Red Hat OpenShift Service on AWS 4 Support

Red Hat OpenShift Service on AWS Support.
Legal Notice

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license (“CC-BY-SA”). An explanation of CC-BY-SA is available at http://creativecommons.org/licenses/by-sa/3.0/. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Offers cluster administrators tools for gathering data for your cluster, monitoring, and troubleshooting.
# Table of Contents

## CHAPTER 1. SUPPORT OVERVIEW
- 1.1. GET SUPPORT ................................................................. 4
- 1.2. REMOTE HEALTH MONITORING ISSUES .................. 4
- 1.3. TROUBLESHOOTING ISSUES ........................................ 4

## CHAPTER 2. MANAGING YOUR CLUSTER RESOURCES
- 2.1. INTERACTING WITH YOUR CLUSTER RESOURCES ......... 6

## CHAPTER 3. GETTING SUPPORT
- 3.1. GETTING SUPPORT .......................................................... 7
- 3.2. ABOUT THE RED HAT KNOWLEDGEBASE ..................... 7
- 3.3. SEARCHING THE RED HAT KNOWLEDGEBASE ............... 7
- 3.4. SUBMITTING A SUPPORT CASE ...................................... 8
- 3.5. ADDITIONAL RESOURCES ................................................ 9

## CHAPTER 4. REMOTE HEALTH MONITORING WITH CONNECTED CLUSTERS
- 4.1. ABOUT REMOTE HEALTH MONITORING ...................... 10
  - 4.1.1. About Telemetry ....................................................... 11
    - 4.1.1.1. Information collected by Telemetry .................... 11
      - 4.1.1.1.1. System information .................................... 11
      - 4.1.1.1.2. Sizing Information ................................... 11
      - 4.1.1.1.3. Usage information ..................................... 12
    - 4.1.1.2. User Telemetry ............................................... 12
  - 4.1.2. About the Insights Operator .................................... 12
  - 4.1.2.1. Information collected by the Insights Operator ....... 13
  - 4.1.3. Understanding Telemetry and Insights Operator data flow 13
  - 4.1.4. Additional details about how remote health monitoring data is used 14
- 4.2. SHOWING DATA COLLECTED BY REMOTE HEALTH MONITORING 14
  - 4.2.1. Showing data collected by Telemetry ...................... 14
- 4.3. USING INSIGHTS TO IDENTIFY ISSUES WITH YOUR CLUSTER 18
  - 4.3.1. About Red Hat Insights Advisor for Red Hat OpenShift Service on AWS 18
  - 4.3.2. Understanding Insights Advisor recommendations ....... 18
  - 4.3.3. Displaying potential issues with your cluster .......... 19
  - 4.3.4. Displaying all Insights Advisor recommendations ....... 19
  - 4.3.5. Disabling Insights Advisor recommendations .......... 20
  - 4.3.6. Enabling a previously disabled Insights Advisor recommendation 21
  - 4.3.7. Displaying the Insights status in the web console .... 21
- 4.4. USING THE INSIGHTS OPERATOR .................................... 22
  - 4.4.1. Understanding Insights Operator alerts .................. 22

## CHAPTER 5. SUMMARIZING CLUSTER SPECIFICATIONS
- 5.1. SUMMARIZING CLUSTER SPECIFICATIONS BY USING A CLUSTER VERSION OBJECT 23

## CHAPTER 6. TROUBLESHOOTING
- 6.1. TROUBLESHOOTING RED HAT OPENSHIFT SERVICE ON AWS INSTALLATIONS 25
  - 6.1.1. Troubleshooting troubleshooting .............................. 25
  - 6.1.1.1. Inspect install or uninstall logs .......................... 25
  - 6.1.1.2. Verify your AWS account permissions for clusters without STS 25
  - 6.1.1.3. Verify your AWS account and quota ..................... 25
  - 6.1.1.4. AWS notification emails .................................... 26
- 6.2. VERIFYING NODE HEALTH ............................................. 26
  - 6.2.1. Reviewing node status, resource usage, and configuration 26
6.3. TROUBLESHOOTING OPERATOR ISSUES
   6.3.1. Operator subscription condition types
   6.3.2. Viewing Operator subscription status by using the CLI
   6.3.3. Viewing Operator catalog source status by using the CLI
   6.3.4. Querying Operator pod status
   6.3.5. Gathering Operator logs
6.4. INVESTIGATING POD ISSUES
   6.4.1. Understanding pod error states
   6.4.2. Reviewing pod status
   6.4.3. Inspecting pod and container logs
   6.4.4. Accessing running pods
   6.4.5. Starting debug pods with root access
   6.4.6. Copying files to and from pods and containers
6.5. TROUBLESHOOTING STORAGE ISSUES
   6.5.1. Resolving multi-attach errors
6.6. INVESTIGATING MONITORING ISSUES
   6.6.1. Investigating why user-defined project metrics are unavailable
   6.6.2. Determining why Prometheus is consuming a lot of disk space
6.7. DIAGNOSING OPENSHIFT CLI (OC) ISSUES
   6.7.1. Understanding OpenShift CLI (oc) log levels
   6.7.2. Specifying OpenShift CLI (oc) log levels
6.8. TROUBLESHOOTING EXPIRED TOKENS
   6.8.1. Troubleshooting expired offline access tokens
6.9. TROUBLESHOOTING IAM ROLES
   6.9.1. Resolving issues with ocm-roles and user-role IAM resources
      6.9.1.1. Creating an ocm-role IAM role
      6.9.1.2. Creating a user-role IAM role
      6.9.1.3. Linking your AWS account
      6.9.1.4. Associating multiple AWS accounts with your Red Hat organization
6.10. TROUBLESHOOTING CLUSTER DEPLOYMENTS
   6.10.1. Obtaining information on a failed cluster
   6.10.2. Failing to create a cluster with an osdCcsAdmin error
   6.10.3. Creating the Elastic Load Balancing (ELB) service-linked role
   6.10.4. Repairing a cluster that cannot be deleted
6.11. RED HAT OPENSHIFT SERVICE ON AWS MANAGED RESOURCES
   6.11.1. Overview
   6.11.2. Hive managed resources
   6.11.3. Red Hat OpenShift Service on AWS add-on namespaces
   6.11.4. Red Hat OpenShift Service on AWS validating webhooks
CHAPTER 1. SUPPORT OVERVIEW

Red Hat offers cluster administrators tools for gathering data for your cluster, monitoring, and troubleshooting.

1.1. GET SUPPORT

Get support: Visit the Red Hat Customer Portal to review knowledge base articles, submit a support case, and review additional product documentation and resources.

1.2. REMOTE HEALTH MONITORING ISSUES

Remote health monitoring issues: Red Hat OpenShift Service on AWS collects telemetry and configuration data about your cluster and reports it to Red Hat by using the Telemeter Client and the Insights Operator. Red Hat uses this data to understand and resolve issues in connected cluster. Red Hat OpenShift Service on AWS collects data and monitors health using the following:

- **Telemetry:** The Telemetry Client gathers and uploads the metrics values to Red Hat every four minutes and thirty seconds. Red Hat uses this data to:
  - Monitor the clusters.
  - Roll out Red Hat OpenShift Service on AWS upgrades.
  - Improve the upgrade experience.

- **Insight Operator:** By default, Red Hat OpenShift Service on AWS installs and enables the Insight Operator, which reports configuration and component failure status every two hours. The Insight Operator helps to:
  - Identify potential cluster issues proactively.
  - Provide a solution and preventive action in Red Hat OpenShift Cluster Manager.

You can review telemetry information.

If you have enabled remote health reporting, Use Insights to identify issues. You can optionally disable remote health reporting.

1.3. TROUBLESHOOTING ISSUES

A cluster administrator can monitor and troubleshoot the following Red Hat OpenShift Service on AWS component issues:

- **Node issues:** A cluster administrator can verify and troubleshoot node-related issues by reviewing the status, resource usage, and configuration of a node. You can query the following:
  - Kubelet’s status on a node.
  - Cluster node journal logs.

- **Operator issues:** A cluster administrator can do the following to resolve Operator issues:
  - Verify Operator subscription status.
  - Check Operator pod health.
• Gather Operator logs.

• **Pod issues**: A cluster administrator can troubleshoot pod-related issues by reviewing the status of a pod and completing the following:
  
  • Review pod and container logs.
  
  • Start debug pods with root access.

• **Storage issues**: A multi-attach storage error occurs when the mounting volume on a new node is not possible because the failed node cannot unmount the attached volume. A cluster administrator can do the following to resolve multi-attach storage issues:
  
  • Enable multiple attachments by using RWX volumes.
  
  • Recover or delete the failed node when using an RWO volume.

• **Monitoring issues**: A cluster administrator can follow the procedures on the troubleshooting page for monitoring. If the metrics for your user-defined projects are unavailable or if Prometheus is consuming a lot of disk space, check the following:
  
  • Investigate why user-defined metrics are unavailable.
  
  • Determine why Prometheus is consuming a lot of disk space.

• **Logging issues**: A cluster administrator can follow the procedures on the troubleshooting page for OpenShift Logging issues. Check the following to resolve logging issues:
  
  • Status of the Logging Operator.
  
  • Status of the Log store.
  
  • Troubleshooting logging alerts.
  
  • Information about your OpenShift logging environment using `oc adm must-gather` command.

• **OpenShift CLI (oc) issues**: Investigate OpenShift CLI (oc) issues by increasing the log level.
CHAPTER 2. MANAGING YOUR CLUSTER RESOURCES

You can apply global configuration options in Red Hat OpenShift Service on AWS. Operators apply these configuration settings across the cluster.

2.1. INTERACTING WITH YOUR CLUSTER RESOURCES

You can interact with cluster resources by using the OpenShift CLI (oc) tool in Red Hat OpenShift Service on AWS. The cluster resources that you see after running the oc api-resources command can be edited.

Prerequisites

- You have access to the cluster as a user with the dedicated-admin role.
- You have access to the web console or you have installed the oc CLI tool.

Procedure

1. To see which configuration Operators have been applied, run the following command:

   ```
   $ oc get config.openshift.io -o name | grep config.openshift.io
   ```

2. To see what cluster resources you can configure, run the following command:

   ```
   $ oc explain <resource_name>.config.openshift.io
   ```

3. To see the configuration of custom resource definition (CRD) objects in the cluster, run the following command:

   ```
   $ oc get <resource_name>.config -o yaml
   ```

4. To edit the cluster resource configuration, run the following command:

   ```
   $ oc edit <resource_name>.config -o yaml
   ```
CHAPTER 3. GETTING SUPPORT

3.1. GETTING SUPPORT

If you experience difficulty with a procedure described in this documentation, or with Red Hat OpenShift Service on AWS in general, visit the Red Hat Customer Portal. From the Customer Portal, you can:

- Search or browse through the Red Hat Knowledgebase of articles and solutions relating to Red Hat products.
- Submit a support case to Red Hat Support.
- Access other product documentation.

To identify issues with your cluster, you can use Insights in OpenShift Cluster Manager Hybrid Cloud Console. Insights provides details about issues and, if available, information on how to solve a problem.

If you have a suggestion for improving this documentation or have found an error, submit a Jira issue for the most relevant documentation component. Please provide specific details, such as the section name and Red Hat OpenShift Service on AWS version.

3.2. ABOUT THE RED HAT KNOWLEDGEBASE

The Red Hat Knowledgebase provides rich content aimed at helping you make the most of Red Hat’s products and technologies. The Red Hat Knowledgebase consists of articles, product documentation, and videos outlining best practices on installing, configuring, and using Red Hat products. In addition, you can search for solutions to known issues, each providing concise root cause descriptions and remedial steps.

3.3. SEARCHING THE RED HAT KNOWLEDGEBASE

In the event of an Red Hat OpenShift Service on AWS issue, you can perform an initial search to determine if a solution already exists within the Red Hat Knowledgebase.

Prerequisites

- You have a Red Hat Customer Portal account.

Procedure

2. Click Search.
3. In the search field, input keywords and strings relating to the problem, including:
   - Red Hat OpenShift Service on AWS components (such as etcd)
   - Related procedure (such as installation)
   - Warnings, error messages, and other outputs related to explicit failures
4. Click the Enter key.
5. Optional: Select the **Red Hat OpenShift Service on AWS** product filter.

6. Optional: Select the **Documentation** content type filter.

### 3.4. Submitting a Support Case

**Prerequisites**

- You have access to the cluster as a user with the `cluster-admin` role.
- You have access to the cluster as a user with the `dedicated-admin` role.
- You have installed the OpenShift CLI (`oc`).
- You have a Red Hat Customer Portal account.
- You have a Red Hat standard or premium Subscription.

**Procedure**


2. Click **Get support**.

3. On the **Cases** tab of the Customer Support page:
   a. Optional: Change the pre-filled account and owner, if needed.
   b. Select the appropriate category for your issue, such as **Bug or Defect**, and click **Continue**.

4. Enter the following information:
   a. In the **Summary** field, enter a concise but descriptive problem summary and further details about the symptoms being experienced, as well as your expectations.
   b. Select **Red Hat OpenShift Service on AWS** from the **Product** drop-down menu.

5. Review the list of suggested Red Hat Knowledgebase solutions for a potential match against the problem that is being reported. If the suggested articles do not address the issue, click **Continue**.

6. Review the updated list of suggested Red Hat Knowledgebase solutions for a potential match against the problem that is being reported. The list is refined as you provide more information during the case creation process. If the suggested articles do not address the issue, click **Continue**.

7. Ensure that the account information presented is as expected, and if not, amend accordingly.

8. Check that the autofilled Red Hat OpenShift Service on AWS Cluster ID is correct. If it is not, manually obtain your cluster ID.
   - To manually obtain your cluster ID using the Red Hat OpenShift Service on AWS web console:
     a. Navigate to **Home → Overview**.
     b. Find the value in the **Cluster ID** field of the **Details** section.
• Alternatively, it is possible to open a new support case through the Red Hat OpenShift Service on AWS web console and have your cluster ID autofilled.

  a. From the toolbar, navigate to (?) Help → Open Support Case.

  b. The Cluster ID value is autofilled.

• To obtain your cluster ID using the OpenShift CLI (oc), run the following command:

  
  $ oc get clusterversion -o jsonpath='{.items[].spec.clusterID}"\n"'

9. Complete the following questions where prompted and then click Continue:

  • Where are you experiencing the behavior? What environment?
  
  • When does the behavior occur? Frequency? Repeatedly? At certain times?
  
  • What information can you provide around time-frames and the business impact?

10. Upload relevant diagnostic data files and click Continue. It is recommended to include data gathered using the oc adm must-gather command as a starting point, plus any issue specific data that is not collected by that command.

11. Input relevant case management details and click Continue.

12. Preview the case details and click Submit.

3.5. ADDITIONAL RESOURCES

• For details about identifying issues with your cluster, see Using Insights to identify issues with your cluster.
CHAPTER 4. REMOTE HEALTH MONITORING WITH CONNECTED CLUSTERS

4.1. ABOUT REMOTE HEALTH MONITORING

Red Hat OpenShift Service on AWS collects telemetry and configuration data about your cluster and reports it to Red Hat by using the Telemeter Client and the Insights Operator. The data that is provided to Red Hat enables the benefits outlined in this document.

A cluster that reports data to Red Hat through Telemetry and the Insights Operator is considered a connected cluster.

Telemetry is the term that Red Hat uses to describe the information being sent to Red Hat by the Red Hat OpenShift Service on AWS Telemeter Client. Lightweight attributes are sent from connected clusters to Red Hat to enable subscription management automation, monitor the health of clusters, assist with support, and improve customer experience.

The Insights Operator gathers Red Hat OpenShift Service on AWS configuration data and sends it to Red Hat. The data is used to produce insights about potential issues that a cluster might be exposed to. These insights are communicated to cluster administrators on OpenShift Cluster Manager Hybrid Cloud Console.

More information is provided in this document about these two processes.

Telemetry and Insights Operator benefits

Telemetry and the Insights Operator enable the following benefits for end-users:

- **Enhanced identification and resolution of issues** Events that might seem normal to an end-user can be observed by Red Hat from a broader perspective across a fleet of clusters. Some issues can be more rapidly identified from this point of view and resolved without an end-user needing to open a support case or file a Jira issue.

- **Advanced release management.** Red Hat OpenShift Service on AWS offers the candidate, fast, and stable release channels, which enable you to choose an update strategy. The graduation of a release from fast to stable is dependent on the success rate of updates and on the events seen during upgrades. With the information provided by connected clusters, Red Hat can improve the quality of releases to stable channels and react more rapidly to issues found in the fast channels.

