OpenShift Dedicated 4

Support

OpenShift Dedicated Support.

Last Updated: 2026-01-16

OpenShift Dedicated 4 Support

OpenShift Dedicated Support.

Legal Notice

Abstract

Offers cluster administrators tools for gathering data for your cluster, monitoring, and troubleshooting.

Table of Contents

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: OpenShift Dedicated 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 a *connected cluster*. OpenShift Dedicated 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 OpenShift Dedicated upgrades.
  - Improve the upgrade experience.
- **Insights Operator**: By default, OpenShift Dedicated installs and enables the Insights Operator, which reports configuration and component failure status every two hours. The Insights 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, Using Red Hat Lightspeed to identify issues with your cluster. You can optionally disable remote health reporting.

1.3. GATHER DATA ABOUT YOUR CLUSTER

Gather data about your cluster: Red Hat recommends gathering your debugging information when opening a support case. This helps Red Hat Support to perform a root cause analysis. A cluster administrator can use the following to gather data about your cluster:

- **must-gather tool**: Use the **must-gather** tool to collect information about your cluster and to debug the issues.
- **sosreport**: Use the **sosreport** tool to collect configuration details, system information, and diagnostic data for debugging purposes.
- **Cluster ID**: Obtain the unique identifier for your cluster, when providing information to Red Hat Support.

- **Cluster node journal logs**: Gather **journald** unit logs and logs within **/var/log** on individual cluster nodes to troubleshoot node-related issues.
- **Network trace**: Provide a network packet trace from a specific OpenShift Dedicated cluster node or a container to Red Hat Support to help troubleshoot network-related issues.

1.4. TROUBLESHOOTING ISSUES

A cluster administrator can monitor and troubleshoot the following OpenShift Dedicated 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.
- Source-to-image issues: A cluster administrator can observe the S2I stages to determine where in the S2I process a failure occurred. Gather the following to resolve Source-to-Image (S2I) issues:
  - Source-to-Image diagnostic data.
  - Application diagnostic data to investigate application failure.
- 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.
- 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 OpenShift Dedicated. 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 OpenShift Dedicated. 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 api-resources -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 OpenShift Dedicated 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 Red Hat Lightspeed in OpenShift Cluster Manager. Red Hat Lightspeed 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 OpenShift Dedicated 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 OpenShift Dedicated 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

1. Log in to the Red Hat Customer Portal .
2. Click **Search**.
3. In the search field, input keywords and strings relating to the problem, including:
   - OpenShift Dedicated 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 **OpenShift Dedicated** 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 **dedicated-admin** role.
- You have installed the OpenShift CLI (**oc**).
- You have access to the Red Hat OpenShift Cluster Manager.

Procedure

1. Log in to the **Customer Support** page of the Red Hat Customer Portal.
2. Click **Get support**.
3. On the **Cases** tab of the **Customer Support** page:
   a. Optional: Change the pre-filled account and owner details 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 **OpenShift Dedicated** 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 OpenShift Dedicated Cluster ID is correct. If it is not, manually obtain your cluster ID.
   - To manually obtain your cluster ID using OpenShift Cluster Manager:
     a. Navigate to **Cluster List**.
     b. Click on the name of the cluster you need to open a support case for.
     c. Find the value in the **Cluster ID** field of the **Details** section of the **Overview** tab.

- To manually obtain your cluster ID using the OpenShift Dedicated 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 OpenShift Dedicated 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**:
    - What are you experiencing? What are you expecting to happen?
    - Define the value or impact to you or the business.
    - Where are you experiencing this behavior? What environment?
    - When does this behavior occur? Frequency? Repeatedly? At certain times?
10. Upload relevant diagnostic data files and click **Continue**.
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 Red Hat Lightspeed to identify issues with your cluster.

CHAPTER 4. REMOTE HEALTH MONITORING WITH CONNECTED CLUSTERS

4.1. ABOUT REMOTE HEALTH MONITORING

OpenShift Dedicated 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 OpenShift Dedicated 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 OpenShift Dedicated 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.

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**. OpenShift Dedicated 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 OpenShift Dedicated 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 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 OpenShift Dedicated clusters are exposed to.

On OpenShift Dedicated, 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 OpenShift Dedicated 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 OpenShift Dedicated better and more intuitive to use.

Additional resources

- See the OpenShift Dedicated upgrade documentation for more information about upgrading a cluster.

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 OpenShift Dedicated 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 OpenShift Dedicated 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 OpenShift Dedicated is deployed on and the data center location

4.1.1.1.2. 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

4.1.1.1.3. 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.

4.1.1.2. 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 OpenShift Dedicated.
- 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 OpenShift Dedicated can display the report of each cluster in the Advisor service on Red Hat Hybrid Cloud Console. If any issues have been identified, Red Hat Lightspeed 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 Lightspeed Data & Application Security for information about Red Hat Lightspeed 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 Advisor service on Red Hat Hybrid Cloud Console
- Improve OpenShift Dedicated by providing aggregated and critical information to product and support teams
- Make OpenShift Dedicated 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 OpenShift Dedicated 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 OpenShift Dedicated is deployed on and the region that the cluster is located in
- If an Operator reports an issue, information is collected about core OpenShift Dedicated pods in the **openshift-*** and **kube-*** projects. This includes state, resource, security context, volume information, and more.

Additional resources

- What data is being collected by the Insights Operator in OpenShift?
- 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

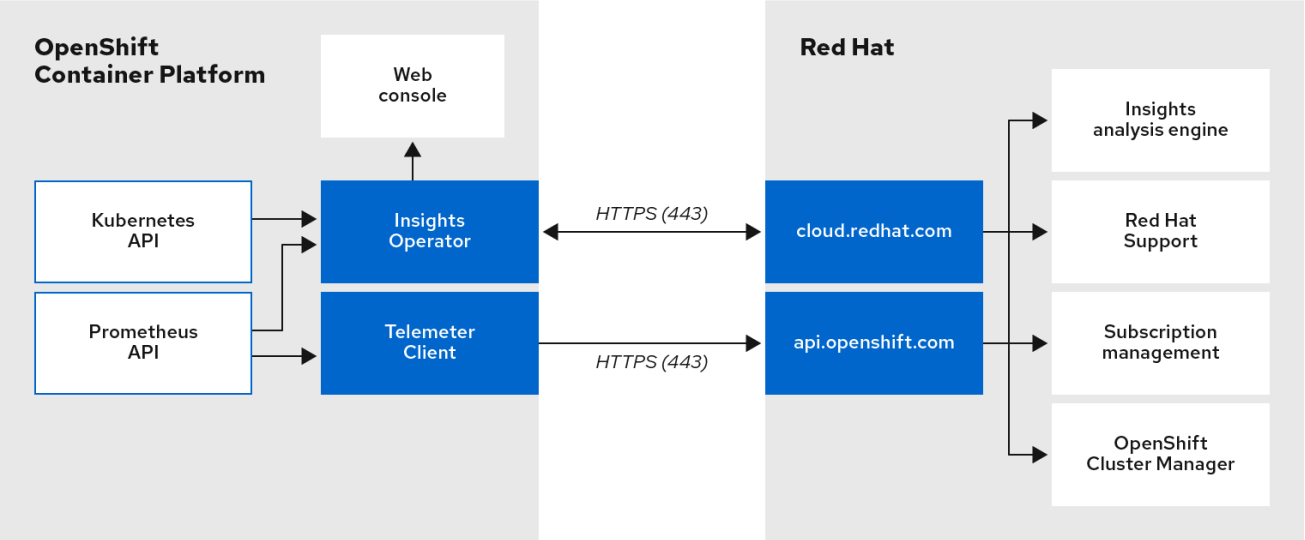
The Telemeter Client collects selected time series data from the Prometheus API. The time series data is uploaded to api.openshift.com every four minutes and thirty seconds for processing.

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 every two hours for processing. The Insights Operator also downloads the latest Red Hat Lightspeed analysis from OpenShift Cluster Manager. This is used to populate the **Red Hat Lightspeed status** pop-up that is included in the **Overview** page in the OpenShift Dedicated 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



Additional resources

- See About OpenShift Dedicated monitoring for more information about the OpenShift Dedicated monitoring stack.

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 might share the data collected through Telemetry and the Insights Operator internally within Red Hat to improve your user experience. Red Hat might 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:

NOTE

The following example contains some values that are specific to OpenShift Dedicated on AWS.

```
$ curl -G -k -H "Authorization: Bearer $(oc whoami -t)" \
https://$(oc get route prometheus-k8s-federate -n \
openshift-monitoring -o jsonpath="{.spec.host}")/federate \
--data-urlencode 'match[]={__name__=~"cluster:usage:.*"}' \
--data-urlencode 'match[]={__name__="count:up0"}' \
--data-urlencode 'match[]={__name__="count:up1"}' \
--data-urlencode 'match[]={__name__="cluster_version"}' \
--data-urlencode 'match[]={__name__="cluster_version_available_updates"}' \
--data-urlencode 'match[]={__name__="cluster_version_capability"}' \
--data-urlencode 'match[]={__name__="cluster_operator_up"}' \
--data-urlencode 'match[]={__name__="cluster_operator_conditions"}' \
--data-urlencode 'match[]={__name__="cluster_version_payload"}' \
--data-urlencode 'match[]={__name__="cluster_installer"}' \
--data-urlencode 'match[]={__name__="cluster_infrastructure_provider"}' \
--data-urlencode 'match[]={__name__="cluster_feature_set"}' \
--data-urlencode 'match[]={__name__="instance:etcd_object_counts:sum"}' \
--data-urlencode 'match[]={__name__="ALERTS",alertstate="firing"}' \
--data-urlencode 'match[]={__name__="code:apiserver_request_total:rate:sum"}' \
--data-urlencode 'match[]={__name__="cluster:capacity_cpu_cores:sum"}' \
--data-urlencode 'match[]={__name__="cluster:capacity_memory_bytes:sum"}' \
--data-urlencode 'match[]={__name__="cluster:cpu_usage_cores:sum"}' \
--data-urlencode 'match[]={__name__="cluster:memory_usage_bytes:sum"}' \
--data-urlencode 'match[]={__name__="openshift:cpu_usage_cores:sum"}' \
--data-urlencode 'match[]={__name__="openshift:memory_usage_bytes:sum"}' \
--data-urlencode 'match[]={__name__="workload:cpu_usage_cores:sum"}' \
--data-urlencode 'match[]={__name__="workload:memory_usage_bytes:sum"}' \
--data-urlencode 'match[]={__name__="cluster:virt_platform_nodes:sum"}' \
--data-urlencode 'match[]={__name__="cluster:node_instance_type_count:sum"}' \
--data-urlencode 'match[]={__name__="cnv:vmi_status_running:count"}' \
--data-urlencode 'match[]={__name__="cluster:vmi_request_cpu_cores:sum"}' \
--data-urlencode 'match[]={__name__="node_role_os_version_machine:cpu_capacity_cores:sum"}' \
--data-urlencode 'match[]=
{__name__="node_role_os_version_machine:cpu_capacity_sockets:sum"}' \
```

```
--data-urlencode 'match[]={__name__="subscription_sync_total"}' \
--data-urlencode 'match[]={__name__="olm_resolution_duration_seconds"}' \
--data-urlencode 'match[]={__name__="csv_succeeded"}' \
--data-urlencode 'match[]={__name__="csv_abnormal"}' \
--data-urlencode 'match[]=
{__name__="cluster:kube_persistentvolumeclaim_resource_requests_storage_bytes:provisioner:sum"}'
\
--data-urlencode 'match[]={__name__="cluster:kubelet_volume_stats_used_bytes:provisioner:sum"}'
\
--data-urlencode 'match[]={__name__="ceph_cluster_total_bytes"}' \
--data-urlencode 'match[]={__name__="ceph_cluster_total_used_raw_bytes"}' \
--data-urlencode 'match[]={__name__="ceph_health_status"}' \
--data-urlencode 'match[]={__name__="odf_system_raw_capacity_total_bytes"}' \
--data-urlencode 'match[]={__name__="odf_system_raw_capacity_used_bytes"}' \
--data-urlencode 'match[]={__name__="odf_system_health_status"}' \
--data-urlencode 'match[]={__name__="job:ceph_osd_metadata:count"}' \
--data-urlencode 'match[]={__name__="job:kube_pv:count"}' \
--data-urlencode 'match[]={__name__="job:odf_system_pvs:count"}' \
--data-urlencode 'match[]={__name__="job:ceph_pools_iops:total"}' \
--data-urlencode 'match[]={__name__="job:ceph_pools_iops_bytes:total"}' \
--data-urlencode 'match[]={__name__="job:ceph_versions_running:count"}' \
--data-urlencode 'match[]={__name__="job:noobaa_total_unhealthy_buckets:sum"}' \
--data-urlencode 'match[]={__name__="job:noobaa_bucket_count:sum"}' \
--data-urlencode 'match[]={__name__="job:noobaa_total_object_count:sum"}' \
--data-urlencode 'match[]={__name__="odf_system_bucket_count", system_type="OCS",
system_vendor="Red Hat"}' \
--data-urlencode 'match[]={__name__="odf_system_objects_total", system_type="OCS",
system_vendor="Red Hat"}' \
--data-urlencode 'match[]={__name__="noobaa_accounts_num"}' \
--data-urlencode 'match[]={__name__="noobaa_total_usage"}' \
--data-urlencode 'match[]={__name__="console_url"}' \
--data-urlencode 'match[]={__name__="cluster:ovnkube_master_egress_routing_via_host:max"}' \
--data-urlencode 'match[]={__name__="cluster:network_attachment_definition_instances:max"}' \
--data-urlencode 'match[]=
{__name__="cluster:network_attachment_definition_enabled_instance_up:max"}' \
--data-urlencode 'match[]={__name__="cluster:ingress_controller_aws_nlb_active:sum"}' \
--data-urlencode 'match[]={__name__="cluster:route_metrics_controller_routes_per_shard:min"}' \
--data-urlencode 'match[]={__name__="cluster:route_metrics_controller_routes_per_shard:max"}' \
--data-urlencode 'match[]={__name__="cluster:route_metrics_controller_routes_per_shard:avg"}' \
--data-urlencode 'match[]={__name__="cluster:route_metrics_controller_routes_per_shard:median"}'
\
--data-urlencode 'match[]={__name__="cluster:openshift_route_info:tls_termination:sum"}' \
--data-urlencode 'match[]={__name__="insightsclient_request_send_total"}' \
--data-urlencode 'match[]={__name__="cam_app_workload_migrations"}' \
--data-urlencode 'match[]=
{__name__="cluster:apiserver_current_inflight_requests:sum:max_over_time:2m"}' \
--data-urlencode 'match[]={__name__="cluster:alertmanager_integrations:max"}' \
--data-urlencode 'match[]={__name__="cluster:telemetry_selected_series:count"}' \
--data-urlencode 'match[]={__name__="openshift:prometheus_tsdb_head_series:sum"}' \
--data-urlencode 'match[]=
{__name__="openshift:prometheus_tsdb_head_samples_appended_total:sum"}' \
--data-urlencode 'match[]={__name__="monitoring:container_memory_working_set_bytes:sum"}' \
--data-urlencode 'match[]={__name__="namespace_job:scrape_series_added:topk3_sum1h"}' \
--data-urlencode 'match[]=
{__name__="namespace_job:scrape_samples_post_metric_relabeling:topk3"}' \
--data-urlencode 'match[]={__name__="monitoring:haproxy_server_http_responses_total:sum"}' \
```

```
--data-urlencode 'match[]={__name__="rhmi_status"}' \
--data-urlencode 'match[]={__name__="status:upgrading:version:rhoam_state:max"}' \
--data-urlencode 'match[]={__name__="state:rhoam_critical_alerts:max"}' \
--data-urlencode 'match[]={__name__="state:rhoam_warning_alerts:max"}' \
--data-urlencode 'match[]={__name__="rhoam_7d_slo_percentile:max"}' \
--data-urlencode 'match[]={__name__="rhoam_7d_slo_remaining_error_budget:max"}' \
--data-urlencode 'match[]={__name__="cluster_legacy_scheduler_policy"}' \
--data-urlencode 'match[]={__name__="cluster_master_schedulable"}' \
--data-urlencode 'match[]={__name__="che_workspace_status"}' \
--data-urlencode 'match[]={__name__="che_workspace_started_total"}' \
--data-urlencode 'match[]={__name__="che_workspace_failure_total"}' \
--data-urlencode 'match[]={__name__="che_workspace_start_time_seconds_sum"}' \
--data-urlencode 'match[]={__name__="che_workspace_start_time_seconds_count"}' \
--data-urlencode 'match[]={__name__="cco_credentials_mode"}' \
--data-urlencode 'match[]={__name__="cluster:kube_persistentvolume_plugin_type_counts:sum"}' \
--data-urlencode 'match[]={__name__="visual_web_terminal_sessions_total"}' \
--data-urlencode 'match[]={__name__="acm_managed_cluster_info"}' \
--data-urlencode 'match[]={__name__="cluster:vsphere_vcenter_info:sum"}' \
--data-urlencode 'match[]={__name__="cluster:vsphere_esxi_version_total:sum"}' \
--data-urlencode 'match[]={__name__="cluster:vsphere_node_hw_version_total:sum"}' \
--data-urlencode 'match[]={__name__="openshift:build_by_strategy:sum"}' \
--data-urlencode 'match[]={__name__="rhods_aggregate_availability"}' \
--data-urlencode 'match[]={__name__="rhods_total_users"}' \
--data-urlencode 'match[]=
{__name__="instance:etcd_disk_wal_fsync_duration_seconds:histogram_quantile",quantile="0.99"}' \
--data-urlencode 'match[]={__name__="instance:etcd_mvcc_db_total_size_in_bytes:sum"}' \
--data-urlencode 'match[]=
{__name__="instance:etcd_network_peer_round_trip_time_seconds:histogram_quantile",quantile="0.99
"}' \
--data-urlencode 'match[]={__name__="instance:etcd_mvcc_db_total_size_in_use_in_bytes:sum"}' \
--data-urlencode 'match[]=
{__name__="instance:etcd_disk_backend_commit_duration_seconds:histogram_quantile",quantile="0.9
9"}' \
--data-urlencode 'match[]={__name__="appsvcs:cores_by_product:sum"}' \
--data-urlencode 'match[]={__name__="nto_custom_profiles:count"}' \
--data-urlencode 'match[]={__name__="openshift_csi_share_configmap"}' \
--data-urlencode 'match[]={__name__="openshift_csi_share_secret"}' \
--data-urlencode 'match[]={__name__="openshift_csi_share_mount_failures_total"}' \
--data-urlencode 'match[]={__name__="openshift_csi_share_mount_requests_total"}' \
--data-urlencode 'match[]={__name__="cluster:velero_backup_total:max"}' \
--data-urlencode 'match[]={__name__="cluster:velero_restore_total:max"}' \
--data-urlencode 'match[]={__name__="eo_es_storage_info"}' \
--data-urlencode 'match[]={__name__="eo_es_redundancy_policy_info"}' \
--data-urlencode 'match[]={__name__="eo_es_defined_delete_namespaces_total"}' \
--data-urlencode 'match[]={__name__="eo_es_misconfigured_memory_resources_info"}' \
--data-urlencode 'match[]={__name__="cluster:eo_es_data_nodes_total:max"}' \
--data-urlencode 'match[]={__name__="cluster:eo_es_documents_created_total:sum"}' \
--data-urlencode 'match[]={__name__="cluster:eo_es_documents_deleted_total:sum"}' \
--data-urlencode 'match[]={__name__="pod:eo_es_shards_total:max"}' \
--data-urlencode 'match[]={__name__="eo_es_cluster_management_state_info"}' \
--data-urlencode 'match[]={__name__="imageregistry:imagestreamtags_count:sum"}' \
--data-urlencode 'match[]={__name__="imageregistry:operations_count:sum"}' \
--data-urlencode 'match[]={__name__="log_logging_info"}' \
--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_unhealthy_namespace_resources:max"}' \
--data-urlencode 'match[]={__name__="namespace:noobaa_namespace_resources:max"}' \
--data-urlencode 'match[]={__name__="namespace:noobaa_unhealthy_namespace_buckets:max"}' \
--data-urlencode 'match[]={__name__="namespace:noobaa_namespace_buckets:max"}' \
--data-urlencode 'match[]={__name__="namespace:noobaa_accounts:max"}' \
--data-urlencode 'match[]={__name__="namespace:noobaa_usage:max"}' \
--data-urlencode 'match[]={__name__="namespace:noobaa_system_health_status:max"}' \
--data-urlencode 'match[]={__name__="ocs_advanced_feature_usage"}' \
--data-urlencode 'match[]={__name__="os_image_url_override:sum"}' \
--data-urlencode 'match[]={__name__="openshift:openshift_network_operator_ipsec_state:info"}'
```

4.3. USING RED HAT LIGHTSPEED TO IDENTIFY ISSUES WITH YOUR CLUSTER

Red Hat Lightspeed repeatedly analyzes the data Insights Operator sends, which includes workload recommendations from Deployment Validation Operator (DVO). Users of OpenShift Dedicated can display the results in the Advisor service on Red Hat Hybrid Cloud Console.

4.3.1. About Red Hat Lightspeed Advisor for OpenShift Dedicated

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

If the cluster has the Deployment Validation Operator (DVO) installed the recommendations also highlight workloads whose configuration might lead to cluster health issues.

The results of the Red Hat Lightspeed analysis are available in the Red Hat Lightspeed advisor service on Red Hat Hybrid Cloud Console. In the Red Hat Hybrid Cloud Console, you can perform the following actions:

- View clusters and workloads affected by specific recommendations.
- 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.

Additional resources

- Using the Deployment Validation Operator in your Red Hat Lightspeed workflow

4.3.2. Understanding Red Hat Lightspeed advisor service recommendations

The Red Hat Lightspeed advisor service bundles information about various cluster states and component configurations that can negatively affect the service availability, fault tolerance, performance, or security of your clusters and workloads. This information set is called a recommendation in the Red Hat Lightspeed advisor service. Recommendations for clusters includes the following information:

- **Name:** A concise description of the recommendation
- **Added:** When the recommendation was published to the Red Hat Lightspeed advisor service archive
- **Category:** Whether the issue has the potential to negatively affect service availability, fault tolerance, performance, or security
- **Total risk:** A value derived from the *likelihood* that the condition will negatively affect your cluster or workload, and the *impact* on operations if that were to happen
- **Clusters:** A list of clusters on which a recommendation is detected
- **Description:** A brief synopsis of the issue, including how it affects your clusters

4.3.3. Displaying potential issues with your cluster

This section describes how to display the Red Hat Lightspeed report in **Red Hat Lightspeed Advisor** on OpenShift Cluster Manager.

Note that Red Hat Lightspeed 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.
- Remote health reporting is enabled, which is the default.
- You are logged in to OpenShift Cluster Manager.

Procedure

1. Navigate to **Advisor → Recommendations** on OpenShift Cluster Manager.
   Depending on the result, the Red Hat Lightspeed advisor service displays one of the following:
   - **No matching recommendations found**, if Red Hat Lightspeed did not identify any issues.
   - A list of issues Red Hat Lightspeed has detected, grouped by risk (low, moderate, important, and critical).
   - **No clusters yet**, if Red Hat Lightspeed 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 Red Hat Lightspeed advisor service 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 service's 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.

Procedure

1. Navigate to **Advisor → Recommendations** on OpenShift Cluster Manager.
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. Advisor recommendation filters

The Red Hat Lightspeed advisor service can return a large number of recommendations. To focus on your most critical recommendations, you can apply filters to the Advisor recommendations list to remove low-priority recommendations.

By default, filters are set to only show enabled recommendations that are impacting one or more clusters. To view all or disabled recommendations in the Red Hat Lightspeed library, you can customize the filters.

To apply a filter, select a filter type and then set its value based on the options that are available in the drop-down list. You can apply multiple filters to the list of recommendations.

You can set the following filter types:

- **Name:** Search for a recommendation by name.
- **Total risk:** Select one or more values from **Critical**, **Important**, **Moderate**, and **Low** indicating the likelihood and the severity of a negative impact on a cluster.
- **Impact:** Select one or more values from **Critical**, **High**, **Medium**, and **Low** indicating the potential impact to the continuity of cluster operations.
- **Likelihood:** Select one or more values from **Critical**, **High**, **Medium**, and **Low** indicating the potential for a negative impact to a cluster if the recommendation comes to fruition.
- **Category:** Select one or more categories from **Service Availability**, **Performance**, **Fault Tolerance**, **Security**, and **Best Practice** to focus your attention on.
- **Status:** Click a radio button to show enabled recommendations (default), disabled recommendations, or all recommendations.
- **Clusters impacted:** Set the filter to show recommendations currently impacting one or more clusters, non-impacting recommendations, or all recommendations.

