SSL certificate.

Retrieve the current certificate

You can retrieve the certificate by using the Red Hat® OpenShift® Container Platform console or command line interface (CLI).

Option 1: Using the console

1. Log on to the Red Hat® OpenShift® Container Platform console as a cluster administrator.

2. From the navigation menu, click **Workloads** > **Secrets**.

3. Locate `platform-auth-ldaps-ca-cert` and click **...** > **Edit Secret**. A window for editing displays.

4. Copy the base64 encoded certificate value from "data" > "certificate" section.

```
"data": {
 "certificate": "LS0tLS1...ASDFASDo="
},
```

5. Convert the existing encoded certificate and save to a file.

```
echo "<copied_cert_value>" | base64 --decode > existing_cert.pem
```

Option 2: using the CLI

1. Access the CLI of your infrastructure node. Install the Kubernetes CLI (kubectl). You need this tool to complete the following tasks.
2. Get the base64 certificate.

```
kubectl -n <your-foundational-services-namespace> get secret platform-auth-ldaps-ca-cert -o "jsonpath=
{.data['certificate']}" | base64 --decode > existing_cert.pem
```

Next, proceed with Encode the SSL certificate.

Encode the SSL certificate

Complete the following steps to encode the certificate in base64:

1. Log on to the infrastructure node of your cluster.
2. Convert the new certificate into PEM format.

```
openssl x509 -in <cert>.cer -outform PEM -out <convertedCert>.pem
```

If your LDAP server uses chain certificates (root CA and intermediate certificates), convert each certificate into PEM format. Then, combine them into one file. Use the following command to combine the converted certificates.

```
cat <convertedFirstCert.pem> <convertedSecondCert.pem> ..<converted_n_Cert.pem> > <convertedCombinedCert>.pem
```

3. Add the existing certificate to the new certificate, or to the combined certificate if you use chain certificates.

```
cat existing_cert.pem <new_cert.pem or convertedCombinedCert.pem> > final_combined_cert.pem
```

4. Encode your certificate in base64.

```
cat <LDAPS SSL certificate name>.pem | base64 -w 0
```

The output resembles the following code:

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdDRENDQS9DZ
0F3SUJBZ01lRS3k1dTZ0bDFObXdVaW03Ym8zeU1CekFOQmdrcWhraUc5
...
lDOHg0OU9oZ1E9Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0KDQo=
```

Next, complete the steps in the Preparing your cluster section.

Preparing your cluster

You can prepare your cluster by using the Red Hat® OpenShift® Container Platform console or CLI.

Option 1: Using the console

1. Log on to the Red Hat® OpenShift® Container Platform console as a cluster administrator.
2. From the navigation menu, click **Workloads** > **Secrets**.
3. Locate `platform-auth-ldaps-ca-cert` and click **...** > **Edit Secret**. A window for editing displays.
4. Paste the base64 certificate in the following section:

```
"data": {
 "certificate": ""
},
```

The updated section resembles the following text:

```
"data": {
 "certificate": "LS0tLS1<very_long_base64_string>ASDFASDo="
},
```

5. Click **Save**.
6. From the navigation menu, click **Workloads > Deployments**.
7. Locate `platform-auth-service`.
8. Click **...** > **Edit Deployment**. A window for editing displays.
9. Click **Save** without making any change. This step is to reload the `platform-auth-service` pods with the latest ConfigMap values.
10. Wait for a minute or two and then check whether the certificate is mounted on the pod.
    1. Get the `platform-auth-service` pods.

```
kubectl -n <your-foundational-services-namespace> get pods | grep platform-auth-service
```

2. Check whether the certificate is mounted on the pod.

```
kubectl -n <your-foundational-services-namespace> exec -it platform-auth-service-<pod-id> -c platform-auth-service cat /opt/ibm/ldaps/ldaps-ca.crt
```

Option 2: Using the CLI

1. Access the CLI of your infrastructure node. You need **kubectl**, the Kubernetes command line tool to complete the following tasks. For more information about installing **kubectl**, see Install tools.
2. Copy the LDAPS CA certificate secret to a file:

```
kubectl get secret platform-auth-ldaps-ca-cert -o yaml -n <your-foundational-services-namespace> > platform-auth-ldaps-ca-cert-secret.yaml
```

3. Copy and paste the base64 certificate to the **platform-auth-ldaps-ca-cert-secret.yaml** file.
4. Get the security service Deployment YAML file:

```
kubectl -n <your-foundational-services-namespace> get deploy platform-auth-service -o yaml > platform-auth-service.yaml
```

5. Apply the changes:
    1. Run the following command to apply the **platform-auth-ldaps-ca-cert-secret.yaml** file:

```
kubectl apply -f platform-auth-ldaps-ca-cert-secret.yaml
```

    2. Run the following command to apply the **platform-auth-service.yaml** file:

```
kubectl apply -f platform-auth-service.yaml
```

Next, complete the steps in the Connecting to your LDAP directory section.

Troubleshooting users and user groups search issues

Verify or test search users and groups by using the **ldapsearch** command tool.

Search users

Use the following **ldapsearch** command:

```
ldapsearch -x -l <TIME_LIMIT> -z <SIZE_LIMIT> -H <LDAP_URL> -b <LDAP_BASEDN> -D <LDAP_BINDDN> -w <LDAP_BINDPASSWORD> -s sub "<search query>"
```

See the following definitions:

- **<LDAP_URL>** is the Lightweight Directory Access Protocol (LDAP) server URL.
- **<LDAP_BASEDN>** is the LDAP Base DN.
- **<LDAP_BINDDN>** is the LDAP Bind DN.
- **<LDAP_BINDPASSWORD>** is the LDAP Bind DN password.
- **<TIME_LIMIT>** is the time limit in seconds for the search. Default value is 5 seconds.
- **<SIZE_LIMIT>** is the size limit for search. Default value is 50 entries.

In IBM Cloud Pak® for AIOps, the search string is based on the **cn** and **User ID map** attributes that are configured in the LDAP connection user filters.

For example, consider the following LDAP user filters configuration:

```
User filter: (&(uid=%v)(objectclass=person))
User ID map: *:uid
```

For the example configuration, following is the search query:

```
(|(&(cn=*<searchstring>*)(objectclass=person))(&(uid=*<searchstring>*)(objectclass=person)))
```

In the search query, user name or user ID value is the **<searchstring>**. For example, **John** or **robbie**.

Following is an example **ldapsearch** command:

```
ldapsearch -x -l 5 -z 50 -H "ldap://X.X.X.X:389" -b "dc=abc,dc=com" -D "cn=admin,dc=abc,dc=com" -w 'password' -s sub "(|(&(cn=*user*)(objectclass=person))(&(uid=*user*)(objectclass=person)))"
```

Here, the **<searchstring>** is **user**.

Following is a sample output:

```
# extended LDIF
#
# LDAPv3
# base <dc=abc,dc=com> with scope subtree
# filter: (|(&(cn=*user*)(objectclass=person))(&(uid=*user*)(objectclass=person)))
# requesting: ALL
```

```
#

# user1, Users, abc.com
dn: uid=user1,ou=Users,dc=abc,dc=com
cn: User One
objectClass: inetOrgPerson
objectClass: person
objectClass: top
sn: One
uid: user1
userPassword:: dXNlcjE=

# user2, Users, abc.com
dn: uid=user2,ou=Users,dc=abc,dc=com
cn: User Two
objectClass: inetOrgPerson
objectClass: person
objectClass: top
sn: Two
uid: user2
userPassword:: dXNlcjI=

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Search groups

Use the following **ldapsearch** command:

```
ldapsearch -x -l <TIME_LIMIT> -z <SIZE_LIMIT> -H <LDAP_URL> -b <LDAP_BASEDN> -D <LDAP_BINDDN> -w <LDAP_BINDPASSWORD> -s
sub "<search query>"
```

See the following list of definitions:

- **<LDAP_URL>** is the LDAP server URL.
- **<LDAP_BASEDN>** is the LDAP Base DN.
- **<LDAP_BINDDN>** is the LDAP Bind DN.
- **<LDAP_BINDPASSWORD>** is the LDAP Bind DN password.
- **<TIME_LIMIT>** is the time limit in seconds for the search. Default value is 5 seconds.
- **<SIZE_LIMIT>** is the size limit for search. Default value is 50 entries.

In IBM Cloud Pak for AIOps, the search string is based on the **cn** and **Group ID map** attributes that are configured in the LDAP connection group filters.

For example, consider the following LDAP group filters configuration:

```
Group filter: (&(cn=%v)(objectclass=groupOfUniqueNames))
Group ID map: *:cn
```

For the example configuration, following is the search query:

```
(&(cn=*<searchstring>*)(objectclass=groupOfUniqueNames))
```

In the search query, group name value is the **<searchstring>**. For example, **security** or **administrators**.

If the **Group ID map** attribute is different than **cn**, for example **gid**, then the following search query can be used:

```
(|(&(cn=*<searchstring>*)(objectclass=groupOfUniqueNames))(&(gid=*<searchstring>*)(objectclass=groupOfUniqueNames)))
```

Here, the group name or **gid** value is the **<searchstring>**.

Following is an example **ldapsearch** command:

```
ldapsearch -x -l 50 -z 100 -H "ldap://X.X.X.X:389" -b "dc=abc,dc=com" -D "cn=admin,dc=abc,dc=com" -w 'password' -s sub "
(&(cn=*gr*)(objectclass=groupOfUniqueNames))"
```

Here, the **<searchstring>** is **gr**.

Following is a sample output:

```
# extended LDIF
#
# LDAPv3
# base <dc=abc,dc=com> with scope subtree
# filter: (&(cn=*gr*)(objectclass=groupOfUniqueNames))
# requesting: ALL
#

# group1, Groups, abc.com
dn: cn=group1,ou=Groups,dc=abc,dc=com
cn: group1
objectClass: groupOfUniqueNames
objectClass: top
uniqueMember: cn=group2,ou=Groups,dc=abc,dc=com
uniqueMember: uid=user1,ou=Users,dc=abc,dc=com
```

```
# group2, Groups, abc.com
dn: cn=group2,ou=Groups,dc=abc,dc=com
cn: group2
objectClass: groupOfUniqueNames
objectClass: top
uniqueMember: uid=user2,ou=Users,dc=abc,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Retrieve users from an LDAP group

Use the following **ldapsearch** command:

```
ldapsearch -x -H "<LDAP_URL>" -b "<LDAP_BASEDN>" -D "<LDAP_BINDDN>" -w "<LDAP_BINDPASSWORD>" -s sub "(cn=<GROUP_NAME>)"
```

See the following list of definition:

- **<LDAP_URL>** is the LDAP server URL.
- **<LDAP_BASEDN>** is the LDAP Base DN.
- **<LDAP_BINDDN>** is the LDAP Bind DN.
- **<LDAP_BINDPASSWORD>** is the LDAP Bind DN password.
- **<GROUP_NAME>** is the group name.

Consider the following example command:

```
ldapsearch -x -H "ldap://X.X.X.X:389" -b "dc=abc,dc=com" -D "cn=admin,dc=abc,dc=com" -w 'password' -s sub "(cn=group2)"
```

Following is a sample output:

```
# extended LDIF
#
# LDAPv3
# base <dc=abc,dc=com> with scope subtree
# filter: (cn=group2)
# requesting: ALL
#

# group2, Groups, abc.com
dn: cn=group2,ou=Groups,dc=abc,dc=com
cn: group2
objectClass: groupOfUniqueNames
objectClass: top
uniqueMember: uid=user2,ou=Users,dc=abc,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Timeout error while setting up an LDAP connection

Cannot set up a connection with your LDAP server.

Symptoms

While setting up an LDAP connection, you might see a **504 Gateway timeout** error.

Cause

The connection needs more time than usual to be established.

Resolving the problem

To resolve the issue, increase the LDAP configuration timeout value in the **platform-auth-idp** configmap to a value in the range 25-60 seconds.

1. Log in to your infrastructure node by using the **oc login** command.
2. Edit the **platform-auth-idp** configmap.

   ```
   oc -n <your-foundational-services-namespace> edit configmap platform-auth-idp
   ```

3. Change the **AUTH_SVC_LDAP_CONFIG_TIMEOUT: "25"** parameter value to a higher value. For example, **"45"**. The value is in seconds.
4. Save the changes.

5. Restart the `platform-auth-service` pods by deleting the pods.

    1. Get the `platform-auth-service` pod names.

       ```
       oc -n <your-foundational-services-namespace> get pods | grep platform-auth-service
       ```

    2. Delete the `platform-auth-service` pod.

       ```
       oc -n <your-foundational-services-namespace> delete pods <pod-name>
       ```

Wait for some time and then check the status of the `platform-auth-service` pods. The status of all `platform-auth-service` pods must show as `Running`.

```
oc -n <your-foundational-services-namespace> get pods | grep platform-auth-service
```

Timeout error after enabling LDAP Nested search for Microsoft Active Directory

Once you enable Nested search option while searching the SCIM users or groups, the time taken for the SCIM API to response might be increased. As a result, sometimes you might see that the SCIM search returns empty or `504 Gateway timeout` error in the logs.

Symptoms

- `504 Gateway timeout` error is displayed.
- Or, you get no response.

Cause

- The connection needs more time than usual to search SCIM users or groups.
- You might be searching many SCIM users or groups in a single query after enabling the Nested search option.
- The underlying LDAP server has many nesting levels.

Resolving the problem

1. Increase the OCP (OpenShift Container Platform) route timeout by using the following code:

   ```
   oc -n <your-foundational-services-namespace> annotate route cp-console --overwrite
   haproxy.router.openshift.io/timeout=60s
   ```

2. Disable the Nested search if it is not required. LDAP Nested Search can be a very expensive operation and it can go worse if the nesting level increases.

Client registration failure in Platform UI console

Client registration fails in Platform UI console When you upgrade the foundational services version 3.22 or version 3.23 to foundational services version 4.x.x.

Symptoms

You might fail to register the client in Platform UI console in the following scenarios:

- When you upgrade the foundational services version 3.22 or version 3.23 to foundational services version 4.x.x and migrate more than one LDAP.
- When you have more than one LDAP before upgrading foundational services version 3.22 or version 3.23 to foundational services version 4.x.x.

Workaround

- Check the `platform-auth-service` pod configuration. You might see that some files have wrong entries. To check the pod configuration, run the following command:

  ```
  for i in $(oc get pod -l app.kubernetes.io/instance=platform-auth-service --no-headers|awk '{print $1}');do oc exec
  -it $i -- ls /config/configDropins/defaults/;done
  ```

  The sample output:

  ```
  Defaulted container "platform-auth-service" out of: platform-auth-service, init-mongodb (init)
  federated.xml  keystore.xml  ldap-cp4ba-prod-default-4776.xml
  ```

- Restart `platform-auth-service` pods.

  ```
  oc delete pod $(oc get pod -l name=platform-auth-service --no-headers|awk '{print $1}')
  ```

  Once the pod is deleted, wait for some time and then restart the `platform-auth-service` pods.

- Then, restart `platform-identity-provider` pods.

  ```
  oc delete pod $(oc get pod -l name=platform-identity-provider --no-headers|awk '{print $1}')
  ```

Once the pod is deleted, wait for some time and then restart the **platform-identity-provider** pods.

Intermittent login failure in Platform UI console

You might fail to login Platform UI console when you upgrade the foundational services version 3.22 or version 3.23 to foundational services version 4.x.x.

Symptoms

There might be intermittent login failure in Platform UI console when you upgrade the foundational services version 3.22 or version 3.23 to foundational services version 4.x.x and then configure the multiple LDAPs.

Workaround

- If you check the **platform-auth-service** pod configuration, you might see that only the **keystore.xml** file is in the pod. Also, some files might be missing in the pod. To check the pod configuration, run the following command:

  ```
  for i in $(oc get pod -l app.kubernetes.io/instance=platform-auth-service --no-headers|awk '{print $1}');do oc exec -it $i -- ls /config/configDropins/defaults/;done
  ```

- Restart **platform-auth-service** pods.

  ```
  oc delete pod $(oc get pod -l name=platform-auth-service --no-headers|awk '{print $1}')
  ```

  Once the pod is deleted, wait for some time and then restart the **platform-auth-service** pods.

LDAP nested search for Microsoft Active Directory does not work correctly

SCIM LDAP Nested search for Microsoft Active Directory is not working properly.

Symptoms

Not all users that match the search query are returned in the search result.

Run these queries to verify whether the nested search is returning correct results.

LDAP queries

- Run an LDAP query to search all nested users that belong to a group: **memberOf:1.2.840.113556.1.4.1941:=<group dn>**.
  - Use the following syntax:

    ```
    ldapsearch -x -H ldap://ip:389 -b <LDAP_BASEDN> -D <LDAP_BINDDN> -w <LDAP_BINDPASSWORD> -s sub "(&(memberOf:1.2.840.113556.1.4.1941:=<group dn>)(objectcategory=user))"
    ```

  - Following is a sample output:

    ```
    ldapsearch -x -H ldap://1.2.3.4:389 -b dc=ibm,dc=com -D cn=root -w password -s sub "(&(memberOf:1.2.840.113556.1.4.1941:=cn=Icpgroup1,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com)(objectcategory=user))"
    ```

- Run an LDAP query to search all nested groups that a user belongs to: **(member:1.2.840.113556.1.4.1941:<user dn>)**.
  - Use the following syntax:

    ```
    ldapsearch -x -H ldap://ip:389 -b <LDAP_BASEDN> -D <LDAP_BINDDN> -w <LDAP_BINDPASSWORD> -s sub "(&(member:1.2.840.113556.1.4.1941:<user dn>)(objectcategory=group))"
    ```

  - Here is a sample output:

    ```
    ldapsearch -x -H ldap://1.2.3.4:389 -b dc=ibm,dc=com -D cn=root -w password -s sub "(&(member:1.2.840.113556.1.4.1941:uid=icpuser1,ou=people,dc=ibm,dc=com)(objectcategory=group))"
    ```

SCIM queries

- Run a SCIM query to get all nested users that belong to a group.
  - Use the following API:

    ```
    curl -sk -X GET --header "Authorization: Bearer $ACCESS_TOKEN" --header "Accept: application/json" "$BCS_URL/idmgmt/identity/api/v1/scim/groups/$GROUP_ID" | jq -r '.members'
    ```

  - See the following example result:

    ```
    curl -sk -X GET --header "Authorization: Bearer $ACCESS_TOKEN" --header "Accept: application/json" "$BCS_URL/idmgmt/identity/api/v1/scim/groups/cn=Icpgroup1,cn=platform,ou=cloud,ou=isl,ou=groups,dc=ibm,dc=com" | jq -r '.members'
    ```

- Run a SCIM query to get all nested groups that a user belongs to.
  - Use the following API:

    ```
    curl -sk -X GET --header "Authorization: Bearer $ACCESS_TOKEN" --header "Accept: application/json"
    "$BCS_URL/idmgmt/identity/api/v1/scim/users/$USER_ID" | jq -r '.groups'
    ```

  - See the following example result:

    ```
    curl -sk -X GET --header "Authorization: Bearer $ACCESS_TOKEN" --header "Accept: application/json"
    "$BCS_URL/idmgmt/identity/api/v1/scim/groups/uid=icpuser1,ou=people,dc=ibm,dc=com"  | jq -r '.groups'
    ```

Cause

Your LDAP directory setup might have problems, or your SCIM configuration might be wrong.