- **Targeted prioritization of new features and functionality** The data collected provides insights about which areas of Red Hat OpenShift Service on AWS are used most. With this information, Red Hat can focus on developing the new features and functionality that have the greatest impact for our customers.

- **A streamlined support experience.** You can provide a cluster ID for a connected cluster when creating a support ticket on the Red Hat Customer Portal. This enables Red Hat to deliver a streamlined support experience that is specific to your cluster, by using the connected information. This document provides more information about that enhanced support experience.

- **Predictive analytics.** The insights displayed for your cluster on OpenShift Cluster Manager Hybrid Cloud Console are enabled by the information collected from connected clusters. Red Hat is investing in applying deep learning, machine learning, and artificial intelligence
automation to help identify issues that Red Hat OpenShift Service on AWS clusters are exposed to.

On Red Hat OpenShift Service on AWS, remote health reporting is always enabled. You cannot opt out of it.

4.1.1. About Telemetry

Telemetry sends a carefully chosen subset of the cluster monitoring metrics to Red Hat. The Telemeter Client fetches the metrics values every four minutes and thirty seconds and uploads the data to Red Hat. These metrics are described in this document.

This stream of data is used by Red Hat to monitor the clusters in real-time and to react as necessary to problems that impact our customers. It also allows Red Hat to roll out Red Hat OpenShift Service on AWS upgrades to customers to minimize service impact and continuously improve the upgrade experience.

This debugging information is available to Red Hat Support and Engineering teams with the same restrictions as accessing data reported through support cases. All connected cluster information is used by Red Hat to help make Red Hat OpenShift Service on AWS better and more intuitive to use.

4.1.1.1. Information collected by Telemetry

The following information is collected by Telemetry:

4.1.1.1.1. System information

- Version information, including the Red Hat OpenShift Service on AWS cluster version and installed update details that are used to determine update version availability

- Update information, including the number of updates available per cluster, the channel and image repository used for an update, update progress information, and the number of errors that occur in an update

- The unique random identifier that is generated during an installation

- Configuration details that help Red Hat Support to provide beneficial support for customers, including node configuration at the cloud infrastructure level, hostnames, IP addresses, Kubernetes pod names, namespaces, and services

- The Red Hat OpenShift Service on AWS framework components installed in a cluster and their condition and status

- Events for all namespaces listed as "related objects" for a degraded Operator

- Information about degraded software

- Information about the validity of certificates

- The name of the provider platform that Red Hat OpenShift Service on AWS is deployed on and the data center location

4.1.1.2. Sizing Information

- Sizing information about clusters, machine types, and machines, including the number of CPU cores and the amount of RAM used for each
- The number of etcd members and the number of objects stored in the etcd cluster
- Number of application builds by build strategy type

4.1.1.3. Usage information

- Usage information about components, features, and extensions
- Usage details about Technology Previews and unsupported configurations

Telemetry does not collect identifying information such as usernames or passwords. Red Hat does not intend to collect personal information. If Red Hat discovers that personal information has been inadvertently received, Red Hat will delete such information. To the extent that any telemetry data constitutes personal data, please refer to the Red Hat Privacy Statement for more information about Red Hat’s privacy practices.

4.1.2. User Telemetry

Red Hat collects anonymized user data from your browser. This anonymized data includes what pages, features, and resource types that the user of all clusters with enabled telemetry uses.

Other considerations:

- User events are grouped as a SHA-1 hash.
- User’s IP address is saved as 0.0.0.0.
- User names and IP addresses are never saved as separate values.

Additional resources

- See Showing data collected by Telemetry for details about how to list the attributes that Telemetry gathers from Prometheus in Red Hat OpenShift Service on AWS.
- See the upstream cluster-monitoring-operator source code for a list of the attributes that Telemetry gathers from Prometheus.

4.1.2. About the Insights Operator

The Insights Operator periodically gathers configuration and component failure status and, by default, reports that data every two hours to Red Hat. This information enables Red Hat to assess configuration and deeper failure data than is reported through Telemetry.

Users of Red Hat OpenShift Service on AWS can display the report of each cluster in the Insights Advisor service on Red Hat Hybrid Cloud Console. If any issues have been identified, Insights provides further details and, if available, steps on how to solve a problem.

The Insights Operator does not collect identifying information, such as user names, passwords, or certificates. See Red Hat Insights Data & Application Security for information about Red Hat Insights data collection and controls.

Red Hat uses all connected cluster information to:

- Identify potential cluster issues and provide a solution and preventive actions in the Insights Advisor service on Red Hat Hybrid Cloud Console
- Improve Red Hat OpenShift Service on AWS by providing aggregated and critical information to product and support teams
- Make Red Hat OpenShift Service on AWS more intuitive

4.1.2.1. Information collected by the Insights Operator

The following information is collected by the Insights Operator:

- General information about your cluster and its components to identify issues that are specific to your Red Hat OpenShift Service on AWS version and environment
- Configuration files, such as the image registry configuration, of your cluster to determine incorrect settings and issues that are specific to parameters you set
- Errors that occur in the cluster components
- Progress information of running updates, and the status of any component upgrades
- Details of the platform that Red Hat OpenShift Service on AWS is deployed on, such as Amazon Web Services, and the region that the cluster is located in
- Cluster workload information transformed into discreet Secure Hash Algorithm (SHA) values, which allows Red Hat to assess workloads for security and version vulnerabilities without disclosing sensitive details
- If an Operator reports an issue, information is collected about core Red Hat OpenShift Service on AWS pods in the openshift-* and kube-* projects. This includes state, resource, security context, volume information, and more.

Additional resources

- The Insights Operator source code is available for review and contribution. See the Insights Operator upstream project for a list of the items collected by the Insights Operator.

4.1.3. Understanding Telemetry and Insights Operator data flow

The Telemeter Client collects selected time series data from the Prometheus API. The time series data is uploaded to api.openshift.com every four minutes and thirty seconds for processing.

The Insights Operator gathers selected data from the Kubernetes API and the Prometheus API into an archive. The archive is uploaded to OpenShift Cluster Manager Hybrid Cloud Console every two hours for processing. The Insights Operator also downloads the latest Insights analysis from OpenShift Cluster Manager Hybrid Cloud Console. This is used to populate the Insights status pop-up that is included in the Overview page in the Red Hat OpenShift Service on AWS web console.

All of the communication with Red Hat occurs over encrypted channels by using Transport Layer Security (TLS) and mutual certificate authentication. All of the data is encrypted in transit and at rest.

Access to the systems that handle customer data is controlled through multi-factor authentication and strict authorization controls. Access is granted on a need-to-know basis and is limited to required operations.

Telemetry and Insights Operator data flow
4.1.4. Additional details about how remote health monitoring data is used

The information collected to enable remote health monitoring is detailed in Information collected by Telemetry and Information collected by the Insights Operator.

As further described in the preceding sections of this document, Red Hat collects data about your use of the Red Hat Product(s) for purposes such as providing support and upgrades, optimizing performance or configuration, minimizing service impacts, identifying and remediating threats, troubleshooting, improving the offerings and user experience, responding to issues, and for billing purposes if applicable.

Collection safeguards

Red Hat employs technical and organizational measures designed to protect the telemetry and configuration data.

Sharing

Red Hat may share the data collected through Telemetry and the Insights Operator internally within Red Hat to improve your user experience. Red Hat may share telemetry and configuration data with its business partners in an aggregated form that does not identify customers to help the partners better understand their markets and their customers' use of Red Hat offerings or to ensure the successful integration of products jointly supported by those partners.

Third parties

Red Hat may engage certain third parties to assist in the collection, analysis, and storage of the Telemetry and configuration data.

4.2. SHOWING DATA COLLECTED BY REMOTE HEALTH MONITORING

User control / enabling and disabling telemetry and configuration data collection

As an administrator, you can review the metrics collected by Telemetry and the Insights Operator.

4.2.1. Showing data collected by Telemetry

You can view the cluster and components time series data captured by Telemetry.

Prerequisites
You have installed the OpenShift Container Platform CLI (oc).

You have access to the cluster as a user with the dedicated-admin role.

**Procedure**

1. Log in to a cluster.

2. Run the following command, which queries a cluster’s Prometheus service and returns the full set of time series data captured by Telemetry:

   ```bash
   $ curl -G -k -H "Authorization: Bearer $(oc whoami -t)" \
   https://$(oc get route prometheus-k8s-federate -n openshift-monitoring -o jsonpath="{.spec.host}")/federate \
   -d-urlencode 'match[]=(_name_="cluster:usage:.*")' \
   -d-urlencode 'match[]=(_name_="count:up0")' \
   -d-urlencode 'match[]=(_name_="cluster_version")' \
   -d-urlencode 'match[]=(_name_="cluster_version_available_updates")' \
   -d-urlencode 'match[]=(_name_="cluster_version_capability")' \
   -d-urlencode 'match[]=(_name_="cluster_operator_up")' \
   -d-urlencode 'match[]=(_name_="cluster_operator_conditions")' \
   -d-urlencode 'match[]=(_name_="cluster_version_payload")' \
   -d-urlencode 'match[]=(_name_="cluster_installer")' \
   -d-urlencode 'match[]=(_name_="cluster_infrastructure_provider")' \
   -d-urlencode 'match[]=(_name_="cluster_feature_set")' \
   -d-urlencode 'match[]=(_name_="instance:etcd_object_counts:sum")' \
   -d-urlencode 'match[]=(_name_="ALERTS",alertstate="firing")' \
   -d-urlencode 'match[]=(_name_="code:apiserver_request_total:rate:sum")' \
   -d-urlencode 'match[]=(_name_="cluster:capacity_cpu_cores:sum")' \
   -d-urlencode 'match[]=(_name_="cluster:capacity_memory_bytes:sum")' \
   -d-urlencode 'match[]=(_name_="cluster:cpu_usage_cores:sum")' \
   -d-urlencode 'match[]=(_name_="cluster:memory_usage_bytes:sum")' \
   -d-urlencode 'match[]=(_name_="openshift:cpu_usage_cores:sum")' \
   -d-urlencode 'match[]=(_name_="openshift:memory_usage_bytes:sum")' \
   -d-urlencode 'match[]=(_name_="workload:cpu_usage_cores:sum")' \
   -d-urlencode 'match[]=(_name_="workload:memory_usage_bytes:sum")' \
   -d-urlencode 'match[]=(_name_="cluster:virt_platform_nodes:sum")' \
   -d-urlencode 'match[]=(_name_="cluster:node_instance_type_count:sum")' \
   -d-urlencode 'match[]=(_name_="cnv:vmi_status_running:count")' \
   -d-urlencode 'match[]=(_name_="node_role_os_version_machine:cpu_capacity_cores:sum")' \
   -d-urlencode 'match[]=(_name_="node_role_os_version_machine:cpu_capacity_sockets:sum")' \
   -d-urlencode 'match[]=(_name_="subscription_sync_total")' \
   -d-urlencode 'match[]=(_name_="olm_resolution_duration_seconds")' \
   -d-urlencode 'match[]=(_name_="csv_succeeded")' \
   -d-urlencode 'match[]=(_name_="csv_abnormal")' \
   -d-urlencode 'match[]=(_name_="cluster:kube_persistentvolumeclaim_resource_requests_storage_bytes:provisioned:sum")' \
   -d-urlencode 'match[]=(_name_="cluster:kubelet_volume_stats_used_bytes:provisioner:sum")' \
   -d-urlencode 'match[]=(_name_="ceph_cluster_total_bytes")' \
   -d-urlencode 'match[]=(_name_="ceph_cluster_total_used_raw_bytes")'
   ```
4.3. USING INSIGHTS TO IDENTIFY ISSUES WITH YOUR CLUSTER

Insights repeatedly analyzes the data Insights Operator sends. Users of Red Hat OpenShift Service on AWS can display the report in the Insights Advisor service on Red Hat Hybrid Cloud Console.

4.3.1. About Red Hat Insights Advisor for Red Hat OpenShift Service on AWS

You can use Insights Advisor to assess and monitor the health of your Red Hat OpenShift Service on AWS clusters. Whether you are concerned about individual clusters, or with your whole infrastructure, it is important to be aware of your exposure to issues that can affect service availability, fault tolerance, performance, or security.

Insights repeatedly analyzes the data that Insights Operator sends using a database of recommendations, which are sets of conditions that can leave your Red Hat OpenShift Service on AWS clusters at risk. Your data is then uploaded to the Insights Advisor service on Red Hat Hybrid Cloud Console where you can perform the following actions:

- See clusters impacted by a specific recommendation.
- Use robust filtering capabilities to refine your results to those recommendations.
- Learn more about individual recommendations, details about the risks they present, and get resolutions tailored to your individual clusters.
- Share results with other stakeholders.

4.3.2. Understanding Insights Advisor recommendations

Insights Advisor bundles information about various cluster states and component configurations that can negatively affect the service availability, fault tolerance, performance, or security of your clusters. This information set is called a recommendation in Insights Advisor and includes the following:

```plaintext
--data-urlencode 'match[]=\{"name__="log_collector_error_count_total"\}'
--data-urlencode 'match[]=\{"name__="log_forwarder_pipeline_info"\}'
--data-urlencode 'match[]=\{"name__="log_forwarder_input_info"\}'
--data-urlencode 'match[]=\{"name__="log_forwarder_output_info"\}'
--data-urlencode 'match[]=\{"name__="cluster:log_collected_bytes_total:sum"\}'
--data-urlencode 'match[]=\{"name__="cluster:log_logged_bytes_total:sum"\}'
--data-urlencode 'match[]=\{"name__="cluster:kata_monitor_running_shim_count:sum"\}'
--data-urlencode 'match[]=\{"name__="platform:hypershift_hostedclusters:max"\}'
--data-urlencode 'match[]=\{"name__="platform:hypershift_nodepools:max"\}'
--data-urlencode 'match[]=\{"name__="namespace:noobaa_unhealthy_bucket_claims:max"\}'
--data-urlencode 'match[]=\{"name__="namespace:noobaa_buckets_claims:max"\}'
--data-urlencode 'match[]=\{"name__="namespace:noobaa_namespace_resources:max"\}'
--data-urlencode 'match[]=\{"name__="namespace:noobaa_namespace_buckets:max"\}'
--data-urlencode 'match[]=\{"name__="namespace:noobaa_accounts:max"\}'
--data-urlencode 'match[]=\{"name__="namespace:noobaa_namespace_buckets:max"\}'
--data-urlencode 'match[]=\{"name__="namespace:noobaa_namespace_buckets:max"\}'
--data-urlencode 'match[]=\{"name__="ocs_advanced_feature_usage"\}'
--data-urlencode 'match[]=\{"name__="os_image_url_override:sum"\}'
```
4.3.3. Displaying potential issues with your cluster

This section describes how to display the Insights report in Insights Advisor on OpenShift Cluster Manager Hybrid Cloud Console.

Note that Insights repeatedly analyzes your cluster and shows the latest results. These results can change, for example, if you fix an issue or a new issue has been detected.

Prerequisites

- Your cluster is registered on OpenShift Cluster Manager Hybrid Cloud Console.
- Remote health reporting is enabled, which is the default.
- You are logged in to OpenShift Cluster Manager Hybrid Cloud Console.

Procedure

1. Navigate to Advisor → Recommendations on OpenShift Cluster Manager Hybrid Cloud Console.
   Depending on the result, Insights Advisor displays one of the following:
   - No matching recommendations found if Insights did not identify any issues.
   - A list of issues Insights has detected, grouped by risk (low, moderate, important, and critical).
   - No clusters yet, if Insights has not yet analyzed the cluster. The analysis starts shortly after the cluster has been installed, registered, and connected to the internet.

2. If any issues are displayed, click the > icon in front of the entry for more details.
   Depending on the issue, the details can also contain a link to more information from Red Hat about the issue.

4.3.4. Displaying all Insights Advisor recommendations

The Recommendations view, by default, only displays the recommendations that are detected on your clusters. However, you can view all of the recommendations in the advisor archive.
Prerequisites

- Remote health reporting is enabled, which is the default.
- Your cluster is registered on Red Hat Hybrid Cloud Console.
- You are logged in to OpenShift Cluster Manager Hybrid Cloud Console.

Procedure

1. Navigate to Advisor → Recommendations on OpenShift Cluster Manager Hybrid Cloud Console.

2. Click the X icons next to the Clusters Impacted and Status filters.
   You can now browse through all of the potential recommendations for your cluster.

4.3.5. Disabling Insights Advisor recommendations

You can disable specific recommendations that affect your clusters, so that they no longer appear in your reports. It is possible to disable a recommendation for a single cluster or all of your clusters.