- **Risk of change:** Select one or more values from **High**, **Moderate**, **Low**, and **Very low** indicating the risk that the implementation of the resolution could have on cluster operations.

4.3.5.1. Filtering Red Hat Lightspeed advisor service recommendations

As an OpenShift Dedicated cluster manager, you can filter the recommendations that are displayed on the recommendations list. By applying filters, you can reduce the number of reported recommendations and concentrate on your highest priority recommendations.

The following procedure demonstrates how to set and remove **Category** filters; however, the procedure is applicable to any of the filter types and respective values.

Prerequisites

You are logged in to the OpenShift Cluster Manager in the Hybrid Cloud Console.

Procedure

1. Go to **OpenShift** > **Advisor** > **Recommendations**.
2. In the main, filter-type drop-down list, select the **Category** filter type.
3. Expand the filter-value drop-down list and select the checkbox next to each category of recommendation you want to view. Leave the checkboxes for unnecessary categories clear.
4. Optional: Add additional filters to further refine the list.

Only recommendations from the selected categories are shown in the list.

Verification

- After applying filters, you can view the updated recommendations list. The applied filters are added next to the default filters.

4.3.5.2. Removing filters from Red Hat Lightspeed advisor service recommendations

You can apply multiple filters to the list of recommendations. When ready, you can remove them individually or completely reset them.

Removing filters individually

- Click the **X** icon next to each filter, including the default filters, to remove them individually.

Removing all non-default filters

- Click **Reset filters** to remove only the filters that you applied, leaving the default filters in place.

4.3.6. Disabling Red Hat Lightspeed advisor service 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.
- You are logged in to OpenShift Cluster Manager.

Procedure

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

4.3.7. Enabling a previously disabled Red Hat Lightspeed advisor service recommendation

When a recommendation is disabled for all clusters, you no longer see the recommendation in the Red Hat Lightspeed advisor service. You can change this behavior.

Prerequisites

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

Procedure

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

4.3.8. About Red Hat Lightspeed advisor service recommendations for workloads

You can use the Red Hat Lightspeed advisor service to view and manage information about recommendations that affect not only your clusters, but also your workloads. The advisor service takes advantage of deployment validation and helps OpenShift cluster administrators to see all runtime violations of deployment policies. You can see recommendations for workloads at OpenShift > Advisor > Workloads on the Red Hat Hybrid Cloud Console. For more information, see these additional resources:

- Information about Kubernetes workloads
- Boost your cluster operations with Deployment Validation and Red Hat Lightspeed Advisor for Workloads
- Identifying workload recommendations for namespaces in your clusters
- Viewing workload recommendations for namespaces in your cluster
- Excluding objects from workload recommendations in your clusters

4.3.9. Displaying the Red Hat Lightspeed status in the web console

Red Hat Lightspeed repeatedly analyzes your cluster and you can display the status of identified potential issues of your cluster in the OpenShift Dedicated 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.

Prerequisites

- Your cluster is registered in OpenShift Cluster Manager.
- Remote health reporting is enabled, which is the default.
- You are logged in to the OpenShift Dedicated web console.

Procedure

1. Navigate to **Home → Overview** in the OpenShift Dedicated web console.
2. Click **Red Hat Lightspeed** on the **Status** card.
   The pop-up window lists potential issues grouped by risk. Click the individual categories or **View all recommendations in Red Hat Lightspeed 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 OpenShift Dedicated can display the report in the Advisor service on Red Hat Hybrid Cloud Console.

Additional resources

- For more information on using the Red Hat Lightspeed advisor service to identify issues with your cluster, see Using Red Hat Lightspeed 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 OpenShift Dedicated 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 4.1. Insights Operator alerts

Alert	Description
InsightsDisabled	Insights Operator is disabled.
SimpleContentAccessNotAvailable	Simple content access is not enabled in Red Hat Subscription Management.
InsightsRecommendationActive	Red Hat Lightspeed has an active recommendation for the cluster.

4.4.2. Obfuscating Deployment Validation Operator data

By default, when you install the Deployment Validation Operator (DVO), the name and unique identifier (UID) of a resource are included in the data that is captured and processed by the Insights Operator for OpenShift Dedicated. If you are a cluster administrator, you can configure the Insights Operator to obfuscate data from the Deployment Validation Operator (DVO). For example, you can obfuscate workload names in the archive file that is then sent to Red Hat.

To obfuscate the name of resources, you must manually set the **obfuscation** attribute in the **insights-config ConfigMap** object to include the **workload_names** value, as outlined in the following procedure.

Prerequisites

- Remote health reporting is enabled, which is the default.
- You are logged in to the OpenShift Dedicated web console with the "cluster-admin" role.
- The **insights-config ConfigMap** object exists in the **openshift-insights** namespace.
- The cluster is self managed and the Deployment Validation Operator is installed.

Procedure

1. Go to **Workloads → ConfigMaps** and select **Project: openshift-insights**.
2. Click the **insights-config ConfigMap** object to open it.
3. Click **Actions** and select **Edit ConfigMap**.
4. Click the **YAML view** radio button.
5. In the file, set the **obfuscation** attribute with the **workload_names** value.

```
apiVersion: v1
kind: ConfigMap
# ...
data:
  config.yaml: |
    dataReporting:
      obfuscation:
        - workload_names
# ...
```

6. Click **Save**. The **insights-config** config-map details page opens.
7. Verify that the value of the **config.yaml obfuscation** attribute is set to **- workload_names**.

CHAPTER 5. GATHERING DATA ABOUT YOUR CLUSTER

When opening a support case, it is helpful to provide debugging information about your cluster to Red Hat Support.

It is recommended to provide:

- Data gathered using the **oc adm must-gather** command
- The unique cluster ID

5.1. ABOUT THE MUST-GATHER TOOL

The **oc adm must-gather** CLI command collects the information from your cluster that is most likely needed for debugging issues, including:

- Resource definitions
- Service logs

By default, the **oc adm must-gather** command uses the default plugin image and writes into **./must-gather.local**.

Alternatively, you can collect specific information by running the command with the appropriate arguments as described in the following sections:

- To collect data related to one or more specific features, use the **--image** argument with an image, as listed in a following section.
  For example:

```
$ oc adm must-gather \
  --image=registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel9:v4.20.1
```

  - To collect the audit logs, use the **-- /usr/bin/gather_audit_logs** argument, as described in a following section.
    For example:

```
$ oc adm must-gather -- /usr/bin/gather_audit_logs
```

NOTE

- Audit logs are not collected as part of the default set of information to reduce the size of the files.
- On a Windows operating system, install the **cwRsync** client and add to the **PATH** variable for use with the **oc rsync** command.

When you run **oc adm must-gather**, a new pod with a random name is created in a new project on the cluster. The data is collected on that pod and saved in a new directory that starts with **must-gather.local** in the current working directory.

For example:

```
NAMESPACE                 NAME              READY   STATUS      RESTARTS      AGE
...
openshift-must-gather-5drcj    must-gather-bklx4    2/2     Running     0          72s
openshift-must-gather-5drcj    must-gather-s8sdh    2/2     Running     0          72s
...
```

Optionally, you can run the **oc adm must-gather** command in a specific namespace by using the **--run-namespace** option.

For example:

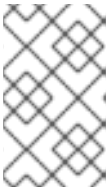
```
$ oc adm must-gather --run-namespace <namespace> \
  --image=registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel9:v4.20.1
```

5.1.1. Gathering data about your cluster for Red Hat Support

You can gather debugging information about your cluster by using the **oc adm must-gather** CLI command.

Prerequisites

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



NOTE

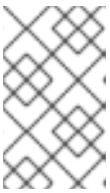
In OpenShift Dedicated deployments, customers who are not using the Customer Cloud Subscription (CCS) model cannot use the **oc adm must-gather** command as it requires **cluster-admin** privileges.

- The OpenShift CLI (**oc**) is installed.

Procedure

1. Navigate to the directory where you want to store the **must-gather** data.
2. Run the **oc adm must-gather** command:

```
$ oc adm must-gather
```



NOTE

Because this command picks a random control plane node by default, the pod might be scheduled to a control plane node that is in the **NotReady** and **SchedulingDisabled** state.

a. If this command fails, for example, if you cannot schedule a pod on your cluster, then use the **oc adm inspect** command to gather information for particular resources.

NOTE

Contact Red Hat Support for the recommended resources to gather.

3. Create a compressed file from the **must-gather** directory that was just created in your working directory. Make sure you provide the date and cluster ID for the unique must-gather data. For more information about how to find the cluster ID, see How to find the cluster-id or name on OpenShift cluster. For example, on a computer that uses a Linux operating system, run the following command:

```
$ tar cvaf must-gather-`date +"%m-%d-%Y-%H-%M-%S"`-<cluster_id>.tar.gz
<must_gather_local_dir> 1
```

1 Replace **<must_gather_local_dir>** with the actual directory name.

4. Attach the compressed file to your support case on the the **Customer Support** page of the Red Hat Customer Portal.

5.1.2. Must-gather flags

The flags listed in the following table are available to use with the **oc adm must-gather** command.

Table 5.1. OpenShift Dedicated flags for oc adm must-gather

Flag	Example command	Description
--all-images	**oc adm must-gather --all-images=false**	Collect **must-gather** data using the default image for all Operators on the cluster that are annotated with **operators.openshift.io/must-gather-image**.
--dest-dir	**oc adm must-gather --dest-dir='<directory_name>'**	Set a specific directory on the local machine where the gathered data is written.
--host-network	**oc adm must-gather --host-network=false**	Run **must-gather** pods as **hostNetwork: true**. Relevant if a specific command and image needs to capture host-level data.
--image	**oc adm must-gather --image=[<plugin_image>]**	Specify a **must-gather** plugin image to run. If not specified, OpenShift Dedicated's default **must-gather** image is used.
--image-stream	**oc adm must-gather --image-stream=[<image_stream>]**	Specify an`<image_stream>` using a namespace or name:tag value containing a **must-gather** plugin image to run.
--node-name	**oc adm must-gather --node-name='<node>'**	Set a specific node to use. If not specified, by default a random master is used.

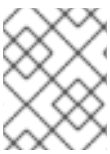
Flag	Example command	Description
--node-selector	**oc adm must-gather --node-selector='<node_selector_name>'**	Set a specific node selector to use. Only relevant when specifying a command and image which needs to capture data on a set of cluster nodes simultaneously.
--run-namespace	**oc adm must-gather --run-namespace='<namespace>'**	An existing privileged namespace where **must-gather** pods should run. If not specified, a temporary namespace is generated.
--since	**oc adm must-gather --since=<time>**	Only return logs newer than the specified duration. Defaults to all logs. Plugins are encouraged but not required to support this. Only one **since-time** or **since** may be used.
--since-time	**oc adm must-gather --since-time='<date_and_time>'**	Only return logs after a specific date and time, expressed in (RFC3339) format. Defaults to all logs. Plugins are encouraged but not required to support this. Only one **since-time** or **since** may be used.
--source-dir	**oc adm must-gather --source-dir='/<directory_name>/'**	Set the specific directory on the pod where you copy the gathered data from.
--timeout	**oc adm must-gather --timeout='<time>'**	The length of time to gather data before timing out, expressed as seconds, minutes, or hours, for example, 3s, 5m, or 2h. Time specified must be higher than zero. Defaults to 10 minutes if not specified.
--volume-percentage	**oc adm must-gather --volume-percentage=<percent>**	Specify maximum percentage of pod's allocated volume that can be used for **must-gather**. If this limit is exceeded,**must-gather** stops gathering, but still copies gathered data. Defaults to 30% if not specified.

5.1.3. Gathering data about specific features

You can gather debugging information about specific features by using the **oc adm must-gather** CLI command with the **--image** or **--image-stream** argument. The **must-gather** tool supports multiple images, so you can gather data about more than one feature by running a single command.