Resolving the problem

For LDAP query issues, check you LDAP directory data setup.

For SCIM query issues, see the available troubleshooting documents.

Pod issues

Troubleshoot security pod issues.

- platform-auth-service pod restarts multiple times
- Pods are not scheduled
- secret-watcher and security-onboarding pods fail to start
- Secret watcher pod shows CrashLoopBackOff status
- IM pods cannot be deployed when foundational services is configured with external EDB PostgreSQL
- The ibm-iam-operator pod fails because of cache sync failures

platform-auth-service pod restarts multiple times

The **platform-auth-service** pod restarts multiple times.

Symptoms

The **platform-auth-service** pod restarts multiple times. The **platform-auth-service** shows the following logs:

```
oc logs platform-auth-service -n <your-foundational-services-namespace> -c platform-auth-service

2019-02-12 18:18:12,170 WARN received SIGTERM indicating exit request
2019-02-12 18:18:12,170 INFO waiting for 01-wlp-platform-auth-service, 02-directory-service to die
2019-02-12 18:18:12,183 INFO stopped: 02-directory-service (terminated by SIGTERM)
[AUDIT   ] CWWKE0085I: The server defaultServer is stopping because the JVM is exiting.
[AUDIT   ] CWWKE1100I: Waiting for up to 30 seconds for the server to quiesce.
[AUDIT   ] CWWKT0017I: Web application removed (default_host): http://platform-auth-service-v9h8q:9080/oidc/
[AUDIT   ] CWWKT0017I: Web application removed (default_host): http://platform-auth-service-v9h8q:9080/oauth2/
2019-02-12 18:18:15,385 INFO waiting for 01-wlp-platform-auth-service to die
2019-02-12 18:18:18,387 INFO waiting for 01-wlp-platform-auth-service to die
2019-02-12 18:18:21,390 INFO waiting for 01-wlp-platform-auth-service to die
2019-02-12 18:18:22,391 WARN killing '01-wlp-platform-auth-service' (9) with SIGKILL
2019-02-12 18:18:22,514 INFO stopped: 01-wlp-platform-auth-service (terminated by SIGKILL)
```

Causes

The **platform-auth-service** is using more resources than the configured limit. This issue is seen more often on Linux on IBM Z and LinuxONE platforms.

Resolving the problem

To resolve the issue, remove the resource limits.

1. Install the Kubernetes CLI (kubectl).
2. Edit the **platform-auth-service** daemonset.

   ```
   kubectl edit ds platform-auth-service -n <your-foundational-services-namespace>
   ```

3. Locate the following section:

   ```
       name: platform-auth-service
       ports:
       - containerPort: 8443
   ```

```
      hostPort: 8443
      name: http
      protocol: TCP
    readinessProbe:
      failureThreshold: 3
      httpGet:
        path: /
        port: 8443
        scheme: HTTPS
      periodSeconds: 10
      successThreshold: 1
      timeoutSeconds: 1
    resources:
      limits:
        cpu: "1"
        memory: 1Gi
      requests:
        cpu: 100m
        memory: 256Mi
```

4. Remove the **limits** section under the **resources** section. The **resources** section must look like the following piece of code after the change:

```
      resources:
        requests:
          cpu: 100m
          memory: 256Mi
```

5. Save the file and wait until all the **platform-auth-service** pods restart. The pods might take a few minutes to restart.

Pods are not scheduled

Security pods are not scheduled since the master nodes do not meet the required memory or CPU requirements.

If you are unable to log in to the console, check whether the following pods are scheduled:

```
platform-identity-provider-xxx
platform-identity-management-xxx
platform-auth-service-xxx
auth-apikeys-xxx
secret-watcher-xxx
security-onboarding-xxx
```

If any of these pods are not scheduled, get information about the pod's controller (daemonset, statefulset, deployment, and other components) and check whether the pod is not getting scheduled due to resource unavailability. Use the following command to get the information:

```
kubectl describe daemonset platform-auth-service -n <your-foundational-services-namespace>
```

The output has the information about memory and CPU requirements. If the resource requirement is not met, increase the master node's memory or CPU.

After you increase the resources, the pods are automatically scheduled.

secret-watcher and security-onboarding pods fail to start

During IBM Cloud Pak foundational services installation, the **secret-watcher** and **security-onboarding** pods fail to start.

Causes

The identity management (IM) pods are not ready.

To get the status of **secret-watcher** and **security-onboarding** pods, run the following command:

```
oc get pod -A | grep -e secret-watcher -e security-onboarding
```

After you run the command, you might get the following output. The **STATUS** in the sample output shows that the **secret-watcher** pod is running successfully and the **security-onboarding** pod is completed.

```
NAMESPACE                      NAME                                  READY   STATUS      RESTARTS   AGE
<your-foundational-services-namespace>          secret-watcher-5c79cb8969-7w4nw   1/1     Running     0          19h
<your-foundational-services-namespace>          security-onboarding-8kh81         0/1     Completed   0          20h
```

For an instance, the **STATUS** of the pods can also display the error:

```
NAMESPACE              NAME                                READY     STATUS                       RESTARTS    AGE
<your-foundational-services-namespace>   secret-watcher-cccd54d9f-vdnn2   0/1       CreateContainerConfigError   0
20m
<your-foundational-services-namespace>   security-onboarding-btlhh        0/1       CreateContainerConfigError   0
20m
```

If you observe the error in the pod's status, you can run the following command to know the cause of the **CreateContainerConfigError** error. The following command display the list of events for the **security-onboarding** pod:

```
oc get event --namespace <your-foundational-services-namespace> --field-selector involvedObject.name=security-onboarding-xxxx
```

After you run the command, you might get the following status that indicates that the "icp-serviceid-apikey-secret" key is missing due to which the pod is showing **CreateContainerConfigError** error. It indicates the issue with the **iam-onboarding** job.

```
108m    Warning  Failed      pod/security-onboarding-8kh81  Error: secret "icp-serviceid-apikey-secret" not found
```

Resolving the problem

Restart the **iam-onboarding** job to resolve the issue.

1. Log in to your boot node by running the **oc login** command.
2. Restart the **iam-onboarding** job.
   - Get the **iam-onboarding** job.

oc get job iam-onboarding

```
  * If the `iam-onboarding` doesn't exist, restart the IM operator.

  ``` {: codeblock}
 oc delete po -l name=ibm-iam-operator

* If the `iam-onboarding` job exists, restart the iam-onboarding job.

``` {: codeblock}
```

oc get job iam-onboarding -o json | jq 'del(.spec.selector)' | jq 'del(.spec.template.metadata.labels)' | kubectl replace --force -f -

Secret watcher pod shows CrashLoopBackOff status

The **secret-watcher** pod keeps crashing and shows a **CrashLoopBackOff** status.

Cause

There might be many Kubernetes secrets that are configured in the cluster. The current memory limit that is configured in the **secret-watcher** pod is not sufficient to manage the Kubernetes secrets. Kubernetes killed the pod due to less memory.

Resolving the problem

To resolve the issue, increase the resource memory limit to a value that is sufficient for the pod to manage all the Kubernetes secrets.

1. Log in to your boot node with the **oc login** command.
2. Check the number of Kubernetes secrets that are in the cluster.

   ```
   oc get secrets --all-namespaces | wc -l
   ```

3. Edit the **CommonService** custom resource.

   ```
   oc -n <your-foundational-services-namespace> edit CommonService common-service
   ```

4. Add the following piece of configuration under the **spec.services** section. Add the required memory limit in the **resources.limits.memory** section.

   ```
   spec:
     services:
     - name: ibm-im-operator
       spec:
         secretwatcher:
           resources:
             limits:
               memory: 220Mi
   ```

5. Save and exit the custom resource.

   Wait for some time and then check the **secret-watcher** pod status. The status must show as **1/1 Running** and the updated memory must show in the pod description.

6. Get the **secret-watcher** pod status and name.

   ```
   oc -n <your-foundational-services-namespace> get pods | grep secret-watcher
   ```

7. Check the pod description.

   ```
   oc -n <your-foundational-services-namespace> describe pod <secret-watcher-pod-name>
   ```

OIDC issues

Troubleshoot OIDC related issues.

- OIDC client creation is not successful
- OIDC registration fails to update

OIDC client creation is not successful

There can be many reasons that why the OIDC client creation is not successful, however, the few obvious reasons are listed following:

- Invalid client CR
- Associated pod is not running properly
- Issue with the dependent services, such as `platform-auth-service` pod, etc.
- OIDC client watcher pod is not Out Of Memory (OOM) or not in the `crashloopback` state.

If you have verified that the client CR is valid, associated pods are running properly or dependent services are also running properly and still the OIDC client creation is still not successful then the possible reason might be that the `oidc-client-watcher` is not able to connect with `auth-service` endpoint properly. This might be because of not having the correct certificate entry at the `auth-service` end.

Confirming the certificate import issue if oidc-client-watcher pod is not able to connect with auth-service endpoint properly

The following symptoms might confirm whether there is certificate import issue:

- If no status is available in the client CR after the client CR is created
- Also, the `oidc-client-watcher` container might show the similar error:

  ```
  oc logs $oidcclient-watcher-xxx -f
  ```

Replace **oidcclient-watcher-xxx** with actual pod and instance names.

```
{"level":"error","ts":1669062335.0758603,"logger":"controller-runtime.controller","msg":"Reconciler
error","controller":"controller_oidc_client","request":"cap4ba-ns3/ibm-bts-oidc-client","error":"Error occurred during
create oidc client regstration".
```

Where, **request** name might vary.

In case you see this error, you can verify whether you are observing the error due to the certificate issue. If there are multiple `platform-auth-service` instances, you need to verify for all the instances by using the following command:

```
oc login
oc project $namespace
oc exec -it $platform-auth-service-xxx /bin/bash
cd /opt/ibm/wlp/output/defaultServer/resources/security
keytool -v -list -keystore key.jks -alias default -storepass insecurePass
```

Replace the **platform-auth-service-xxx** with the `platform-auth-service` instance names. And, provide the namespace where `common-services` are installed.

If you find the default alias entry that contains the following information as `Issuer` and `Owner`, it means that the certificate import is an issue.

```
Owner: OU=ISL, O=IBM India, L=BANGALORE, ST=KARNATAKA, C=IN
Issuer: EMAILADDRESS=test@test.local, CN=CFSSL TEST Root CA, O=CFSSL TEST, L=San Francisco, ST=California, C=US
```

Note: If certificate import is the issue, see Steps to follow if certificate import is an issue. If certificate import is not the issue then there might be different reason that why the client creation is not happening and that needs to be checked.

Steps to follow if certificate import is an issue

1. Login to the cluster by using `oc` CLI.
2. Save the content of shell as `updateJKS.sh`.
3. Run the shell script as `updateJKS.sh $namespace`. Provide the namespace where `common-services` are installed.

```
oc project $1
for pod in $(oc get pod -o name | grep platform-auth-service);
do
  echo "Going to update jks for pod Name: ${pod}"
  oc exec -it ${pod} /bin/bash -c 'platform-auth-service' -- keytool -importkeystore -destkeystore
/opt/ibm/wlp/output/defaultServer/resources/security/key.jks -srckeystore /certs/generated/platformauth-ks.p12 -
srcstoretype pkcs12 -alias default -storepass insecurePass -srcstorepass insecurePass -noprompt
  echo  "Done"
done
```

Note: It is possible that after restart you might see the same issue so it is recommended to re-run the script.

OIDC registration fails to update

Symptoms

Before you register the OIDC clients by using IdP V3 API, you need to login into third party ID provider. And, then you can register the OIDC clients in the application. While registering, you use application url as cp-console url and redirect URL as **`https://<cp-console-url>/ibm/api/social-login/redirect/<name of the oidc>`**. However, you might face issue while opening the cp-console browser. When you click the configured ID provider name, you might not be redirected to the authentication page of that IdP.

Causes

Issue with configuring the OIDC XML files in multi-cluster environment.

Troubleshooting the issue

Restart all the replicas of **`platform-auth-service`** pod.

To restart the **`platform-auth-service`** pod:

1. Get the platform-auth-service pod ID.

   ```
   oc -n <your-foundational-services-namespace> get pods | grep platform-auth-service
   ```

   Following is a sample output of the command:

   ```
   platform-auth-service-t6sfm          4/4     Running      0     1d
   ```

2. Delete the **`platform-auth-service`** pod.

   ```
   oc -n <your-foundational-services-namespace> delete pod pod-name
   ```

   Where, **pod-name** specifies the name of the **`platform-auth-service`** pod that you want to delete.

3. Once the pod is deleted, you might see a similar message:

   ```
   pod "pod-name" deleted
   ```

   Where, **pod-name** specifies the name of the **`platform-auth-service`** pod that has been deleted.

4. Verify the status of the **`platform-auth-service`** pod that has been deleted.

   ```
   oc -n <your-foundational-services-namespace> get pods | grep pod-name
   ```

   Where, **pod-name** specifies the name of the **`platform-auth-service`** pod that you want to verify whether the pod is deleted or not.

5. The **`platform-auth-service`** pod will restart. The pod gets new ID after the restart.

API call issues

Troubleshoot OIDC related issues.

- Cannot generate IAM access token through API for external OIDC and SAML users
- Unable to find and add the IDP users or user groups using the console

Error 403 as a response while calling `/iam-token/serviceids/` APIs

While calling **`/iam-token/serviceids/`** APIs, you get **403** as a response.

Causes

The deployment of the old version of the Cloud Pak might not be properly cleaned up while you upgrading the Cloud Pak version. To know the reason for getting **403** as a response while calling **`/iam-token/serviceids/`** APIs, enable **`platform-auth-service`** pod debug level traces and liberty traces by using the following steps:

1. Open **`platform-auth-idp`** configmap to edit it by using the following command:

   ```
   oc edit cm platform-auth-idp
   ```

   After you run this command, the editor view of the configmap is displayed.

2. In **`platform-auth-service`** container log:

   - Change the value of **`LOG_LEVEL_AUTHSVC`** parameter to **`debug`**.

   Note: By default, the value of **`LOG_LEVEL_AUTHSVC`** parameter is **`warning`** or **`info`**. Changing its value to **`debug`** provides you the exact cause of error **403** that you get while calling **`/iam-token/serviceids/`** APIs.

   - Change the value of **`LIBERTY_DEBUG_ENABLED`** parameter to **`true`** to view liberty debug traces.

You might see the following trace sample. The sample displays the error, **Authorization failed. User is not authorized**.

```
[12/8/22 17:29:46:580 UTC] 00000099 id=00000000 com.ibm.cloud.iam.core.ServiceIdCore                    I
filterServiceIdByUserNamespace crnNamespace namespace : cp4s
[12/8/22 17:29:46:580 UTC] 00000099 id=00000000 com.ibm.cloud.iam.core.ServiceIdCore                    I
filterServiceIdByUserNamespace User namespace : cert-manager
[12/8/22 17:29:46:580 UTC] 00000099 id=00000000 com.ibm.cloud.iam.core.ServiceIdCore                    I
filterServiceIdByUserNamespace User namespace : cp4s-sandbox
[12/8/22 17:29:46:580 UTC] 00000099 id=00000000 com.ibm.cloud.iam.core.ServiceIdCore                    I
filterServiceIdByUserNamespace User namespace : cp4sec
[12/8/22 17:29:46:580 UTC] 00000099 id=00000000 com.ibm.cloud.iam.core.ServiceIdCore                    I
filterServiceIdByUserNamespace User namespace : default
.
.
.
[12/8/22 17:29:46:588 UTC] 00000099 id=00000000 com.ibm.cloud.iam.core.ServiceIdCore                    I
filterServiceIdByUserNamespace User namespace : services
[12/8/22 17:29:46:588 UTC] 00000099 id=00000000 com.ibm.cloud.iam.serviceid.rest.ServiceidsApi          1
listServiceid
com.ibm.cloud.iam.exception.authorization.AuthorizationServiceException: BXNIM0500E: Authorization failed. User is not
authorized..
at com.ibm.cloud.iam.core.ServiceIdCore.filterServiceIdsByUserNamespace(ServiceIdCore.java:406)
at com.ibm.cloud.iam.core.ServiceIdCore.listServiceId(ServiceIdCore.java:353)
```

IMPORTANT: After performing the steps, restore the value of **LOG_LEVEL_AUTHSVC** and **LIBERTY_DEBUG_ENABLED** parameters value to its default values to prevent unintended adverse effects, including degraded performance and system instability.

Troubleshooting the issue

To troubleshoot the issue, restore the old namespace where Cloud Pak was deployed.

Cannot generate IAM access tokens through API for external OIDC and SAML users

An LDAP user can generate an IAM access token with API. You cannot generate the IAM access token through APIs for external OIDC and SAML users in the Platform UI.

For example:

Curl command to generate access token:

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -
d "grant_type=password&username=alfredo.cappariello@ie.ibm.com&password=*********&scope=openid" https://cp-
console.roks.ibm-stratus.com/idprovider/v1/auth/identitytoken
```

The sample output:

```
Failed to get access token, invalid token request parameters server_error
```

Causes

By default, IAM checks for the username in the LDAP backend repository to validate the user. The following error is displayed in the liberty logs if an LDAP identity provider is not configured in the Cloud Pak console.

```
com.ibm.wsspi.security.wim.exception.PasswordCheckFailedException: CWIML4537E: The login operation could not be
completed. The specified principal name 270001635h is not found in the back-end repository.
```

To enable debugging for **platform-auth-service** liberty, see Enable debugging for user authentication issues.

Workaround

Follow these steps to obtain an access token from Platform UI:

1. Login to the Platform UI. For more information, see Accessing your cluster by using the console.
2. Open Developer tools in your web browser and click **Application** tab.
3. Go to the **Cookies** section and obtain the access token in the **ibm-private-cloud-session** cookie.
4. Use the obtained token to make an API call.

Troubleshooting certificates

Learn how to isolate and resolve problems with certificates.

- Expired certificate secrets fail to rotate

Expired certificate secrets fail to rotate

When a cluster is restored from a snapshot or rebooted after a long period of shutdown, the certificate manager might need to rotate many certificates at the same time. When the IBM Cert Manager is used, this can lead to the Cert Manager failing to rotate some secrets associated with the certificates.

Solution: Use the following commands to identify if any certificate secrets contain expired certificate data:

1. Set the project (namespace):

   ```
   oc project <namespace>
   ```

   Where **<namespace>** is the project (namespace) where IBM Cloud Pak for AIOps is installed.

2. Run the following script to check for expired certificate secrets:

   Note: The script requires version 4 of bash.

   ```
   convert_date () {
   local months=( Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec )
   local i
   for (( i=0; i<11; i++ )); do
      [[ $2 = ${months[$i]} ]] && break
   done
   printf "%4d-%02d-%02d\n" $3 $(( i+1 )) $1
   }

   declare -A CERT_TO_EXPIRY

   for secret in $(oc get certificate -o jsonpath='{.items[*].spec.secretName}'); do CERT_TO_EXPIRY["$(oc get secret
   $secret -o jsonpath='{.metadata.name}')"]="$(oc get secret $secret -o go-template='{{index .data "ca.crt" |
   base64decode}}' | openssl x509 -text -noout | grep "Not After" | awk '{print $5 " " $4 " " $7}')"; done

   NOW=$(date +%s)

   for secret in ${!CERT_TO_EXPIRY[@]}; do
   CERT_DATE=${CERT_TO_EXPIRY[$secret]}
   if [[ "$OSTYPE" == "linux-gnu"* ]]; then
      CERT_DATE_SEC=$(date -d "$(convert_date $CERT_DATE)" +%s)
   elif [[ "$OSTYPE" == "darwin"* ]]; then
      CERT_DATE_SEC=$(date -j -f "%Y-%m-%d" "$(convert_date $CERT_DATE)" +%s)
   fi
   if [ $CERT_DATE_SEC -lt $NOW ]; then echo "$secret expired on $CERT_DATE"; fi
   done
   ```

3. If no secret names are printed from running the preceding script, no action is required. If secret names are printed, check the state of the owning certificates using the following command:

   ```
   oc get certificates.cert-manager.io
   ```

   The output can resemble the following sample output:

   ```
   NAME                                        READY   SECRET                                      AGE
   aiops-ibm-elasticsearch-tls-secret          True    aiops-ibm-elasticsearch-tls-secret          4h2m
   aiops-installation-redis-client-cert        True    aiops-installation-redis-client-cert        4h7m
   aiops-installation-redis-server-cert        True    aiops-installation-redis-server-cert        4h7m
   ```

   If any certificates show as `False` in the **READY** column, contact IBM Support to resolve the Cert Manager failure. If all certificates show as `True` in the **READY** column, continue with the resolution steps.

4. Use the following script to force the certificate secrets to be rotated.

   Note: The script requires version 4 of bash.