**NOTE**
Disabling a recommendation for all of your clusters also applies to any future clusters.

Prerequisites

- Remote health reporting is enabled, which is the default.
- Your cluster is registered on OpenShift Cluster Manager Hybrid Cloud Console.
- You are logged in to OpenShift Cluster Manager Hybrid Cloud Console.

Procedure

1. Navigate to Advisor → Recommendations on OpenShift Cluster Manager Hybrid Cloud Console.

2. Optional: Use the Clusters Impacted and Status filters as needed.

3. Disable an alert by using one of the following methods:

   - To disable an alert:
     a. Click the Options menu for that alert, and then click Disable recommendation.
     b. Enter a justification note and click Save.

   - To view the clusters affected by this alert before disabling the alert:
     a. Click the name of the recommendation to disable. You are directed to the single recommendation page.
     b. Review the list of clusters in the Affected clusters section.
c. Click **Actions → Disable recommendation** to disable the alert for all of your clusters.

d. Enter a justification note and click **Save**.

### 4.3.6. Enabling a previously disabled Insights Advisor recommendation

When a recommendation is disabled for all clusters, you no longer see the recommendation in the Insights Advisor. You can change this behavior.

#### Prerequisites

- Remote health reporting is enabled, which is the default.
- Your cluster is registered on **OpenShift Cluster Manager Hybrid Cloud Console**.
- You are logged in to **OpenShift Cluster Manager Hybrid Cloud Console**.

#### Procedure

1. Navigate to **Advisor → Recommendations** on **OpenShift Cluster Manager Hybrid Cloud Console**.

2. Filter the recommendations to display on the disabled recommendations:
   
   a. From the **Status** drop-down menu, select **Status**.
   
   b. From the **Filter by status** drop-down menu, select **Disabled**.
   
   c. Optional: Clear the **Clusters impacted** filter.

3. Locate the recommendation to enable.

4. Click the **Options** menu, and then click **Enable recommendation**.

### 4.3.7. Displaying the Insights status in the web console

Insights repeatedly analyzes your cluster and you can display the status of identified potential issues of your cluster in the Red Hat OpenShift Service on AWS web console. This status shows the number of issues in the different categories and, for further details, links to the reports in **OpenShift Cluster Manager Hybrid Cloud Console**.

#### Prerequisites

- Your cluster is registered in **OpenShift Cluster Manager Hybrid Cloud Console**.
- Remote health reporting is enabled, which is the default.
- You are logged in to the Red Hat OpenShift Service on AWS web console.

#### Procedure

1. Navigate to **Home → Overview** in the Red Hat OpenShift Service on AWS web console.

2. Click **Insights** on the **Status** card.
The pop-up window lists potential issues grouped by risk. Click the individual categories or View all recommendations in Insights Advisor to display more details.

4.4. USING THE INSIGHTS OPERATOR

The Insights Operator periodically gathers configuration and component failure status and, by default, reports that data every two hours to Red Hat. This information enables Red Hat to assess configuration and deeper failure data than is reported through Telemetry. Users of Red Hat OpenShift Service on AWS can display the report in the Insights Advisor service on Red Hat Hybrid Cloud Console.

Additional resources

- For more information on using Insights Advisor to identify issues with your cluster, see Using Insights to identify issues with your cluster.

4.4.1. Understanding Insights Operator alerts

The Insights Operator declares alerts through the Prometheus monitoring system to the Alertmanager. You can view these alerts in the Alerting UI in the Red Hat OpenShift Service on AWS web console by using one of the following methods:

- In the Administrator perspective, click Observe → Alerting.
- In the Developer perspective, click Observe → <project_name> → Alerts tab.

Currently, Insights Operator sends the following alerts when the conditions are met:

<table>
<thead>
<tr>
<th>Alert</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InsightsDisabled</td>
<td>Insights Operator is disabled.</td>
</tr>
<tr>
<td>SimpleContentAccessNotAvailable</td>
<td>Simple content access is not enabled in Red Hat Subscription Management.</td>
</tr>
<tr>
<td>InsightsRecommendationActive</td>
<td>Insights has an active recommendation for the cluster.</td>
</tr>
</tbody>
</table>
CHAPTER 5. SUMMARIZING CLUSTER SPECIFICATIONS

5.1. SUMMARIZING CLUSTER SPECIFICATIONS BY USING A CLUSTER VERSION OBJECT

You can obtain a summary of Red Hat OpenShift Service on AWS cluster specifications by querying the `clusterversion` resource.

**Prerequisites**

- You have access to the cluster as a user with the `dedicated-admin` role.
- You have installed the OpenShift CLI (`oc`).

**Procedure**

1. Query cluster version, availability, uptime, and general status:

   ```
   $ oc get clusterversion
   NAME      VERSION   AVAILABLE   PROGRESSING   SINCE       STATUS
   version   4.13.8    True        False         8h         Cluster version is 4.13.8
   ```

   **Example output**

2. Obtain a detailed summary of cluster specifications, update availability, and update history:

   ```
   $ oc describe clusterversion
   ```

   **Example output**

Name: version
Namespace: 
Labels: <none>
Annotations: <none>
API Version: config.openshift.io/v1
Kind: ClusterVersion
# ...
   Image: quay.io/openshift-release-dev/ocp-release@sha256:a956488d295fe5a59c8663a4d9992b9b5d0950f510a7387dbbfb8d20fc5970ce
   URL: https://access.redhat.com/errata/RHSA-2023:4456
   Version: 4.13.8
   History:
   Completion Time: 2023-08-17T13:20:21Z
   Image: quay.io/openshift-release-dev/ocp-release@sha256:a956488d295fe5a59c8663a4d9992b9b5d0950f510a7387dbbfb8d20fc5970ce
   Started Time: 2023-08-17T12:59:45Z
   State: Completed
6.1. TROUBLESHOOTING RED HAT OPENShift SERVICE ON AWS INSTALLATIONS

6.1.1. Installation troubleshooting

6.1.1.1. Inspect install or uninstall logs

To display install logs:

- Run the following command, replacing `<cluster_name>` with the name of your cluster:
  
  ```
  $ rosa logs install --cluster=<cluster_name>
  ```

- To watch the logs, include the `--watch` flag:
  
  ```
  $ rosa logs install --cluster=<cluster_name> --watch
  ```

To display uninstall logs:

- Run the following command, replacing `<cluster_name>` with the name of your cluster:
  
  ```
  $ rosa logs uninstall --cluster=<cluster_name>
  ```

- To watch the logs, include the `--watch` flag:
  
  ```
  $ rosa logs uninstall --cluster=<cluster_name> --watch
  ```

6.1.1.2. Verify your AWS account permissions for clusters without STS

Run the following command to verify if your AWS account has the correct permissions. This command verifies permissions only for clusters that do not use the AWS Security Token Service (STS):

```
$ rosa verify permissions
```

If you receive any errors, double check to ensure than an SCP is not applied to your AWS account. If you are required to use an SCP, see Red Hat Requirements for Customer Cloud Subscriptions for details on the minimum required SCP.

6.1.1.3. Verify your AWS account and quota

Run the following command to verify you have the available quota on your AWS account:

```
$ rosa verify quota
```

AWS quotas change based on region. Be sure you are verifying your quota for the correct AWS region. If you need to increase your quota, navigate to your AWS console, and request a quota increase for the service that failed.
6.1.1.4. AWS notification emails

When creating a cluster, the Red Hat OpenShift Service on AWS service creates small instances in all supported regions. This check ensures the AWS account being used can deploy to each supported region.

For AWS accounts that are not using all supported regions, AWS may send one or more emails confirming that "Your Request For Accessing AWS Resources Has Been Validated". Typically the sender of this email is aws-verification@amazon.com.

This is expected behavior as the Red Hat OpenShift Service on AWS service is validating your AWS account configuration.

6.2. VERIFYING NODE HEALTH

6.2.1. Reviewing node status, resource usage, and configuration

Review cluster node health status, resource consumption statistics, and node logs. Additionally, query kubelet status on individual nodes.

Prerequisites

- You have access to the cluster as a user with the dedicated-admin role.
- You have installed the OpenShift CLI (oc).

Procedure

- List the name, status, and role for all nodes in the cluster:
  
  $ oc get nodes

- Summarize CPU and memory usage for each node within the cluster:
  
  $ oc adm top nodes

- Summarize CPU and memory usage for a specific node:
  
  $ oc adm top node my-node

6.3. TROUBLESHOOTING OPERATOR ISSUES

Operators are a method of packaging, deploying, and managing an Red Hat OpenShift Service on AWS application. They act like an extension of the software vendor’s engineering team, watching over an Red Hat OpenShift Service on AWS environment and using its current state to make decisions in real time. Operators are designed to handle upgrades seamlessly, react to failures automatically, and not take shortcuts, such as skipping a software backup process to save time.

Red Hat OpenShift Service on AWS 4 includes a default set of Operators that are required for proper functioning of the cluster. These default Operators are managed by the Cluster Version Operator (CVO).

As a cluster administrator, you can install application Operators from the OperatorHub using the Red
Hat OpenShift Service on AWS web console or the CLI. You can then subscribe the Operator to one or more namespaces to make it available for developers on your cluster. Application Operators are managed by Operator Lifecycle Manager (OLM).

If you experience Operator issues, verify Operator subscription status. Check Operator pod health across the cluster and gather Operator logs for diagnosis.

### 6.3.1. Operator subscription condition types

Subscriptions can report the following condition types:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CatalogSourcesUnhealthy</td>
<td>Some or all of the catalog sources to be used in resolution are unhealthy.</td>
</tr>
<tr>
<td>InstallPlanMissing</td>
<td>An install plan for a subscription is missing.</td>
</tr>
<tr>
<td>InstallPlanPending</td>
<td>An install plan for a subscription is pending installation.</td>
</tr>
<tr>
<td>InstallPlanFailed</td>
<td>An install plan for a subscription has failed.</td>
</tr>
<tr>
<td>ResolutionFailed</td>
<td>The dependency resolution for a subscription has failed.</td>
</tr>
</tbody>
</table>

**NOTE**

Default Red Hat OpenShift Service on AWS cluster Operators are managed by the Cluster Version Operator (CVO) and they do not have a Subscription object. Application Operators are managed by Operator Lifecycle Manager (OLM) and they have a Subscription object.

### Additional resources

- [Catalog health requirements](#)

### 6.3.2. Viewing Operator subscription status by using the CLI

You can view Operator subscription status by using the CLI.

#### Prerequisites

- You have access to the cluster as a user with the dedicated-admin role.
- You have installed the OpenShift CLI (oc).

#### Procedure

1. List Operator subscriptions:

   ```bash
   $ oc get subs -n <operator_namespace>
   ```
2. Use the `oc describe` command to inspect a **Subscription** resource:

```
$ oc describe sub <subscription_name> -n <operator_namespace>
```

3. In the command output, find the **Conditions** section for the status of Operator subscription condition types. In the following example, the **CatalogSourcesUnhealthy** condition type has a status of **false** because all available catalog sources are healthy:

**Example output**

```
Name:         cluster-logging
Namespace:    openshift-logging
Labels:       operators.coreos.com/cluster-logging.openshift-logging=
Annotations:  <none>
API Version:  operators.coreos.com/v1alpha1
Kind:         Subscription
# ...
Conditions:   
Message:       all available catalogsources are healthy
Reason:        AllCatalogSourcesHealthy
Status:        False
Type:          CatalogSourcesUnhealthy
# ...
```

**NOTE**

Default Red Hat OpenShift Service on AWS cluster Operators are managed by the Cluster Version Operator (CVO) and they do not have a **Subscription** object. Application Operators are managed by Operator Lifecycle Manager (OLM) and they have a **Subscription** object.

### 6.3.3. Viewing Operator catalog source status by using the CLI

You can view the status of an Operator catalog source by using the CLI.

**Prerequisites**

- You have access to the cluster as a user with the **dedicated-admin** role.
- You have installed the OpenShift CLI (`oc`).

**Procedure**

1. List the catalog sources in a namespace. For example, you can check the **openshift-marketplace** namespace, which is used for cluster-wide catalog sources:

```
$ oc get catalogsources -n openshift-marketplace
```

**Example output**

```
NAME                  DISPLAY               TYPE   PUBLISHER   AGE
Red Hat OpenShift Service on AWS 4 Support
```
2. Use the `oc describe` command to get more details and status about a catalog source:

```
$ oc describe catalogsource example-catalog -n openshift-marketplace
```

**Example output**

```
Name:         example-catalog
Namespace:    openshift-marketplace
Labels:       <none>
Annotations:  operatorframework.io/managed-by: marketplace-operator
target.workload.openshift.io/management: {
  "effect": "PreferredDuringScheduling"
}  
API Version:  operators.coreos.com/v1alpha1
Kind:         CatalogSource
# ...
Status:
  Connection State:
    Address:              example-catalog.openshift-marketplace.svc:50051
    Last Connect:         2021-09-09T17:07:35Z
    Last Observed State:  TRANSIENT_FAILURE
  Registry Service:
    Created At:         2021-09-09T17:05:45Z
    Port:               50051
    Protocol:           grpc
    Service Name:       example-catalog
    Service Namespace:  openshift-marketplace
# ...
```

In the preceding example output, the last observed state is `TRANSIENT_FAILURE`. This state indicates that there is a problem establishing a connection for the catalog source.

3. List the pods in the namespace where your catalog source was created:

```
$ oc get pods -n openshift-marketplace
```

**Example output**

```
NAME                                    READY   STATUS             RESTARTS   AGE
certified-operators-cv9nn               1/1     Running            0          36m
community-operators-6v8lp               1/1     Running            0          36m
marketplace-operator-86bf75f9b-jkgbc    1/1     Running            0          42m
example-catalog-bwt8z                   0/1     ImagePullBackOff   0          3m55s
redhat-marketplace-57p8c                1/1     Running            0          36m
redhat-operators-smxx8                  1/1     Running            0          36m
```

When a catalog source is created in a namespace, a pod for the catalog source is created in that namespace. In the preceding example output, the status for the `example-catalog-bwt8z` pod is `ImagePullBackOff`. This status indicates that there is an issue pulling the catalog source’s index image.
4. Use the **oc describe** command to inspect a pod for more detailed information:

```
$ oc describe pod example-catalog-bwt8z -n openshift-marketplace
```

**Example output**

```
Name:         example-catalog-bwt8z
Namespace:    openshift-marketplace
Priority:     0
Node:         ci-ln-jryyy2-f76d1-ggdbq-worker-b-vsxjd/10.0.128.2

Events:

<table>
<thead>
<tr>
<th>Type</th>
<th>Reason</th>
<th>Age</th>
<th>From</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Scheduled</td>
<td>48s</td>
<td>default-scheduler</td>
<td>Successfully assigned openshift-marketplace/example-catalog-bwt8z to ci-ln-jryyy2-f76d1-ffgdbq-worker-b-vsxjd</td>
</tr>
<tr>
<td>Normal</td>
<td>AddedInterface</td>
<td>47s</td>
<td>multus</td>
<td>Add eth0 [10.131.0.40/23] from openshift-sdn</td>
</tr>
<tr>
<td>Normal</td>
<td>BackOff</td>
<td>20s (x2 over 46s)</td>
<td>kubelet</td>
<td>Back-off pulling image &quot;quay.io/example-org/example-catalog:v1&quot;</td>
</tr>
<tr>
<td>Warning</td>
<td>Failed</td>
<td>20s (x2 over 46s)</td>
<td>kubelet</td>
<td>Error: ImagePullBackOff</td>
</tr>
<tr>
<td>Normal</td>
<td>Pulling</td>
<td>8s (x3 over 47s)</td>
<td>kubelet</td>
<td>Pulling image &quot;quay.io/example-org/example-catalog:v1&quot;</td>
</tr>
<tr>
<td>Warning</td>
<td>Failed</td>
<td>8s (x3 over 47s)</td>
<td>kubelet</td>
<td>Failed to pull image &quot;quay.io/example-org/example-catalog:v1&quot;: rpc error: code = Unknown desc = reading manifest v1 in quay.io/example-org/example-catalog: unauthorized: access to the requested resource is not authorized</td>
</tr>
<tr>
<td>Warning</td>
<td>Failed</td>
<td>8s (x3 over 47s)</td>
<td>kubelet</td>
<td>Error: ErrImagePull</td>
</tr>
</tbody>
</table>
```

In the preceding example output, the error messages indicate that the catalog source’s index image is failing to pull successfully because of an authorization issue. For example, the index image might be stored in a registry that requires login credentials.

**Additional resources**

- gRPC documentation: States of Connectivity

**6.3.4. Querying Operator pod status**

You can list Operator pods within a cluster and their status. You can also collect a detailed Operator pod summary.