Table 5.2. Supported must-gather images

Image	Purpose
registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel9:v4.20.1	Data collection for OpenShift Virtualization.
registry.redhat.io/openshift-serverless-1/svls-must-gather-rhel8	Data collection for OpenShift Serverless.
registry.redhat.io/openshift-service-mesh/istio-must-gather-rhel8:<installed_version_service_mesh>	Data collection for Red Hat OpenShift Service Mesh.
registry.redhat.io/multicluster-engine/must-gather-rhel8	Data collection for hosted control planes.
registry.redhat.io/rhmtc/openshift-migration-must-gather-rhel8:v<installed_version_migration_toolkit>	Data collection for the Migration Toolkit for Containers.
registry.redhat.io/openshift-logging/cluster-logging-rhel9-operator:v<installed_version_logging>	Data collection for logging.
quay.io/netobserv/must-gather	Data collection for the Network Observability Operator.
registry.redhat.io/openshift-gitops-1/must-gather-rhel8:v<installed_version_GitOps>	Data collection for Red Hat OpenShift GitOps.
registry.redhat.io/openshift4/ose-secrets-store-csi-mustgather-rhel9:v<installed_version_secret_store>	Data collection for the Secrets Store CSI Driver Operator.



NOTE

To determine the latest version for an OpenShift Dedicated component's image, see the OpenShift Operator Life Cycles web page on the Red Hat Customer Portal.

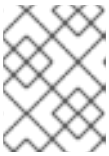
Prerequisites

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

Procedure

1. Navigate to the directory where you want to store the **must-gather** data.

2. Run the **oc adm must-gather** command with one or more **--image** or **--image-stream** arguments.



NOTE

- To collect the default **must-gather** data in addition to specific feature data, add the **--image-stream=openshift/must-gather** argument.

For example, the following command gathers both the default cluster data and information specific to OpenShift Virtualization:

```
$ oc adm must-gather \
  --image-stream=openshift/must-gather \ 1
  --image=registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel9:v4.20.1 2
```

1 The default OpenShift Dedicated **must-gather** image

2 The must-gather image for OpenShift Virtualization

You can use the **must-gather** tool with additional arguments to gather data that is specifically related to OpenShift Logging and the Cluster Logging Operator in your cluster. For OpenShift Logging, run the following command:

```
$ oc adm must-gather --image=$(oc -n openshift-logging get deployment.apps/cluster-logging-operator \
  -o jsonpath='{.spec.template.spec.containers[?(@.name == "cluster-logging-operator")].image}')
```

Example 5.1. Example must-gather output for OpenShift Logging

```
├── cluster-logging
│   ├── clo
│   │   ├── cluster-logging-operator-74dd5994f-6ttgt
│   │   ├── clusterlogforwarder_cr
│   │   ├── cr
│   │   ├── csv
│   │   ├── deployment
│   │   └── logforwarding_cr
│   ├── collector
│   │   ├── fluentd-2tr64
│   ├── curator
│   │   └── curator-1596028500-zkz4s
│   ├── eo
│   │   ├── csv
│   │   ├── deployment
│   │   └── elasticsearch-operator-7dc7d97b9d-jb4r4
│   ├── es
│   │   ├── cluster-elasticsearch
│   │   │   ├── aliases
│   │   │   ├── health
│   │   │   ├── indices
│   │   │   ├── latest_documents.json
│   │   │   ├── nodes
```

```
│   │   │   ├── nodes_stats.json
│   │   │   └── thread_pool
│   │   ├── cr
│   │   ├── elasticsearch-cdm-lp8l38m0-1-794d6dd989-4jxms
│   │   └── logs
│   │       ├── elasticsearch-cdm-lp8l38m0-1-794d6dd989-4jxms
│   ├── install
│   │   ├── co_logs
│   │   ├── install_plan
│   │   ├── olmo_logs
│   │   └── subscription
│   └── kibana
│       ├── cr
│       ├── kibana-9d69668d4-2rkvz
├── cluster-scoped-resources
│   └── core
│       ├── nodes
│       │   ├── ip-10-0-146-180.eu-west-1.compute.internal.yaml
│       └── persistentvolumes
│           ├── pvc-0a8d65d9-54aa-4c44-9ecc-33d9381e41c1.yaml
├── event-filter.html
├── gather-debug.log
└── namespaces
    ├── openshift-logging
    │   ├── apps
    │   │   ├── daemonsets.yaml
    │   │   ├── deployments.yaml
    │   │   ├── replicasets.yaml
    │   │   └── statefulsets.yaml
    │   ├── batch
    │   │   ├── cronjobs.yaml
    │   │   └── jobs.yaml
    │   ├── core
    │   │   ├── configmaps.yaml
    │   │   ├── endpoints.yaml
    │   │   ├── events
    │   │   │   ├── curator-1596021300-wn2ks.162634ebf0055a94.yaml
    │   │   │   ├── curator.162638330681bee2.yaml
    │   │   │   ├── elasticsearch-delete-app-1596020400-
gm6nl.1626341a296c16a1.yaml
    │   │   │   ├── elasticsearch-delete-audit-1596020400-
9l9n4.1626341a2af81bbd.yaml
    │   │   │   ├── elasticsearch-delete-infra-1596020400-
v98tk.1626341a2d821069.yaml
    │   │   │   ├── elasticsearch-rollover-app-1596020400-
cc5vc.1626341a3019b238.yaml
    │   │   │   ├── elasticsearch-rollover-audit-1596020400-
s8d5s.1626341a31f7b315.yaml
    │   │   │   ├── elasticsearch-rollover-infra-1596020400-
7mgv8.1626341a35ea59ed.yaml
    │   │   ├── events.yaml
    │   │   ├── persistentvolumeclaims.yaml
    │   │   ├── pods.yaml
    │   │   ├── replicationcontrollers.yaml
    │   │   ├── secrets.yaml
    │   │   └── services.yaml
```

```
│   ├── openshift-logging.yaml
│   ├── pods
│   │   ├── cluster-logging-operator-74dd5994f-6ttgt
│   │   │   ├── cluster-logging-operator
│   │   │   │   └── cluster-logging-operator
│   │   │   │       └── logs
│   │   │   │           ├── current.log
│   │   │   │           ├── previous.insecure.log
│   │   │   │           └── previous.log
│   │   │   └── cluster-logging-operator-74dd5994f-6ttgt.yaml
│   │   ├── cluster-logging-operator-registry-6df49d7d4-mxxff
│   │   │   ├── cluster-logging-operator-registry
│   │   │   │   └── cluster-logging-operator-registry
│   │   │   │       └── logs
│   │   │   │           ├── current.log
│   │   │   │           ├── previous.insecure.log
│   │   │   │           └── previous.log
│   │   │   ├── cluster-logging-operator-registry-6df49d7d4-mxxff.yaml
│   │   │   └── mutate-csv-and-generate-sqlite-db
│   │   │       └── mutate-csv-and-generate-sqlite-db
│   │   │           └── logs
│   │   │               ├── current.log
│   │   │               ├── previous.insecure.log
│   │   │               └── previous.log
│   │   ├── curator-1596028500-zkz4s
│   │   ├── elasticsearch-cdm-lp8l38m0-1-794d6dd989-4jxms
│   │   ├── elasticsearch-delete-app-1596030300-bpgcx
│   │   │   ├── elasticsearch-delete-app-1596030300-bpgcx.yaml
│   │   │   └── indexmanagement
│   │   │       └── indexmanagement
│   │   │           └── logs
│   │   │               ├── current.log
│   │   │               ├── previous.insecure.log
│   │   │               └── previous.log
│   │   ├── fluentd-2tr64
│   │   │   ├── fluentd
│   │   │   │   └── fluentd
│   │   │   │       └── logs
│   │   │   │           ├── current.log
│   │   │   │           ├── previous.insecure.log
│   │   │   │           └── previous.log
│   │   │   ├── fluentd-2tr64.yaml
│   │   │   └── fluentd-init
│   │   │       └── fluentd-init
│   │   │           └── logs
│   │   │               ├── current.log
│   │   │               ├── previous.insecure.log
│   │   │               └── previous.log
│   │   ├── kibana-9d69668d4-2rkvz
│   │   │   ├── kibana
│   │   │   │   └── kibana
│   │   │   │       └── logs
│   │   │   │           ├── current.log
│   │   │   │           ├── previous.insecure.log
│   │   │   │           └── previous.log
│   │   │   └── kibana-9d69668d4-2rkvz.yaml
```

```
│   │   │   └── kibana-proxy
│   │   │       └── kibana-proxy
│   │   │           └── logs
│   │   │               ├── current.log
│   │   │               ├── previous.insecure.log
│   │   │               └── previous.log
│   │   └── route.openshift.io
│   │       └── routes.yaml
│   └── openshift-operators-redhat
│       ├── ...
```

3. Run the **oc adm must-gather** command with one or more **--image** or **--image-stream** arguments. For example, the following command gathers both the default cluster data and information specific to KubeVirt:

```
$ oc adm must-gather \
 --image-stream=openshift/must-gather \ 1
 --image=quay.io/kubevirt/must-gather 2
```

1. The default OpenShift Dedicated **must-gather** image
2. The must-gather image for KubeVirt

4. Create a compressed file from the **must-gather** directory that was just created in your working directory. Make sure you provide the date and cluster ID for the unique must-gather data. For more information about how to find the cluster ID, see How to find the cluster-id or name on OpenShift cluster. For example, on a computer that uses a Linux operating system, run the following command:

```
$ tar cvaf must-gather-`date +"%m-%d-%Y-%H-%M-%S"`-<cluster_id>.tar.gz
<must_gather_local_dir> 1
```

1. Replace **<must_gather_local_dir>** with the actual directory name.

5. Attach the compressed file to your support case on the the **Customer Support** page of the Red Hat Customer Portal.

Additional resources

- OpenShift Dedicated update life cycle

5.1.4. Gathering network logs

You can gather network logs on all nodes in a cluster.

Procedure

1. Run the **oc adm must-gather** command with **-- gather_network_logs**:

```
$ oc adm must-gather -- gather_network_logs
```

NOTE

By default, the **must-gather** tool collects the OVN **nbdb** and **sbdb** databases from all of the nodes in the cluster. Adding the **-- gather_network_logs** option to include additional logs that contain OVN-Kubernetes transactions for OVN **nbdb** database.

2. Create a compressed file from the **must-gather** directory that was just created in your working directory. Make sure you provide the date and cluster ID for the unique must-gather data. For more information about how to find the cluster ID, see How to find the cluster-id or name on OpenShift cluster. For example, on a computer that uses a Linux operating system, run the following command:

```
$ tar cvaf must-gather-`date +"%m-%d-%Y-%H-%M-%S"`-<cluster_id>.tar.gz
<must_gather_local_dir>1
```

1 Replace **<must_gather_local_dir>** with the actual directory name.

3. Attach the compressed file to your support case on the the **Customer Support** page of the Red Hat Customer Portal.

5.1.5. Changing the must-gather storage limit

When using the **oc adm must-gather** command to collect data the default maximum storage for the information is 30% of the storage capacity of the container. After the 30% limit is reached the container is killed and the gathering process stops. Information already gathered is downloaded to your local storage. To run the must-gather command again, you need either a container with more storage capacity or to adjust the maximum volume percentage.

If the container reaches the storage limit, an error message similar to the following example is generated.

Example output

```
Disk usage exceeds the volume percentage of 30% for mounted directory. Exiting...
```

Prerequisites

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

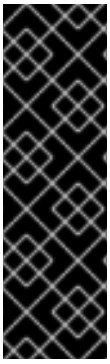
Procedure

- Run the **oc adm must-gather** command with the **volume-percentage** flag. The new value cannot exceed 100.

```
$ oc adm must-gather --volume-percentage <storage_percentage>
```

5.2. ABOUT SUPPORT LOG GATHER

Support Log Gather Operator builds on the functionality of the traditional **must-gather** tool to automate the collection of debugging data. It streamlines troubleshooting by packaging the collected information into a single **.tar** file and automatically uploading it to the specified Red Hat Support case.



IMPORTANT

Support Log Gather is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

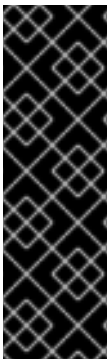
For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

The key features of Support Log Gather include the following:

- **No administrator privileges required**: Enables you to collect and upload logs without needing elevated permissions, making it easier for non-administrators to gather data securely.
- **Simplified log collection**: Collects debugging data from the cluster, such as resource definitions and service logs.
- **Configurable data upload**: Provides configuration options to either automatically upload the **.tar** file to a support case, or store it locally for manual upload.