```
convert_date () {
   local months=( Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec )
   local i
   for (( i=0; i<11; i++ )); do
      [[ $2 = ${months[$i]} ]] && break
   done
   printf "%4d-%02d-%02d\n" $3 $(( i+1 )) $1
}

declare -A CERT_TO_EXPIRY

for secret in $(oc get certificate -o jsonpath='{.items[*].spec.secretName}'); do CERT_TO_EXPIRY["$(oc get secret $secret
-o jsonpath='{.metadata.name}')"]="$(oc get secret $secret -o go-template='{{index .data "ca.crt" | base64decode}}' |
openssl x509 -text -noout | grep "Not After" | awk '{print $5 " " $4 " " $7}')"; done

NOW=$(date +%s)

for secret in ${!CERT_TO_EXPIRY[@]}; do
  CERT_DATE=${CERT_TO_EXPIRY[$secret]}
  if [[ "$OSTYPE" == "linux-gnu"* ]]; then
    CERT_DATE_SEC=$(date -d "$(convert_date $CERT_DATE)" +%s)
  elif [[ "$OSTYPE" == "darwin"* ]]; then
    CERT_DATE_SEC=$(date -j -f "%Y-%m-%d" "$(convert_date $CERT_DATE)" +%s)
  fi
  if [ $CERT_DATE_SEC -lt $NOW ]; then oc delete secret $secret; fi
done
```

Troubleshooting Infrastructure Automation

Review the details about frequently encountered issues that are related to Infrastructure Automation.

- Troubleshooting installing and uninstalling Infrastructure Automation
- Troubleshooting Infrastructure Management
- Troubleshooting Managed services

Troubleshooting installing and uninstalling Infrastructure Automation

Review frequently encountered issues related to installing, upgrading and uninstalling Infrastructure Automation.

Installation issues

- Database is stuck on retrying on Infrastructure Automation IMInstall Power Environment
- The `cam-tenant-api` pod is not in a ready state after installing the IAConfig CR
- Offline install or upgrade throws 'invalid character' error
- Offline install or upgrade stuck because the 'oc ibm-pak generate mirror-manifests' command fails with 'no space left on device'

Uninstall issues

- Uninstall fails to remove Infrastructure Management cleanly after deleting IAConfig instance
- Uninstall hangs when uninstalling Managed services

Troubleshooting installation

Database is stuck on retrying on Infrastructure Management IMInstall Power Environment

Sometimes when Infrastructure Management is deployed, the orchestrator pod fails to start properly with the following in the oc logs:

```
Cannot connect to the database!
Deployment status is
```

Check the postgresql pod to ensure it's running and listening. Then, verify the **vmdb_production** database was created by rushing into the postresql pod:

```
sh-4.4$ psql -U postgres
psql (10.19)
Type "help" for help.

postgres=# \l
                                  List of databases
      Name       |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------------+----------+----------+------------+------------+
 postgres        | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
                 |          |          |            |            | postgres=CTc/postgres
 template1       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
                 |          |          |            |            | postgres=CTc/postgres
(3 rows)
```

Solution: If the **vmdb_production** database doesn't exist, like above, it is possible that a remnant postgresql.conf in the DATA directory is preventing the postgresql image from initializing the new database on boot.

You will need to check the DATA directory and either clean it manually or recreate the PVC/PV and check that it is empty.

You can always find the location of the DATA directory using the psql command below:

```
postgres=# show data_directory;
        data_directory
------------------------------
 /var/lib/pgsql/data/userdata
(1 row)
```

Ensure that the data directory is empty before the postgresql pod is restarted.

The `cam-tenant-api` pod is not in a ready state after installing the IAConfig CR

After you install Infrastructure Automation, you can encounter an error where the **cam-tenant-api** pod displays as running, but not in a ready state. When this error occurs, you can see the following message:

```
[ERROR] init-platform-security - >>>>>>>>>> Failed to configure Platform Security. Will retry in 60 seconds <<<<<<<<<<<<
OperationalError: [object Object]
```

If this error occurs, delete the **cam-tenant-api** pod to cause the pod to restart and attempt to enter a ready state.

Offline install or upgrade throws 'invalid character' error.

When doing an offline install or upgrade, running the **oc ibm-pak generate mirror-manifests <..>** command throws an error similar to the following:

```
Error: failed to load the catalog FBC at cp.stg.icr.io/cp/ <...> invalid character '<' in string escape code
```

Solution: You must have **IBM Catalog Management Plug-in for IBM Cloud Pak** (**ibm-pak-plugin**) v1.10 or higher installed. Run the following commands to ensure that **ibm-pak-plugin** is at the required level.

1. Check which version of **ibm-pak-plugin** you have installed.

   Run the following command on your bastion host, portable compute device, or connected compute device if you are using a portable storage device.

   ```
   oc ibm-pak --version
   ```

   Example output:

   ```
   oc ibm-pak --version
   v1.11.0
   ```

2. If the **ibm-pak-plugin** version is lower than v1.10.0, then you must download and install the most recent version.

   Follow the steps for your installation approach:

   - Bastion host: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.
   - Portable device: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.

Offline install or upgrade stuck because the 'oc ibm-pak generate mirror-manifests' command fails with 'no space left on device'

The **oc ibm-pak generate mirror-manifests $IA_CASE_NAME $TARGET_REGISTRY --version $IA_CASE_VERSION** command fails with a message similar to the following in **$IBMPAK_HOME/.ibm-pak/logs/oc-ibm_pak.log**:

```
write /tmp/render-unpack-4002583241/var/lib/rpm/Packages: no space left on device
```

Solution: The default temporary directory does not have enough space to run the **ibm-pak** tool. You must set the **TMPDIR** environment variable to a different directory with more space before running the **oc ibm-pak generate mirror-manifests** command.

```
TMPDIR=<new_temp_dir> oc ibm-pak generate mirror-manifests $IA_CASE_NAME $TARGET_REGISTRY --version $IA_CASE_VERSION
```

Where **<new_temp_dir>** is the path for a directory with more space.

Troubleshooting uninstall

Uninstall fails to remove Infrastructure Management cleanly after deleting IAConfig instance

After deleting Infrastructure Management IAConfig CR in the namespace to uninstall Infrastructure Management from Red Hat OpenShift Container Platform console, some of the Infrastructure Management pods were still there in the namespace. You can workaround the problem by deleting IMInstall custom resource from the Red Hat OpenShift Container Platform console. This will remove the pods from the namespace.

You should also delete the **clients.oidc.security.ibm.com** customer resource that was used by Infrastructure Management if it still exists. Then, You can uninstall the operators remaining on the namespace.

To delete IMInstall Custom Resource (CR), you need to edit the CR to remove the Finalizer. Save the CR. Then delete the IMInstall Custom Resource (CR). Run the following commands or perform the same from the Red Hat OpenShift Container Platform console.

1. Identify the CR name, for example, **im-iminstall**

   ```
   oc -n <project> get iminstall
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed.

2. Edit the statement "finalizers:" and the one below it. Save and exit.

   ```
   oc -n <project> edit iminstall im-iminstall
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

3. Delete the CR.

   ```
   oc -n <project> delete iminstall im-iminstall
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

After a few minutes, the Infrastructure Management pods should no longer exist. The clients.oidc.security.ibm.com Custom Resource (CR) could still exist, and to delete the CR, do the following steps,

1. Check if the CR exist.

   ```
   oc -n <project> get clients.oidc.security.ibm.com
   ```

Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

2. If the CR exists, edit the CR. Delete the statement "finalizers:" and the one below. Save and exit.

   ```
   oc -n <project> edit clients.oidc.security.ibm.com ibm-infra-management-application-client
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

3. Delete the CR.

   ```
   oc -n <project> delete clients.oidc.security.ibm.com ibm-infra-management-application-client
   ```

   Where **<project>** is the project (namespace) where Infrastructure Automation is deployed

Uninstall hangs when uninstalling Managed services

The Infrastructure Automation uninstallation sometimes hangs when trying to delete the manageservice instance.

Describing the Managed services instance shows output similar to the following:

```
# oc describe manageservice cam
Name:         cam
Namespace:    cp4aiops
Labels:       operator.ibm.com/opreq-control=true
Annotations:  operator-sdk/primary-resource: /cam-services-sa-csb-patch-rb-pod
              operator-sdk/primary-resource-type: ClusterRoleBinding.rbac.authorization.k8s.io
API Version:  cam.management.ibm.com/v1alpha1
Kind:         ManageService
Metadata:
  Creation Timestamp:             2024-03-05T21:46:56Z
  Deletion Grace Period Seconds:  0
  Deletion Timestamp:             2024-04-15T01:11:29Z
  Finalizers:
    helm.sdk.operatorframework.io/uninstall-release
  Generation:         2
  Resource Version:   55077252
  UID:                87e6b8cf-3123-4914-bf64-a2b209cbe22d
```

Solution: Edit the Managed services instance to remove the finalizer entry.

1. Run the following command:

   ```
   oc edit ManageService cam
   ```

2. Delete the following lines, and then save your changes.

   ```
   Finalizers:
       helm.sdk.operatorframework.io/uninstall-release
   ```

Troubleshooting Infrastructure Automation

Review the details about frequently encountered issues that are related to Infrastructure Automation.

- Troubleshooting installing and uninstalling Infrastructure Automation
- Troubleshooting Infrastructure Management
- Troubleshooting Managed services

Support

If you need help with using IBM Cloud Pak® for AIOps, visit the IBM Support page.

Additionally, you can see whether the problem is documented in Known issues or in Troubleshooting.

Learn more about IBM Support from the following topics:

- Opening an IBM Support case
- Installing and running the MustGather tool (Linux)
- Installing the MustGather tool (Openshift)
- Installing the MustGather tool
- Running the MustGather tool

Collecting support information for IBM Cloud Pak for AIOps

Learn how to gather information that is needed and relevant to a problem before you open a case with IBM® Support. Additionally, learn how to open a meaningful support case with IBM Support for IBM Cloud Pak® for AIOps.

Before you open an IBM Support case

Check the troubleshooting and known issues sections

Before you contact IBM Support, review the following topics to check whether your problem is already reported and a workaround or solution exists:

- Troubleshooting and support
- Known issues

Collect support information with MustGather tools

If you cannot find a solution for your problem in the documentation, complete the following steps to use the MustGather tool and gather information from the cluster for opening an IBM Support case. The collected data facilitates more effective support.

- Data collection for IBM Support cases
- Infrastructure Automation data collection for IBM Support cases

Data collection for IBM Support cases

1. Ensure that you meet the prerequisites for running the MustGather tool for collecting data
   - Access to the cluster as a user with a cluster-admin role.
   - The Red Hat® OpenShift® Container Platform CLI (oc) installed. For more information, see Installing the Red Hat OpenShift CLI.
2. Download and install the MustGather tool
3. Run the MustGather tool

Infrastructure Automation data collection for IBM Support cases

1. Ensure that you meet the prerequisites for running the MustGather tool for collecting data
   - Access to the cluster as a user with a cluster-admin role.
   - The Red Hat® OpenShift® Container Platform CLI (oc) installed. For more information, see Installing the Red Hat OpenShift CLI.
2. Run the MustGather to collect data and logs

   Follow the instructions on the following IBM Support page to run the MustGather tool to collect the data and logs for IBM Cloud Pak for AIOps: Collecting Data: MustGather for IBM Cloud Pak for AIOps - Infrastructure Automation

Opening an IBM Support case

If you cannot find a solution to your problem in product documentation, open a support case with IBM Support. Complete the following steps

1. Go to the IBM Support site and click **Open a case**.
2. Log in with your IBM ID and password, and consent to the IBM Privacy Statement.
3. Enter the case details. Enter the meaningful title that summarizes your problem. Then, select the Product Manufacturer, Product, and Severity.

   Note: The case severity is based on the business impact of the problem. If you set the case severity as 1, you need to have 24x7 availability to work with IBM Support on this issue.
4. Provide the detailed description of your problem. A detailed description can help IBM Support understand your problem more accurately and thus provide solutions or answers more quickly. The following information is crucial:
   - Product version
   - Platform
   - Red Hat OpenShift Container Platform version
   - Virtualization platform (VMWARE, Azure, AWS, IBM Cloud)
   - High Availability (HA)
   - Steps to reproduce the issue
5. Optional: Upload the `.tar.gz` file with the results of the diagnostic scan. For more information, see Collecting support information.

   Note: If you encounter problems with uploading the file, see Uploading a file during case creation.
6. Select the language preferences and click **Submit a case**.

Installing and running the MustGather tool (IBM Cloud Pak for AIOps on Linux)

Learn about installing and running the MustGather tool for IBM Cloud Pak for AIOps on Linux.

Installing and running the MustGather tool

The MustGather tool is used for troubleshooting and support purposes. It collects data and logs about a deployment of IBM Cloud Pak for AIOps. To install and run the latest version of the MustGather tool, run the following command on the control plane node:

```
aiopsctl mustgather
```

Advanced configuration

You can also supply parameters to the MustGather tool. For more information, see Running the MustGather tool. In the example code blocks in the MustGather topics, replace **waiops-mustgather.sh** with **aiopsctl mustgather** for a deployment of IBM Cloud Pak for AIOps on Linux.

There are additional options available for deployments of IBM Cloud Pak for AIOps on Linux. Run the following command to seee usage:

```
aiopsctl mustgather -h
```

Example output:

```
root@acme-server:~# aiopsctl mustgather -h
Run the AIOps MustGather script to collect server information for support

Usage:
  aiopsctl mustgather [flags]

Flags:
  -a, --alias string     MustGather configuration alias
  -f, --force-install    Force MustGather to be reinstalled
  -h, --help             help for mustgather
  -o, --options string   Additional options that will be supplied to the MustGather script
  -u, --update           Check for MustGather updates

Global Flags:
      --accept-license     Accept the license
  -c, --config string      Config file location (default "~/.aiopsctl.yaml")
  -l, --log-file string    Log filename, set to stdout to print to console
      --show-secrets       Allow the display of secrets in output
  -v, --verbose            Verbose logging
      --wait-timeout int   Wait timeout for services to come ready in minutes (default 180)
```

Installing the IBM Cloud Pak for AIOps MustGather tool (IBM Cloud Pak for AIOps on Red Hat OpenShift)

You must download and install the MustGather tool before you can run any commands for the tool. You must download the installation package for the MustGather tool from IBM Support. You can download the package to any machine that has valid access to the target cluster for running **oc** commands.

Important: This MustGather tool gathers data for IBM Cloud Pak® for AIOps deployments. For gathering data for Infrastructure Automation, see Infrastructure Automation data collection for IBM Support cases.

Procedure

Complete either of the following steps to download the tool:

- Download the (**waiops-mustgather-<version>.tar.gz**) package at the end of the following IBM Support page:

  Collecting Data: MustGather for IBM Cloud Pak for AIOps

- Open a command-line terminal and use the following command to download the package:

  ```
  wget https://www.ibm.com/support/pages/system/files/inline-files/waiops-mustgather-<version>.tar.gz
  ```

  Where **<version>** is the version of the MustGather tool. Refer to the preceding IBM Support page to determine the latest available version. For v.1.16.6, the command is:

  ```
  wget https://www.ibm.com/support/pages/system/files/inline-files/waiops-mustgather-v1.16.6.tar.gz
  ```

Important:

- Unpack the downloaded **waiops-mustgather-<version>.tar.gz** package in its own directory. Ensure that the directory is not shared with any other files or directories.
- Keep the MustGather tool up to date. By default the tool can run a **new-version-check** automatically. To disable this check, include the **-M** option when you run the MustGather tool to gather data. If you want to directly check for, and download, the latest version run the following command

  ```
  waiops-mustgather.sh -UD
  ```

MustGather tool use cases

Learn how to run the tool, and learn about the different use cases and command options that are available for use with the tool. For more information, see:

- Running the MustGather tool

- Data collection modes
- Advanced tips for using the MustGather tool

IBM Cloud Pak for AIOps MustGather tool

Learn how to use the IBM Cloud Pak® for AIOps MustGather tool to gather information that is needed and relevant to a problem before you open a case with IBM Support.

Important: This MustGather tool gathers data for IBM Cloud Pak for AIOps deployments. For gathering data for Infrastructure Automation, see Infrastructure Automation data collection for IBM Support cases.

The IBM Cloud Pak for AIOps MustGather tool is a wrapper for running **oc** commands. With this tool, you can complete data collection tasks and view analytics on the gathered data. After you run the tool to gather data, upload the output data file that is under your **/tmp** (or another directory if you used the ``-o` option) to the IBM Support website.

The tool generates a report that shows critical data about the status and health of the cluster and installation. This report allows the user view an overview of their cluster to help spot and solve multiple problems. The report can provide basic cluster analysis to identify cluster node resource utilization, warnings, and problematic resources. The report can also provide incident analysis, such as to identify whether Pod A is in CrashLoopBackOff.

The tool supports data collection for the following types of data:

Data type collection mode

Data type	Collection mode
oc get and **oc describe** (or **kubectl**) commands	This data is collected through the primary data collection modes.
Pod logs	This data is collected through the primary data collection modes.
Application logs within pods	This data collection is supported by the CPFILES secondary data collection mode.
Command output for commands that run through pods	This data collection is supported by the MANUALCOLLECT secondary data collection mode.
Command output that run on client machines (custom script)	This data collection is supported by the CMDEXEC secondary data collection mode.
Logs or command output that run on individual cluster nodes, such as journal, systemctl, iptables	This data collection is supported by the CLUSTERNODES secondary data collection mode.

Command syntax