**Prerequisites**

- You have access to the cluster as a user with the **dedicated-admin** role.
- Your API service is still functional.
- You have installed the OpenShift CLI (**oc**).

**Procedure**

1. List Operators running in the cluster. The output includes Operator version, availability, and uptime information:
2. List Operator pods running in the Operator’s namespace, plus pod status, restarts, and age:

   $ oc get pods -n <operator_namespace>

3. Output a detailed Operator pod summary:

   $ oc describe pod <operator_pod_name> -n <operator_namespace>

### 6.3.5. Gathering Operator logs

If you experience Operator issues, you can gather detailed diagnostic information from Operator pod logs.

#### Prerequisites

- You have access to the cluster as a user with the dedicated-admin role.
- Your API service is still functional.
- You have installed the OpenShift CLI (oc).
- You have the fully qualified domain names of the control plane or control plane machines.

#### Procedure

1. List the Operator pods that are running in the Operator’s namespace, plus the pod status, restarts, and age:

   $ oc get pods -n <operator_namespace>

2. Review logs for an Operator pod:

   $ oc logs pod/<pod_name> -n <operator_namespace>

   If an Operator pod has multiple containers, the preceding command will produce an error that includes the name of each container. Query logs from an individual container:

   $ oc logs pod/<operator_pod_name> -c <container_name> -n <operator_namespace>

3. If the API is not functional, review Operator pod and container logs on each control plane node by using SSH instead. Replace `<master-node>.<cluster_name>.<base_domain>` with appropriate values.
   
   a. List pods on each control plane node:

      $ ssh core@<master-node>.<cluster_name>.<base_domain> sudo crictl pods

   b. For any Operator pods not showing a Ready status, inspect the pod’s status in detail. Replace `<operator_pod_id>` with the Operator pod’s ID listed in the output of the preceding command:
6.4. INVESTIGATING POD ISSUES

Red Hat OpenShift Service on AWS leverages the Kubernetes concept of a pod, which is one or more containers deployed together on one host. A pod is the smallest compute unit that can be defined, deployed, and managed on Red Hat OpenShift Service on AWS 4.

After a pod is defined, it is assigned to run on a node until its containers exit, or until it is removed. Depending on policy and exit code, Pods are either removed after exiting or retained so that their logs can be accessed.

The first thing to check when pod issues arise is the pod’s status. If an explicit pod failure has occurred, observe the pod’s error state to identify specific image, container, or pod network issues. Focus diagnostic data collection according to the error state. Review pod event messages, as well as pod and container log information. Diagnose issues dynamically by accessing running Pods on the command line, or start a debug pod with root access based on a problematic pod’s deployment configuration.

6.4.1. Understanding pod error states

NOTE

Red Hat OpenShift Service on AWS 4 cluster nodes running Red Hat Enterprise Linux CoreOS (RHCOS) are immutable and rely on Operators to apply cluster changes. Accessing cluster nodes by using SSH is not recommended. Before attempting to collect diagnostic data over SSH, review whether the data collected by running `oc adm must gather` and other `oc` commands is sufficient instead. However, if the Red Hat OpenShift Service on AWS API is not available, or the kubelet is not properly functioning on the target node, `oc` operations will be impacted. In such situations, it is possible to access nodes using `ssh core@<node>.<cluster_name>.<base_domain>`.

```bash
$ ssh core@<master-node>.<cluster_name>.<base_domain> sudo crictl inspectp <operator_pod_id>
c. List containers related to an Operator pod:
$ ssh core@<master-node>.<cluster_name>.<base_domain> sudo crictl ps --pod= <operator_pod_id>
d. For any Operator container not showing a `Ready` status, inspect the container’s status in detail. Replace `<container_id>` with a container ID listed in the output of the preceding command:
$ ssh core@<master-node>.<cluster_name>.<base_domain> sudo crictl inspect <container_id>
e. Review the logs for any Operator containers not showing a `Ready` status. Replace `<container_id>` with a container ID listed in the output of the preceding command:
$ ssh core@<master-node>.<cluster_name>.<base_domain> sudo crictl logs -f <container_id>
```
Pod failures return explicit error states that can be observed in the `status` field in the output of `oc get pods`. Pod error states cover image, container, and container network related failures.

The following table provides a list of pod error states along with their descriptions.

**Table 6.2. Pod error states**

<table>
<thead>
<tr>
<th>Pod error state</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ErrImagePull</strong></td>
<td>Generic image retrieval error.</td>
</tr>
<tr>
<td><strong>ErrImagePullBackOff</strong></td>
<td>Image retrieval failed and is backed off.</td>
</tr>
<tr>
<td><strong>ErrInvalidImageName</strong></td>
<td>The specified image name was invalid.</td>
</tr>
<tr>
<td><strong>ErrImageInspect</strong></td>
<td>Image inspection did not succeed.</td>
</tr>
<tr>
<td><strong>ErrImageNeverPull</strong></td>
<td><code>PullPolicy</code> is set to <code>NeverPullImage</code> and the target image is not present locally on the host.</td>
</tr>
<tr>
<td><strong>ErrRegistryUnavailable</strong></td>
<td>When attempting to retrieve an image from a registry, an HTTP error was encountered.</td>
</tr>
<tr>
<td><strong>ErrContainerNotFound</strong></td>
<td>The specified container is either not present or not managed by the kubelet, within the declared pod.</td>
</tr>
<tr>
<td><strong>ErrRunInitContainer</strong></td>
<td>Container initialization failed.</td>
</tr>
<tr>
<td><strong>ErrRunContainer</strong></td>
<td>None of the pod's containers started successfully.</td>
</tr>
<tr>
<td><strong>ErrKillContainer</strong></td>
<td>None of the pod's containers were killed successfully.</td>
</tr>
<tr>
<td><strong>ErrCrashLoopBackOff</strong></td>
<td>A container has terminated. The kubelet will not attempt to restart it.</td>
</tr>
<tr>
<td><strong>ErrVerifyNonRoot</strong></td>
<td>A container or image attempted to run with root privileges.</td>
</tr>
<tr>
<td><strong>ErrCreatePodSandbox</strong></td>
<td>Pod sandbox creation did not succeed.</td>
</tr>
<tr>
<td><strong>ErrConfigPodSandbox</strong></td>
<td>Pod sandbox configuration was not obtained.</td>
</tr>
</tbody>
</table>
### 6.4.2. Reviewing pod status

You can query pod status and error states. You can also query a pod’s associated deployment configuration and review base image availability.

**Prerequisites**

- You have access to the cluster as a user with the `dedicated-admin` role.
- You have installed the OpenShift CLI (`oc`).
- `skopeo` is installed.

**Procedure**

1. Switch into a project:

   ```bash
   $ oc project <project_name>
   ```

2. List pods running within the namespace, as well as pod status, error states, restarts, and age:

   ```bash
   $ oc get pods
   ```

3. Determine whether the namespace is managed by a deployment configuration:

   ```bash
   $ oc status
   ```

   If the namespace is managed by a deployment configuration, the output includes the deployment configuration name and a base image reference.

4. Inspect the base image referenced in the preceding command’s output:

   ```bash
   $ skopeo inspect docker://<image_reference>
   ```

5. If the base image reference is not correct, update the reference in the deployment configuration:

   ```bash
   $ oc edit deployment/my-deployment
   ```

---

<table>
<thead>
<tr>
<th>Pod error state</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrKillPodSandbox</td>
<td>A pod sandbox did not stop successfully.</td>
</tr>
<tr>
<td>ErrSetupNetwork</td>
<td>Network initialization failed.</td>
</tr>
<tr>
<td>ErrTeardownNetwork</td>
<td>Network termination failed.</td>
</tr>
</tbody>
</table>

---
6. When deployment configuration changes on exit, the configuration will automatically redeploy. Watch pod status as the deployment progresses, to determine whether the issue has been resolved:

```bash
$ oc get pods -w
```

7. Review events within the namespace for diagnostic information relating to pod failures:

```bash
$ oc get events
```

### 6.4.3. Inspecting pod and container logs

You can inspect pod and container logs for warnings and error messages related to explicit pod failures. Depending on policy and exit code, pod and container logs remain available after pods have been terminated.

#### Prerequisites

- You have access to the cluster as a user with the `dedicated-admin` role.
- Your API service is still functional.
- You have installed the OpenShift CLI (`oc`).

#### Procedure

1. Query logs for a specific pod:

   ```bash
   $ oc logs <pod_name>
   ```

2. Query logs for a specific container within a pod:

   ```bash
   $ oc logs <pod_name> -c <container_name>
   ```

   Logs retrieved using the preceding `oc logs` commands are composed of messages sent to stdout within pods or containers.

3. Inspect logs contained in `/var/log/` within a pod.

   a. List log files and subdirectories contained in `/var/log` within a pod:

   ```bash
   $ oc exec <pod_name> -- ls -alh /var/log
   ```

#### Example output

```
total 124K
drwxr-xr-x. 1 root root   33 Aug 11 11:23 .
drwxr-xr-x. 1 root root   28 Sep  6 2022 ..
-rw-rw----. 1 root utmp    0 Jul 10 10:31 btmp
-rw-r--r--. 1 root root  33K Jul 17 10:07 dnf.librepo.log
-rw-r--r--. 1 root root  69K Jul 17 10:07 dnf.log
-rw-r--r--. 1 root root  8.8K Jul 17 10:07 dnf.rpm.log
-rw-r--r--. 1 root root  480 Jul 17 10:07 hawkey.log
```
b. Query a specific log file contained in `/var/log` within a pod:

```
$ oc exec <pod_name> cat /var/log/<path_to_log>
```

**Example output**

```
2023-07-10T10:29:38+0000 INFO --- logging initialized ---
2023-07-10T10:29:38+0000 DDEBUG timer: config: 13 ms
2023-07-10T10:29:38+0000 DEBUG Loaded plugins: builddep, changelog, config-manager, copr, debug, debuginfo-install, download, generate_completion_cache, groups-manager, needs-restarting, playground, product-id, repoclosure, repodiff, repograph, repomanage, reposync, subscription-manager, uploadprofile
2023-07-10T10:29:38+0000 INFO Updating Subscription Management repositories.
2023-07-10T10:29:38+0000 INFO Unable to read consumer identity
2023-07-10T10:29:38+0000 INFO Subscription Manager is operating in container mode.
2023-07-10T10:29:38+0000 INFO
```

c. List log files and subdirectories contained in `/var/log` within a specific container:

```
$ oc exec <pod_name> -c <container_name> ls /var/log
```

d. Query a specific log file contained in `/var/log` within a specific container:

```
$ oc exec <pod_name> -c <container_name> cat /var/log/<path_to_log>
```

### 6.4.4. Accessing running pods

You can review running pods dynamically by opening a shell inside a pod or by gaining network access through port forwarding.

**Prerequisites**

- You have access to the cluster as a user with the `dedicated-admin` role.
- Your API service is still functional.
- You have installed the OpenShift CLI (`oc`).

**Procedure**

1. Switch into the project that contains the pod you would like to access. This is necessary because the `oc rsh` command does not accept the `-n` namespace option:

   ```
   $ oc project <namespace>
   ```

2. Start a remote shell into a pod:
If a pod has multiple containers, `oc rsh` defaults to the first container unless `-c <container_name>` is specified.

3. Start a remote shell into a specific container within a pod:

```
$ oc rsh -c <container_name> pod/<pod_name>
```

4. Create a port forwarding session to a port on a pod:

```
$ oc port-forward <pod_name> <host_port>:<pod_port>
```

Enter Ctrl+C to cancel the port forwarding session.

### 6.4.5. Starting debug pods with root access

You can start a debug pod with root access, based on a problematic pod’s deployment or deployment configuration. Pod users typically run with non-root privileges, but running troubleshooting pods with temporary root privileges can be useful during issue investigation.

**Prerequisites**

- You have access to the cluster as a user with the dedicated-admin role.
- Your API service is still functional.
- You have installed the OpenShift CLI (`oc`).

**Procedure**

1. Start a debug pod with root access, based on a deployment.
   a. Obtain a project’s deployment name:
      
      ```
      $ oc get deployment -n <project_name>
      ```
   b. Start a debug pod with root privileges, based on the deployment:
      
      ```
      $ oc debug deployment/my-deployment --as-root -n <project_name>
      ```

2. Start a debug pod with root access, based on a deployment configuration.
   a. Obtain a project’s deployment configuration name:
      
      ```
      $ oc get deploymentconfigs -n <project_name>
      ```
   b. Start a debug pod with root privileges, based on the deployment configuration:
      
      ```
      $ oc debug deploymentconfig/my-deployment-configuration --as-root -n <project_name>
      ```
NOTE

You can append -- <command> to the preceding oc debug commands to run individual commands within a debug pod, instead of running an interactive shell.

6.4.6. Copying files to and from pods and containers

You can copy files to and from a pod to test configuration changes or gather diagnostic information.

Prerequisites

- You have access to the cluster as a user with the dedicated-admin role.
- Your API service is still functional.
- You have installed the OpenShift CLI (oc).

Procedure

1. Copy a file to a pod:

   $ oc cp <local_path> <pod_name>:/<path> -c <container_name> 1

   The first container in a pod is selected if the -c option is not specified.

2. Copy a file from a pod:

   $ oc cp <pod_name>:/<path> -c <container_name> <local_path> 1

   The first container in a pod is selected if the -c option is not specified.

   NOTE

   For oc cp to function, the tar binary must be available within the container.

6.5. TROUBLESHOOTING STORAGE ISSUES

6.5.1. Resolving multi-attach errors

When a node crashes or shuts down abruptly, the attached ReadWriteOnce (RWO) volume is expected to be unmounted from the node so that it can be used by a pod scheduled on another node.

However, mounting on a new node is not possible because the failed node is unable to unmount the attached volume.

A multi-attach error is reported:

Example output

Unable to attach or mount volumes: unmounted volumes=[sso-mysql-pvol], unattached volumes=[sso-mysql-pvol default-token-x4rzc]: timed out waiting for the condition
Multi-Attach error for volume "pvc-8837384d-69d7-40b2-b2e6-5df86943eef9" Volume is already used by pod(s) sso-mysql-1-ns6b4

Procedure

To resolve the multi-attach issue, use one of the following solutions:

- Enable multiple attachments by using RWX volumes.
  For most storage solutions, you can use ReadWriteMany (RWX) volumes to prevent multi-attach errors.

- Recover or delete the failed node when using an RWO volume.
  For storage that does not support RWX, such as VMware vSphere, RWO volumes must be used instead. However, RWO volumes cannot be mounted on multiple nodes.

If you encounter a multi-attach error message with an RWO volume, force delete the pod on a shutdown or crashed node to avoid data loss in critical workloads, such as when dynamic persistent volumes are attached.

```bash
$ oc delete pod <old_pod> --force=true --grace-period=0
```

This command deletes the volumes stuck on shutdown or crashed nodes after six minutes.

6.6. INVESTIGATING MONITORING ISSUES

Red Hat OpenShift Service on AWS includes a preconfigured, preinstalled, and self-updating monitoring stack that provides monitoring for core platform components. In Red Hat OpenShift Service on AWS 4, cluster administrators can optionally enable monitoring for user-defined projects.

You can follow these procedures if your own metrics are unavailable or if Prometheus is consuming a lot of disk space.

6.6.1. Investigating why user-defined project metrics are unavailable

**ServiceMonitor** resources enable you to determine how to use the metrics exposed by a service in user-defined projects. Follow the steps outlined in this procedure if you have created a **ServiceMonitor** resource but cannot see any corresponding metrics in the Metrics UI.

**Prerequisites**

- You have access to the cluster as a user with the **dedicated-admin** role.
- You have installed the OpenShift CLI (**oc**).
- You have enabled and configured monitoring for user-defined workloads.
- You have created the **user-workload-monitoring-config** **ConfigMap** object.
- You have created a **ServiceMonitor** resource.

**Procedure**

1. **Check that the corresponding labels match** in the service and **ServiceMonitor** resource configurations.
a. Obtain the label defined in the service. The following example queries the `prometheus-example-app` service in the `ns1` project:

```shell
$ oc -n ns1 get service prometheus-example-app -o yaml
```

Example output

```yaml
labels:
  app: prometheus-example-app
```

b. Check that the `matchLabels` app label in the `ServiceMonitor` resource configuration matches the label output in the preceding step:

```shell
$ oc -n ns1 get servicemonitor prometheus-example-monitor -o yaml
```

Example output

```yaml
apiVersion: v1
kind: Service
# ...
spec:
  endpoints:
    - interval: 30s
      port: web
      scheme: http
      selector:
        matchLabels:
          app: prometheus-example-app
# ...
```

**NOTE**

You can check service and `ServiceMonitor` resource labels as a developer with view permissions for the project.