5.2.1. Installing Support Log Gather by using the web console

You can use the web console to install the Support Log Gather.



IMPORTANT

Support Log Gather is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

Prerequisites

- You have access to the cluster with **cluster-admin** privileges.
- You have access to the OpenShift Dedicated web console.

Procedure

1. Log in to the OpenShift Dedicated web console.
2. Navigate to **Ecosystem → Software Catalog**.
3. In the filter box, enter **Support Log Gather**.

4. Select **Support Log Gather**.
5. From **Version** list, select the Support Log Gather version , and click **Install**.
6. On the **Install Operator** page, configure the installation settings:
    a. Choose the **Installed Namespace** for the Operator.
    The default Operator namespace is **must-gather-operator**. The **must-gather-operator** namespace is created automatically if it does not exist.
    b. Select an **Update approval** strategy:
        - Select **Automatic** to have the Operator Lifecycle Manager (OLM) update the Operator automatically when a newer version is available.
        - Select **Manual** if Operator updates must be approved by a user with appropriate credentials.
    c. Click **Install**.

Verification

1. Verify that the Operator is installed successfully:
    a. Navigate to **Ecosystem → Software Catalog**.
    b. Verify that **Support Log Gather** is listed with a **Status** of **Succeeded** in the **must-gather-operator** namespace.
2. Verify that Support Log Gather pods are running:
    a. Navigate to **Workloads → Pods**
    b. Verify that the status of the Support Log Gather pods is **Running**.
    You can use the Support Log Gather only after the pods are up and running.

5.2.2. Installing Support Log Gather by using the CLI

To enable automated log collection for support cases, you can install Support Log Gather from the command-line interface (CLI).

IMPORTANT

Support Log Gather is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

Prerequisites

- You have access to the cluster with **cluster-admin** privileges.

Procedure

1. Create a new project named **must-gather-operator** by running the following command:

```
$ oc new-project must-gather-operator
```

2. Create an **OperatorGroup** object:

   a. Create a YAML file, for example, **operatorGroup.yaml**, that defines the **OperatorGroup** object:

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: must-gather-operator
  namespace: must-gather-operator
spec: {}
```

   b. Create the **OperatorGroup** object by running the following command:

```
$ oc create -f operatorGroup.yaml
```

3. Create a **Subscription** object:

   a. Create a YAML file, for example, **subscription.yaml**, that defines the **Subscription** object:

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: support-log-gather-operator
  namespace: must-gather-operator
spec:
  channel: tech-preview
  name: support-log-gather-operator
  source: redhat-operators
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
```

   b. Create the **Subscription** object by running the following command:

```
$ oc create -f subscription.yaml
```

Verification

1. Verify the status of the pods in the Operator namespace by running the following command.

```
$ oc get pods
```

Example output

```
NAME                                    READY   STATUS    RESTARTS   AGE
must-gather-operator-657fc74d64-2gg2w                1/1     Running   0          13m
```

The status of all the pods must be **Running**.

2. Verify that the subscription is created by running the following command:

```
$ oc get subscription -n must-gather-operator
```

Example output

```
NAME                        PACKAGE                     SOURCE          CHANNEL
support-log-gather-operator  support-log-gather-operator  redhat-operators  tech-preview
```

3. Verify that the Operator is installed by running the following command:

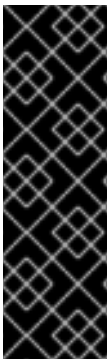
```
$ oc get csv -n must-gather-operator
```

Example output

```
NAME                                DISPLAY             VERSION   REPLACES   PHASE
support-log-gather-operator.v4.20.0  support log gather  4.20.0              Succeeded
```

5.2.3. Configuring a Support Log Gather instance

You must create a **MustGather** custom resource (CR) from the command-line interface (CLI) to automate the collection of diagnostic data from your cluster. This process also automatically uploads the data to a Red Hat Support case.



IMPORTANT

Support Log Gather is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

Prerequisites

- You have installed the OpenShift CLI (**oc**) tool.
- You have installed Support Log Gather in your cluster.
- You have a Red Hat Support case ID.
- You have created a Kubernetes secret containing your Red Hat Customer Portal credentials. The secret must contain a username field and a password field.
- You have created a service account.

Procedure

1. Create a YAML file for the **MustGather** CR, such as **support-log-gather.yaml**, that contains the following basic configuration::

   Example support-log-gather.yaml

```
apiVersion: operator.openshift.io/v1alpha1
kind: MustGather
metadata:
  name: example-mg
  namespace: must-gather-operator
spec:
  serviceAccountName: must-gather-operator
  audit: true
  proxyConfig:
    httpProxy: "http://proxy.example.com:8080"
    httpsProxy: "https://proxy.example.com:8443"
    noProxy: ".example.com,localhost"
  mustGatherTimeout: "1h30m9s"
  uploadTarget:
    type: SFTP
    sftp:
      caseID: "04230315"
      caseManagementAccountSecretRef:
        name: mustgather-creds
      host: "sftp.access.redhat.com"
  retainResourcesOnCompletion: true
  storage:
    type: PersistentVolume
    persistentVolume:
      claim:
        name: mustgather-pvc
      subPath: must-gather-bundles/case-04230315
```

   For more information on the configuration parameters, see "Configuration parameters for MustGather custom resource".

2. Create the **MustGather** object by running the following command:

```
$ oc create -f support-log-gather.yaml
```

Verification

1. Verify that the **MustGather** CR was created by running the following command:

```
$ oc get mustgather
```

   Example output

```
NAME         AGE
example-mg   7s
```

2. Verify the status of the pods in the Operator namespace by running the following command.

```
$ oc get pods
```

Example output

```
NAME                                    READY   STATUS      RESTARTS   AGE
must-gather-operator-657fc74d64-2gg2w   1/1     Running     0          13m
example-mg-gk8m8                        2/2     Running     0          13s
```

A new pod with a name based on the **MustGather** CR must be created. The status of all the pods must be **Running**.

3. To monitor the progress of the file upload, view the logs of the upload container in the job pod by running the following command:

```
oc logs -f pod/example-mg-gk8m8 -c upload
```

When successful, the process must create an archive and upload it to the Red Hat Secure File Transfer Protocol (SFTP) server for the specified case.

Additional resources

- Understanding and creating service accounts

5.2.4. Configuration parameters for MustGather custom resource

You can manage your **MustGather** custom resource (CR) by creating a YAML file that specifies the parameters for data collection and the upload process. The following table provides an overview of the parameters that you can configure in the **MustGather** CR.

Parameter name	Description	Type
spec.audit	Optional: Specifies whether to collect audit logs. The valid values are **true** and **false**. The default value is **false**.	**boolean**
spec.mustGatherTimeout	Optional: Specifies the time limit for the **must-gather** command to complete.	The value must be a floating-point number with a time unit. The valid units are **s** (seconds), **m** (minutes), or **h** (hours). By default, no time is limit set.
spec.proxyConfig	Optional: Defines the proxy configuration to be used. The default value is set to the cluster-level proxy configuration.	**Object**
spec.proxyConfig.httpProxy	Specifies the URL of the proxy for HTTP requests.	URL

Parameter name	Description	Type
spec.proxyConfig.httpsProxy	Specifies the URL of the proxy for HTTPS requests.	
spec.proxyConfig.noProxy	Specifies a comma-separated list of domains for which the proxy must not be used.	List of URLs
spec.retainResourcesOnCompletion	Optional: Specifies whether to retain the **must-gather** job and its related resources after the completion of data collection. The valid values are **true** and **false**. The default value is **false**.	**boolean**
spec.serviceAccountName	Optional: Specifies the name of the service account. The default value is **default**.	**string**
spec.storage	Optional: Defines the storage configuration for the **must-gather** bundle.	**Object**
spec.storage.persistentVolume	Defines the details of the persistent volume.	**Object**
spec.storage.persistentVolume.claim	Defines the details of the persistent volume claim (PVC).	**Object**
spec.storage.persistentVolume.claim.name	Specifies the name of the PVC to be used for storage.	**string**
spec.storage.persistentVolume.subPath	Optional: Specifies the path within the PVC to store the bundle.	**string**
spec.storage.type	Defines the type of storage. The only supported value is **PersistentVolume**.	**string**
spec.uploadTarget	Optional: Defines the upload location for the **must-gather** bundle.	**Object**
spec.uploadTarget.host	Optional: Specifies the destination server for the bundle upload. By default, the bundle is uploaded to **sftp.access.redhat.com**.	By default, the bundle is uploaded to **sftp.access.redhat.com**.
spec.uploadTarget.sftp.caseID	Specifies the Red Hat Support case ID for which the diagnostic data is collected.	**string**

Parameter name	Description	Type
spec.uploadTarget.sftp.caseManagementAccountSecretRef	Defines the credentials required for authenticating and uploading the files to the Red Hat Customer Portal support case. The value must contain a **username** and **password** field.	**Object**
spec.uploadTarget.sftp.caseManagementAccountSecretRef.name	Specifies the name of the Kubernetes secret that contains the credentials.	**string**
spec.uploadTarget.sftp.internalUser	Optional: Specifies whether the user provided in the **caseManagementAccountSecretRef** is a Red Hat internal user. The valid values are **true** and **false**. The default value is **false**.	**boolean**
spec.uploadTarget.type	Specifies the type of upload location for the **must-gather** bundle. The only supported value is **SFTP**.	**string**

NOTE

If you do not specify **spec.uploadTarget** or **spec.storage**, the pod saves the data to an ephemeral volume and the data is permanently deleted when the pod terminates.

5.2.5. Uninstalling Support Log Gather

You can uninstall the Support Log Gather by using the web console.

Prerequisites

- You have access to the cluster with **cluster-admin** privileges.
- You have access to the OpenShift Dedicated web console.
- The Support Log Gather is installed.

Procedure

1. Log in to the OpenShift Dedicated web console.
2. Uninstall the Support Log Gather Operator.
   a. Navigate to **Ecosystem → Installed Operators**.
   b. Click the Options menu  next to the **Support Log Gather** entry and click **Uninstall Operator**.

c. In the confirmation dialog, click **Uninstall**.

5.2.6. Removing Support Log Gather resources

Once you have uninstalled the Support Log Gather, you can remove the associated resources from your cluster.

Prerequisites

- You have access to the cluster with **cluster-admin** privileges.
- You have access to the OpenShift Dedicated web console.

Procedure

1. Log in to the OpenShift Dedicated web console.
2. Delete the component deployments in the must-gather-operator namespace.:
   a. Click the **Project** drop-down menu to view the list of all available projects, and select the **must-gather-operator** project.
   b. Navigate to **Workloads → Deployments**.
   c. Select the deployment that you want to delete.
   d. Click the **Actions** drop-down menu, and select **Delete Deployment**
   e. In the confirmation dialog box, click **Delete** to delete the deployment.
   f. Alternatively, delete deployments of the components present in the **must-gather-operator** namespace by using the command-line interface (CLI).

```
$ oc delete deployment -n must-gather-operator -l operators.coreos.com/support-log-gather-operator.must-gather-operator
```

3. Optional: Remove the custom resource definitions (CRDs) that were installed by the Support Log Gather:
   a. Navigate to **Administration → CustomResourceDefinitions**.
   b. Enter **MustGather** in the **Name** field to filter the CRDs.
   c. Click the Options menu  next to each of the following CRDs, and select **Delete Custom Resource Definition**:
      - **MustGather**
4. Optional: Remove the **must-gather-operator** namespace.
   a. Navigate to **Administration → Namespaces**.

b. Click the Options menu  next to the **must-gather-operator** and select **Delete Namespace**.

c. In the confirmation dialog box, enter **must-gather-operator** and click **Delete**.

5.3. OBTAINING YOUR CLUSTER ID

When providing information to Red Hat Support, it is helpful to provide the unique identifier for your cluster. You can have your cluster ID autofilled by using the OpenShift Dedicated web console. You can also manually obtain your cluster ID by using the web console or the OpenShift CLI (**oc**).