```
waiops-mustgather.sh -[a|b|c|l] [-k resourname] [-z modules] [-d] [-L numlines] [-C cmdexec-file:cmdexec-env] [-E cmdexec-envvar] [-T timeout-value] [-m manual-collection-param] [-n manual-collection-envvar] [-j objname] [-w manual-collection-exec-timeout] [-e extra-namespaces:extra-resources] [-x product-version/path-to-config-file] [-A compliance-param] [-P plugins] [-Z plugins-envvar] [-Q plugins-timeout] [-F cpfiles-param] [-K] [-V view-only-module] [-y] [-J] [-p] [-t] [-g] [-f] [-R] [-S] [-s] [-u node-admin-username] [-i path-to-sshkey] [-o output-directory] [-G filename-regex:pattern] [-W primary-mode-var] [-N] [-I] [-X] [-Y] [-U] [-M] [-D] [-v] [-h]
```

- For details about the command options, see Data collection modes for the IBM Cloud Pak for AIOps MustGather tool.
- For details about running a custom script with the tool, see Run a custom script to complete corrective actions

Running the tool

- Primary use cases
- Additional use cases
- Miscellaneous use cases

Primary use cases

- Installation related problems
- Application related problems
- IBM Cloud Pak for AIOps related problems
- Log anomaly detection related problems
- Metric anomaly detection related problems
- Lifecycle related problems
- Datalayer related problems
- ChatOps related problems
- Healthcheck related problems

Installation related problems

To gather data for installation related problems, run the following command:

```
waiops-mustgather.sh -O install
```

The alias **-O install** is expanded to **-cypDfd** that uses the primary comprehensive (**-c**) mode and clusternode (**d**) mode to cause the tool to gather data for installation related problems.

To gather data for general non-installation related problems, run the following command:

```
waiops-mustgather.sh -O general
```

The alias **-O general** is expanded to **-cypDf** that is used to gather data for common problems.

Overall operation related problems

To gather data for overall IBM Cloud Pak for AIOps operation related problems, run the following command:

```
waiops-mustgather.sh -O aimgr
```

To gather raw data for IBM Cloud Pak for AIOps from its various data sources, such as Kafka, Flink, ElasticSearch and Postgres, run the following command:

```
waiops-mustgather.sh -O aimgr-data
```

To learn more about what aliases are available, use the command **waiops-mustgather.sh -O .listalias**

Important: The ALIAS mode (**-O xxx**) is used to expand an alias into a predefined command. If there is a need to change or add any option, you need to revert to the conventional way of running the MustGather tool by providing all the options or flags manually. For example, if a secret YAML is needed, you need to use **-cypDfS** (plus any other options or flags).

Log anomaly detection related problems

To gather data for log anomaly related problems, run the following command:

```
waiops-mustgather.sh -O lad
```

Metric anomaly detection related problems

To gather data for metric anomaly related problems, run the following command:

```
waiops-mustgather.sh -O mad
```

Lifecycle related problems

To gather data for Lifecycle related problems, run the following command:

```
waiops-mustgather.sh -O lifecycle
```

Datalayer related problems

To gather data for Datalayer related problems, run the following command:

```
waiops-mustgather.sh -O datalayer
```

ChatOps related problems

To gather data for ChatOps related problems, run the following command:

```
waiops-mustgather.sh -O chatops
```

Healthcheck related problems

Use the healtcheck option to gather data to check whether your installation was successful or to analyze how IBM Cloud Pak for AIOps is running.

For more information about using this option, see Running the MustGather Healthcheck.

Additional use cases

- Run script to complete corrective actions
- Run baseline data collection
- Pod specific data collection
- Product specific data collection
- Installation verification
- Copying pod files and directories
- Complex command

Run a custom script to complete corrective actions

If you want to complete a **corrective action**, such as to delete a pod to restart it, you can write a script and run the following command to run the script:

```
waiops-mustgather.sh -C <script>
waiops-mustgather.sh -R -C <script>
```

Where **<script>** is the script that contains the corrective action to run.

The preceding command will only run your script. To perform data collection before running the script, you can opt for any of the primary modes."

Notes: If needed, you can run the secondary data collection modes **MANUALCOLLECT**, **EXTRA**, **CPFILES** within your custom script.

The following examples show how to trigger the MANUALCOLLECT, EXTRA, and CPFILES modes:

- To make sure the correction actions are successful, you can trigger MANUALCOLLECT to collect some data within the script

  ```
  echo "ALL##NS4PROD=aimanager##pod##teams######" >> $CMD_EXEC_RESULT_MCCFG_FILE
  ```

- You can also trigger EXTRA mode to collect namespace data within the script

  ```
  echo "openshift-insights" > $CMD_EXEC_RESULT_EXTRANS_FILE
  ```

- If you need to copy some files from some pods to make sure your script ran successfully:

  ```
  echo "NS4PROD=aimanager##^iaf-system-kafka####/opt/kafka/config/log4j.properties" >> $CMD_EXEC_RESULT_CPFILES_FILE
  ```

For an example custom script and some of the variables and functions that you can include within it, see Example: custom script.

Run baseline data collection

To run a baseline data collection, use any of the primary data collection modes, such as **-cypDf**, for example:

```
waiops-mustgather.sh -cypf -C <script> -e <namespace> -F 'NS4PROD=<prodname>,<podname-regex>,<container-name>,<filename>'
```

With the preceding command, you can run a baseline data collection that is followed by running a corrective action through a custom script. You can also check the result of the corrective action by using the **-e** option where **<namespace>** is the namespace where the custom script runs. You can also opt to copy files from the restarted pod through the **-F** option.

Notes:

- Only **-c** is the primary data collection mode. Other options are supported, but should be used together.
- If you want to run another round of data collection after your custom script runs, use a secondary data collection mode (other than **clusternodes** or **cmdexec**).

Pod specific data collection

If you have to run commands on a specific pod, but you do not know the full name of the pod, such when you know only the UUID portion, you can use the **manualcollect** mode:

```
waiops-mustgather.sh -R -m ps-out:[configfile]
```

Example (**<configfile>**):

```
@ TAG##NAMESPACE##OBJTYPE##OBJNAME##CONTAINER##EXECPREP##EXECCMD
ps-out##NS4PROD=aimanager##pod##^cp4waiops-postgres-keeper######hostname;ps -ef
```

The podname is a regex. The tool determines the full name of the pod automatically.

```
[OUTPUT - found in file 4-MANUALCOLLECT/<namespace>/<podname>.bash.exec]
cp4waiops-postgres-keeper-0
UID          PID    PPID  C STIME TTY          TIME CMD
stolon         1       0  1 Nov09 ?        02:09:11 stolon-keeper --data-dir /stolon-data
stolon        29       1  0 Nov09 ?        00:00:00 [create-template] <defunct>
stolon        54       1  0 Nov09 ?        00:15:06 postgres -D /stolon-data/postgres -c unix_socket_directories=/tmp
stolon        59      54  0 Nov09 ?        00:00:09 postgres: checkpointer
stolon        60      54  0 Nov09 ?        00:00:10 postgres: background writer
stolon        61      54  0 Nov09 ?        00:01:00 postgres: walwriter
stolon        62      54  0 Nov09 ?        00:00:19 postgres: autovacuum launcher
stolon        63      54  0 Nov09 ?        00:02:47 postgres: stats collector
stolon        64      54  0 Nov09 ?        00:00:00 postgres: logical replication launcher
stolon   2905619      54  0 05:39 ?        00:00:00 postgres: cp4aiops cp4aiops 10.254.12.46(50562) idle
stolon   2906589      54  0 05:43 ?        00:00:00 postgres: cp4aiops cp4aiops 10.254.12.46(53384) idle
stolon   2918996       0  0 06:34 ?        00:00:00 bash -c hostname;ps -ef
stolon   2919006 2918996  0 06:34 ?        00:00:00 ps -ef
```

Product specific data collection

If you need to collect product-specific data, you can run the *plugins* secondary collection mode:

```
waiops-mustgather.sh -P aimanager
```

Installation verification

If you just want to check whether an installation is successful:

```
waiops-mustgather.sh -x X -DR
```

The preceding command scans the product namespaces for missing resources or objects based on the configuration file that is specified in the **<MUSTGATHER-DIR>/missingobj/config/missingobj-waiops-<prodver>.cfg** file.

Copying pod files and directories

If you need to copy files from a pod, you can run a command similar to the following command:

```
waiops-mustgather.sh -F 'NS4PROD=aimanager##aimanager-aio-log-anomaly-detector-####/opt/ai4it/*.log*'
```

The preceding command copies files and directories that match the filename wildcard of **/opt/ai4it/*.log*** from pods that match regex **aimanager-aio-log-anomaly-detector-** in the auto-resolved IBM Cloud Pak for AIOps namespace.

Complex command

If needed, you can run a more complex command:

```
waiops-mustgather.sh -aypf -e ibm-common-services,cs-control,openshift-operators -P aimanager -G
'ALL:error|fail|exception|login' -C /tmp/test.sh -E 'TEST1=uname -a##TEST2=date' -m cqlsh:manualcollect.csv -n
'SQL="describe tables"' -F spark
```

Miscellaneous use cases

- Scale down the MustGather data collection

Scale down the MustGather data collection

If the cluster that you are working on is slow or throttles a lot, you can scale down the MustGather through the *selective* primary collection mode:

```
waiops-mustgather.sh -Xy -k deployments,sts,pods,jobs,pvc
```

The preceding command disables custom data collection (**-X**), enables YAML collection (**-y**) and runs the MustGather tool with the *selective* primary collection mode (**-k**) to collect data on resources such as **deployments**, **statefulsets**, **pods**, **jobs**, and **pvc**.

Command help

If you need help with running the MustGather tool, you can use the tool **help** command when you run the tool to view more details about the command:

```
waiops-mustgather.sh -h
```

Running the MustGather Healthcheck

You can run the IBM Cloud Pak for AIOps MustGather tool with a healthcheck option to gather data to check whether your installation was successful or to analyze how IBM Cloud Pak for AIOps is running.

Important: Ensure that your installed version of the tool is **v1.17.6** or newer. Some healthcheck data is only gathered with version **v1.17.6** and newer.

To gather data for healthcheck related problems, run the following command:

```
waiops-mustgather.sh -O healthcheck
```

This option gathers the following data for checking the current health of your IBM Cloud Pak for AIOps deployment and Red Hat OpenShift Container Platform cluster:

The healthcheck gathers the following data:

1. Node status, CPU utilization, and memory utilization
2. Red Hat OpenShift etcd status
3. Cluster service version (CSV) phase for IBM Cloud Pak for AIOps CSVs
4. CustomResourceDefinitions (CRDs) phase/status and version for IBM Cloud Pak for AIOps projects
5. Persistent volumes and persistent volume claims (PVC) status
6. Deployments, statefulsets, pods, daemonsets, and jobs statuses
7. Key missing resources
8. Basic compliance check results

After the tool finishes gathering data, review the **0-README-FIRST/SUMMARY.log** log file.

Node status, CPU utilization, and memory utilization

For the list of all nodes in the cluster with their respective status, CPU utilization, and memory utilization, review the **HEALTH REPORT OF NODES** section of the log.

This report shows the total CPU and MEMORY available in the cluster, and the CPU and memory request and limit of individual nodes. This report also indicates a warning when the CPU or memory utlization exceeds 95%.

The following log example shows a node health report for a IBM Cloud Pak for AIOps deployment:

```
================================================
HEALTH REPORT OF NODES
================================================
NODENAME                                          CPU_COUNT     CPU_USAGE                          MEM_SIZE
MEM_USAGE                          AGE      ROLES                STATUS
master0.waiops42b.mysite.ibm.com                  8             top=14% req=23% limit=0%           16015816Ki
top=66% req=51% limit=5%           39d      control-plane,master Ready
master1.waiops42b.mysite.ibm.com                  8             top=18% req=27% limit=0%           16015816Ki
top=84% req=60% limit=5%           39d      control-plane,master Ready
master2.waiops42b.mysite.ibm.com                  8             top=29% req=26% limit=0%           16015808Ki
top=90% req=61% limit=5%           39d      control-plane,master Ready
worker0.waiops42b.mysite.ibm.com                  16            top=14% req=68% limit=145%         48270036Ki
top=51% req=81% limit=98%          39d      worker               Ready
worker1.waiops42b.mysite.ibm.com                  16            top=6% req=78% limit=196%          48270036Ki
top=35% req=68% limit=93%          39d      worker               Ready
worker10.waiops42b.mysite.ibm.com                 16            top=7% req=76% limit=188%          48270036Ki
top=26% req=70% limit=114%         39d      worker               Ready
worker11.waiops42b.mysite.ibm.com                 16            top=4% req=78% limit=116%          48270036Ki
top=23% req=68% limit=71%          39d      worker               Ready
worker12.waiops42b.mysite.ibm.com                 16            top=13% req=91% limit=309%         48270036Ki
top=42% req=81% limit=149%         39d      worker               Ready
worker13.waiops42b.mysite.ibm.com                 16            top=15% req=56% limit=89%          48270036Ki
top=59% req=81% limit=78%          39d      worker               Ready
```

```
worker14.waiops42b.mysite.ibm.com                              16          top=6% req=70% limit=151%        48270036Ki
top=24% req=70% limit=88%        39d      worker                Ready
worker15.waiops42b.mysite.ibm.com                              16          top=6% req=73% limit=229%        48270036Ki
top=38% req=74% limit=107%       39d      worker                Ready
worker16.waiops42b.mysite.ibm.com                              16          top=4% req=72% limit=133%        48270036Ki
top=29% req=60% limit=80%        39d      worker                Ready
worker17.waiops42b.mysite.ibm.com                              16          top=8% req=74% limit=148%        48270036Ki
top=42% req=80% limit=97%        39d      worker                Ready
worker2.waiops42b.mysite.ibm.com                               16          top=10% req=70% limit=189%       48270036Ki
top=30% req=68% limit=92%        39d      worker                Ready
worker3.waiops42b.mysite.ibm.com                               16          top=6% req=79% limit=193%        48270036Ki
top=37% req=76% limit=91%        39d      worker                Ready
worker4.waiops42b.mysite.ibm.com                               16          top=9% req=73% limit=202%        48270036Ki
top=50% req=72% limit=97%        39d      worker                Ready
worker5.waiops42b.mysite.ibm.com                               16          top=4% req=98% limit=154%        48270036Ki
top=27% req=89% limit=94%        39d      worker                Ready
worker6.waiops42b.mysite.ibm.com                               16          top=5% req=78% limit=171%        48270036Ki
top=29% req=65% limit=79%        39d      worker                Ready
worker7.waiops42b.mysite.ibm.com                               16          top=6% req=75% limit=134%        48270036Ki
top=29% req=61% limit=72%        39d      worker                Ready
worker8.waiops42b.mysite.ibm.com                               16          top=6% req=91% limit=223%        48270036Ki
top=38% req=61% limit=93%        39d      worker                Ready
worker9.waiops42b.mysite.ibm.com                               16          top=3% req=16% limit=25%         48270036Ki
top=15% req=17% limit=15%        39d      worker                Ready
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
TOTAL                                                          312                                           938.91GB
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------

>>> [TOTAL OK] Running = 21
>>> [GRAND TOTAL] Node = 21
>>> [CPU WARNING >=95%]
worker5.waiops42b.mysite.ibm.com: req=98%
[Sun Nov 12 20:53:25 PST 2023] All nodes are up and running!
```

Red Hat OpenShift etcd status

For the Red Hat OpenShift etcd status, review the **HEALTH AND STATUS REPORT OF ETCD** section of the log.

The following log example shows a health and status report of ETCD for a IBM Cloud Pak for AIOps deployment:

```
=================================================
HEALTH AND STATUS REPORT OF ETCD
=================================================
[HEALTH]
https://10.20.3.321:2379 is healthy: successfully committed proposal: took = 7.704087ms
https://10.20.4.136:2379 is healthy: successfully committed proposal: took = 10.624038ms
https://10.20.3.246:2379 is healthy: successfully committed proposal: took = 10.661189ms

[MEMBER]
427c4b2527db01b6, started, master1.waiops42b.mysite.ibm.com, https://10.20.3.246:2380, https://10.20.3.246:2379, false
5c7cacd390f3a1e0, started, master0.waiops42b.mysite.ibm.com, https://10.20.3.321:2380, https://10.20.3.321:2379, false
5fd7600cb4fe0ee5, started, master2.waiops42b.mysite.ibm.com, https://10.20.4.136:2380, https://10.20.4.136:2379, false

[STATUS]
+--------------------------+------------------+---------+---------+-----------+------------+-----------+------------+----
---------------+--------+
|         ENDPOINT         |        ID        | VERSION | DB SIZE | IS LEADER | IS LEARNER | RAFT TERM | RAFT INDEX |
RAFT APPLIED INDEX | ERRORS |
+--------------------------+------------------+---------+---------+-----------+------------+-----------+------------+----
---------------+--------+
| https://10.20.3.321:2379 | 5c7cacd390f3a1e0 |   3.5.9 |  452 MB |     false |      false |        25 |  123057248 |
123057248 |        |
| https://10.20.3.246:2379 | 427c4b2527db01b6 |   3.5.9 |  400 MB |     false |      false |        25 |  123057248 |
123057248 |        |
| https://10.20.4.136:2379 | 5fd7600cb4fe0ee5 |   3.5.9 |  429 MB |      true |      false |        25 |  123057248 |
123057248 |        |
+--------------------------+------------------+---------+---------+-----------+------------+-----------+------------+----
---------------+--------+
```

Cluster service version (CSV) phase for IBM Cloud Pak for AIOps CSVs

For a report that lists the cluster service versions (CSVs) for a IBM Cloud Pak for AIOps project (namespace), including the CSV phase, review the **STATUS REPORT OF CLUSTER SERVICE VERSION (namespace = <namespace>)** section of the log, where **<namespace>** is the project (namespace) where IBM Cloud Pak for AIOps is installed.

This report lists the version and phase for each CSV. Within the report check the **PHASE** column and value for each CSV. The value should be **Succeeded**.

The following log example shows a cluster service version status report for a IBM Cloud Pak for AIOps deployment:

```
=================================================
STATUS REPORT OF CLUSTER SERVICE VERSION (namespace = cp4aiops)
=================================================
```