2. **Inspect the logs for the Prometheus Operator in the `openshift-user-workload-monitoring` project.**

   a. List the pods in the `openshift-user-workload-monitoring` project:

   ```shell
   $ oc -n openshift-user-workload-monitoring get pods
   ```

   Example output

   ```plaintext
   NAME                                   READY   STATUS    RESTARTS   AGE
   prometheus-operator-776fcbbd56-2nbfm   2/2     Running   0          132m
   prometheus-user-workload-0             5/5     Running   1          132m
   prometheus-user-workload-1             5/5     Running   1          132m
   thanos-ruler-user-workload-0           3/3     Running   0          132m
   thanos-ruler-user-workload-1           3/3     Running   0          132m
   ```
b. Obtain the logs from the `prometheus-operator` container in the `prometheus-operator` pod. In the following example, the pod is called `prometheus-operator-776fcbbd56-2nbfm`:

```
$ oc -n openshift-user-workload-monitoring logs prometheus-operator-776fcbbd56-2nbfm -c prometheus-operator
```

If there is a issue with the service monitor, the logs might include an error similar to this example:

```
level=warn ts=2020-08-10T11:48:20.906739623Z caller=operator.go:1829 component=prometheusoperator msg="skipping servicemonitor" error="it accesses file system via bearer token file which Prometheus specification prohibits"
```

3. **Review the target status for your endpoint** on the Metrics targets page in the Red Hat OpenShift Service on AWS web console UI.

   a. Log in to the Red Hat OpenShift Service on AWS web console and navigate to **Observe → Targets** in the **Administrator** perspective.

   b. Locate the metrics endpoint in the list, and review the status of the target in the **Status** column.

   c. If the **Status** is **Down**, click the URL for the endpoint to view more information on the **Target Details** page for that metrics target.

4. **Configure debug level logging for the Prometheus Operator** in the `openshift-user-workload-monitoring` project.

   a. Edit the `user-workload-monitoring-config` ConfigMap object in the `openshift-user-workload-monitoring` project:

```
$ oc -n openshift-user-workload-monitoring edit configmap user-workload-monitoring-config
```

   b. Add `logLevel: debug` for `prometheusOperator` under `data/config.yaml` to set the log level to `debug`:

```
apiVersion: v1
gkind: ConfigMap
metadata:
  name: user-workload-monitoring-config
namespace: openshift-user-workload-monitoring
data:
  config.yaml:
    prometheusOperator:
      logLevel: debug
# ...
```

   c. Save the file to apply the changes.
The prometheus-operator in the openshift-user-workload-monitoring project restarts automatically when you apply the log-level change.

d. Confirm that the debug log-level has been applied to the prometheus-operator deployment in the openshift-user-workload-monitoring project:

```bash
$ oc -n openshift-user-workload-monitoring get deploy prometheus-operator -o yaml | grep "log-level"
```

Example output

```
- --log-level=debug
```

Debug level logging will show all calls made by the Prometheus Operator.

e. Check that the prometheus-operator pod is running:

```bash
$ oc -n openshift-user-workload-monitoring get pods
```

NOTE

If an unrecognized Prometheus Operator loglevel value is included in the config map, the prometheus-operator pod might not restart successfully.

f. Review the debug logs to see if the Prometheus Operator is using the ServiceMonitor resource. Review the logs for other related errors.

Additional resources

- Creating a user-defined workload monitoring config map
- See Specifying how a service is monitored for details on how to create a service monitor or pod monitor
- See Getting detailed information about a metrics target

6.6.2. Determining why Prometheus is consuming a lot of disk space

Developers can create labels to define attributes for metrics in the form of key-value pairs. The number of potential key-value pairs corresponds to the number of possible values for an attribute. An attribute that has an unlimited number of potential values is called an unbound attribute. For example, a customer_id attribute is unbound because it has an infinite number of possible values.

Every assigned key-value pair has a unique time series. The use of many unbound attributes in labels can result in an exponential increase in the number of time series created. This can impact Prometheus performance and can consume a lot of disk space.

You can use the following measures when Prometheus consumes a lot of disk:

- Check the number of scrape samples that are being collected.
• Check the time series database (TSDB) status using the Prometheus HTTP API for more information about which labels are creating the most time series. Doing so requires cluster administrator privileges.

• **Reduce the number of unique time series that are created** by reducing the number of unbound attributes that are assigned to user-defined metrics.

  **NOTE**
  Using attributes that are bound to a limited set of possible values reduces the number of potential key-value pair combinations.

• **Enforce limits on the number of samples that can be scraped** across user-defined projects. This requires cluster administrator privileges.

Prerequisites

• You have access to the cluster as a user with the dedicated-admin role.

• You have installed the OpenShift CLI (`oc`).

Procedure

1. In the Administrator perspective, navigate to Observe → Metrics.

2. Run the following Prometheus Query Language (PromQL) query in the Expression field. This returns the ten metrics that have the highest number of scrape samples:

   ```
   topk(10,count by (job)({__name__=~".+"}))
   ```

3. Investigate the number of unbound label values assigned to metrics with higher than expected scrape sample counts.

   • If the metrics relate to a user-defined project review the metrics key-value pairs assigned to your workload. These are implemented through Prometheus client libraries at the application level. Try to limit the number of unbound attributes referenced in your labels.

   • If the metrics relate to a core Red Hat OpenShift Service on AWS project create a Red Hat support case on the Red Hat Customer Portal.

4. Review the TSDB status using the Prometheus HTTP API by running the following commands as a dedicated-admin:

   ```
   $ oc login -u <username> -p <password>
   $ host=$(oc -n openshift-monitoring get route prometheus-k8s -ojsonpath={.spec.host})
   $ token=$(oc whoami -t)
   $ curl -H "Authorization: Bearer $token" -k "https://$host/api/v1/status/tsdb"
   ```

Example output
Additional resources

- See Setting a scrape sample limit for user-defined projects for details on how to set a scrape sample limit and create related alerting rules

6.7. DIAGNOSING OPENSHIFT CLI (oc) ISSUES

6.7.1. Understanding OpenShift CLI (oc) log levels

With the OpenShift CLI (oc), you can create applications and manage Red Hat OpenShift Service on AWS projects from a terminal.

If oc command-specific issues arise, increase the oc log level to output API request, API response, and curl request details generated by the command. This provides a granular view of a particular oc command’s underlying operation, which in turn might provide insight into the nature of a failure.

oc log levels range from 1 to 10. The following table provides a list of oc log levels, along with their descriptions.

Table 6.3. OpenShift CLI (oc) log levels

<table>
<thead>
<tr>
<th>Log level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 5</td>
<td>No additional logging to stderr.</td>
</tr>
<tr>
<td>6</td>
<td>Log API requests to stderr.</td>
</tr>
<tr>
<td>7</td>
<td>Log API requests and headers to stderr.</td>
</tr>
<tr>
<td>8</td>
<td>Log API requests, headers, and body, plus API response headers and body to stderr.</td>
</tr>
<tr>
<td>9</td>
<td>Log API requests, headers, and body, API response headers and body, plus curl requests to stderr.</td>
</tr>
<tr>
<td>10</td>
<td>Log API requests, headers, and body, API response headers and body, plus curl requests to stderr, in verbose detail.</td>
</tr>
</tbody>
</table>

6.7.2. Specifying OpenShift CLI (oc) log levels

You can investigate OpenShift CLI (oc) issues by increasing the command’s log level.

The Red Hat OpenShift Service on AWS user’s current session token is typically included in logged curl requests where required. You can also obtain the current user’s session token manually, for use when testing aspects of an oc command’s underlying process step-by-step.

Prerequisites

- Install the OpenShift CLI (oc).
Procedure

- Specify the `oc` log level when running an `oc` command:

  ```bash
  $ oc <command> --loglevel <log_level>
  ```

  where:

  `<command>`
  Specifies the command you are running.

  `<log_level>`
  Specifies the log level to apply to the command.

- To obtain the current user’s session token, run the following command:

  ```bash
  $ oc whoami -t
  ```

Example output

```
sha256~RCV3Qcn7H-OEfqCGVI0CvnZ6...
```

### 6.8. TROUBLESHOOTING EXPIRED TOKENS

#### 6.8.1. Troubleshooting expired offline access tokens

If you use the Red Hat OpenShift Service on AWS (ROSA) CLI, `rosa`, and your api.openshift.com offline access token expires, an error message appears. This happens when sso.redhat.com invalidates the token.

Example output

```
Can't get tokens ....
Can't get access tokens ....
```

Procedure

- Generate a new offline access token at the following URL. A new offline access token is generated every time you visit the URL.

  - Red Hat OpenShift Service on AWS (ROSA):
    https://console.redhat.com/openshift/token/rosa

### 6.9. TROUBLESHOOTING IAM ROLES

#### 6.9.1. Resolving issues with ocm-roles and user-role IAM resources

You may receive an error when trying to create a cluster using the Red Hat OpenShift Service on AWS (ROSA) CLI, `rosa`.

Sample output
This error means that the user-role IAM role is not linked to your AWS account. The most likely cause of this error is that another user in your Red Hat organization created the ocm-role IAM role. Your user-role IAM role needs to be created.

NOTE

After any user sets up an ocm-role IAM resource linked to a Red Hat account, any subsequent users wishing to create a cluster in that Red Hat organization must have a user-role IAM role to provision a cluster.

Procedure

- Assess the status of your ocm-role and user-role IAM roles with the following commands:

  ```
  $ rosa list ocm-role
  Sample output
  
  I: Fetching ocm roles
  ROLE NAME                           ROLE ARN                                          LINKED  ADMIN
  ManagedOpenShift-OCM-Role-1158  arn:aws:iam::2066:role/ManagedOpenShift-OCM-Role-1158   No      No
  
  $ rosa list user-role
  Sample output
  
  I: Fetching user roles
  ROLE NAME                                   ROLE ARN                                        LINKED
  ManagedOpenShift-User.osdocs-Role  arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role  Yes
  
  With the results of these commands, you can create and link the missing IAM resources.

6.9.1.1. Creating an ocm-role IAM role

You create your ocm-role IAM roles by using the command-line interface (CLI).

Prerequisites

- You have an AWS account.
- You have Red Hat Organization Administrator privileges in the OpenShift Cluster Manager organization.
- You have the permissions required to install AWS account-wide roles.
- You have installed and configured the latest Red Hat OpenShift Service on AWS (ROSA) CLI, rosa, on your installation host.
Procedure

- To create an ocm-role IAM role with basic privileges, run the following command:

```sh
$ rosa create ocm-role
```

- To create an ocm-role IAM role with admin privileges, run the following command:

```sh
$ rosa create ocm-role --admin
```

This command allows you to create the role by specifying specific attributes. The following example output shows the “auto mode” selected, which lets the ROSA CLI (rosa) create your Operator roles and policies. See "Methods of account-wide role creation" in the Additional resources for more information.

Example output

```
I: Creating ocm role
? Role prefix: ManagedOpenShift
? Enable admin capabilities for the OCM role (optional): No
? Permissions boundary ARN (optional):
? Role Path (optional):
? Role creation mode: auto
I: Creating role using 'arn:aws:iam::<ARN>:user/<UserName>'
? Create the 'ManagedOpenShift-OCM-Role-182' role? Yes
I: Created role 'ManagedOpenShift-OCM-Role-182' with ARN 'arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182'
I: Linking OCM role
? OCM Role ARN: arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182
? Link the 'arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182' role with organization '<AWS ARN>'? Yes
I: Successfully linked role-arn 'arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182' with organization account '<AWS ARN>'
```

1. A prefix value for all of the created AWS resources. In this example, ManagedOpenShift prepends all of the AWS resources.

2. Choose if you want this role to have the additional admin permissions.

3. The Amazon Resource Name (ARN) of the policy to set permission boundaries.

4. Specify an IAM path for the user name.

5. Choose the method to create your AWS roles. Using auto, the ROSA CLI generates and links the roles and policies. In the auto mode, you receive some different prompts to create the AWS roles.

6. The auto method asks if you want to create a specific ocm-role using your prefix.

7. Confirm that you want to associate your IAM role with your OpenShift Cluster Manager.
6.9.1.2. Creating a user-role IAM role

You can create your user-role IAM roles by using the command-line interface (CLI).

**Prerequisites**

- You have an AWS account.
- You have installed and configured the latest Red Hat OpenShift Service on AWS (ROSA) CLI, `rosa`, on your installation host.

**Procedure**

- To create a user-role IAM role with basic privileges, run the following command:

  ```
  $ rosa create user-role
  ```

  This command allows you to create the role by specifying specific attributes. The following example output shows the "auto mode" selected, which lets the ROSA CLI (`rosa`) to create your Operator roles and policies. See "Understanding the auto and manual deployment modes" in the Additional resources for more information.

**Example output**

```
I: Creating User role
? Role prefix: ManagedOpenShift
? Permissions boundary ARN (optional):
? Role Path (optional):
? Role creation mode: auto
I: Creating ocm user role using 'arn:aws:iam::2066:user'
? Create the 'ManagedOpenShift-User.osdocs-Role' role? Yes
I: Created role 'ManagedOpenShift-User.osdocs-Role' with ARN 'arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role'
I: Linking User role
? User Role ARN: arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role
? Link the 'arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role' role with account '1AGE'? Yes
I: Successfully linked role ARN 'arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role' with account '1AGE'
```

1. A prefix value for all of the created AWS resources. In this example, `ManagedOpenShift` prepends all of the AWS resources.

2. The Amazon Resource Name (ARN) of the policy to set permission boundaries.

3. Specify an IAM path for the user name.

4. Choose the method to create your AWS roles. Using **auto**, the ROSA CLI generates and links the roles and policies. In the **auto** mode, you receive some different prompts to create the AWS roles.

5. The **auto** method asks if you want to create a specific user-role using your prefix.
Links the created role with your AWS organization.

6.9.1.3. Linking your AWS account

You can link your AWS account to existing IAM roles by using the Red Hat OpenShift Service on AWS (ROSA) CLI, `rosa`.

**Prerequisites**

- You have an AWS account.
- You are using OpenShift Cluster Manager Hybrid Cloud Console to create clusters.
- You have the permissions required to install AWS account-wide roles. See the "Additional resources" of this section for more information.
- You have installed and configured the latest AWS (`aws`) and ROSA (`rosa`) CLIs on your installation host.
- You have created your `ocm-role` and `user-role` IAM roles, but have not yet linked them to your AWS account. You can check whether your IAM roles are already linked by running the following commands:
  
  ```
  $ rosa list ocm-role
  $ rosa list user-role
  
  If Yes is displayed in the **Linked** column for both roles, you have already linked the roles to an AWS account.
  ```

**Procedure**

1. From the CLI, link your `ocm-role` resource to your Red Hat organization by using your Amazon Resource Name (ARN):

   ```
   $ rosa link ocm-role --role-arn <arn>
   ```

   **NOTE**

   You must have Red Hat Organization Administrator privileges to run the `rosa link` command. After you link the `ocm-role` resource with your AWS account, it is visible for all users in the organization.

   ```
   $ rosa link ocm-role --role-arn <arn>
   ```

   **Example output**

   ```
   I: Linking OCM role
   ? Link the `<AWS ACCOUNT ID>` role with organization `<ORG ID>`? Yes
   I: Successfully linked role-arn `<AWS ACCOUNT ID>` with organization account `<ORG ID>`
   ```

2. From the CLI, link your `user-role` resource to your Red Hat user account by using your Amazon Resource Name (ARN):
6.9.1.4. Associating multiple AWS accounts with your Red Hat organization

You can associate multiple AWS accounts with your Red Hat organization. Associating multiple accounts lets you create Red Hat OpenShift Service on AWS (ROSA) clusters on any of the associated AWS accounts from your Red Hat organization.

With this feature, you can create clusters in different AWS regions by using multiple AWS profiles as region-bound environments.

Prerequisites

- You have an AWS account.
- You are using OpenShift Cluster Manager Hybrid Cloud Console to create clusters.
- You have the permissions required to install AWS account-wide roles.
- You have installed and configured the latest AWS (aws) and ROSA (rosa) CLIs on your installation host.
- You have created your ocm-role and user-role IAM roles.

Procedure

To associate an additional AWS account, first create a profile in your local AWS configuration. Then, associate the account with your Red Hat organization by creating the ocm-role, user, and account roles in the additional AWS account.

To create the roles in an additional region, specify the --profile <aws_profile> parameter when running the rosa create commands and replace <aws_profile> with the additional account profile name:

- To specify an AWS account profile when creating an OpenShift Cluster Manager role:
  
  $ rosa create --profile <aws_profile> ocm-role

- To specify an AWS account profile when creating a user role:
  
  $ rosa create --profile <aws_profile> user-role

- To specify an AWS account profile when creating the account roles:
  
  $ rosa create --profile <aws_profile> account-roles
NOTE
If you do not specify a profile, the default AWS profile is used.

6.10. TROUBLESHOOTING CLUSTER DEPLOYMENTS
This document describes how to troubleshoot cluster deployment errors.

6.10.1. Obtaining information on a failed cluster
If a cluster deployment fails, the cluster is put into an "error" state.

Procedure
Run the following command to get more information:

```bash
$ rosa describe cluster -c <my_cluster_name> --debug
```

6.10.2. Failing to create a cluster with an osdCcsAdmin error
If a cluster creation action fails, you can receive the following error message.

Example output

```text
Failed to create cluster: Unable to create cluster spec: Failed to get access keys for user 'osdCcsAdmin': NoSuchEntity: The user with name osdCcsAdmin cannot be found.
```

Procedure
To fix this issue:

1. Delete the stack:
   ```bash
   $ rosa init --delete
   ```
2. Reinitialize your account:
   ```bash
   $ rosa init
   ```

6.10.3. Creating the Elastic Load Balancing (ELB) service-linked role
If you have not created a load balancer in your AWS account, it is possible that the service-linked role for Elastic Load Balancing (ELB) might not exist yet. You may receive the following error:

```text
```

Procedure
To resolve this issue, ensure that the role exists on your AWS account. If not, create this role with the following command:

```
aws iam get-role --role-name "AWSServiceRoleForElasticLoadBalancing" || aws iam create-service-linked-role --aws-service-name "elasticloadbalancing.amazonaws.com"
```

**NOTE**
This command only needs to be executed once per account.

### 6.10.4. Repairing a cluster that cannot be deleted

In specific cases, the following error appears in *OpenShift Cluster Manager Hybrid Cloud Console* if you attempt to delete your cluster.

#### Error deleting cluster

```
CLUSTERS-MGMT-400: Failed to delete cluster <hash>: sts_user_role is not linked to your account. sts_ocm_role is linked to your organization <org number> which requires sts_user_role to be linked to your Red Hat account <account ID>. Please create a user role and link it to the account: User Account <account ID> is not authorized to perform STS cluster operations
```

If you try to delete your cluster from the CLI, the following error appears.

```
E: Failed to delete cluster <hash>: sts_user_role is not linked to your account. sts_ocm_role is linked to your organization <org_number> which requires sts_user_role to be linked to your Red Hat account <account_id>. Please create a user role and link it to the account: User Account <account ID> is not authorized to perform STS cluster operations
```

This error occurs when the **user-role** is unlinked or deleted.

#### Procedure

1. Run the following command to create the **user-role** IAM resource:

```
$ rosa create user-role
```

2. After you see that the role has been created, you can delete the cluster. The following confirms that the role was created and linked:

```
I: Successfully linked role ARN <user role ARN> with account <account ID>
```

### 6.11. RED HAT OPENSHIFT SERVICE ON AWS MANAGED RESOURCES

#### 6.11.1. Overview

The following covers all resources managed or protected by the Service Reliability Engineering Platform (SRE-P) Team. Customers should not attempt to modify these resources because doing so can lead to cluster instability.
6.11.2. Hive managed resources

The following list displays the Red Hat OpenShift Service on AWS resources managed by OpenShift Hive, the centralized fleet configuration management system. These resources are in addition to the OpenShift Container Platform resources created during installation. OpenShift Hive continually attempts to maintain consistency across all Red Hat OpenShift Service on AWS clusters. Changes to Red Hat OpenShift Service on AWS resources should be made through OpenShift Cluster Manager so that OpenShift Cluster Manager and Hive are synchronized. Contact ocm-feedback@redhat.com if OpenShift Cluster Manager does not support modifying the resources in question.

Example 6.1. List of Hive managed resources

Resources:

ConfigMap:
- namespace: openshift-config
  name: rosa-brand-logo
- namespace: openshift-console
  name: custom-logo
- namespace: openshift-deployment-validation-operator
  name: deployment-validation-operator-config
- namespace: openshift-file-integrity
  name: fr-aide-conf
- namespace: openshift-managed-upgrade-operator
  name: managed-upgrade-operator-config
- namespace: openshift-monitoring
  name: cluster-monitoring-config
- namespace: openshift-monitoring
  name: managed-namespaces
- namespace: openshift-monitoring
  name: ocp-namespaces
- namespace: openshift-monitoring
  name: osd-rebalance-infra-nodes
- namespace: openshift-monitoring
  name: sre-dns-latency-exporter-code
- namespace: openshift-monitoring
  name: sre-dns-latency-exporter-trusted-ca-bundle
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter-code
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter-trusted-ca-bundle
- namespace: openshift-monitoring
  name: sre-stuck-ebs-vols-code
- namespace: openshift-monitoring
  name: sre-stuck-ebs-volstrusted-ca-bundle
- namespace: openshift-monitoring
  name: token-refresher-trusted-ca-bundle
- namespace: openshift-security
  name: osd-audit-policy
- namespace: openshift-validation-webhook
  name: webhook-cert
- namespace: openshift
  name: motd

Endpoints:
- namespace: openshift-deployment-validation-operator
  name: deployment-validation-operator-metrics
- namespace: openshift-monitoring
name: sre-dns-latency-exporter
- namespace: openshift-monitoring
name: sre-ebs-iops-reporter
- namespace: openshift-monitoring
name: sre-stuck-ebs-vols
- namespace: openshift-monitoring
name: token-refresher
- namespace: openshift-scanning
name: loggerservice
- namespace: openshift-security
name: audit-exporter
- namespace: openshift-validation-webhook
name: validation-webhook

Namespace:
- name: dedicated-admin
- name: openshift-addon-operator
- name: openshift-aqua
- name: openshift-aws-vpce-operator
- name: openshift-backplane
- name: openshift-backplane-cee
- name: openshift-backplane-csa
- name: openshift-backplane-cse
- name: openshift-backplane-csm
- name: openshift-backplane-managed-scripts
- name: openshift-backplane-mobb
- name: openshift-backplane-srep
- name: openshift-backplane-tam
- name: openshift-cloud-ingress-operator
- name: openshift-codeready-workspaces
- name: openshift-compliance
- name: openshift-container-security
- name: openshift-custom-domains-operator
- name: openshift-customer-monitoring
- name: openshift-deployment-validation-operator
- name: openshift-managed-node-metadata-operator
- name: openshift-file-integrity
- name: openshift-logging
- name: openshift-managed-upgrade-operator
- name: openshift-must-gather-operator
- name: openshift-observability-operator
- name: openshift-ocm-agent-operator
- name: openshift-operators-redhat
- name: openshift-osd-metrics
- name: openshift-rbac-permissions
- name: openshift-route-monitor-operator
- name: openshift-scanning
- name: openshift-security
- name: openshift-splunk-forwarder-operator
- name: openshift-sre-pruning
- name: openshift-strimzi
- name: openshift-suricata
- name: openshift-validation-webhook
- name: openshift-velero
- name: openshift-monitoring
- name: openshift
- name: openshift-cluster-version
ReplicationController:
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter-1
- namespace: openshift-monitoring
  name: sre-stuck-ebs-vols-1
Secret:
- namespace: openshift-authentication
  name: v4-0-config-user-idp-0-file-data
- namespace: openshift-authentication
  name: v4-0-config-user-template-error
- namespace: openshift-authentication
  name: v4-0-config-user-template-login
- namespace: openshift-authentication
  name: v4-0-config-user-template-provider-selection
- namespace: openshift-config
  name: htpasswd-secret
- namespace: openshift-config
  name: osd-oauth-templates-errors
- namespace: openshift-config
  name: osd-oauth-templates-login
- namespace: openshift-config
  name: osd-oauth-templates-providers
- namespace: openshift-config
  name: rosa-oauth-templates-errors
- namespace: openshift-config
  name: rosa-oauth-templates-login
- namespace: openshift-config
  name: rosa-oauth-templates-providers
- namespace: openshift-config
  name: support
- namespace: openshift-config
  name: tony-devlab-primary-cert-bundle-secret
- namespace: openshift-ingress
  name: tony-devlab-primary-cert-bundle-secret
- namespace: openshift-kube-apiserver
  name: user-serving-cert-000
- namespace: openshift-kube-apiserver
  name: user-serving-cert-001
- namespace: openshift-monitoring
  name: dms-secret
- namespace: openshift-monitoring
  name: observatorium-credentials
- namespace: openshift-monitoring
  name: pd-secret
- namespace: openshift-scanning
  name: clam-secrets
- namespace: openshift-scanning
  name: logger-secrets
- namespace: openshift-security
  name: splunk-auth
ServiceAccount:
- namespace: openshift-backplane-managed-scripts
  name: osd-backplane
- namespace: openshift-backplane-srep
  name: 6804d07fb268b8285b023bcf65392f0e
- namespace: openshift-backplane-srep
name: osd-delete-ownerrefs-serviceaccounts
- namespace: openshift-backplane
  name: osd-delete-backplane-serviceaccounts
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator
- namespace: openshift-custom-domains-operator
  name: custom-domains-operator
- namespace: openshift-managed-upgrade-operator
  name: managed-upgrade-operator
- namespace: openshift-machine-api
  name: osd-disable-cpms
- namespace: openshift-marketplace
  name: osd-patch-subscription-source
- namespace: openshift-monitoring
  name: configure-alertmanager-operator
- namespace: openshift-monitoring
  name: osd-cluster-ready
- namespace: openshift-monitoring
  name: osd-rebalance-infra-nodes
- namespace: openshift-monitoring
  name: sre-dns-latency-exporter
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter
- namespace: openshift-monitoring
  name: sre-stuck-ebs-vols
- namespace: openshift-monitoring
  name: sre-pod-network-connectivity-check-pruner
- namespace: openshift-ocm-agent-operator
  name: ocm-agent-operator
- namespace: openshift-rbac-permissions
  name: rbac-permissions-operator
- namespace: openshift-sre-pruning
  name: bz1980755
- namespace: openshift-scanning
  name: logger-sa
- namespace: openshift-scanning
  name: scanner-sa
- namespace: openshift-sre-pruning
  name: sre-pruner-sa
- namespace: openshift-suricata
  name: ids-test
- namespace: openshift-suricata
  name: suricata-sa
- namespace: openshift-validation-webhook
  name: validation-webhook
- namespace: openshift-velero
  name: managed-velero-operator
- namespace: openshift-velero
  name: velero
- namespace: openshift-backplane-srep
  name: UNIQUE_BACKPLANE_SERVICEACCOUNT_ID
Service:
- namespace: openshift-deployment-validation-operator
  name: deployment-validation-operator-metrics
- namespace: openshift-monitoring
  name: sre-dns-latency-exporter
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter
- namespace: openshift-monitoring
  name: sre-stuck-ebs-vols
- namespace: openshift-monitoring
  name: token-refresher
- namespace: openshift-scanning
  name: loggerservice
- namespace: openshift-security
  name: audit-exporter
- namespace: openshift-validation-webhook
  name: validation-webhook
AddonOperator:
  - name: addon-operator
ValidatingWebhookConfiguration:
  - name: sre-hiveownership-validation
  - name: sre-namespace-validation
  - name: sre-pod-validation
  - name: sre-prometheusrule-validation
  - name: sre-regular-user-validation
  - name: sre-scc-validation
  - name: sre-techpreviounoupgrade-validation
DaemonSet:
  - namespace: openshift-monitoring
    name: sre-dns-latency-exporter
  - namespace: openshift-scanning
    name: logger
  - namespace: openshift-scanning
    name: scanner
  - namespace: openshift-security
    name: audit-exporter
  - namespace: openshift-suricata
    name: suricata
  - namespace: openshift-validation-webhook
    name: validation-webhook
Deployment:
  - namespace: openshift-monitoring
    name: token-refresher
DeploymentConfig:
  - namespace: openshift-monitoring
    name: sre-ebs-iops-reporter
  - namespace: openshift-monitoring
    name: sre-stuck-ebs-vols
ClusterRoleBinding:
  - name: aqua-scanner-binding
  - name: backplane-cluster-admin
  - name: backplane-impersonate-cluster-admin
  - name: bz1980755
  - name: configure-alertmanager-operator-prom
  - name: dedicated-admins-cluster
  - name: dedicated-admins-registry-cas-cluster
  - name: logger-clusterrolebinding
  - name: openshift-backplane-managed-scripts-reader
  - name: osd-cluster-admin
- name: osd-cluster-ready
- name: osd-delete-backplane-script-resources
- name: osd-delete-ownerrefs-serviceaccounts
- name: osd-patch-subscription-source
- name: osd-rebalance-infra-nodes
- name: pcap-dedicated-admins
- name: splunk-forwarder-operator
- name: splunk-forwarder-operator-clusterrolebinding
- name: sre-pod-network-connectivity-check-pruner
- name: sre-pruner-buildsdeploys-pruning
- name: velero
- name: webhook-validation

ClusterRole:
- name: backplane-cee-readers-cluster
- name: backplane-impersonate-cluster-admin
- name: backplane-readers-cluster
- name: backplane-srep-admins-cluster
- name: backplane-srep-admins-project
- name: bz1980755
- name: dedicated-admins-aggregate-cluster
- name: dedicated-admins-aggregate-project
- name: dedicated-admins-cluster
- name: dedicated-admins-manage-operators
- name: dedicated-admins-project
- name: dedicated-admins-registry-cas-cluster
- name: dedicated-readers
- name: image-scanner
- name: logger-clusterrole
- name: openshift-backplane-managed-scripts-reader
- name: openshift-splunk-forwarder-operator
- name: osd-cluster-ready
- name: osd-custom-domains-dedicated-admin-cluster
- name: osd-delete-backplane-script-resources
- name: osd-delete-backplane-serviceaccounts
- name: osd-delete-ownerrefs-serviceaccounts
- name: osd-get-namespace
- name: osd-netnamespaces-dedicated-admin-cluster
- name: osd-patch-subscription-source
- name: osd-readers-aggregate
- name: osd-rebalance-infra-nodes
- name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- name: pcap-dedicated-admins
- name: splunk-forwarder-operator
- name: sre-allow-read-machine-info
- name: sre-pruner-buildsdeploys-cr
- name: webhook-validation-cr

RoleBinding:
- namespace: kube-system
  name: cloud-ingress-operator-cluster-config-v1-reader
- namespace: kube-system
  name: managed-velero-operator-cluster-config-v1-reader
- namespace: openshift-aqua
  name: dedicated-admins-openshift-aqua
- namespace: openshift-backplane-managed-scripts
  name: backplane-cee-mustgather
- namespace: openshift-backplane-managed-scripts
name: backplane-srep-mustgather
- namespace: openshift-backplane-managed-scripts
  name: osd-delete-backplane-script-resources
- namespace: openshift-cloud-ingress-operator
  name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-codeready-workspaces
  name: dedicated-admins-openshift-codeready-workspaces
- namespace: openshift-config
  name: dedicated-admins-project-request
- namespace: openshift-config
  name: dedicated-admins-registry-cas-project
- namespace: openshift-cloud-ingress-operator
  name: muo-pullsecret-reader
- namespace: openshift-config
  name: oao-openshiftconfig-reader
- namespace: openshift-config
  name: osd-cluster-ready
- namespace: openshift-custom-domains-operator
  name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-customer-monitoring
  name: dedicated-admins-openshift-customer-monitoring
- namespace: openshift-customer-monitoring
  name: prometheus-k8s-openshift-customer-monitoring
- namespace: openshift-dns
  name: dedicated-admins-openshift-dns
- namespace: openshift-dns
  name: osd-rebalance-infra-nodes-openshift-dns
- namespace: openshift-image-registry
  name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-ingress-operator
  name: cloud-ingress-operator
- namespace: openshift-ingress
  name: cloud-ingress-operator
- namespace: openshift-kube-apiserver
  name: cloud-ingress-operator
- namespace: openshift-machine-api
  name: cloud-ingress-operator
- namespace: openshift-logging
  name: admin-dedicated-admins
- namespace: openshift-logging
  name: admin-system:serviceaccounts:dedicated-admin
- namespace: openshift-logging
  name: openshift-logging-dedicated-admins
- namespace: openshift-logging
  name: openshift-logging:serviceaccounts:dedicated-admin
- namespace: openshift-machine-api
  name: osd-cluster-ready
- namespace: openshift-machine-api
  name: sre-ebs-iops-reporter-read-machine-info
- namespace: openshift-machine-api
  name: sre-stuck-ebs-vols-read-machine-info
- namespace: openshift-managed-node-metadata-operator
  name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-machine-api
  name: osd-disable-cpms
- namespace: openshift-marketplace
- namespace: openshift-customer-monitoring
- namespace: openshift-dns
- namespace: openshift-image-registry
name: dedicated-admins-openshift-marketplace
- namespace: openshift-monitoring
ame: backplane-cee
- namespace: openshift-monitoring
name: muo-monitoring-reader
- namespace: openshift-monitoring
name: oao-monitoring-manager
- namespace: openshift-monitoring
name: osd-cluster-ready
- namespace: openshift-monitoring
name: osd-rebalance-infra-nodes-openshift-monitoring
- namespace: openshift-monitoring
name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-monitoring
name: sre-dns-latency-exporter
- namespace: openshift-monitoring
name: sre-ebs-iops-reporter
- namespace: openshift-monitoring
name: sre-stuck-ebs-vols
- namespace: openshift-must-gather-operator
name: backplane-cee-mustgather
- namespace: openshift-must-gather-operator
name: backplane-srep-mustgather
- namespace: openshift-must-gather-operator
name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-network-diagnostics
name: sre-pod-network-connectivity-check-pruner
- namespace: openshift-network-operator
name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-operators-redhat
name: admin-dedicated-admins
- namespace: openshift-operators-redhat
name: admin-system:serviceaccounts:dedicated-admin
- namespace: openshift-operators-redhat
name: openshift-operators-redhat-dedicated-admins
- namespace: openshift-operators-redhat
name: openshift-operators-redhat:serviceaccounts:dedicated-admin
- namespace: openshift-operators
name: dedicated-admins-openshift-operators
- namespace: openshift-osd-metrics
name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-osd-metrics
name: prometheus-k8s
- namespace: openshift-rbac-permissions
name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-rbac-permissions
name: prometheus-k8s
- namespace: openshift-route-monitor-operator
name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-scanning
name: scanner-rolebinding
- namespace: openshift-security
name: osd-rebalance-infra-nodes-openshift-security
- namespace: openshift-security
name: osd-rebalance-infra-nodes-openshift-security
name: prometheus-k8s
- namespace: openshift-splunk-forwarder-operator
  name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-strimzi
  name: dedicated-admins-openshift-strimzi
- namespace: openshift-suricata
  name: suricata-rolebinding
- namespace: openshift-user-workload-monitoring
  name: dedicated-admins-uwm-config-create
- namespace: openshift-user-workload-monitoring
  name: dedicated-admins-uwm-config-edit
- namespace: openshift-user-workload-monitoring
  name: dedicated-admins-uwm-managed-am-secret
- namespace: openshift-user-workload-monitoring
  name: osd-rebalance-infra-nodes-openshift-user-workload-monitoring
- namespace: openshift-velero
  name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- namespace: openshift-velero
  name: prometheus-k8s
Role:
- namespace: kube-system
  name: cluster-config-v1-reader
- namespace: kube-system
  name: cluster-config-v1-reader-cio
- namespace: openshift-aqua
  name: dedicated-admins-openshift-aqua
- namespace: openshift-backplane-managed-scripts
  name: backplane-cee-pcap-collector
- namespace: openshift-backplane-managed-scripts
  name: backplane-srep-pcap-collector
- namespace: openshift-backplane-managed-scripts
  name: osd-delete-backplane-script-resources
- namespace: openshift-codeready-workspaces
  name: dedicated-admins-openshift-codeready-workspaces
- namespace: openshift-config
  name: dedicated-admins-project-request
- namespace: openshift-config
  name: dedicated-admins-registry-cas-project
- namespace: openshift-config
  name: muo-pullsecret-reader
- namespace: openshift-config
  name: oao-openshiftconfig-reader
- namespace: openshift-config
  name: osd-cluster-ready
- namespace: openshift-customer-monitoring
  name: dedicated-admins-openshift-customer-monitoring
- namespace: openshift-customer-monitoring
  name: prometheus-k8s-openshift-customer-monitoring
- namespace: openshift-dns
  name: dedicated-admins-openshift-dns
- namespace: openshift-dns
  name: osd-rebalance-infra-nodes-openshift-dns
- namespace: openshift-ingress-operator
  name: cloud-ingress-operator
- namespace: openshift-ingress
  name: cloud-ingress-operator
- namespace: openshift-kube-apiserver
  name: cloud-ingress-operator
- namespace: openshift-machine-api
  name: cloud-ingress-operator
- namespace: openshift-logging
  name: dedicated-admins-openshift-logging
- namespace: openshift-machine-api
  name: osd-cluster-ready
- namespace: openshift-machine-api
  name: osd-disable-cpms
- namespace: openshift-marketplace
  name: dedicated-admins-openshift-marketplace
- namespace: openshift-monitoring
  name: backplane-cee
- namespace: openshift-monitoring
  name: muo-monitoring-reader
- namespace: openshift-monitoring
  name: oao-monitoring-manager
- namespace: openshift-monitoring
  name: osd-cluster-ready
- namespace: openshift-monitoring
  name: osd-rebalance-infra-nodes-openshift-monitoring
- namespace: openshift-must-gather-operator
  name: backplane-cee-mustgather
- namespace: openshift-must-gather-operator
  name: backplane-srep-mustgather
- namespace: openshift-network-diagnostics
  name: sre-pod-network-connectivity-check-pruner
- namespace: openshift-operators
  name: dedicated-admins-openshift-operators
- namespace: openshift-osd-metrics
  name: prometheus-k8s
- namespace: openshift-rbac-permissions
  name: prometheus-k8s
- namespace: openshift-scanning
  name: scanner-role
- namespace: openshift-security
  name: osd-rebalance-infra-nodes-openshift-security
- namespace: openshift-security
  name: prometheus-k8s
- namespace: openshift-strimzi
  name: dedicated-admins-openshift-strimzi
- namespace: openshift-suricata
  name: suricata-role
- namespace: openshift-user-workload-monitoring
  name: dedicated-admins-user-workload-monitoring-create-cm
- namespace: openshift-user-workload-monitoring
  name: dedicated-admins-user-workload-monitoring-manage-am-secret
- namespace: openshift-user-workload-monitoring
  name: osd-rebalance-infra-nodes-openshift-user-workload-monitoring
- namespace: openshift-velero
  name: prometheus-k8s
CronJob:
- namespace: openshift-backplane-managed-scripts
  name: osd-delete-backplane-script-resources
- namespace: openshift-backplane-srep
name: osd-delete-ownerrrefs-serviceaccounts
- namespace: openshift-backplane
  name: osd-delete-backplane-serviceaccounts
- namespace: openshift-machine-api
  name: osd-disable-cpms
- namespace: openshift-marketplace
  name: osd-patch-subscription-source
- namespace: openshift-monitoring
  name: osd-rebalance-infra-nodes
- namespace: openshift-network-diagnostics
  name: sre-pod-network-connectivity-check-pruner
- namespace: openshift-sre-pruning
  name: builds-pruner
- namespace: openshift-sre-pruning
  name: bz1980755
- namespace: openshift-sre-pruning
  name: deployments-pruner
- namespace: openshift-suricata
  name: ids-tester
Job:
- namespace: openshift-monitoring
  name: osd-cluster-ready
CredentialsRequest:
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator-credentials-aws
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator-credentials-gcp
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter-aws-credentials
- namespace: openshift-monitoring
  name: sre-stuck-ebs-vols-aws-credentials
- namespace: openshift-velero
  name: managed-velero-operator-iam-credentials-aws
- namespace: openshift-velero
  name: managed-velero-operator-iam-credentials-gcp
APIScheme:
- namespace: openshift-cloud-ingress-operator
  name: rh-api
PublishingStrategy:
- namespace: openshift-cloud-ingress-operator
  name: publishingstrategy
ScanSettingBinding:
- namespace: openshift-compliance
  name: fedramp-high-ocp
- namespace: openshift-compliance
  name: fedramp-high-rhcos
ScanSetting:
- namespace: openshift-compliance
  name: osd
TailoredProfile:
- namespace: openshift-compliance
  name: rhcos4-high-rosa
OAuth:
- name: cluster
EndpointSlice:
name: deployment-validation-operator-metrics-rhtwg
- namespace: openshift-monitoring
  name: sre-dns-latency-exporter-4cw9r
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter-6tx5g
- namespace: openshift-monitoring
  name: sre-stuck-ebs-vols-gmdhs
- namespace: openshift-monitoring
  name: token-refresher-v5cpg
- namespace: openshift-scanning
  name: loggerservice-zprbq
- namespace: openshift-security
  name: audit-exporter-nqfdk
- namespace: openshift-validation-webhook
  name: validation-webhook-97b8t
FileIntegrity:
- namespace: openshift-file-integrity
  name: osd-fileintegrity
MachineHealthCheck:
- namespace: openshift-machine-api
  name: srep-infra-healthcheck
- namespace: openshift-machine-api
  name: srep-metal-worker-healthcheck
- namespace: openshift-machine-api
  name: srep-worker-healthcheck
MachineSet:
- namespace: openshift-machine-api
  name: sbasabat-mc-qhqkn-infra-us-east-1a
- namespace: openshift-machine-api
  name: sbasabat-mc-qhqkn-worker-us-east-1a
ContainerRuntimeConfig:
- name: custom-crio
KubeletConfig:
- name: custom-kubelet
MachineConfig:
- name: 00-master-chrony
- name: 00-worker-chrony
SubjectPermission:
- namespace: openshift-rbac-permissions
  name: backplane-cee
- namespace: openshift-rbac-permissions
  name: backplane-csa
- namespace: openshift-rbac-permissions
  name: backplane-cse
- namespace: openshift-rbac-permissions
  name: backplane-csm
- namespace: openshift-rbac-permissions
  name: backplane-mobb
- namespace: openshift-rbac-permissions
  name: backplane-srep
- namespace: openshift-rbac-permissions
  name: backplane-tam
- namespace: openshift-rbac-permissions
  name: dedicated-admin-serviceaccounts
- namespace: openshift-rbac-permissions
  name: dedicated-admin-serviceaccounts-core-ns
CHAPTER 6. TROUBLESHOOTING

- namespace: openshift-rbac-permissions
  name: dedicated-admins
- namespace: openshift-rbac-permissions
  name: dedicated-admins-alert-routing-edit
- namespace: openshift-rbac-permissions
  name: dedicated-admins-core-ns
- namespace: openshift-rbac-permissions
  name: dedicated-admins-customer-monitoring
- namespace: openshift-rbac-permissions
  name: osd-delete-backplane-serviceaccounts

VeleroInstall:
- namespace: openshift-velero
  name: cluster

PrometheusRule:
- namespace: openshift-monitoring
  name: rhmi-sre-cluster-admins
- namespace: openshift-monitoring
  name: rhoam-sre-cluster-admins
- namespace: openshift-monitoring
  name: sre-alertmanager-silences-active
- namespace: openshift-monitoring
  name: sre-alerts-stuck-builds
- namespace: openshift-monitoring
  name: sre-alerts-stuck-volumes
- namespace: openshift-monitoring
  name: sre-cloud-ingress-operator-offline-alerts
- namespace: openshift-monitoring
  name: sre-avo-pendingacceptance
- namespace: openshift-monitoring
  name: sre-configure-alertmanager-operator-offline-alerts
- namespace: openshift-monitoring
  name: sre-control-plane-resizing-alerts
- namespace: openshift-monitoring
  name: sre-dns-alerts
- namespace: openshift-monitoring
  name: sre-ebs-iops-burstbalance
- namespace: openshift-monitoring
  name: sre-elasticsearch-jobs
- namespace: openshift-monitoring
  name: sre-elasticsearch-managed-notification-alerts
- namespace: openshift-monitoring
  name: sre-excessive-memory
- namespace: openshift-monitoring
  name: sre-fr-alerts-low-disk-space
- namespace: openshift-monitoring
  name: sre-haproxy-reload-fail
- namespace: openshift-monitoring
  name: sre-internal-slo-recording-rules
- namespace: openshift-monitoring
  name: sre-leader-election-master-status-alerts
- namespace: openshift-monitoring
  name: sre-managed-kube-apiserver-missing-on-node
- namespace: openshift-monitoring
  name: sre-managed-kube-controller-manager-missing-on-node
- namespace: openshift-monitoring
  name: sre-managed-kube-scheduler-missing-on-node
- namespace: openshift-monitoring
  name: sre-managed-node-metadata-operator-alerts
- namespace: openshift-monitoring
  name: sre-managed-notification-alerts
- namespace: openshift-monitoring
  name: sre-managed-upgrade-operator-alerts
- namespace: openshift-monitoring
  name: sre-managed-velero-operator-alerts
- namespace: openshift-monitoring
  name: sre-node-unschedulable
- namespace: openshift-monitoring
  name: sre-oauth-server
- namespace: openshift-monitoring
  name: sre-pending-csr-alert
- namespace: openshift-monitoring
  name: sre-proxy-managed-notification-alerts
- namespace: openshift-monitoring
  name: sre-pruning
- namespace: openshift-monitoring
  name: sre-pv
- namespace: openshift-monitoring
  name: sre-router-health
- namespace: openshift-monitoring
  name: sre-runaway-sdn-preventing-container-creation
- namespace: openshift-monitoring
  name: sre-slo-recording-rules
- namespace: openshift-monitoring
  name: sre-telemeter-client
- namespace: openshift-monitoring
  name: sre-telemetry-managed-labels-recording-rules
- namespace: openshift-monitoring
  name: sre-upgrade-send-managed-notification-alerts
- namespace: openshift-monitoring
  name: sre-uptime-sla
ServiceMonitor:
- namespace: openshift-monitoring
  name: sre-dns-latency-exporter
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter
- namespace: openshift-monitoring
  name: sre-stuck-ebs-vols
ClusterUrlMonitor:
- namespace: openshift-route-monitor-operator
  name: api
RouteMonitor:
- namespace: openshift-route-monitor-operator
  name: console
NetworkPolicy:
- namespace: openshift-deployment-validation-operator
  name: allow-from-openshift-insights
- namespace: openshift-deployment-validation-operator
  name: allow-from-openshift-olm
- namespace: openshift-monitoring
  name: token-refresher
ManagedNotification:
- namespace: openshift-ocm-agent-operator
  name: sre-elasticsearch-managed-notifications
- namespace: openshift-ocm-agent-operator
  name: sre-managed-notifications
- namespace: openshift-ocm-agent-operator
  name: sre-proxy-managed-notifications
- namespace: openshift-ocm-agent-operator
  name: sre-upgrade-managed-notifications
OcmAgent:
- namespace: openshift-ocm-agent-operator
  name: ocmagent
- namespace: openshift-security
  name: audit-exporter
Console:
- name: cluster
CatalogSource:
- namespace: openshift-addon-operator
  name: addon-operator-catalog
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator-registry
- namespace: openshift-compliance
  name: compliance-operator-registry
- namespace: openshift-container-security
  name: container-security-operator-registry
- namespace: openshift-custom-domains-operator
  name: custom-domains-operator-registry
- namespace: openshift-deployment-validation-operator
  name: deployment-validation-operator-catalog
- namespace: openshift-managed-node-metadata-operator
  name: managed-node-metadata-operator-registry
- namespace: openshift-file-integrity
  name: file-integrity-operator-registry
- namespace: openshift-managed-upgrade-operator
  name: managed-upgrade-operator-catalog
- namespace: openshift-monitoring
  name: configure-alertmanager-operator-registry
- namespace: openshift-must-gather-operator
  name: must-gather-operator-registry
- namespace: openshift-observability-operator
  name: observability-operator-catalog
- namespace: openshift-ocm-agent-operator
  name: ocm-agent-operator-registry
- namespace: openshift-osd-metrics
  name: osd-metrics-exporter-registry
- namespace: openshift-rbac-permissions
  name: rbac-permissions-operator-registry
- namespace: openshift-route-monitor-operator
  name: route-monitor-operator-registry
- namespace: openshift-splunk-forwarder-operator
  name: splunk-forwarder-operator-catalog
- namespace: openshift-velero
  name: managed-velero-operator-registry
OperatorGroup:
- namespace: openshift-aqua
  name: openshift-aqua
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator
- namespace: openshift-codeready-workspaces
  name: openshift-codeready-workspaces
- namespace: openshift-compliance
  name: compliance-operator
- namespace: openshift-container-security
  name: container-security-operator
- namespace: openshift-custom-domains-operator
  name: custom-domains-operator
- namespace: openshift-customer-monitoring
  name: openshift-customer-monitoring
- namespace: openshift-deployment-validation-operator
  name: deployment-validation-operator-og
- namespace: openshift-managed-node-metadata-operator
  name: managed-node-metadata-operator
- namespace: openshift-file-integrity
  name: file-integrity-operator
- namespace: openshift-logging
  name: openshift-logging
- namespace: openshift-managed-upgrade-operator
  name: managed-upgrade-operator-og
- namespace: openshift-must-gather-operator
  name: must-gather-operator
- namespace: openshift-observability-operator
  name: observability-operator-og
- namespace: openshift-ocm-agent-operator
  name: ocm-agent-operator-og
- namespace: openshift-osd-metrics
  name: osd-metrics-exporter
- namespace: openshift-rbac-permissions
  name: rbac-permissions-operator
- namespace: openshift-route-monitor-operator
  name: route-monitor-operator
- namespace: openshift-splunk-forwarder-operator
  name: splunk-forwarder-operator-og
- namespace: openshift-strimzi
  name: openshift-strimzi
- namespace: openshift-velero
  name: managed-velero-operator
Subscription:
- namespace: openshift-addon-operator
  name: addon-operator
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator
- namespace: openshift-compliance
  name: compliance-operator-sub
- namespace: openshift-container-security
  name: container-security-operator-sub
- namespace: openshift-custom-domains-operator
  name: custom-domains-operator
- namespace: openshift-deployment-validation-operator
  name: deployment-validation-operator
- namespace: openshift-managed-node-metadata-operator
  name: openshift-managed-node-metadata-operator
name: managed-node-metadata-operator
- namespace: openshift-file-integrity
name: file-integrity-operator-sub
- namespace: openshift-managed-upgrade-operator
name: managed-upgrade-operator
- namespace: openshift-monitoring
name: configure-alertmanager-operator
- namespace: openshift-must-gather-operator
name: must-gather-operator
- namespace: openshift-observability-operator
name: observability-operator
- namespace: openshift-ocm-agent-operator
name: ocm-agent-operator
- namespace: openshift-osd-metrics
name: osd-metrics-exporter
- namespace: openshift-rbac-permissions
name: rbac-permissions-operator
- namespace: openshift-route-monitor-operator
name: route-monitor-operator
- namespace: openshift-splunk-forwarder-operator
name: openshift-splunk-forwarder-operator
- namespace: openshift-velero
name: managed-velero-operator
PackageManifest:
- namespace: openshift-splunk-forwarder-operator
name: splunk-forwarder-operator
- namespace: openshift-addon-operator
name: addon-operator
- namespace: openshift-rbac-permissions
name: rbac-permissions-operator
- namespace: openshift-cloud-ingress-operator
name: cloud-ingress-operator
- namespace: openshift-managed-node-metadata-operator
name: managed-node-metadata-operator
- namespace: openshift-velero
name: managed-velero-operator
- namespace: openshift-deployment-validation-operator
name: managed-upgrade-operator
- namespace: openshift-managed-upgrade-operator
name: managed-upgrade-operator
- namespace: openshift-container-security
name: container-security-operator
- namespace: openshift-route-monitor-operator
name: route-monitor-operator
- namespace: openshift-file-integrity
name: file-integrity-operator
- namespace: openshift-custom-domains-operator
name: managed-node-metadata-operator
- namespace: openshift-route-monitor-operator
name: custom-domains-operator
- namespace: openshift-managed-upgrade-operator
name: managed-upgrade-operator
- namespace: openshift-ocm-agent-operator
name: ocm-agent-operator
- namespace: openshift-observability-operator
name: observability-operator
- namespace: openshift-monitoring
  name: configure-alertmanager-operator
- namespace: openshift-must-gather-operator
  name: deployment-validation-operator
- namespace: openshift-osd-metrics
  name: osd-metrics-exporter
- namespace: openshift-compliance
  name: compliance-operator
- namespace: openshift-rbac-permissions
  name: rbac-permissions-operator

Status:
- {}

Project:
- name: dedicated-admin
- name: openshift-addon-operator
- name: openshift-aqua
- name: openshift-backplane
- name: openshift-backplane-cee
- name: openshift-backplane-csa
- name: openshift-backplane-cse
- name: openshift-backplane-csm
- name: openshift-backplane-managed-scripts
- name: openshift-backplane-mobb
- name: openshift-backplane-srep
- name: openshift-backplane-tam
- name: openshift-cloud-ingress-operator
- name: openshift-codeready-workspaces
- name: openshift-compliance
- name: openshift-container-security
- name: openshift-custom-domains-operator
- name: openshift-customer-monitoring
- name: openshift-deployment-validation-operator
- name: openshift-managed-node-metadata-operator
- name: openshift-file-integrity
- name: openshift-logging
- name: openshift-managed-upgrade-operator
- name: openshift-must-gather-operator
- name: openshift-observability-operator
- name: openshift-ocm-agent-operator
- name: openshift-operators-redhat
- name: openshift-osd-metrics
- name: openshift-rbac-permissions
- name: openshift-route-monitor-operator
- name: openshift-scanning
- name: openshift-security
- name: openshift-splunk-forwarder-operator
- name: openshift-sre-pruning
- name: openshift-strimzi
- name: openshift-suricata
- name: openshift-validation-webhook
- name: openshift-velero

ClusterResourceQuota:
- name: loadbalancer-quota
- name: persistent-volume-quota

SecurityContextConstraints:
- name: osd-scanning-scc
- name: osd-suricata-scc
- name: pcap-dedicated-admins
- name: splunkforwarder
SplunkForwarder:
  - namespace: openshift-security
    name: splunkforwarder
Group:
  - name: clusteradmins
  - name: dedicated-admins
User:
  - name: backplane-cluster-admin
Backup:
  - namespace: openshift-velero
    name: daily-full-backup-20221123112305
  - namespace: openshift-velero
    name: daily-full-backup-20221125042537
  - namespace: openshift-velero
    name: daily-full-backup-20221126010038
  - namespace: openshift-velero
    name: daily-full-backup-20221127010039
  - namespace: openshift-velero
    name: daily-full-backup-20221128010040
  - namespace: openshift-velero
    name: daily-full-backup-20221129050847
  - namespace: openshift-velero
    name: hourly-object-backup-20221128051740
  - namespace: openshift-velero
    name: hourly-object-backup-20221128061740
  - namespace: openshift-velero
    name: hourly-object-backup-20221128071740
  - namespace: openshift-velero
    name: hourly-object-backup-20221128081740
  - namespace: openshift-velero
    name: hourly-object-backup-20221128091740
  - namespace: openshift-velero
    name: hourly-object-backup-20221129050852
  - namespace: openshift-velero
    name: hourly-object-backup-20221129051747
  - namespace: openshift-velero
    name: weekly-full-backup-20221116184315
  - namespace: openshift-velero
    name: weekly-full-backup-20221121033854
  - namespace: openshift-velero
    name: weekly-full-backup-20221128020040
Schedule:
  - namespace: openshift-velero
    name: daily-full-backup
  - namespace: openshift-velero
    name: hourly-object-backup
  - namespace: openshift-velero
    name: weekly-full-backup

6.11.3. Red Hat OpenShift Service on AWS add-on namespaces
Red Hat OpenShift Service on AWS add-ons are services available for installation after cluster installation. These additional services include Red Hat OpenShift Dev Spaces, Red Hat OpenShift API Management, and Cluster Logging Operator. Any changes to resources within the following namespaces can be overridden by the add-on during upgrades, which can lead to unsupported configurations for the add-on functionality.

**Example 6.2. List of add-on managed namespaces**