Prerequisites

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

Procedure

- To manually obtain your cluster ID using OpenShift Cluster Manager:
  a. Navigate to **Cluster List**.
  b. Click on the name of the cluster you need to open a support case for.
  c. Find the value in the **Cluster ID** field of the **Details** section of the **Overview** tab.
- To open a support case and have your cluster ID autofilled using the web console:
  a. From the toolbar, navigate to **(?) Help** and select **Share Feedback** from the list.
  b. Click **Open a support case** from the **Tell us about your experience** window.
- To manually obtain your cluster ID using the web console:
  a. Navigate to **Home → Overview**.
  b. The value is available in the **Cluster ID** field of the **Details** section.
- To obtain your cluster ID using the OpenShift CLI (**oc**), run the following command:

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

5.4. QUERYING CLUSTER NODE JOURNAL LOGS

You can gather **journald** unit logs and other logs within **/var/log** on individual cluster nodes.

Prerequisites

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

NOTE

In OpenShift Dedicated deployments, customers who are not using the Customer Cloud Subscription (CCS) model cannot use the **oc adm node-logs** command as it requires **cluster-admin** privileges.

- You have installed the OpenShift CLI (**oc**).

Procedure

1. Query **kubelet journald** unit logs from OpenShift Dedicated cluster nodes. The following example queries control plane nodes only:

```
$ oc adm node-logs --role=master -u kubelet 1
```

   - **kubelet**: Replace as appropriate to query other unit logs.

2. Collect logs from specific subdirectories under **/var/log/** on cluster nodes.

   a. Retrieve a list of logs contained within a **/var/log/** subdirectory. The following example lists files in **/var/log/openshift-apiserver/** on all control plane nodes:

```
$ oc adm node-logs --role=master --path=openshift-apiserver
```

   b. Inspect a specific log within a **/var/log/** subdirectory. The following example outputs **/var/log/openshift-apiserver/audit.log** contents from all control plane nodes:

```
$ oc adm node-logs --role=master --path=openshift-apiserver/audit.log
```

5.5. NETWORK TRACE METHODS

Collecting network traces, in the form of packet capture records, can assist Red Hat Support with troubleshooting network issues.

OpenShift Dedicated supports two ways of performing a network trace. Review the following table and choose the method that meets your needs.

Table 5.3. Supported methods of collecting a network trace

Method	Benefits and capabilities
Collecting a host network trace	You perform a packet capture for a duration that you specify on one or more nodes at the same time. The packet capture files are transferred from nodes to the client machine when the specified duration is met. You can troubleshoot why a specific action triggers network communication issues. Run the packet capture, perform the action that triggers the issue, and use the logs to diagnose the issue.

Method	Benefits and capabilities
Collecting a network trace from an OpenShift Dedicated node or container	You perform a packet capture on one node or one container. You run the **tcpdump** command interactively, so you can control the duration of the packet capture. You can start the packet capture manually, trigger the network communication issue, and then stop the packet capture manually. This method uses the **cat** command and shell redirection to copy the packet capture data from the node or container to the client machine.

5.6. COLLECTING A HOST NETWORK TRACE

Sometimes, troubleshooting a network-related issue is simplified by tracing network communication and capturing packets on multiple nodes at the same time.

You can use a combination of the **oc adm must-gather** command and the **registry.redhat.io/openshift4/network-tools-rhel8** container image to gather packet captures from nodes. Analyzing packet captures can help you troubleshoot network communication issues.

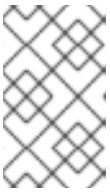
The **oc adm must-gather** command is used to run the **tcpdump** command in pods on specific nodes. The **tcpdump** command records the packet captures in the pods. When the **tcpdump** command exits, the **oc adm must-gather** command transfers the files with the packet captures from the pods to your client machine.

TIP

The sample command in the following procedure demonstrates performing a packet capture with the **tcpdump** command. However, you can run any command in the container image that is specified in the **--image** argument to gather troubleshooting information from multiple nodes at the same time.

Prerequisites

- You are logged in to OpenShift Dedicated as a user with the **cluster-admin** role.



NOTE

In OpenShift Dedicated deployments, customers who are not using the Customer Cloud Subscription (CCS) model cannot use the **oc adm must-gather** command as it requires **cluster-admin** privileges.

- You have installed the OpenShift CLI (**oc**).

Procedure

1. Run a packet capture from the host network on some nodes by running the following command:

```
$ oc adm must-gather \
    --dest-dir /tmp/captures \// <.>
    --source-dir '/tmp/tcpdump/' \// <.>
```

```
--image registry.redhat.io/openshift4/network-tools-rhel8:latest \// <.>
--node-selector 'node-role.kubernetes.io/worker' \// <.>
--host-network=true \// <.>
--timeout 30s \// <.>
-- \
tcpdump -i any \// <.>
-w /tmp/tcpdump/%Y-%m-%dT%H:%M:%S.pcap -W 1 -G 300
```

<.> The **--dest-dir** argument specifies that **oc adm must-gather** stores the packet captures in directories that are relative to **/tmp/captures** on the client machine. You can specify any writable directory. <.> When **tcpdump** is run in the debug pod that **oc adm must-gather** starts, the **--source-dir** argument specifies that the packet captures are temporarily stored in the **/tmp/tcpdump** directory on the pod. <.> The **--image** argument specifies a container image that includes the **tcpdump** command. <.> The **--node-selector** argument and example value specifies to perform the packet captures on the worker nodes. As an alternative, you can specify the **--node-name** argument instead to run the packet capture on a single node. If you omit both the **--node-selector** and the **--node-name** argument, the packet captures are performed on all nodes. <.> The **--host-network=true** argument is required so that the packet captures are performed on the network interfaces of the node. <.> The **--timeout** argument and value specify to run the debug pod for 30 seconds. If you do not specify the **--timeout** argument and a duration, the debug pod runs for 10 minutes. <.> The **-i any** argument for the **tcpdump** command specifies to capture packets on all network interfaces. As an alternative, you can specify a network interface name.

2. Perform the action, such as accessing a web application, that triggers the network communication issue while the network trace captures packets.

3. Review the packet capture files that **oc adm must-gather** transferred from the pods to your client machine:

```
tmp/captures
├──── event-filter.html
├──── ip-10-0-192-217-ec2-internal 1
│     └──── registry-redhat-io-openshift4-network-tools-rhel8-sha256-bca...
│           └──── 2022-01-13T19:31:31.pcap
├──── ip-10-0-201-178-ec2-internal 2
│     └──── registry-redhat-io-openshift4-network-tools-rhel8-sha256-bca...
│           └──── 2022-01-13T19:31:30.pcap
├──── ip-...
└──── timestamp
```

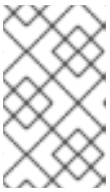
1 1 2 The packet captures are stored in directories that identify the hostname, container, and file name. If you did not specify the **--node-selector** argument, then the directory level for the hostname is not present.

5.7. COLLECTING A NETWORK TRACE FROM AN OPENSHIFT DEDICATED NODE OR CONTAINER

When investigating potential network-related OpenShift Dedicated issues, Red Hat Support might request a network packet trace from a specific OpenShift Dedicated cluster node or from a specific container. The recommended method to capture a network trace in OpenShift Dedicated is through a debug pod.

Prerequisites

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



NOTE

In OpenShift Dedicated deployments, customers who are not using the Customer Cloud Subscription (CCS) model cannot use the **oc debug** command as it requires **cluster-admin** privileges.

- You have installed the OpenShift CLI (**oc**).
- You have an existing Red Hat Support case ID.

Procedure

1. Obtain a list of cluster nodes:

```
$ oc get nodes
```

2. Enter into a debug session on the target node. This step instantiates a debug pod called **<node_name>-debug**:

```
$ oc debug node/my-cluster-node
```

3. Set **/host** as the root directory within the debug shell. The debug pod mounts the host's root file system in **/host** within the pod. By changing the root directory to **/host**, you can run binaries contained in the host's executable paths:

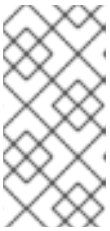
```
# chroot /host
```

4. From within the **chroot** environment console, obtain the node's interface names:

```
# ip ad
```

5. Start a **toolbox** container, which includes the required binaries and plugins to run **sosreport**:

```
# toolbox
```



NOTE

If an existing **toolbox** pod is already running, the **toolbox** command outputs **'toolbox-' already exists. Trying to start…**. To avoid **tcpdump** issues, remove the running toolbox container with **podman rm toolbox-** and spawn a new toolbox container.

6. Initiate a **tcpdump** session on the cluster node and redirect output to a capture file. This example uses **ens5** as the interface name:

```
$ tcpdump -nn -s 0 -i ens5 -w /host/var/tmp/my-cluster-node_$(date +%d_%m_%Y-%H_%M_%S-%Z).pcap
```

1

1 The **tcpdump** capture file's path is outside of the **chroot** environment because the toolbox container mounts the host's root directory at **/host**.

7. If a **tcpdump** capture is required for a specific container on the node, follow these steps.

   a. Determine the target container ID. The **chroot host** command precedes the **crictl** command in this step because the toolbox container mounts the host's root directory at **/host**:

   ```
   # chroot /host crictl ps
   ```

   b. Determine the container's process ID. In this example, the container ID is **a7fe32346b120**:

   ```
   # chroot /host crictl inspect --output yaml a7fe32346b120 | grep 'pid' | awk '{print $2}'
   ```

   c. Initiate a **tcpdump** session on the container and redirect output to a capture file. This example uses **49628** as the container's process ID and **ens5** as the interface name. The **nsenter** command enters the namespace of a target process and runs a command in its namespace. because the target process in this example is a container's process ID, the **tcpdump** command is run in the container's namespace from the host:

   ```
   # nsenter -n -t 49628 -- tcpdump -nn -i ens5 -w /host/var/tmp/my-cluster-node-my-container_$(date +%d_%m_%Y-%H_%M_%S-%Z).pcap 1
   ```

   1 The **tcpdump** capture file's path is outside of the **chroot** environment because the toolbox container mounts the host's root directory at **/host**.

8. Provide the **tcpdump** capture file to Red Hat Support for analysis, using one of the following methods.

   - Upload the file to an existing Red Hat support case.

     a. Concatenate the **sosreport** archive by running the **oc debug node/<node_name>** command and redirect the output to a file. This command assumes you have exited the previous **oc debug** session:

     ```
     $ oc debug node/my-cluster-node -- bash -c 'cat /host/var/tmp/my-tcpdump-capture-file.pcap' > /tmp/my-tcpdump-capture-file.pcap 1
     ```

     1 The debug container mounts the host's root directory at **/host**. Reference the absolute path from the debug container's root directory, including **/host**, when specifying target files for concatenation.

     b. Navigate to an existing support case within the **Customer Support** page of the Red Hat Customer Portal.

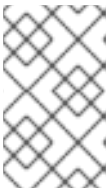
     c. Select **Attach files** and follow the prompts to upload the file.

5.8. PROVIDING DIAGNOSTIC DATA TO RED HAT SUPPORT

When investigating OpenShift Dedicated issues, Red Hat Support might ask you to upload diagnostic data to a support case. Files can be uploaded to a support case through the Red Hat Customer Portal.

Prerequisites

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



NOTE

In OpenShift Dedicated deployments, customers who are not using the Customer Cloud Subscription (CCS) model cannot use the **oc debug** command as it requires **cluster-admin** privileges.

- You have installed the OpenShift CLI (**oc**).
- You have an existing Red Hat Support case ID.

Procedure

- Upload diagnostic data to an existing Red Hat support case through the Red Hat Customer Portal.

  a. Concatenate a diagnostic file contained on an OpenShift Dedicated node by using the **oc debug node/<node_name>** command and redirect the output to a file. The following example copies **/host/var/tmp/my-diagnostic-data.tar.gz** from a debug container to **/var/tmp/my-diagnostic-data.tar.gz**:

```
$ oc debug node/my-cluster-node -- bash -c 'cat /host/var/tmp/my-diagnostic-data.tar.gz'
> /var/tmp/my-diagnostic-data.tar.gz 1
```

  1 The debug container mounts the host's root directory at **/host**. Reference the absolute path from the debug container's root directory, including **/host**, when specifying target files for concatenation.

  b. Navigate to an existing support case within the **Customer Support** page of the Red Hat Customer Portal.

  c. Select **Attach files** and follow the prompts to upload the file.

5.9. ABOUT TOOLBOX

toolbox is a tool that starts a container on a Red Hat Enterprise Linux CoreOS (RHCOS) system. The tool is primarily used to start a container that includes the required binaries and plugins that are needed to run commands such as **sosreport**.