```
NAME                                          DISPLAY                                        VERSION  REPLACES
PHASE
aimanager-operator.v4.4.0                     IBM AIOps AI Manager                           4.4.0
Succeeded
aiopsedge-operator.v4.4.0                     IBM AIOps Edge                                 4.4.0
Succeeded
asm-operator.v4.4.0                           IBM Netcool Agile Service Manager              4.4.0
Succeeded
cloud-native-postgresql.v1.18.7               EDB Postgres for Kubernetes                    1.18.7   cloud-native-
postgresql.v1.18.6   Succeeded
ibm-aiops-ir-ai.v4.4.0                        IBM Watson AIOps Issue Resolution AI & Analytics  4.4.0
Succeeded
ibm-aiops-ir-core.v4.4.0                      IBM Watson AIOps Issue Resolution Core         4.4.0
Succeeded
ibm-aiops-ir-lifecycle.v4.4.0                 IBM Cloud Pak for Watson AIOps Lifecycle       4.4.0
Succeeded
ibm-aiops-orchestrator.v4.4.0                 IBM Cloud Pak for AIOps                        4.4.0
Succeeded
ibm-automation-elastic.v1.3.15                IBM Elastic                                    1.3.15
Succeeded
ibm-automation-flink.v1.3.15                  IBM Automation Foundation Flink                1.3.15
Succeeded
ibm-cert-manager-operator.v4.2.1              IBM Cert Manager                               4.2.1
Succeeded
ibm-common-service-operator.v4.3.1            IBM Cloud Pak foundational services            4.3.1
Succeeded
ibm-commonui-operator.v4.3.0                  Ibm Common UI                                  4.3.0
Succeeded
ibm-events-operator.v4.9.0                    IBM Events Operator                            4.9.0
Succeeded
ibm-iam-operator.v4.3.1                       IBM IM Operator                                4.3.1
Succeeded
ibm-mongodb-operator.v4.2.1                   IBM MongoDB Operator                           4.2.1
Succeeded
ibm-redis-cp.v1.1.4                           ibm-redis-cp-operator                          1.1.4
Succeeded
ibm-secure-tunnel-operator.v4.4.0             IBM Secure Tunnel                              4.4.0
Succeeded
ibm-watson-aiops-ui-operator.v4.4.0           IBM AIOps UI                                   4.4.0
Succeeded
ibm-zen-operator.v5.1.1                       IBM Zen Service                                5.1.1
Succeeded
operand-deployment-lifecycle-manager.v4.2.2   Operand Deployment Lifecycle Manager           4.2.2
```

CustomResourceDefinitions (CRDs) phase/status and version

For a report that lists all important CustomResourceDefinitions (CRDs) for a IBM Cloud Pak for AIOps project (namespace), including the CRD phase/status and version, review the **`STATUS REPORT OF AIOPS PRODUCT COMPONENTS`** section of the log.

This report shows the version, phase/status, and any error for each key CRD. Within the report check the **`VERSION`** column for each CRD:

- The string value **`<none>`** indicates that no version details are available. This value is not a cause for alarm.
- The string value **`N/A`** indicates that the MustGather tool is unable to detect the CRD. The healthcheck option for the MustGather scans for a predefined list of CRDs, if a CRD is installed, but shows as **`N/A`**, further investigation is required as there might be a problem with that component.

The following log example shows a product components status report for a IBM Cloud Pak for AIOps deployment:

```
================================================
STATUS REPORT OF WAIOPS PRODUCT COMPONENTS
================================================
KIND                                  NAME                          NAMESPACE                VERSION
PHASE/STATUS
AIManager                             aimanager                     cp4waiops                4.2.0
Completed
AIOpsAnalyticsOrchestrator            aiops                         cp4waiops                4.2.0
Ready
AIOpsEdge                             aiopsedge                     cp4waiops                <none>
Healthy
AIOpsUI                               aiopsui-instance              cp4waiops                4.2.0
Ready
ASMFormation                          aiops-topology                cp4waiops                2.17.0
OK
ASMFormation                          evtmanager-topology           cp4waiops-emgr           <none>
OK
automationbase                        N/A                           N/A                      N/A
N/A
automationuiconfig                    N/A                           N/A                      N/A
N/A
BaseUI                                baseui-instance               cp4waiops                4.2.0
Ready
cartridge                             N/A                           N/A                      N/A
N/A
cartridgerequirements                 N/A                           N/A                      N/A
N/A
CEMFormation                          evtmanager                    cp4waiops-emgr           2.12.7
OK
eventprocessor                        N/A                           N/A                      N/A
```

```
N/A
Installation                    ibm-cp-watson-aiops          cp4waiops                <none>
Running
IssueResolutionCore             aiops                        cp4waiops                4.2.0
Ready
LifecycleService                aiops                        cp4waiops                4.2.0
Ready
NOI                             evtmanager                   cp4waiops-emgr           1.6.10
OK
NOIFormation                    evtmanager                   cp4waiops-emgr           1.6.10
OK
noihybrid                       N/A                          N/A                      N/A
N/A
ZenService                      iaf-zen-cpdservice           cp4waiops                4.8.9
Completed
```

Persistent volumes and Persistent volume claims (PVC) status

For node disk status (disk utilization details), Persistent volume status, and Persistent volume claims status, review the following sections of the log:

- **STATUS REPORT OF PERSISTENTVOLUMECLAIMS (namespace = <namespace>)**
- **PVC UTILIZATION REPORT (namespace = <namespace>)**

The PVC utilization report lists all PVCs and their respective **df -h** output. The report also shows a warning if the disk utilization is over 95% or if the **df -h** output is not available for a PVC.

Note: **<namespace>** is the project (namespace) where IBM Cloud Pak for AIOps is installed.

The following log example shows a PERSISTENTVOLUMECLAIMS report for a IBM Cloud Pak for AIOps deployment:

```
=================================================
STATUS REPORT OF PERSISTENTVOLUMECLAIMS (namespace = cp4waiops)
=================================================
[Sun Nov 12 20:54:41 PST 2023] [cp4waiops] All PVCs seem to be operating fine!
>>> [TOTAL OK] [cp4waiops] Bound (pvc) = 40
>>> [TOTAL PVC] = 40
```

The following log example shows a PVC utilization report for a IBM Cloud Pak for AIOps deployment:

```
=================================================
PVC UTILIZATION REPORT (namespace = cp4waiops)
=================================================

[1] [ PVC = aiops-ir-analytics-spark-shared-state / PV = pvc-70d738c1-45ea-4654-9dba-8259c05e69cf ]
>> [ POD = aiops-ir-analytics-metric-spark-7b7cd75c85-s2sr6 | CONTAINER = metric-spark | VOLUME = shared-state-store |
MOUNTPATH = /opt/spark/store ]
Filesystem
Size  Used Avail Use% Mounted on
172.30.19.136:6789,172.30.165.234:6789,172.30.203.51:6789:/volumes/csi/csi-vol-12ae3465-2321-4f8c-b952-
15987bef9f3e/2f4c98f9-e8e0-4094-91d3-af3d198339a9  100G     0  100G   0% /opt/spark/store

>> [ POD = aiops-ir-analytics-spark-worker-0 | CONTAINER = spark-worker | VOLUME = shared-state-store | MOUNTPATH =
/opt/spark/store ]
Filesystem
Size  Used Avail Use% Mounted on
172.30.203.51:6789,172.30.19.136:6789,172.30.165.234:6789:/volumes/csi/csi-vol-12ae3465-2321-4f8c-b952-
15987bef9f3e/2f4c98f9-e8e0-4094-91d3-af3d198339a9  100G     0  100G   0% /opt/spark/store

>> [ POD = aiops-ir-analytics-spark-worker-1 | CONTAINER = spark-worker | VOLUME = shared-state-store | MOUNTPATH =
/opt/spark/store ]
Filesystem
Size  Used Avail Use% Mounted on
172.30.19.136:6789,172.30.165.234:6789,172.30.203.51:6789:/volumes/csi/csi-vol-12ae3465-2321-4f8c-b952-
15987bef9f3e/2f4c98f9-e8e0-4094-91d3-af3d198339a9  100G     0  100G   0% /opt/spark/store

-------------------------------------------------
[2] [ PVC = back-aiops-topology-cassandra-0 / PV = pvc-74c5f0a3-a845-427c-b333-e4eac70ffcab ]
>> [ POD = aiops-topology-cassandra-0 | CONTAINER = aiops-topology-cassandra | VOLUME = back | MOUNTPATH =
/opt/ibm/cassandra/data/backup_tar ]
Filesystem      Size  Used Avail Use% Mounted on
/dev/rbd2        49G   53M   49G   1% /opt/ibm/cassandra/data/backup_tar

-------------------------------------------------
[3] [ PVC = back-aiops-topology-cassandra-1 / PV = pvc-82d323b0-56d8-4563-bce2-7d3132a883cb ]
>> [ POD = aiops-topology-cassandra-1 | CONTAINER = aiops-topology-cassandra | VOLUME = back | MOUNTPATH =
/opt/ibm/cassandra/data/backup_tar ]
Filesystem      Size  Used Avail Use% Mounted on
/dev/rbd1        49G   53M   49G   1% /opt/ibm/cassandra/data/backup_tar

...
...
...

>> [ POD = zen-metastoredb-1 | CONTAINER = zen-metastoredb | VOLUME = user-home-mount | MOUNTPATH = /user-home ]
Filesystem
Size  Used Avail Use% Mounted on
```

```
172.30.19.136:6789,172.30.165.234:6789,172.30.203.51:6789:/volumes/csi/csi-vol-f81f5a47-351b-4c19-800f-
338905326861/9881d7c5-6e2e-46f5-a7cb-93bf70afe200   10G  160M  9.9G   2% /user-home

>> [ POD = zen-metastoredb-2 | CONTAINER = zen-metastoredb | VOLUME = user-home-mount | MOUNTPATH = /user-home ]
Filesystem
Size  Used Avail Use% Mounted on
172.30.19.136:6789,172.30.165.234:6789,172.30.203.51:6789:/volumes/csi/csi-vol-f81f5a47-351b-4c19-800f-
338905326861/9881d7c5-6e2e-46f5-a7cb-93bf70afe200   10G  160M  9.9G   2% /user-home

>> Pod [zen-pre-requisite-job-696hj] in namespace [cp4waiops] for PVC [user-home-pvc] is not in 'Running' state [STATUS =
Bound]
>> [ POD = zen-watcher-659bcf6fc9-nzv91 | CONTAINER = zen-watcher-container | VOLUME = user-home-mount | MOUNTPATH =
/user-home ]
time="2023-11-13T04:54:41Z" level=error msg="exec failed: unable to start container process: exec: \"timeout\":
executable file not found in $PATH"
command terminated with exit code 255

--------------------------------------------------
Total PVC: 40
Total PVC with 'df' output: 40
Total PVC without 'df' output: 0
```

Deployments, statefulsets, pods, daemonsets, and jobs statuses

For deployment, statefulset, pod, daemonsets, and job statuses, review the **HEALTH REPORT OF NAMESPACE = <namespace>** section of the log, where **<namespace>** is the project (namespace) where IBM Cloud Pak for AIOps is installed.

This resource analysis report includes a simple analysis of the total resources (deployments, statefulsets, daemonsetspods, pods, and jobs) that are running properly and not running properly. The report identifies any of these resources that are problematic. For problematic pods, the report shows the events for the pods.

The following log example shows a project (namespace) health report of namespace for a IBM Cloud Pak for AIOps deployment:

```
================================================
HEALTH REPORT OF NAMESPACE = cp4waiops
================================================
[Sun Nov 12 20:53:38 PST 2023] [analyzeOcAll] Analyzing 'oc get all -o wide -n cp4waiops' => /tmp/mg/waiops-modules-
12112023-205309/PROD_NAMESPACES/cp4waiops/oc_all.out
[Sun Nov 12 20:53:38 PST 2023] [analyzeOcAll] Analyzing DEPLOYMENTS [cp4waiops]
>>> [TOTAL OK] [cp4waiops] Deployment = 84
>>> [TOTAL DEPLOYMENT] = 84

[Sun Nov 12 20:53:41 PST 2023] [analyzeOcAll] Analyzing STATEFULESETS [cp4waiops]
>>> [TOTAL OK] [cp4waiops] Statefulset = 14
>>> [TOTAL STATEFULSET] = 14

[Sun Nov 12 20:53:41 PST 2023] [analyzeOcAll] Analyzing DAEMONSET [cp4waiops]
>>> [cp4waiops] There is no 'daemonset' found!
>>> [TOTAL DAEMONSET] = 0

[Sun Nov 12 20:53:41 PST 2023] [analyzeOcAll] Analyzing PODS [cp4waiops]
>>> [TOTAL OK] [cp4waiops] Running = 166 / Completed = 20
>>> [TOTAL POD] = 186

[Sun Nov 12 20:53:46 PST 2023] [analyzeOcAll] Analyzing JOBS [cp4waiops]
COUNT  JOB                                                                  READY      DURATION   STATUS
1      aimanager-aio-curator-job-28281060                                   0/1        34d        NOK
>>> [TOTAL OK] [cp4waiops] Job = 40
>>> [TOTAL NOK] [cp4waiops] Job = 1
>>> [TOTAL JOB] = 41
```

Key missing resources

For the list of missing resources, review the **REPORT OF MISSING OBJECTS (PROD_VER=<version>)** section of the log, where **<version>** is the version of IBM Cloud Pak for AIOps that is installed.

This report includes the results of a scan for the resources that are expected to exist after a successful installation. The list of resources that are scanned is configurable, and can include any resource type (deployments, statefulsets, pods, strimzipodset, and more).

When reviewing the results, review the **[SCAN RESULT]** and **[SCAN SUMMARY]** sections for details. The results list:

- Any missing resources and their expected count.
- Resources that exist, but have mismatched counts, such as when 2 are expected, but only 1 exists

For any **[Objects Not Matched]** section for a context in the report, the section lists resources that are found, but are not configured in the scan as needed or critical resources. These results are not cause for concern.

The following log example shows a missing objects report for a IBM Cloud Pak for AIOps deployment:

```
================================================
REPORT OF MISSING OBJECTS (PROD_GROUP = cp4aiops@ibm-cp-aiops / PROD_VER = 4.4.0)
================================================
NOTE(1): Please be as accurate as possible with the regex of OBJNAME. Otherwise, an object is considered found as long as
```

```
there is at least one instance!
NOTE(2): The algorithm is based on matching 'oc get' returned object name against AWK regular expression defined by
'OBJNAME'!
NOTE(3): For missing object with label [Count condition OK], it is likely to be either optional or deleted job pod. To
investigate further!

[INPUT FILE]
/tmp/waiops-mustgather/config/missingobj/4.4.0/config/missingobj.cfg

[INPUT FILE PROCESSING DETAILS]
Total config line read = 642
Total config line skipped = 344
Total config line excluded = 62
Total config line re-generated = 236
Total config line untouched = 0
Total config line (final) = 236

[SCAN RESULT]
No object is found missing!

[SCAN CONFIGURATION DETAILS]
Total configuration line = 236
Total skipped configuration = 0

[STATISTICS]
Context [cp4aiops/deployment] : Total Config = 96 / Total Config Hit = 96 / Total Config Missed = 0 / Total Objects Found
= 96 / Total Objects Matched = 96 / Total Objects Not Matched = 0
Context [cp4aiops/pod] : Total Config = 112 / Total Config Hit = 112 / Total Config Missed = 0 / Total Objects Found =
242 / Total Objects Matched = 210 / Total Objects Not Matched = 0 / Total Excluded Objects Removed = 32
Context [cp4aiops/statefulset] : Total Config = 14 / Total Config Hit = 14 / Total Config Missed = 0 / Total Objects
Found = 14 / Total Objects Matched = 14 / Total Objects Not Matched = 0
Context [cp4aiops/strimzipodset] : Total Config = 2 / Total Config Hit = 2 / Total Config Missed = 0 / Total Objects
Found = 2 / Total Objects Matched = 2 / Total Objects Not Matched = 0
Context [ibm-cert-manager/deployment] : Total Config = 4 / Total Config Hit = 4 / Total Config Missed = 0 / Total Objects
Found = 4 / Total Objects Matched = 4 / Total Objects Not Matched = 0
Context [ibm-cert-manager/pod] : Total Config = 4 / Total Config Hit = 4 / Total Config Missed = 0 / Total Objects Found
= 4 / Total Objects Matched = 4 / Total Objects Not Matched = 0
Context [ibm-licensing/deployment] : Total Config = 2 / Total Config Hit = 2 / Total Config Missed = 0 / Total Objects
Found = 2 / Total Objects Matched = 2 / Total Objects Not Matched = 0
Context [ibm-licensing/pod] : Total Config = 2 / Total Config Hit = 2 / Total Config Missed = 0 / Total Objects Found = 2
/ Total Objects Matched = 2 / Total Objects Not Matched = 0

[SCAN SUMMARY]
Total namespace(s) scanned = [3] [cp4aiops ibm-cert-manager ibm-licensing]
Total objects matched = 334
No object is found missing!
No object with mismatched count is found!
Scan for missing objects is completed.
```

Basic compliance check results

For the results from a basic compliance check, review the **[AIOPs <version>] COMPLIANCE REPORT** section of the log, where **<version>** is the version of IBM Cloud Pak for AIOps that is installed.

This report indicates the results from the compliance check, which which checks for adherence to configured compliance rules

The following log example shows a complaince report for a IBM Cloud Pak for AIOps deployment:

```
==================================================
[PRODUCT GROUP = cp4aiops@aiops-installation / WAIOPS v4.4.0] COMPLIANCE SUMMARY REPORT
==================================================
Total PROCESSED compliance rules = [2]
Total PASSED compliance rules = [2]
>>> [ocpversion,aimgr-subs-health]
Total FAILED compliance rules = [0]

[Mon Jan  8 10:27:48 CST 2024] [printSummaryLog] Displaying COMPLIANCE_REPORT.log [/tmp/mg/COMPLIANCE_REPORT.log.56430]
on screen...

==================================================
[PRODUCT GROUP = cp4aiops@aiops-installation / WAIOPS v4.4.0] COMPLIANCE REPORT
==================================================
[TAGNAME] = ocpversion
[SOURCE OBJECT] =
[TARGET OBJECT] =
[COMPLIANCE SCRIPT] = ~/waiops-mustgather/config/compliance/4.4.0/scripts/ocpversion.sh
[COMPLIANCE MESSAGE] = Openshift version [4.12.36] is supported!
[COMPLIANCE RESULT] = 0 [PASSED]

==================================================
[TAGNAME] = aimgr-subs-health
[SOURCE OBJECT] = cp4aiops/subscriptions/aimanager-operator,cp4aiops/subscriptions/aiops-ibm-common-
services,cp4aiops/subscriptions/aiopsedge-operator,cp4aiops/subscriptions/asm-operator,cp4aiops/subscriptions/cloud-
native-postgresql,cp4aiops/subscriptions/ibm-aiops-orchestrator,cp4aiops/subscriptions/ibm-automation-
elastic,cp4aiops/subscriptions/ibm-automation-flink,cp4aiops/subscriptions/ibm-events-
operator,cp4aiops/subscriptions/ibm-idp-config-ui-operator,cp4aiops/subscriptions/ibm-im-mongodb-
operator,cp4aiops/subscriptions/ibm-im-operator,cp4aiops/subscriptions/ibm-platformui-
operator,cp4aiops/subscriptions/ibm-redis-cp,cp4aiops/subscriptions/ibm-secure-tunnel-
```

```
operator,cp4aiops/subscriptions/ibm-watson-aiops-ui-operator,cp4aiops/subscriptions/ir-ai-
operator,cp4aiops/subscriptions/ir-core-operator,cp4aiops/subscriptions/ir-lifecycle-
operator,cp4aiops/subscriptions/operand-deployment-lifecycle-manager-app
[TARGET OBJECT] =
[COMPLIANCE SCRIPT] = ~/waiops-mustgather/config/compliance/4.4.0/scripts/subs-health.sh
[COMPLIANCE MESSAGE] = (1) aimanager-operator = [OK]
(2) aiops-ibm-common-services = [OK]
(3) aiopsedge-operator = [OK]
(4) asm-operator = [OK]
(5) cloud-native-postgresql = [OK]
(6) ibm-aiops-orchestrator = [OK]
(7) ibm-automation-elastic = [OK]
(8) ibm-automation-flink = [OK]
(9) ibm-events-operator = [OK]
(10) ibm-idp-config-ui-operator = [OK]
(11) ibm-im-mongodb-operator = [OK]
(12) ibm-im-operator = [OK]
(14) ibm-platformui-operator = [OK]
(15) ibm-redis-cp = [OK]
(16) ibm-secure-tunnel-operator = [OK]
(17) ibm-watson-aiops-ui-operator = [OK]
(18) ir-ai-operator = [OK]
(19) ir-core-operator = [OK]
(20) ir-lifecycle-operator = [OK]
(21) operand-deployment-lifecycle-manager-app = [OK]
[COMPLIANCE RESULT] = 0 [PASSED]
```

Data collection modes for the IBM Cloud Pak for AIOps MustGather tool

Learn more about the different commands that you can use with the IBM Cloud Pak® for AIOps MustGather tool to gather information for opening a case with IBM® Support.

There are 2 sets of data collection modes (primary and secondary). The MustGather tool has multiple (6) primary collection modes and multiple advanced modes for collecting specific sets of data.

Rules:

- Users can select zero or one primary mode.
- User can select zero, some or all secondary modes.
- User can combine one primary mode with zero, some or all secondary modes.
- Secondary modes can run independently without any primary mode
- The secondary modes selected (independently, or with any primary mode) are run with the following precedence: **clusternodes** > **cmdexec** > **manualcollect** > **extra** > **missingobj** > **compliance** > **plugins** > **cpfiles** > **snapshots**

Example commands:

```
waiops-mustgather.sh -aypD -C /tmp/cmdexec.input -m /tmp/manualcollect.csv -e cp4aiops
```

Since a primary mode **-a** is selected, all data collection that is configured for the *all* primary mode is run first. Then the data collection for the secondary modes are run in the sequence of **cmdexec** > **manualcollect** > **extra**.