```
addon-namespaces:
  ocs-converged-dev: openshift-storage
  managed-api-service-internal: redhat-rhoami-operator
  codeready-workspaces-operator: codeready-workspaces-operator
  managed-odh: redhat-ods-operator
  codeready-workspaces-operator-qe: codeready-workspaces-operator-qe
  integrally-operator: redhat-rhmi-operator
  nvidia-gpu-addon: redhat-nvidia-gpu-addon
  integrally-operator-internal: redhat-rhmi-operator
  rhosak-qe: redhat-managed-kafka-operator-qe
  rhoams: redhat-rhoam-operator
  ocs-converged: openshift-storage
  addon-operator: redhat-addon-operator
  rhosak: redhat-managed-kafka-operator
  kas-fleetshard-operator-qe: redhat-kas-fleetshard-operator-qe
  prow-operator: prow
  cluster-logging-operator: openshift-logging
  advanced-cluster-management: redhat-open-cluster-management
  cert-manager-operator: redhat-cert-manager-operator
  dba-operator: addon-dba-operator
  reference-addon: redhat-reference-addon
  ocm-addon-test-operator: redhat-ocm-addon-test-operator
  kas-fleetshard-operator: redhat-kas-fleetshard-operator
  connectors-operator: redhat-openshift-connectors
```

**6.11.4. Red Hat OpenShift Service on AWS validating webhooks**

Red Hat OpenShift Service on AWS validating webhooks are a set of dynamic admission controls maintained by the OpenShift SRE team. These HTTP callbacks, also known as webhooks, are called for various types of requests to ensure cluster stability. The following list describes the various webhooks with rules containing the registered operations and resources that are controlled. Any attempt to circumvent these validating webhooks could affect the stability and supportability of the cluster.

**Example 6.3. List of validating webhooks**

```
[
  {
    "webhookName": "clusterlogging-validation",
    "rules": [
      {
        "operations": [
          "CREATE",
          "UPDATE"
        ],
        "apiGroups": [
```
"logging.openshift.io",
"apiVersions": [
  "v1"
],
"resources": [
  "clusterloggings"
],
"scope": "Namespaced"
},
"documentString": "Managed OpenShift Customers may set log retention outside the allowed range of 0-7 days"
},
{
  "webhookName": "clusterrolebindings-validation",
  "rules": [
    {
      "operations": [
        "DELETE"
      ],
      "apiGroups": [
        "rbac.authorization.k8s.io"
      ],
      "apiVersions": [
        "v1"
      ],
      "resources": [
        "clusterrolebindings"
      ],
      "scope": "Cluster"
    }
  ],
  "documentString": "Managed OpenShift Customers may not delete the cluster role bindings under the managed namespaces: (^openshift-.|kube-system)"
},
{
  "webhookName": "hiveownership-validation",
  "rules": [
    {
      "operations": [
        "UPDATE",
        "DELETE"
      ],
      "apiGroups": [
        "quota.openshift.io"
      ],
      "apiVersions": [
        "*"
      ],
      "resources": [
        "clusterresourcequotas"
      ],
      "scope": "Cluster"
    }
  ]
}
"webhookObjectSelector": {
  "matchLabels": {
    "hive.openshift.io/managed": "true"
  }
},
"documentString": "Managed OpenShift customers may not edit certain managed resources. A managed resource has a \"hive.openshift.io/managed\": \"true\" label."
},
{  "webhookName": "imagecontentpolicies-validation",
  "rules": [
    {
      "operations": [
        "CREATE",
        "UPDATE"
      ],
      "apiGroups": [
        "config.openshift.io"
      ],
      "apiVersions": [
        "*"
      ],
      "resources": [
        "imagedigestmirrorsets",
        "imagetagmirrorsets"
      ],
      "scope": "Cluster"
    },
    {
      "operations": [
        "CREATE",
        "UPDATE"
      ],
      "apiGroups": [
        "operator.openshift.io"
      ],
      "apiVersions": [
        "*"
      ],
      "resources": [
        "imagecontentsourcepolicies"
      ],
      "scope": "Cluster"
    }
  ],
  "documentString": "Managed OpenShift customers may not create ImageContentSourcePolicy, ImageDigestMirrorSet, or ImageTagMirrorSet resources that configure mirrors for the entirety of quay.io, registry.redhat.io, nor registry.access.redhat.com. If needed, specific repositories can have mirrors configured, such as quay.io/example."
},
{  "webhookName": "ingresscontroller-validation",
  "rules": [
    {
      "operations": [
        "CREATE",
        "UPDATE"
      ],
      "apiGroups": [
        "networking\.openstack\.
        "k8s\.io"
      ],
      "apiVersions": [
        "*"
      ],
      "resources": [
        "ingresses",
        "networkpods"
      ],
      "scope": "Cluster"
    }
  ]}
"UPDATE"
],
"apiGroups": [
"operator.openshift.io"
],
"apiVersions": [
"*"
],
"resources": [
"ingresscontroller",
"ingresscontrollers"
],
"scope": "Namespaced"
}
]
"documentString": "Managed OpenShift Customer may create IngressControllers without necessary taints. This can cause those workloads to be provisioned on infra or master nodes."
},
{ "webhookName": "namespace-validation",
"rules": [
{ "operations": [
"CREATE",
"UPDATE",
"DELETE"
],
"apiGroups": [
""
],
"apiVersions": [
"*"
],
"resources": [
"namespaces"
],
"scope": "Cluster"
}
],
"documentString": "Managed OpenShift Customers may not modify namespaces specified in the [openshift-monitoring/addons-namespaces openshift-monitoring/managed-namespaces openshift-monitoring/ocp-namespaces] ConfigMaps because customer workloads should be placed in customer-created namespaces. Customers may not create namespaces identified by this regular expression (^com$|^io$|^in$) because it could interfere with critical DNS resolution. Additionally, customers may not set or change the values of these Namespace labels [managed.openshift.io/storage-pv-quota-exempt managed.openshift.io/service-lb-quota-exempt]."
},
{ "webhookName": "pod-validation",
"rules": [
{ "operations": [
"*
],
"apiGroups": [
"v1"
]
Managed OpenShift Customers may use tolerations on Pods that could cause those Pods to be scheduled on infra or master nodes.

Managed OpenShift Customers may not create PrometheusRule in namespaces managed by Red Hat.

Managed OpenShift Customers may not create PrometheusRule in namespaces managed by Red Hat.
"resources": [
  "autoscaling.openshift.io",
  "config.openshift.io"
],
"apiGroups": [
  "autoscaling.openshift.io",
  "config.openshift.io"
],
"apiVersions": [
  "*",
  "*"
],
"resources": [
  "clusterautoscalers",
  "machineautoscalers",
  "clusterversions",
  "clusterversions/status",
  "schedulers",
  "apiservers",
  "proxies",
  "configmaps"
],
"scope": "*",
"operations": [
  "CREATE",
  "UPDATE",
  "DELETE"
]
},    "scope": "*"
]
],
"apiGroups": [
  "machineconfiguration.openshift.io"
],
"apiVersions": [
  "*"
],
"resources": [
  "machineconfigs",
  "machineconfigpools"
],
"scope": "*"
},
{
  "operations": [  
  "*"
  ],
  "apiGroups": [
    "operator.openshift.io"
  ],
  "apiVersions": [
    "*"
  ],
  "resources": [
    "kubeapiservers",
    "openshiftapiservers"
  ],
  "scope": "*"
],
{
  "operations": [  
  "*"
  ],
  "apiGroups": [
    "managed.openshift.io"
  ],
  "apiVersions": [
    "*"
  ],
  "resources": [
    "subjectpermissions",
    "subjectpermissions/*"
  ],
  "scope": "*"
],
{
  "operations": [  
  "*"
  ],
  "apiGroups": [
  ],
  "apiVersions": [  
  ]
}
Managed OpenShift customers may not manage any objects in the following APIgroups [operator.openshift.io splunkforwarder.managed.openshift.io config.openshift.io upgrade.managed.openshift.io autoscaling.openshift.io machineconfiguration.openshift.io network.openshift.io cloudcredential.openshift.io managed.openshift.io addons.managed.openshift.io cloudingress.managed.openshift.io ocmagent.managed.openshift.io machine.openshift.io admissionregistration.k8s.io], nor may Managed OpenShift customers alter the APIServer, KubeAPIServer, OpenShiftAPIServer, ClusterVersion, Proxy or SubjectPermission objects.

Managed OpenShift customers may not manage any objects in the following APIgroups [], nor may Managed OpenShift customers alter the Node objects.

Managed OpenShift customers may not manage any objects in the following APIgroups [], nor may Managed OpenShift customers alter the Node objects.
Managed OpenShift Customers may not modify the following default SCCs:
[anyuid hostaccess hostmount-anyuid hostnetwork hostnetwork-v2 node-exporter nonroot nonroot-v2 privileged restricted restricted-v2]

Managed OpenShift Customers may not delete the service accounts under the managed namespaces.

Managed OpenShift Customers may not use TechPreviewNoUpgrade
FeatureGate that could prevent any future ability to do a y-stream upgrade to their clusters."