The primary purpose for a **toolbox** container is to gather diagnostic information and to provide it to Red Hat Support. However, if additional diagnostic tools are required, you can add RPM packages or run an image that is an alternative to the standard support tools image.

5.9.1. Installing packages to a toolbox container

By default, running the **toolbox** command starts a container with the **registry.redhat.io/rhel9/support-**

tools:latest image. This image contains the most frequently used support tools. If you need to collect node-specific data that requires a support tool that is not part of the image, you can install additional packages.

Prerequisites

- You have accessed a node with the **oc debug node/<node_name>** command.
- You can access your system as a user with root privileges.

Procedure

1. Set **/host** as the root directory within the debug shell. The debug pod mounts the host's root file system in **/host** within the pod. By changing the root directory to **/host**, you can run binaries contained in the host's executable paths:

```
# chroot /host
```

2. Start the toolbox container:

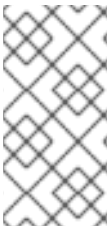
```
# toolbox
```

3. Install the additional package, such as **wget**:

```
# dnf install -y <package_name>
```

5.9.2. Starting an alternative image with toolbox

By default, running the **toolbox** command starts a container with the **registry.redhat.io/rhel9/support-tools:latest** image.



NOTE

You can start an alternative image by creating a **.toolboxrc** file and specifying the image to run. However, running an older version of the **support-tools** image, such as **registry.redhat.io/rhel8/support-tools:latest**, is not supported on OpenShift Dedicated 4.

Prerequisites

- You have accessed a node with the **oc debug node/<node_name>** command.
- You can access your system as a user with root privileges.

Procedure

1. Set **/host** as the root directory within the debug shell. The debug pod mounts the host's root file system in **/host** within the pod. By changing the root directory to **/host**, you can run binaries contained in the host's executable paths:

```
# chroot /host
```

2. Optional: If you need to use an alternative image instead of the default image, create a **.toolboxrc** file in the home directory for the root user ID, and specify the image metadata:

```
REGISTRY=quay.io 1
IMAGE=fedora/fedora:latest 2
TOOLBOX_NAME=toolbox-fedora-latest 3
```

1. Optional: Specify an alternative container registry.
2. Specify an alternative image to start.
3. Optional: Specify an alternative name for the toolbox container.

3. Start a toolbox container by entering the following command:

```
# toolbox
```

NOTE

If an existing **toolbox** pod is already running, the **toolbox** command outputs **'toolbox-' already exists. Trying to start…**. To avoid issues with **sosreport** plugins, remove the running toolbox container with **podman rm toolbox-** and then spawn a new toolbox container.

CHAPTER 6. SUMMARIZING CLUSTER SPECIFICATIONS

6.1. SUMMARIZING CLUSTER SPECIFICATIONS BY USING A CLUSTER VERSION OBJECT

You can obtain a summary of OpenShift Dedicated 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
```

Example output

```
NAME      VERSION   AVAILABLE   PROGRESSING   SINCE   STATUS
version   4.13.8    True        False         8h      Cluster version is 4.13.8
```

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
```

```
    Verified:          false
    Version:           4.13.8
# ...
```

CHAPTER 7. TROUBLESHOOTING

7.1. TROUBLESHOOTING AN OPENSHIFT DEDICATED ON GOOGLE CLOUD CLUSTER DEPLOYMENT

OpenShift Dedicated on Google Cloud cluster deployment errors can occur for several reasons, including insufficient quota limits and settings, incorrectly inputted data, incompatible configurations, and so on.

Learn how to resolve common OpenShift Dedicated on Google Cloud cluster installation errors in the following sections.

7.1.1. Troubleshooting OpenShift Dedicated on Google Cloud installation error codes

The following table lists OpenShift Dedicated on Google Cloud installation error codes and what you can do to resolve these errors.

Table 7.1. OpenShift Dedicated on Google Cloud installation error codes

Error code	Description	Resolution
OCM3022	Invalid Google Cloud project ID.	Verify the project ID in the Google cloud console and retry cluster creation.
OCM3023	Google Cloud instance type not found.	Verify the instance type and retry cluster creation. For more information about OpenShift Dedicated on Google Cloud instance types, see *Google Cloud instance types* in the *Additional resources* section.
OCM3024	Google Cloud precondition failed.	Verify the organization policy constraints and retry cluster creation. For more information about organization policy constraints, see Organization policy constraints.

Error code	Description	Resolution
OCM3025	Google Cloud SSD quota limit exceeded.	Check your available persistent disk SSD quota either in the Google Cloud console or in the **gcloud** CLI. There must be at least 896 GB of SSD available. Increase the SSD quota limit and retry cluster creation. For more information about managing persistent disk SSD quota, see Allocation quotas.
OCM3026	Google Cloud compute quota limit exceeded.	Increase your CPU compute quota and retry cluster installation. For more information about the CPU compute quota, see Compute Engine quota and limits overview.
OCM3027	Google Cloud service account quota limit exceeded.	Ensure your quota allows for additional unused service accounts. Check your current usage for quotas in your Google Cloud account and try again. For more information about managing your quotas, see Manage your quotas using the console.

Additional resources

- For more information about OpenShift Dedicated on Google Cloud instance types, see Google Cloud instance types.

7.2. VERIFYING NODE HEALTH

7.2.1. 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
```

7.3. TROUBLESHOOTING OPERATOR ISSUES

Operators are a method of packaging, deploying, and managing an OpenShift Dedicated application. They act like an extension of the software vendor's engineering team, watching over an OpenShift Dedicated 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.

OpenShift Dedicated 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 software catalog using the OpenShift Dedicated 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).

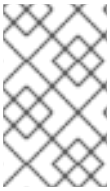
If you experience Operator issues, verify Operator subscription status. Check Operator pod health across the cluster and gather Operator logs for diagnosis.

7.3.1. Operator subscription condition types

Subscriptions can report the following condition types:

Table 7.2. Subscription condition types

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



NOTE

Default OpenShift Dedicated 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

7.3.2. 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:

```
$ oc get subs -n <operator_namespace>
```

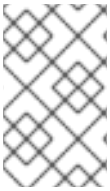
2. Use the **oc describe** command to inspect a **Subscription** resource:

```
$ 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:
  Last Transition Time:  2019-07-29T13:42:57Z
  Message:               all available catalogsources are healthy
  Reason:                AllCatalogSourcesHealthy
  Status:                False
  Type:                  CatalogSourcesUnhealthy
# ...
```



NOTE

Default OpenShift Dedicated 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.

7.3.3. 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
certified-operators   Certified Operators   grpc   Red Hat     55m
community-operators   Community Operators   grpc   Red Hat     55m
example-catalog       Example Catalog       grpc   Example Org 2m25s
redhat-operators      Red Hat Operators     grpc   Red Hat     55m
```

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-86bfc75f9b-jkgbc 1/1     Running            0          42m
example-catalog-bwt8z                 0/1     ImagePullBackOff   0          3m55s
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-jyryyg2-f76d1-ggdbq-worker-b-vsxjd/10.0.128.2
...
Events:
  Type     Reason          Age                From               Message
  ----     ------          ----               ----               -------
  Normal   Scheduled       48s                default-scheduler  Successfully assigned openshift-
marketplace/example-catalog-bwt8z to ci-ln-jyryyf2-f76d1-fgdbq-worker-b-vsxjd
  Normal   AddedInterface  47s                multus             Add eth0 [10.131.0.40/23] from
openshift-sdn
  Normal   BackOff         20s (x2 over 46s)  kubelet            Back-off pulling image
"quay.io/example-org/example-catalog:v1"
  Warning  Failed          20s (x2 over 46s)  kubelet            Error: ImagePullBackOff
  Normal   Pulling         8s (x3 over 47s)   kubelet            Pulling image "quay.io/example-
org/example-catalog:v1"
  Warning  Failed          8s (x3 over 47s)   kubelet            Failed to pull image
"quay.io/example-org/example-catalog:v1": rpc error: code = Unknown desc = reading
manifest v1 in quay.io/example-org/example-catalog: unauthorized: access to the requested
resource is not authorized
  Warning  Failed          8s (x3 over 47s)   kubelet            Error: ErrImagePull
```

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

- Operator Lifecycle Manager concepts and resources → Catalog source
- States of Connectivity (gRPC documentation)

7.3.4. 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 up-time information:

```
$ oc get clusteroperators
```

2. List Operator pods running in the Operator's namespace, plus pod status, restarts, and age:

```
$ oc get pod -n <operator_namespace>
```

3. Output a detailed Operator pod summary:

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

7.3.5. 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

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:

```
$ 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>
```

NOTE

OpenShift Dedicated 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 OpenShift Dedicated 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>**.

7.4. INVESTIGATING POD ISSUES

OpenShift Dedicated 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 OpenShift Dedicated 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.

7.4.1. Understanding pod error states

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 7.3. Pod error states

Pod error state	Description
ErrImagePull	Generic image retrieval error.
ErrImagePullBackOff	Image retrieval failed and is backed off.
ErrInvalidImageName	The specified image name was invalid.
ErrImageInspect	Image inspection did not succeed.
ErrImageNeverPull	**PullPolicy** is set to **NeverPullImage** and the target image is not present locally on the host.

Pod error state	Description
ErrRegistryUnavailable	When attempting to retrieve an image from a registry, an HTTP error was encountered.
ErrContainerNotFound	The specified container is either not present or not managed by the kubelet, within the declared pod.
ErrRunInitContainer	Container initialization failed.
ErrRunContainer	None of the pod's containers started successfully.
ErrKillContainer	None of the pod's containers were killed successfully.
ErrCrashLoopBackOff	A container has terminated. The kubelet will not attempt to restart it.
ErrVerifyNonRoot	A container or image attempted to run with root privileges.
ErrCreatePodSandbox	Pod sandbox creation did not succeed.
ErrConfigPodSandbox	Pod sandbox configuration was not obtained.
ErrKillPodSandbox	A pod sandbox did not stop successfully.
ErrSetupNetwork	Network initialization failed.
ErrTeardownNetwork	Network termination failed.

7.4.2. 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:

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

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

```
$ oc get pods
```

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

```
$ 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:

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

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

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

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:

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

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

```
$ oc get events
```

7.4.3. 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

Procedure

1. Query logs for a specific pod:

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

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

```
$ 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:

```
$ 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
-rw-rw-r--. 1 root utmp    0 Jul 10 10:31 lastlog
drwx------. 2 root root   23 Aug 11 11:14 openshift-apiserver
drwx------. 2 root root    6 Jul 10 10:31 private
drwxr-xr-x. 1 root root   22 Mar  9 08:05 rhsm
-rw-rw-r--. 1 root utmp    0 Jul 10 10:31 wtmp
```

   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>
```

7.4.4. 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:

```
$ oc rsh <pod_name> 1
```

1 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> 1
```

1 Enter **Ctrl+C** to cancel the port forwarding session.

7.4.5. 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.

7.4.6. 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
   ```

   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)

(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.

7.5. TROUBLESHOOTING THE SOURCE-TO-IMAGE PROCESS

7.5.1. Strategies for Source-to-Image troubleshooting

Use Source-to-Image (S2I) to build reproducible, Docker-formatted container images. You can create ready-to-run images by injecting application source code into a container image and assembling a new image. The new image incorporates the base image (the builder) and built source.

Procedure

1. To determine where in the S2I process a failure occurs, you can observe the state of the pods relating to each of the following S2I stages:

   a. **During the build configuration stage**, a build pod is used to create an application container image from a base image and application source code.

   b. **During the deployment configuration stage**, a deployment pod is used to deploy application pods from the application container image that was built in the build configuration stage. The deployment pod also deploys other resources such as services and routes. The deployment configuration begins after the build configuration succeeds.

   c. **After the deployment pod has started the application pods**, application failures can occur within the running application pods. For instance, an application might not behave as expected even though the application pods are in a **Running** state. In this scenario, you can access running application pods to investigate application failures within a pod.

2. When troubleshooting S2I issues, follow this strategy:

   a. Monitor build, deployment, and application pod status.

   b. Determine the stage of the S2I process where the problem occurred.

   c. Review logs corresponding to the failed stage.