```
waiops-mustgather.sh -C /tmp/cmdexec.input -m /tmp/manualcollect.csv -e cp4aiops
```

Since no primary mode is selected, the secondary modes are run in the sequence **cmdexec** > **manualcollect** > **extra**.

Primary modes

Primary mode options

Option	Description
-a	All mode, which gets all data (data volume is high).
-c	Comprehensive mode, which gets comprehensive data (data volume is moderate).
-b	Basic mode, which gets basic data (data volume is OK).
-l	Limited mode, which gets limited data (data volume is low).
-k	Selective mode, which gets data for selective resources (separate resources with comma), for example, pods,deployment,secrets (data volume is varied). To get cluster-wide (all-namespaces) data for namespace-scoped resource, such as a pod, you can use the special keyword **allns** (**pod/allns**,**ingress**,**cm/allns**). **allns** data is available in the **CLUSTER_DATA** directory. **namespaced** data is available in the **PROD_NAMESPACES** directory.
-z	Module mode, which gets data based on predefined modules separated by comma (clusterinfo productinfo storage networking installation csv pvcusage nodeinfo) (data volume is varied). To list all available modules, use **-z .listmodules**.

Secondary modes

Clusternode modes

Clusternode mode options

Option	Description
`-d`	Gets logs and data from all cluster nodes by using `oc debug` or SSH (if `-i` or `-u` is used).
`-l`	Number of lines of journal to collect from cluster nodes. The default is `2000`.

cmdexec mode

cmdexec mode options

Option	Description
`-C`	`[script file]##[environment file]` Where `[environment file]` is optional. The format of `[environment file]` is `KEY=VALUE`, which is similar to the variable definition format in shell script.
`-E`	Environment variable for the `cmdexec` script. For example, `'VAR1=VAL1##VAR2=VAL2##...'` - Enclose the entire input string with single quotation marks. For example `'VAR1=hello##VAR2=uname -a'`
`-T`	Timeout value for `cmdexec [-C]`. The default is `5m`. For example, `10s` where `s`=second; `m`=minute; `h`=hour.

Manualcollect mode

Manualcollect mode options

Option	Description
`-m`	`<manual-collection-param>` = `[TAGS]:input file` for the `getManualCollection()` function where `TAGS` is optional. The default is all config lines. - Separate multiple tags with a comma.
`-n`	Environment variables for the `manualcollect` `EXECCMD` command. For example, `TAG1~>VAR1=VAL1##VAR2=VAL2^^TAG2~>VAR1=VAL1...` - Use the symbol `^^` to separate environment variables for different tags. - Use the symbol `~>` to define an environment variable for a tag. Use the symbol `##` to separate multiple environment variables. - Enclose the entire input string with single quotation marks and each value with double quotation marks. For example, `VAR1="hello"##VAR2="uname -a"`.
`-j`	Provided string `[tag:<objname>##tag:<objname>##...]` is used to substitute the `OBJNAME` keyword under the `OBJNAME` column that is based on the `tagname` in the manual collect input configuration file, such as `tag1=pod-abc##tag2=pod-xyz`.
`-w`	Timeout value for `manualcollect [-m]`. The default is `5m`. For example, `10s` where `s`=second; `m`=minute; `h`=hour.

Extra namespace mode

Extra namespace mode options

Option	Description
`-e`	Get data for extra namespaces (`oc get all,pvc,configmap,secret,serviceaccount`, `oc describe`, `oc logs`) - Separate namespaces with a comma (`,`)

The `[:extra-resouces]` is optional and is a file that can be used to specify additional resource types, such as catalog sources, operand configs, and more). Specify one resource type per line in the file.

Missing object mode

Missing object mode options

Option	Description
`-x`	- Product version to use for `getMissingObj` function, such `-x 3.2.0` or `-x X` to auto-detect. Defaults to the product version that is detected by tool if a version is not provided Or - The absolute path for an input config file, such as `-x /tmp/myfile.cfg` for the `getMissingObj` function to use.

Compliance mode

Compliance mode options

Option	Description
`-A`	`<compliance-param> = TAGS:[input file for the getCompliance() function]` where `input file` is defaulted to `compliance/<prod-version>/compliance.cfg` (use keyword `ALL` to select all config lines. Separate multiple tags with a comma) [EXPERIMENTAL]

plugins mode

Plugins mode options

Option	Description
`-P`	Product plugin `[supported=aimanager]:action [supported=data fix]`. The default action is `data`. For example, `aimanager` or `aimanager:data` or `aimanager:fix`. - Separate plugins with a comma (`,`).
`-Z`	Environment variables for the plugins script. For example, `VAR1=VAL1##VAR2=VAL2##...` - Enclose the entire input string with single quotation marks. For example, `VAR1=hello##VAR2=uname -a`.
`-Q`	Timeout value for plugins `[-P]` The default is `15m`. For example, `10s` where `s`=second; `m`=minute; `h`=hour.

cpfiles mode

cpfiles mode options

Option	Description

Option	Description
`-F`	Copy files in a pod. `cpfiles-param` = `TAGS` or `NAMESPACE##PODNAME##CONTAINER##FILES` where multiple `TAGS` can be delimited by `@@`. `CONTAINER` is optional and `FILES` can be delimited by a colon. - `-F aimanager` refers to `cpfiles/cpfiles-<prodname>-<version>.cfg` and copies all the files configured as `TAG=aimanager`. `-F NS4PROD=aimanager##iaf-system-kafka-0####/tmp/abc.txt` is the manual command to copy a file.

Notes:

- You can use keyword `NS4PROD=<prodname>` to obtain namespace for those products automatically.
- You can use asterisk(`*`), square brackets(`[]`), question mark(`?`), and curly braces(`{}`) wildcards in `FILES`.
- Since the `tar` (needed by `oc cp`) command might not be available in the target pod or container, use this option only for text files (`oc exec/cat` is used as a secondary option).

Snapshots mode

Snapshots mode options

Option	Description
`-K`	Complete snapshots comparison.

View mode

View mode options

Option	Description
`-V`	`<view-only-module>` is a script that can be created under the `<mustgather-install-dir>/config/view/<PRODUCT_VERSION>` (precedence) or `<mustgather-install-dir>/view` directory. You can use `-V .listviews` to list down all available modules and their descriptions.

Supporting Options

Supporting options

Option	Description
`-y`	Collect YAML output (`oc get -o yaml`).
`-J`	Collect JSON output (`oc get -o json`).
`-p`	Collect logs for the previous instance of the container in a pod if it exists
`-t`	Do not collect `oc describe`.
`-g`	Do not collect pod logs (`oc logs` and `oc logs -p`).
`-f`	Collect PVC usage data (`df -h`) - Available only with options `-a`, `-c`, `-b`, `-l`, `-z` (productinfo/storage/pvcusage), `-k` (pvc) and `-e`.
`-R`	Enable report-on-screen where SUMMARY.log is printed on display.
`-S`	Explicitly collect YAML output of secret objects.
`-s`	Use SSH to collect logs and data from all cluster nodes without a common admin username across the cluster.
`-u`	Provide the admin username to used to SSH to all cluster nodes for data collection purpose. A passwordless SSH is needed for this option.
`-i`	Set the path to the SSH key. This path is needed for clusters on a cloud (experimental).
`-o`	Set the output directory. The default directory is `/tmp`.
`-G`	Use a filename regex and pattern, that is delimited by a colon, to run a grep operation. This option is applicable to pod logs, events, cpfiles, and node-related data files. - Separate patterns with double asperands (`@@`). - Separate `FILENAME_REGEX:PATTERNS` key value pairs with double hashes (`##`). - Use the keyword `ALL` as the filename regex as an instruction to run `grep` for all detected files. - Use a tilde (~) to run a piping grep. For example, `grep A <file>` | `grep B` | `grep C`. - To escape, use a tilde (~) with a backward slash (\) for the literal value. There is no need to escape by using a colon (`:`). For example, `ALL:error@@failure` or `^aimanager:error~fail` or `^aimanager:error##ALL:failure` or `ALL:error\~fail##ALL:error\/fail`
`-W`	Set global environment variable that all primary data collection modes can use for further processing. For example, `-W 'VAR1=123##VAR2=456##VAR3=789'`; this option delimits multiple variables with the `##` characters. All variables are suffixed with the string `PRIMODE_` when used internally (`VAR1` > `PRIMODE_VAR1`).
`-N`	Turn on data collection for problematic non-product namespaces. Data collection for problematic Red Hat OpenShift namespaces is always enabled.
`-I`	Enable incident reporting (experimental).
`-X`	Turn off custom data collection.
`-Y`	Auto-answer `YES` to question prompts.
`-U`	Check for updates.
`-M`	Turn of the automatic new version check and download for the tool.
`-D`	Turn on Debug mode.
`-v`	View the version details.
`-h`	View the tool help details.

Advanced tips for using the IBM Cloud Pak for AIOps MustGather tool

Learn more about advanced techniques that you can use with the IBM Cloud Pak® for AIOps MustGather tool to gather information for opening a case with IBM® Support.

Tip: Use manual collection mode to run commands

If you need to run **`oc exec`** commands on pods, you can use the manual collection (**`MANUALCOLLECT`**) mode. This mode can be configured to complete specific tasks, such as to "get all Elasticsearch indexes".

The MustGather Tool does include some default task configurations. For more information, review the **`manualcollect/<version>/mc.cfg`** file within the MustGather tool installation package.

Example command:

```
# ./waiops-mustgather.sh -R -m aimgr-indices:mc.cfg
```

Example output:

```
==================================================
[MANUALCOLLECT] OUTPUT OF EXECCMD
==================================================

[CFGLINE = lifecycle-policy##cp4waiops##pod##^aiops-ir-lifecycle-policy-registry-svc-######curl -u$(cat
$SYSTEMAUTH_BINDING_DIR/username):$(cat $SYSTEMAUTH_BINDING_DIR/password) http://localhost:$PORT/policyregistry.ibm-
netcool-prod.aiops.io/v1alpha/system/cfd95b7e-3bc7-4006-a4a8-a73a79c71255]

[FILE = /tmp/waiops-manualcollect-07112022-190154/4-MANUALCOLLECT/TAG=lifecycle-policy/cp4waiops/aiops-ir-lifecycle-
policy-registry-svc-7587b88cdb-rk594.bash.exec]

[POD = aiops-ir-lifecycle-policy-registry-svc-7587b88cdb-rk594]

Defaulted container "aiops-ir-lifecycle-policy-registry-svc" out of: aiops-ir-lifecycle-policy-registry-svc,
policyregistry-checkdb (init)

[

  {

    "tenantid": "cfd95b7e-3bc7-4006-a4a8-a73a79c71255",

    "policyid": "80301820-406a-11ed-bf66-4dfc9e76db68",

    "configuration": "{\"executionPriority\":80,\"state\":\"enabled\",\"spec\":{\"triggers\":
[{\"entityId\":\"alert\",\"triggerId\":\"aiops.ibm.com/trigger/alert-pre-update\",\"arguments\":{\"condition\":\"{{#if
(exists alert.sender.name)}} {{ alert.sender.name }} == \\\"Log Anomaly\\\" and not {{ isEmpty alert.insights }} {{else}}
false {{/if}}\"}},{\"entityId\":\"alert\",\"triggerId\":\"aiops.ibm.com/trigger/alert-pre-create\",\"arguments\":
{\"condition\":\"{{#if (exists alert.sender.name)}} {{ alert.sender.name }} == \\\"Log Anomaly\\\" and not {{ isEmpty
alert.insights }} {{else}} false {{/if}}\"}}],\"actions\":
[{\"actionId\":\"aiops.ibm.com/action/internal/partition\",\"arguments\":{\"partitionKey\":
{\"$variable\":\"alert.id\"},\"actions\":[{\"actionId\":\"aiops.ibm.com/action/internal/array/foreach\",\"arguments\":
{\"array\":{\"$variable\":\"alert.insights\"},\"condition\":\"{{element.type}} == \\\"aiops.ibm.com/insight-
type/lad/templates\\\"\",\"foreach\":[{\"actionId\":\"aiops.ibm.com/action/internal/insights/aggregate\",\"arguments\":
{\"aggregationKey\":{\"$variable\":\"alert.id\"},\"insightDetails\":{\"$variable\":\"element.details\"},\"insightType\":
{\"$variable\":\"element.type\"},\"insightId\":{\"$variable\":\"element.id\"}},\"output\":\"alert.insights.[@id=insight-
lad]\"}]}}]}}]},\"hash\":\"7d1f1b4fbb433259aaf0bf2ffc3305de06e41d98\",\"revision\":\"ee305a657825c73944447ec2ffdedd6d6519
530d\",\"hotfields\":[]}",

    "entityid": "[\"alert\",\"alert\"]",

    "metadata": "{\"labels\":{\"ibm.com/is-default\":\"true\"},\"name\":\"Aggregate log anomaly detection count
vectors\",\"description\":\"Aggregates the log message count vectors output by log anomaly events, across the event
occurences contributing to a given alert.\\n\",\"createdBy\":{\"id\":\"system\",\"type\":\"system\"},\"lastUpdatedBy\":
{\"id\":\"system\",\"type\":\"system\",\"changeDetails\":\"Created\"},\"lastUpdatedTimestamp\":\"2022-09-
30T02:49:27.969Z\",\"creationTimestamp\":\"2022-09-30T02:49:27.969Z\"}",

    "triggerid": "[\"aiops.ibm.com/trigger/alert-pre-update\",\"aiops.ibm.com/trigger/alert-pre-create\"]"
  },
...
```

Tip: Use `CPFILES` to copy files from pods

If you need to copy specific files from pods, you can use the **`CPFILES`** option.

For example, the following command uses the MustGather tool to pull the **`/conf/redis/redis.conf`** file from all pods that match regex expression **`^c-example-redis-m-`**. This example uses the following parameters:

- **`NS4PROD=`** is a special keyword that instructs the tool to auto-resolve the namespace for product **`aimanager`**.
- **`keyword`** and **`regex`** (avoid hardcoding) are used to allow the configuration to be used across different environments without the need to reconfigure.

Example command:

```
[file cpfiles/cpfiles-waiops-320.cfg in the package]
test##NS4PROD=aimanager##^c-example-redis-m-####/conf/redis/redis.conf
```

Example output:

```
# pwd
/tmp/waiops-cpfiles-01042022-173635/8-CPFILES/waiops32
```

```
# ls -l c-example-redis-m-0/
total 56
-rw-r--r-- 1 root root 51281 Apr  1 17:36 '^conf^redis^redis.conf'
-rw-r--r-- 1 root root    78 Apr  1 17:36 '^conf^redis^redis.conf.ls'
# ls -l c-example-redis-m-1
total 56
-rw-r--r-- 1 root root 51306 Apr  1 17:36 '^conf^redis^redis.conf'
-rw-r--r-- 1 root root    78 Apr  1 17:36 '^conf^redis^redis.conf.ls'
# ls -l c-example-redis-m-2
total 56
-rw-r--r-- 1 root root 51306 Apr  1 17:36 '^conf^redis^redis.conf'
-rw-r--r-- 1 root root    78 Apr  1 17:36 '^conf^redis^redis.conf.ls'
```

Tip: Run custom scripts with the MustGather tool

You can run your own script with the MustGather Tool, such as for completing corrective actions before the tool collects data. After completing the custom script actions, the tool then proceeds with the data collection.

- Custom script variable
- Custom script functions
- Example Command
- Example script
- Example script functions
- Example script results

Custom script variable

The following tables list some of the variables that you can include within a custom script:

Custom script variables

Variable	Description
`$OC_CMD`	The system command used to collect data. You can use either an `oc` or `kubectl` command.
`$OC_WHOAMI`	The current user.
`$AWK`	The `AWK` used in the tool, which can be either `awk` or `gawk` (preferred).
`$OLDIFS`	The original IFS.
`$PRODUCT_VERSION`	The product version detected during runtime (on site).
`$MUSTGATHER_DIRNAME`	The directory name where the MustGather tool is installed.
`$OUTDIR`	The directory where the output data is written.
`$PROD_NAMESPACES_REFFILE`	The product-to-namespace mapping reference file (Do not change).
`$PROD_NAMESPACES_UNIQUE_REFFILE`	The unique product namespaces reference file (Do not change).
`$PROD_NAMESPACES_INVALID_STR`	A constant string value that represents the failed `getNamespaceToProdMapping()` function.
`$CMD_EXEC_OUTPUT_DIR`	The directory where the output of `CMDEXEC` is written to.
`$CMD_EXEC_PRODUCT_NSFILE`	A copy of the original product namespaces reference file that needs to be used by the inherited `getNamespaceToProdMapping()` function.
`$CMD_EXEC_RESULT_MCCFG_FILE`	A file that the custom script can write to for running the *MANUALCOLLECT* secondary mode.
`$CMD_EXEC_RESULT_EXTRANS_FILE`	A file that the custom script can write to for running the *EXTRA* (namespace) secondary mode.
`$CMD_EXEC_RESULT_CPFILES_FILE`	A file that the custom script can write to for running the *CPFILES* secondary mode.

Custom script functions

The following tables list some of the functions that you can include within a custom script:

Custom script functions

Functions	Description
`getProdToNamespacesMapping()`	A function to retrieve product namespace through provided product name. **Syntax:** `getProdToNamespacesMapping <product-name> $CMD_EXEC_PRODUCT_NSFILE` **Returns:** - Product namespaces of the given product name or `PROD_NAMESPACES_INVALID_STR`. - Return status `0` for successful execution and `1` for a failed execution. Product namespaces: - `aimanager` = IBM Cloud Pak for AIOps - `noihybrid` = IBM Netcool Operations Insight (hybrid) - `ics` = IBM Cloud Pak foundational services
`getNamespaceToProdMapping()`	A function to retrieve product name through provided product namespace. **Syntax:** `getNamespaceToProdMapping <product-namespace> $CMD_EXEC_PRODUCT_NSFILE` **Returns:** Product name of the given product namespace.
`getResourceInstances()`	A function to retrieve a list of resource instances. **Syntax:** `getResourceInstances <resource type> <resource name regex> <resource namespace>` Where `<resource namespace>` is optional and defaults to `-all-namespaces`. **Returns:** - A list of resources that match the provided details or a blank string if the command fails or no resource is detected. - Return status `0` for successful execution and `1` for a failed execution.

Functions	Description
`getUniqueProdNamespaces()`	A function to retrieve a list of unique product namespaces. **Syntax:** `getUniqueProdNamespaces $CMD_EXEC_PRODUCT_NSFILE` **Returns:** A list of unique product namespaces or blank string if failed.
`checkUserPriv()`	A function to check whether the current user has privilege to complete specific actions (verb) on a specific resource in a specific namespace. **Syntax** `checkUserPriv <verb> <resource type> <resource namespace>` **Returns:** `0` to indicate the user has the required privileges and `1` to indicate that user does not have the required privilege. `<verb>` reference: Kubernetes API Concepts.
`isPodOk()`	A function to determine whether a pod is in `RUNNING` state and all its containers are up. **Syntax:** `isPodOk <namespace> <podname>` **Returns:** `0` for an OK pod and `1` for a pod that is not OK.
`isPodLineOk()`	A function to determine whether a pod is in RUNNING state and all its containers are up. **Syntax:** `isPodLineOk <'oc get –no-headers' output of the pod>` **Returns:** `0` for an OK pod and `1` for an pod that is not OK.

Example Command

```
waiops-mustgather.sh -R -C /tmp/elite.sh##/tmp/env.in -Y
```

The previous command uses an environment variable file `/tmp/env.in` to pass in variables into the script.

```
POD=teams
CMD=date > /tmp/date.out
```

Example script

The following code shows an example custom script:

```
NOI_NAMESPACE=$(getProdToNamespacesMapping noi $CMD_EXEC_PRODUCT_NSFILE)

if [ ${#NOI_NAMESPACE} -gt 0 ]
then
  echo "[$(date)] Detected namespace = [$NOI_NAMESPACE]"

  IMPACTGUI_POD=$(getResourceInstances pod impactgui $NOI_NAMESPACE)

  if [ ${#IMPACTGUI_POD} -gt 0 ]
  then
    echo "[$(date)] Detected IMPACT GUI pod = [$IMPACTGUI_POD]"
    oc delete pod -n $NOI_NAMESPACE $IMPACTGUI_POD

    WAITFORPOD=0

    while [ $WAITFORPOD -eq 0 ]
    do
      if isPodOk "$NOI_NAMESPACE" "$IMPACTGUI_POD"
      then
        OUTFILE=$($OC_CMD exec -n $NOI_NAMESPACE $IMPACTGUI_POD -- /opt/IBM/tivoli/impact/bin/nci_collect_logs | grep
'Netcool/Impact log file is' | $AWK '{ print $7 }')

        if [ ${#OUTFILE} -gt 0 ]
        then
          echo "[$(date)] Creating COLLECTLOG timestamp file..."
          echo "ALL##$NOI_NAMESPACE##pod##$(getResourceInstances pod impactgui $NOI_NAMESPACE)####touch /tmp/collectlog"
> $CMD_EXEC_RESULT_MCCFG_FILE

          echo "[$(date)] Detected IMPACT GUI logfile = [$OUTFILE]"
          echo "$NOI_NAMESPACE##$IMPACTGUI_POD####$OUTFILE" > $CMD_EXEC_RESULT_CPFILES_FILE

          WAITFORPOD=1
        else
          echo "[$(date)] Unable to locate IMPACT GUI logfiles!"
          exit 1
        fi
      else
        echo "[$(date)] IMPACT GUI pod is not running!"
        sleep 5
      fi
    done
  else
    echo "[$(date)] Unable to locate IMPACT GUI pod!"
    exit 1
  fi
else
  echo "[$(date)] Unable to determine NOI namespace!"
  exit 1
fi
```

Example script functions

When you are creating your custom script, you can use the following functions:

- The function `getProdToNamespacesMapping` can be used in any `CMDEXEC` script to obtain product namespaces:

  ```
  NS=$(getProdToNamespacesMapping aimanager $CMD_EXEC_PRODUCT_NSFILE)
  echo "NAMESPACE=$NS"
  ```

- The Function **getResourceInstances** can be used in any **CMDEXEC** script to obtain resource instances based on a provided regex expression such to complete a corrective action:

```
for MYPOD in $(getResourceInstances pod $POD $NS)
do
MYCMD="oc exec $MYPOD -n $NS -- bash -c '$CMD'"
echo $MYCMD
eval $MYCMD
done
```

- The **MANUALCOLLECT** mode can be used to collect data within the script, such as to ensure that the correction actions are successful:

```
echo "ALL##NS4PROD=aimanager##pod##teams####" >> $CMD_EXEC_RESULT_MCCFG_FILE
```

- The **EXTRA** mode can be used to collect namespace data within the script:

```
echo "openshift-insights" > $CMD_EXEC_RESULT_EXTRANS_FILE
```

- The **CPFILES** option can be used to copy files from pods to verify that your script ran successfully:

```
echo "NS4PROD=aimanager##^iaf-system-kafka####/opt/kafka/config/log4j.properties" >> $CMD_EXEC_RESULT_CPFILES_FILE
```

Example script results

- When the command **date > /tmp/date.out** runs in the pod:

  ```
  oc exec -it
  ```

  Output:

  ```
  aimanager-aio-chatops-teams-integrator-545bc7d4d-9w8fp -- bash -c 'ls -l /tmp; cat /tmp/date.out'
  Defaulted container "chatops-teams-integrator" out of: chatops-teams-integrator, controller-is-ready (init)
  total 12
  -rw-r--r--. 1 default root  29 Apr  2 01:00 date.out
  -rwx------. 1 root    root 701 Sep 14  2021 ks-script-vzv5jj6c
  -rwx------. 1 root    root 291 Sep 14  2021 ks-script-y5xeen0d
  Sat Apr  2 01:00:25 UTC 2022
  ```

- Through **MANUALCOLLECT**, the data of the pod that matched the regex **teams** is collected. You can verify this by running the following commands:

  1. Print the working directory:

     ```
     # pwd
     ```

     Output:

     ```
     /tmp/waiops-cmdexec-01042022-175950/4-MANUALCOLLECT/TAG=ALL/waiops32
     ```

  2. List the directory files:

     ```
     # ls -l
     ```

     Output:

     ```
     total 4
     -rw-r--r-- 1 root root 180 Apr  1 18:00 oc_pod-aimanager-aio-chatops-teams-integrator-545bc7d4d-9w8fp.out
     drwxr-xr-x 2 root root 164 Apr  1 18:00 pod
     ```

  3. List the pods files:

     ```
     total 12
     -rw-r--r-- 1 root root 1551 Apr  1 18:00 aimanager-aio-chatops-teams-integrator-545bc7d4d-9w8fp~chatops-teams-
     integrator.log
     -rw-r--r-- 1 root root 7315 Apr  1 18:00 aimanager-aio-chatops-teams-integrator-545bc7d4d-9w8fp.desc
     ```

- Through the **EXTRA** mode, data about the namespace **openshift-insights** is collected. You can verify this by running the following commands:

  1. Print the working directory:

     ```
     # pwd
     ```

     Output:

     ```
     /tmp/waiops-cmdexec-01042022-175950/5-EXTRA/openshift-insights
     ```

  2. List the directory files:

     ```
     # ls -l
     ```

     Output:

     ```
     total 28
     drwxr-xr-x 2 root root  132 Apr  1 18:00 configmaps
     drwxr-xr-x 2 root root   36 Apr  1 18:00 deployment.apps
     -rw-r--r-- 1 root root 1738 Apr  1 18:00 oc_all.out
     -rw-r--r-- 1 root root  194 Apr  1 18:00 oc_configmaps.out
     -rw-r--r-- 1 root root  328 Apr  1 18:00 oc_pod_scc.out
     -rw-r--r-- 1 root root   52 Apr  1 18:00 oc_pvc.out
     -rw-r--r-- 1 root root 1427 Apr  1 18:00 oc_secrets.out
     -rw-r--r-- 1 root root  155 Apr  1 18:00 oc_serviceaccounts.out
     drwxr-xr-x 2 root root  117 Apr  1 18:00 pod
     ```

```
drwxr-xr-x 2 root root   35 Apr  1 18:00 pvc
drwxr-xr-x 2 root root   86 Apr  1 18:00 replicaset.apps
drwxr-xr-x 2 root root 4096 Apr  1 18:00 secrets
drwxr-xr-x 2 root root   26 Apr  1 18:00 service
drwxr-xr-x 2 root root  107 Apr  1 18:00 serviceaccounts
```

- Through CPFILES, the file **/opt/kafka/config/log4j.properties** is copied from the pods that matches the regex expression **^iaf-system-kafka**. You can verify this by running the following commands:

  1. Print the working directory:

     ```
     # pwd
     ```

     Output:

     ```
     /tmp/waiops-cmdexec-01042022-175950/8-CPFILES/waiops32/iaf-system-kafka-0
     ```

  2. List the directory files:

     ```
     # ls -l
     ```

     Output:

     ```
     total 12
     -rw-r--r-- 1 root root 4674 Apr  1 18:00 '^opt^kafka^config^log4j.properties'
     -rw-r--r-- 1 root root   77 Apr  1 18:00 '^opt^kafka^config^log4j.properties.ls'
     ```

  3. View the first lines of the file to verify the file contents:

     ```
     # head ^opt^kafka^config^log4j.properties
     ```

     Output:

     ```
     # Licensed to the Apache Software Foundation (ASF) under one or more
     # contributor license agreements.  See the NOTICE file distributed with
     # this work for additional information regarding copyright ownership.
     # The ASF licenses this file to You under the Apache License, Version 2.0
     # (the "License"); you may not use this file except in compliance with
     # the License.  You may obtain a copy of the License at
     #
     #    http://www.apache.org/licenses/LICENSE-2.0
     #
     # Unless required by applicable law or agreed to in writing, software
     ```

Security bulletins and fixes

Stay informed about known security vulnerabilities and fixes for IBM Cloud Pak® for AIOps by subscribing to the security bulletins and by reviewing the list of fixed security-related vulnerabilities.

- Security bulletins
- Fixed security-related vulnerabilities in version 4.8.0
- Fixed security-related vulnerabilities in previous versions

Security bulletins

Subscribe to IBM Cloud Pak for AIOps notifications by following these steps:

1. Go to the IBM Support site.
2. Scroll to the **Support basics** section. Then, click the **Notification settings** card.
3. Log in to IBM with your IBM ID and password to continue.
4. Enter **IBM Cloud Pak for AIOps** in the **Product lookup** field. Click **Subscribe**.
5. In the **Select document types** page, select **Security bulletin** and **Fixes** > **Security Vulnerability (Sec/Int)**. You can also select any other document types that you need to keep informed about.
6. Click **Submit**.
7. To configure how you receive notifications, click **Delivery preferences** in the banner at the beginning of the page. Edit your settings as needed.

Fixed security-related vulnerabilities in version 4.8.0

Review the following tables, which lists the fixed reported security-related vulnerabilities with IBM Cloud Pak for AIOps, and any included IBM or third-party software.

Table. Fixed Common Vulnerabilities and Exposures in Version 4.8.0

CVE-ID	Issue	Description

CVE-ID	Issue	Description
CVE-2022-21221	Go fasthttp package directory traversal	Go fasthttp package could allow a remote attacker to traverse directories on the system, caused by improper validation of user requests. by the ServeFile function. An attacker could send a specially-crafted URL request containing the backslash (%5c) character to read or write arbitrary files on the system.
CVE-2022-48773	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by pointer derefs in Error Cases Of Rpcrdma_ep_create. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2023-38709	Apache HTTP Server response splitting	Apache HTTP Server is vulnerable to HTTP response splitting attacks, caused by improper input validation in the core. A remote attacker could exploit this vulnerability to inject arbitrary HTTP headers and cause the server to return a split response, once the URL is clicked. This would allow the attacker to perform further attacks, such as Web cache poisoning or cross-site scripting, and possibly obtain sensitive information.
CVE-2023-49290	JWx denial of service	JWx is vulnerable to a denial of service, caused by a flaw when p2c parameter in JWE's alg PBES2-* is set too high. By sending a specially crafted request, a remote attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2023-52492	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a NULL pointer dereference flaw in the channel unregistration function. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-0874	CoreDNS security bypass	CoreDNS could allow a remote attacker to bypass security restrictions, caused by a flaw when CD bit is set in query. By sending a specially-crafted request, an attacker could exploit this vulnerability to allow CD queries to pass.
CVE-2024-3817	HashiCorp go-getter code execution	HashiCorp go-getter could allow a remote attacker to execute arbitrary code on the system, caused by an argument injection flaw. By sending a specially crafted request, an attacker could exploit this vulnerability to execute arbitrary code on the system.
CVE-2024-6257	HashiCorp go-getter code execution	HashiCorp go-getter could allow a remote authenticated attacker to execute arbitrary code on the system, caused by improper input validation.. By using a specially crafted Git Configuration, an attacker could exploit this vulnerability to execute arbitrary code on the system.
CVE-2024-9143	OpenSSL code execution	OpenSSL could allow a remote attacker to execute arbitrary code on the system, caused by an out-of-bounds memory read or write flaw due to the use of the low-level GF(2^m) elliptic curve APIs with untrusted explicit values for the field polynomial. By sending a specially crafted request, an attacker could exploit this vulnerability to execute arbitrary code or cause the application to crash.
CVE-2024-21208		Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: Networking). Supported versions that are affected are Oracle Java SE: 8u421, 8u421-perf, 11.0.24, 17.0.12, 21.0.4, 23; Oracle GraalVM for JDK: 17.0.12, 21.0.4, 23; Oracle GraalVM Enterprise Edition: 20.3.15 and 21.3.11. Difficult to exploit vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition.
CVE-2024-21210		Vulnerability in Oracle Java SE (component: Hotspot). Supported versions that are affected are Oracle Java SE: 8u421, 8u421-perf, 11.0.24, 17.0.12, 21.0.4 and 23. Difficult to exploit vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Oracle Java SE.
CVE-2024-21217		Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: Serialization). Supported versions that are affected are Oracle Java SE: 8u421, 8u421-perf, 11.0.24, 17.0.12, 21.0.4, 23; Oracle GraalVM for JDK: 17.0.12, 21.0.4, 23; Oracle GraalVM Enterprise Edition: 20.3.15 and 21.3.11. Difficult to exploit vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition.
CVE-2024-21235		Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: Hotspot). Supported versions that are affected are Oracle Java SE: 8u421, 8u421-perf, 11.0.24, 17.0.12, 21.0.4, 23; Oracle GraalVM for JDK: 17.0.12, 21.0.4, 23; Oracle GraalVM Enterprise Edition: 20.3.15 and 21.3.11. Difficult to exploit vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition.
CVE-2024-21534	jsonpath-plus code execution	Jsonpath-plus could allow a remote attacker to execute arbitrary code on the system, caused by improper input sanitization and unsafe default usage of the vm module in Node.js. By exploiting the unsafe default usage of the vm module in Node.js, an attacker could exploit this vulnerability to inject and execute arbitrary code on the system.
CVE-2024-21664	jwx denial of service	jwx is vulnerable to a denial of service, caused by a NULL pointer dereference flaw when parsing JSON serialized payload without protected field. By sending a specially crafted payload, a remote attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-24814	OpenIDC mod_auth_openidc denial of service	OpenIDC mod_auth_openidc is vulnerable to a denial of service, caused by missing input validation on mod_auth_openidc_session_chunks cookie value. By sending a specially crafted request, a remote attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-24857	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a race condition in the conn_info_{min,max}_age_set() function in net/bluetooth. By sending a specially crafted request, a remote authenticated attacker could exploit this vulnerability to cause bluetooth connection abnormality or a denial of service condition.
CVE-2024-26851	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a bmp length out of range flaw in nf_conntrack_h323. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-26924	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a flaw with the _remove function when more than one element that share the same key. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-26976	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a flaw with when a vCPU is clearing its completion queue. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-27017	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by an error in netfilter: nft_set_pipapo. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-27062	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a race condition in the nouveau module. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.

CVE-ID	Issue	Description
CVE-2024-28122	JWx denial of service	JWx is vulnerable to a denial of service, caused by a flaw when using a compressed JWE message. By crafting a specially crafted JSON Web Encryption (JWE) token with an exceptionally high compression ratio, a remote attacker with a trusted public key could exploit this vulnerability to cause a denial of service.
CVE-2024-29025	Netty denial of service	Netty is vulnerable to a denial of service, caused by a flaw when using the HttpPostRequestDecoder to decode a form. By sending a specially crafted request, a remote attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-29857	The Bouncy Castle Crypto Package For Java denial of service	The Bouncy Castle Crypto Package For Java is vulnerable to a denial of service, caused by improper input validation. By importing an EC certificate with crafted F2m parameters, a remote attacker could exploit this vulnerability to cause excessive CPU consumption.
CVE-2024-30171	The Bouncy Castle Crypto Package For Java information disclosure	The Bouncy Castle Crypto Package For Java could allow a remote authenticated attacker to obtain sensitive information, caused by a flaw in the RSA decryption (both PKCS#1v1.5 and OAEP) feature. By utilize timing side-channel attack techniques, an attacker could exploit this vulnerability to obtain sensitive information, and use this information to launch further attacks against the affected system.
CVE-2024-30172	The Bouncy Castle Crypto Package For Java denial of service	The Bouncy Castle Crypto Package For Java is vulnerable to a denial of service, caused by an infinite loop in the Ed25519 verification code. By persuading a victim to use a specially crafted signature and public key, a remote attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-35839	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by an error related to netfilter: bridge: replace physindev with physinif in nf_bridge_info. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-35898	netfilter: nf_tables: Fix potential data-race in __nft_flowtable_type_get()	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: Fix potential data-race in __nft_flowtable_type_get() nft_unregister_flowtable_type() within nf_flow_inet_module_exit() can concurrent with __nft_flowtable_type_get() within nf_tables_newflowtable(). And thhere is not any protection when iterate over nf_tables_flowtables list in __nft_flowtable_type_get().
CVE-2024-35939	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by the leaking of pages on dma_set_decrypted() failure. A local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-37298	Gorilla web toolkit schema denial of service	Gorilla web toolkit schema is vulnerable to a denial of service, caused by a memory exhaustion flaw due to sparse slice deserialization. By sending a specially crafted request, a remote attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-37370	MIT Kerberos 5 (aka krb5) security bypass	MIT Kerberos 5 (aka krb5) could allow a remote attacker to bypass security restrictions, caused by improper access control. By sending a specially crafted request to modify the plaintext Extra Count field of a confidential GSS krb5 wrap token, an attacker could exploit this vulnerability to cause the unwrapped token to appear truncated to the application.
CVE-2024-37371	MIT Kerberos 5 (aka krb5) denial of service	MIT Kerberos 5 (aka krb5) is vulnerable to a denial of service, caused by an invalid memory reads during GSS message token handling. By sending specially crafted message tokens, a remote authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-38540	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by an out-of-bounds flaw when bnxt_qplib_alloc_init_hwq is called with hwq_attr->aux_depth != 0 and hwq_attr->aux_stride == 0. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-38541	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a stack-based buffer overflow in of_modalias(). By sending an overly long argument, a local authenticated attacker could exploit this vulnerability to overflow a buffer and cause a denial of service.
CVE-2024-38586	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a ring buffer corruption on fragmented Tx packets in r8169. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-38608	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a NULL pointer dereference in netif state handling in net/mlx5e. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-39503	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a race condition between namespace cleanup in ipset and the garbage collection of the list:set type. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-39689	Certifi python-certifi weak security	Certifi python-certifi could provide weaker than expected security, caused by the use of GLOBALTRUST root certificate. An attacker could exploit this vulnerability to launch further attacks on the system.
CVE-2024-39908	Ruby REXML denial of service	Ruby REXML is vulnerable to a denial of service, caused by an uncontrolled resource consumption flaw. By sending a specially crafted request, a remote attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-40924	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by an error related to making DPT object unshrinkable. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-40961	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a NULL pointer dereference in fib6_nh_init(). By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-40983	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a dst refcount before doing decryption. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-40984	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a NULL pointer dereference error. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-41009	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a stack-based buffer overflow in ringbuf. By sending an overly long argument, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-41042	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by stack-based buffer overflow inf_tables_api.c. A local authenticated attacker could exploit this vulnerability to cause a denial of service.

CVE-ID	Issue	Description
CVE-2024-41066	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by an skb leak. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-41092	Linux Kernel code execution	Linux Kernel could allow a local authenticated attacker to execute arbitrary code on the system, caused by a use-after-free error. An attacker could exploit this vulnerability to execute arbitrary code on the system.
CVE-2024-41093	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by the use of null object of framebuffer. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-41123	Ruby REXML denial of service	Ruby REXML is vulnerable to a denial of service, caused by improper input validation. By sending a specially crafted XML content contains many specific characters, a remote attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-41946	Ruby REXML denial of service	Ruby REXML is vulnerable to a denial of service, caused by improper input validation. By sending a specially crafted XML content, a remote attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-42070	Linux Kernel information disclosure	Linux Kernel could allow a local authenticated attacker to obtain sensitive information, caused by a flaw in Netfilter: Nf_tables. By sending a specially crafted request, an attacker could exploit this vulnerability to obtain sensitive information.
CVE-2024-42079	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a NULL pointer dereference in gfs2_log_flush. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-42244		In the Linux kernel, the following vulnerability has been resolved: USB: serial: mos7840: fix crash on resume Since commit c49cfa917025 ("USB: serial: use generic method if no alternative is provided in usb serial layer"), USB serial core calls the generic resume implementation when the driver has not provided one. This can trigger a crash on resume with mos7840 since support for multiple read URBs was added back in 2011. Specifically, both port read URBs are now submitted on resume for open ports, but the context pointer of the second URB is left set to the core rather than mos7840 port structure.
CVE-2024-42284	Linux Kernel buffer overflow	Linux Kernel is vulnerable to a buffer overflow, caused by improper bounds checking by the tipc_udp_addr2str() function. By sending a specially crafted request, a local attacker could overflow a buffer and execute arbitrary code on the system.
CVE-2024-42292	Linux Kernel information disclosure	Linux Kernel could allow a local authenticated attacker to obtain sensitive information, caused by an out-of-bounds read flaw in the zap_modalias_env() function. By sending a specially crafted request, an attacker could exploit this vulnerability to obtain sensitive information or cause a denial of service condition.
CVE-2024-42301	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by array out-of-bounds issues in sprintf. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-43398	Ruby REXML denial of service	Ruby REXML is vulnerable to a denial of service, caused by improper input validation. By using a specially crafted XML content, a remote attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-43854	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a flaw with metadata added by bio_integrity_prep is using plain kmalloc. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-43880	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a object nesting flaw. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-43889	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a divide-by-zero panic in padata_mt_helper(). By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-43892	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a race condition between multiple idr_remove() calls or between idr_alloc()/idr_replace() and idr_remove() functions. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-44935	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a NULL pointer dereference flaw in the reuseport_add_sock() function. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-44989	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a NULL pointer dereferences in xfrm real_dev. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service condition.
CVE-2024-44990	bonding: fix null pointer deref in bond_ipsec_offload_ok	In the Linux kernel, the following vulnerability has been resolved: bonding: fix null pointer deref in bond_ipsec_offload_ok We must check if there is an active slave before dereferencing the pointer.
CVE-2024-45018	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by missing initialization of extack in flow offload. By sending a specially crafted request, a local authenticated attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-45321	cpanminus code execution	cpanminus could allow a remote attacker to execute arbitrary code on the system, caused by downloading code via insecure HTTP. By sending a specially crafted request, an attacker could exploit this vulnerability to execute arbitrary code on the system.
CVE-2024-46826	ELF: fix kernel.randomize_va_space double read	In the Linux kernel, the following vulnerability has been resolved: ELF: fix kernel.randomize_va_space double read ELF loader uses "randomize_va_space" twice. It is sysctl and can change at any moment, so 2 loads could see 2 different values in theory with unpredictable consequences. Issue exactly one load for consistent value across one exec.
CVE-2024-47178	basic-auth-connect's callback uses time unsafe string comparison	basic-auth-connect is Connect's Basic Auth middleware in its own module. basic-auth-connect < 1.1.0 uses a timing-unsafe equality comparison that can leak timing information. This issue has been fixed in basic-auth-connect 1.1.0.
CVE-2024-47668	Linux Kernel denial of service	Linux Kernel is vulnerable to a denial of service, caused by a rare race in __genradix_ptr_alloc(). By sending a specially crafted request, a local attacker could exploit this vulnerability to cause a denial of service.
CVE-2024-49761	Ruby REXML denial of service	Ruby REXML is vulnerable to a denial of service, caused by a regular expression denial of service (ReDoS) flaw. By sending a specially crafted regex input, a remote attacker could exploit this vulnerability to cause a denial of service condition.

CVE-ID	Issue	Description
CVE-2024-49766	Werkzeug safe_join not safe on Windows	Werkzeug is a Web Server Gateway Interface web application library. On Python < 3.11 on Windows, os.path.isabs() does not catch UNC paths like //server/share. Werkzeug's safe_join() relies on this check, and so can produce a path that is not safe, potentially allowing unintended access to data. Applications using Python >= 3.11, or not using Windows, are not vulnerable. Werkzeug version 3.0.6 contains a patch.
CVE-2024-49767	Werkzeug possible resource exhaustion when parsing file data in forms	Werkzeug is a Web Server Gateway Interface web application library. Applications using `werkzeug.formparser.MultiPartParser` corresponding to a version of Werkzeug prior to 3.0.6 to parse `multipart/form-data` requests (e.g. all flask applications) are vulnerable to a relatively simple but effective resource exhaustion (denial of service) attack. A specifically crafted form submission request can cause the parser to allocate and block 3 to 8 times the upload size in main memory. There is no upper limit; a single upload at 1 Gbit/s can exhaust 32 GB of RAM in less than 60 seconds. Werkzeug version 3.0.6 fixes this issue.
CVE-2024-49768	Waitress has request processing race condition in HTTP pipelining with invalid first request	Waitress is a Web Server Gateway Interface server for Python 2 and 3. A remote client may send a request that is exactly recv_bytes (defaults to 8192) long, followed by a secondary request using HTTP pipelining. When request lookahead is disabled (default) we won't read any more requests, and when the first request fails due to a parsing error, we simply close the connection. However when request lookahead is enabled, it is possible to process and receive the first request, start sending the error message back to the client while we read the next request and queue it. This will allow the secondary request to be serviced by the worker thread while the connection should be closed. Waitress 3.0.1 fixes the race condition. As a workaround, disable channel_request_lookahead, this is set to 0 by default disabling this feature.
CVE-2024-49769	Pylons Project Waitress denial of service	Pylons Project Waitress is vulnerable to a denial of service, caused by improper error handling when a socket that no longer exists. By sending a specially crafted request, a remote attacker could exploit this vulnerability to cause a high CPU usage, and results in a denial of service condition.
CVE-2024-52303	aio-libs aiohttp denial of service	aio-libs aiohttp is vulnerable to a denial of service, caused by a memory leak when middlewares are used. By sending a specially crafted request, a remote attacker could exploit this vulnerability to exhaust available memory resources on the server, and results in a denial of service condition.

Fixed security-related vulnerabilities in previous versions

Review the following documentation, which includes the list of fixed reported security-related vulnerabilities in previous versions of IBM Cloud Pak for AIOps:

- Fixed security-related vulnerabilities in version 4.7.1
- Fixed security-related vulnerabilities in version 4.7.0
- Fixed security-related vulnerabilities in version 4.6.1
- Fixed security-related vulnerabilities in version 4.6.0
- Fixed security-related vulnerabilities in version 4.5.1
- Fixed security-related vulnerabilities in version 4.5.0
- Fixed security-related vulnerabilities in version 4.4.1
- Fixed security-related vulnerabilities in version 4.4.0
- Fixed security-related vulnerabilities in version 4.3.0
- Fixed security-related vulnerabilities in version 4.2.1
- Fixed security-related vulnerabilities in version 4.2.0
- Fixed security-related vulnerabilities in version 4.1.2
- Fixed security-related vulnerabilities in version 4.1.1
- Fixed security-related vulnerabilities in version 4.1.0

Fixed Support originated issues in IBM Cloud Pak for AIOps 4.8.0

Review the following list of fixed issues that were reported through IBM Support cases.

Fixed Support originated issues

Fixed install and upgrade issues

Fixed issue	Problem description
Installing - Incorrect registry configuration can break mirroring for offline installs	TBD
ELK Integration	TBD
Instana connector - List of services entity type metrics is not retrieved	Integration does not query all pages with the API
Netcool Connector - Mis-matched events between Netcool and Cloud Pak for AIOps	The integration connector thread stops running, while the pod still runs with an *Unknown* status being shown in the UI.
Netcool Connector - Performance issue with queries (mapping JSON payload)	TBD
Netcool connector - Missing alerts and parsing issues	TBD
Netcool Connector - Missing alerts	TBD
Netcool Connector - PVC capacity overflow	TBD
Netcool Connector - PVC capacity Overflow	TBD

Fixed issue	Problem description
event filtering	TBD
event filtering not working	TBD
cp4waiops-eventprocessor-eve-29ee-ep-taskmanager pods continuously restarting	TBD
Infrastructure Automation - Synchronization with Azure cloud provider does not work	TBD
Infrastructure Automation - Service provisioning stopped working	TBD
Messaging keystore password with special characters fails appliance setup	TBD
Infrastructure Automation - Memcached pod failed as certificate as expired	IBM Cloud Pak for AIOps is up and running but Infrastructure Automation is down, with the login page showing `504 Gateway Time-out`.
Incorrect ServiceMonitor resource crashes monitoring	When Red Hat OpenShift is configured for user monitoring to fetch metrics, the Prometheus monitoring ended in a CrashLoopBackOff. This issue is related to a reported Red Hat issue: https://issues.redhat.com/browse/OCPBUGS-23252.
Temporal and Seasonality trainings failed when there were 0 alerts in the database	When 0 alerts are in the database, training failed and models were not created. As a workaround, verify that you have at least three months of historical event data so the algorithm can learn alert patterns and discover groups.

Fixed Support originated issues in previous versions

Review the following documentation, which includes the list of fixed Support originated issues in previous versions of IBM Cloud Pak for AIOps:

- Fixed Support originated issues in version 4.6.1
- Fixed Support originated issues in version 4.6.0

Fixed security vulnerabilities

For the list of fixed security-related vulnerabilities, see Security bulletins and fixes.

Known issues and limitations

For more information about current known issues in IBM Cloud Pak® for AIOps, see Known issues and limitations.

Notices

These legal notices pertain to the IBM Cloud Pak® for AIOps product and its documentation.

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any third party product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information here; these changes will be incorporated in new editions of the publication. IBM may make improvements or changes, or both, in the products or the programs, or both, described in this publication at any time without notice.

Any references in this information to third party websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials on those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information, which has been exchanged, should contact:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning third party products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to third party products. Questions on the capabilities of third party products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE: This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM®, the IBM logo, and ibm.com® are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at *Copyright and trademark information* at www.ibm.com/legal/copytrade.shtml.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

ITIL is a Registered Trade Mark of AXELOS Limited.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and its affiliates, or both.

Red Hat®, JBoss®, OpenShift®, Fedora®, Hibernate®, Ansible®, CloudForms®, RHCA®, RHCE®, RHCSA®, Ceph®, and Gluster® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Terms and conditions

Personal use

You can reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside of your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the preceding instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Licensing

For full details, see the IBM License Information documents page.

Glossary

This glossary provides terms and definitions for IBM Cloud Pak® for AIOps.

The following cross-references are used in this glossary:

- *See* refers you from a nonpreferred term to the preferred term or from an abbreviation to the spelled-out form.
- *See also* refers you to a related or contrasting term.

While IBM values the use of inclusive language, terms that are outside of IBM's direct influence are sometimes required for the sake of maintaining user understanding. As other industry leaders join IBM in embracing the use of inclusive language, IBM will continue to update the documentation to reflect those changes.

A

accelerator

A set of assets that enable customers to understand core concepts and implement solutions.

AI model

In a machine learning context, a set of functions and algorithms that are trained and tested on a data set to provide predictions or decisions. See also training definition training definition.

alert

A record of an event indicating a fault in the managed environment.

anomaly

A deviation from the expected baseline behaviors.

application

One or more computer programs or software components that provide a function in direct support of a specific business process or processes.

application mapping

Used to expose the on-premises IBM Cloud Pak for AIOps instance to the public cloud. After the application mapping is created, IBM Cloud Pak for AIOps can be accessed from Slack.

auto-remediation

The process of resolving an issue with minimal or no human intervention.

B

boot node

The OpenShift cluster node that is used for running installation, configuration, node scaling, and cluster updates.

C

capability

The core functions of a service. For example, the XYZ service provides "monitoring" capability.

CD

See continuous delivery.

change risk

The risk associated with implementing any type of change, whether it is to code, configurations, or data. See also risk score.

CI

See continuous integration.

connector

The means by which a data source is connected to a product or service. See also integration.

continuous delivery (CD)

A software development practice that employs techniques such as continuous testing, continuous integration, and continuous deployment so that new features and fixes are packaged and deployed rapidly and at low risk to test environments and then to customers.

continuous integration (CI)

A software development practice where members of a team integrate their work frequently so that there are multiple integrations each day. Integrations are verified by an automated build to detect integration errors as quickly as possible.

coverage

A qualitative determination of the AI's ability to produce insights from your incoming data flow as a result of it's training. See also training definition.

CRD

See custom resource definition.

custom resource

An instance of a custom resource definition. See also custom resource definition.

custom resource definition (CRD)

A customizable YAML file that defines a logically related group of objects in a cluster. Custom resource definitions enable a custom resource to be used like any native Kubernetes object in the cluster. See also custom resource.

D

data set

Data that is extracted and organized from data streams. Data sets serve as inputs for pipelines to train, retrain, or tune models.

data stream

A set of processes that convert extracted data into a normalized data set.

E

edge

The relationship, or link, between resources. See also hop, seed.

error

The fault in an event that causes a failure.

event

A change in state of a service or system.

F

failure

The surfacing of an error to the user.

fault

An error in the internal state of a component in a system.

field mapping

The assignment of values from one integration type to a standard set of values. When one maps source fields, one replaces or supplements those values with new or different values.

H

hop

A step along a single edge from one resource to another. See also edge, seed.

I

incident

An unplanned interruption that causes, may cause, or reduces the quality of an IT service. See also incident.

integration

Any independent application that provides additional capabilities to an IBM service or base product. See also connector.

issue

A cause or potential cause of one or more recurring or similar tickets or incidents.

L

log

Data output from an application or service that describes a transaction or state occurring in that application or service.

M

mean time between failures (MTBF)

The average time between failures.

mean time to failure (MTTF)

The average amount of time before a system lapses into a failed state.

mean time to detect (MTTD)

The average amount of time it takes to identify a failure.

mean time to resolve (MTTR)

The total time period from the start of a failure to when the failure resolves and operations resume.

O

observer

A service that extracts resource information and provides topology data for integrations and localization and blast radius information for events published to ChatOps integrations.

observer job

An automated scheduled task that retrieves topological details for a target system.

operand

The service component used by an operator to perform actions.

operator

A system design that simulates user behavior by linking a controller to custom resources, simulating the act of observing differences in states then acting on them.

P

pipeline

A set of iterative processes that transforms data into insights. For example, retraining pipelines take additional training data sets and improves existing models based on new insights.

proactive channel

A ChatOps channel that is used for change risk notifications. See also reactive channel.

probable cause

A rank-based indicator of causal likelihood that is based on the analysis of alerts and associated topology within the context of an incident.

problem

See issue.

R

reactive channel

A ChatOps channel that is used for outgoing incidents and alerts. See also proactive channel.

resource

A physical or logical component that can be provisioned or reserved for an application or service instance. Examples of resources can include storage, processors, memory, clusters, and VMs.

risk score

A value derived to describe risk across several dimensions associated with change. See also change risk.

RPO (Recovery point objective)

The application loss tolerance, such as the amount of data that can be lost before significant harm to the application occurs.

RTA (Recovery Time Actual)

The actual time that is required for an application to recover completely.

RTO (Recovery Time Objective)

The time that an application can be down without causing significant damage. RTA should be less than RTO.

S

secure tunnel

A feature to package TCP requests and responses, encrypt them with HTTPS, and transport the payloads between IBM Cloud Pak for AIOps and Slack. With Secure Tunnel, you can integrate IBM Cloud Pak for AIOps with Slack without creating firewall rules and policies.

seed

A single resource that serves as the starting point of a topology. Once defined, the topology view is expanded one hop at a time. See also edge, hop.

service

In IBM Cloud Pak solutions, a set of capabilities that provide functionality to a product.

service catalog

A curated repository of services that users can discover and instantiate for use within their organization's applications.

incident

A collection of insights derived from different data sources (logs, events, and alerts) that represent the products' determination of an incident. Incidents build understanding and drive remediation.

T

ticket

A formal record of identified issues or requests that are created against an item and assigned to appropriate users to resolve those issues or complete the requests.

topology

An arrangement of specific, interconnected assets and resources within a network or application.

topology template

A defined topology pattern that is used to dynamically generate topologies based on matching conditions from previously established topologies in the topology database.

training definition

The established model that serves as the baseline for your data to be measured against. For example, your AI model for log anomalies acts as the training definition for all of your log anomaly integrations. See also AI model, coverage.

tunnel worker

Server side of a TCP-over-HTTPS tunnel. It is developed based on OSS project, and is installed in the IBM Cloud Pak for AIOps cluster.

tunnel connection

Used to manage IBM Cloud Pak for AIOps instance that you want to expose. If you want to expose on-premises IBM Cloud Pak for AIOps for Slack integration, first you need to create a tunnel connection.

tunnel connector

Client side of a TCP-over-HTTPS tunnel. It is developed based on OSS project, and can be installed in an OpenShift cluster. Ensure that the OpenShift cluster can be accessed from the public network, and the cluster domain certificate is signed.