7.5.2. Gathering Source-to-Image diagnostic data

The S2I tool runs a build pod and a deployment pod in sequence. The deployment pod is responsible for deploying the application pods based on the application container image created in the build stage. Watch build, deployment and application pod status to determine where in the S2I process a failure occurs. Then, focus diagnostic data collection accordingly.

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

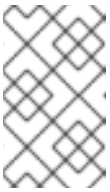
1. Watch the pod status throughout the S2I process to determine at which stage a failure occurs:

```
$ oc get pods -w 1
```

1 Use **-w** to monitor pods for changes until you quit the command using **Ctrl+C**.

2. Review a failed pod's logs for errors.
   - **If the build pod fails**, review the build pod's logs:

```
$ oc logs -f pod/<application_name>-<build_number>-build
```



NOTE

Alternatively, you can review the build configuration's logs using **oc logs -f bc/<application_name>**. The build configuration's logs include the logs from the build pod.

   - **If the deployment pod fails**, review the deployment pod's logs:

```
$ oc logs -f pod/<application_name>-<build_number>-deploy
```

NOTE

Alternatively, you can review the deployment configuration's logs using **oc logs -f dc/<application_name>**. This outputs logs from the deployment pod until the deployment pod completes successfully. The command outputs logs from the application pods if you run it after the deployment pod has completed. After a deployment pod completes, its logs can still be accessed by running **oc logs -f pod/<application_name>-<build_number>-deploy**.

   - **If an application pod fails, or if an application is not behaving as expected within a running application pod**, review the application pod's logs:

```
$ oc logs -f pod/<application_name>-<build_number>-<random_string>
```

7.5.3. Gathering application diagnostic data to investigate application failures

Application failures can occur within running application pods. In these situations, you can retrieve diagnostic information with these strategies:

- Review events relating to the application pods.

- Review the logs from the application pods, including application-specific log files that are not collected by the OpenShift Logging framework.
- Test application functionality interactively and run diagnostic tools in an application container.

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 events relating to a specific application pod. The following example retrieves events for an application pod named **my-app-1-akdlg**:

```
$ oc describe pod/my-app-1-akdlg
```

2. Review logs from an application pod:

```
$ oc logs -f pod/my-app-1-akdlg
```

3. Query specific logs within a running application pod. Logs that are sent to stdout are collected by the OpenShift Logging framework and are included in the output of the preceding command. The following query is only required for logs that are not sent to stdout.

   a. If an application log can be accessed without root privileges within a pod, concatenate the log file as follows:

```
$ oc exec my-app-1-akdlg -- cat /var/log/my-application.log
```

   b. If root access is required to view an application log, you can start a debug container with root privileges and then view the log file from within the container. Start the debug container from the project's **DeploymentConfig** object. Pod users typically run with non-root privileges, but running troubleshooting pods with temporary root privileges can be useful during issue investigation:

```
$ oc debug dc/my-deployment-configuration --as-root -- cat /var/log/my-application.log
```

NOTE

You can access an interactive shell with root access within the debug pod if you run **oc debug dc/<deployment_configuration> --as-root** without appending **-- <command>**.

4. Test application functionality interactively and run diagnostic tools, in an application container with an interactive shell.

   a. Start an interactive shell on the application container:

```
$ oc exec -it my-app-1-akdlg /bin/bash
```

   b. Test application functionality interactively from within the shell. For example, you can run

the container's entry point command and observe the results. Then, test changes from the command line directly, before updating the source code and rebuilding the application container through the S2I process.

c. Run diagnostic binaries available within the container.

NOTE

Root privileges are required to run some diagnostic binaries. In these situations you can start a debug pod with root access, based on a problematic pod's **DeploymentConfig** object, by running **oc debug dc/<deployment_configuration> --as-root**. Then, you can run diagnostic binaries as root from within the debug pod.

7.6. TROUBLESHOOTING STORAGE ISSUES

7.6.1. 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.

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

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

7.7. INVESTIGATING MONITORING ISSUES

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

Use these procedures if the following issues occur:

- Your own metrics are unavailable.
- Prometheus is consuming a lot of disk space.
- The **KubePersistentVolumeFillingUp** alert is firing for Prometheus.

7.7.1. 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

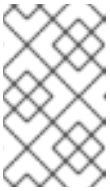
- You have access to the cluster as a user with the **dedicated-admin** role.
- You have installed the OpenShift CLI (**oc**).
- You have enabled and configured monitoring for user-defined projects.
- You have created a **ServiceMonitor** resource.

Procedure

1. Ensure that your project and resources are not excluded from user workload monitoring. The following examples use the **ns1** project.

   a. Verify that the project *does not* have the **openshift.io/user-monitoring=false** label attached:

   ```
   $ oc get namespace ns1 --show-labels | grep 'openshift.io/user-monitoring=false'
   ```

   NOTE

   The default label set for user workload projects is **openshift.io/user-monitoring=true**. However, the label is not visible unless you manually apply it.

   b. Verify that the **ServiceMonitor** and **PodMonitor** resources *do not* have the **openshift.io/user-monitoring=false** label attached. The following example checks the **prometheus-example-monitor** service monitor.

   ```
   $ oc -n ns1 get servicemonitor prometheus-example-monitor --show-labels | grep
   'openshift.io/user-monitoring=false'
   ```

c. If the label is attached, remove the label:

Example of removing the label from the project

```
$ oc label namespace ns1 'openshift.io/user-monitoring-'
```

Example of removing the label from the resource

```
$ oc -n ns1 label servicemonitor prometheus-example-monitor 'openshift.io/user-monitoring-'
```

Example output

```
namespace/ns1 unlabeled
```

2. Check that the corresponding labels match in the service and **ServiceMonitor** resource configurations. The following examples use the **prometheus-example-app** service, the **prometheus-example-monitor** service monitor, and the **ns1** project.

a. Obtain the label defined in the service.

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

Example output

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

b. Check that the **matchLabels** definition in the **ServiceMonitor** resource configuration matches the label output in the previous step.

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

Example output

```
apiVersion: v1
kind: ServiceMonitor
metadata:
  name: prometheus-example-monitor
  namespace: ns1
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.

3. 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:

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

   Example output

   ```
   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"
   servicemonitor=eagle/eagle namespace=openshift-user-workload-monitoring
   prometheus=user-workload
   ```

4. Review the target status for your endpoint on the **Metrics targets** page in the OpenShift Dedicated web console UI.

   a. Log in to the OpenShift Dedicated web console and go to **Observe → Targets**.

   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.

5. 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
kind: 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 affected **prometheus-operator** pod is automatically redeployed.

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

```
$ 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:

```
$ 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

- Specifying how a service is monitored
- Getting detailed information about a metrics target

7.7.2. 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 time series database (TSDB) status using the Prometheus HTTP API** for more information about which labels are creating the most time series data. Doing so requires cluster administrator privileges.
- **Check the number of scrape samples** that are being collected.
- **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 OpenShift Dedicated web console, go to **Observe → Metrics**.
2. Enter a Prometheus Query Language (PromQL) query in the **Expression** field. The following example queries help to identify high cardinality metrics that might result in high disk space consumption:
   - By running the following query, you can identify the ten jobs that have the highest number of scrape samples:

     ```
     topk(10, max by(namespace, job) (topk by(namespace, job) (1,
     scrape_samples_post_metric_relabeling)))
     ```

   - By running the following query, you can pinpoint time series churn by identifying the ten jobs that have created the most time series data in the last hour:

     ```
     topk(10, sum by(namespace, job) (sum_over_time(scrape_series_added[1h])))
     ```

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 OpenShift Dedicated project**, create a Red Hat support case on the Red Hat Customer Portal.

4. Review the TSDB status using the Prometheus HTTP API by following these steps when logged in as a **dedicated-admin**:

   a. Get the Prometheus API route URL by running the following command:

   ```
   $ HOST=$(oc -n openshift-monitoring get route prometheus-k8s -
   ojsonpath='{.status.ingress[].host}')
   ```

   b. Extract an authentication token by running the following command:

   ```
   $ TOKEN=$(oc whoami -t)
   ```

   c. Query the TSDB status for Prometheus by running the following command:

   ```
   $ curl -H "Authorization: Bearer $TOKEN" -k "https://$HOST/api/v1/status/tsdb"
   ```

   Example output

   ```
   "status": "success","data":{"headStats":{"numSeries":507473,
   "numLabelPairs":19832,"chunkCount":946298,"minTime":1712253600010,
   "maxTime":1712257935346},"seriesCountByMetricName":
   [{"name":"etcd_request_duration_seconds_bucket","value":51840},
   {"name":"apiserver_request_sli_duration_seconds_bucket","value":47718},
   ...
   ```

Additional resources

- Setting scrape intervals, evaluation intervals, and enforced limits for user-defined projects

7.8. DIAGNOSING OPENSHIFT CLI (oc) ISSUES

7.8.1. Understanding OpenShift CLI (oc) log levels

With the OpenShift CLI (**oc**), you can create applications and manage OpenShift Dedicated 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 7.4. OpenShift CLI (oc) log levels

Log level	Description
1 to 5	No additional logging to stderr.
6	Log API requests to stderr.
7	Log API requests and headers to stderr.
8	Log API requests, headers, and body, plus API response headers and body to stderr.
9	Log API requests, headers, and body, API response headers and body, plus **curl** requests to stderr.
10	Log API requests, headers, and body, API response headers and body, plus **curl** requests to stderr, in verbose detail.

7.8.2. Specifying OpenShift CLI (oc) log levels

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

The OpenShift Dedicated 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:

```
$ 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:

```
$ oc whoami -t
```

Example output

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

7.9. RED HAT MANAGED RESOURCES

7.9.1. Overview

The following covers all OpenShift Dedicated resources that are managed or protected by the Service Reliability Engineering Platform (SRE-P) Team. Customers must not modify these resources because doing so can lead to cluster instability.

7.9.2. Hive managed resources

The following list displays the OpenShift Dedicated 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 OpenShift Dedicated clusters. Changes to OpenShift Dedicated 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 7.1. 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-vols-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-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-compliance-monkey
  - 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-suricata
```

```
    - name: openshift-validation-webhook
    - name: openshift-velero
    - name: openshift-monitoring
    - name: openshift
    - name: openshift-cluster-version
    - name: keycloak
    - name: goalert
    - name: configure-goalert-operator
    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-network-diagnostics
    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-splunk-forwarder-operator
    name: splunk-forwarder-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: 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-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-techpreviewnoupgrade-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
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-config
  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
  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-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-ocm-agent-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: prometheus-k8s
- namespace: openshift-splunk-forwarder-operator
  name: osd-rebalance-infra-nodes-openshift-pod-rebalance
- 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-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-ownerrefs-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
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:
  - namespace: openshift-deployment-validation-operator
    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-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
- 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-kubequotaexceeded
  - 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
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-addon-operator
  name: addon-operator-og
- 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-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: 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-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: cluster-admins
- 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
```

7.9.3. OpenShift Dedicated core namespaces

OpenShift Dedicated core namespaces are installed by default during cluster installation.

Example 7.2. List of core namespaces

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: ocp-namespaces
  namespace: openshift-monitoring
data:
  managed_namespaces.yaml: |
    Resources:
```

```
Namespace:
- name: kube-system
- name: openshift-apiserver
- name: openshift-apiserver-operator
- name: openshift-authentication
- name: openshift-authentication-operator
- name: openshift-cloud-controller-manager
- name: openshift-cloud-controller-manager-operator
- name: openshift-cloud-credential-operator
- name: openshift-cloud-network-config-controller
- name: openshift-cluster-api
- name: openshift-cluster-csi-drivers
- name: openshift-cluster-machine-approver
- name: openshift-cluster-node-tuning-operator
- name: openshift-cluster-samples-operator
- name: openshift-cluster-storage-operator
- name: openshift-config
- name: openshift-config-managed
- name: openshift-config-operator
- name: openshift-console
- name: openshift-console-operator
- name: openshift-console-user-settings
- name: openshift-controller-manager
- name: openshift-controller-manager-operator
- name: openshift-dns
- name: openshift-dns-operator
- name: openshift-etcd
- name: openshift-etcd-operator
- name: openshift-host-network
- name: openshift-image-registry
- name: openshift-ingress
- name: openshift-ingress-canary
- name: openshift-ingress-operator
- name: openshift-insights
- name: openshift-kni-infra
- name: openshift-kube-apiserver
- name: openshift-kube-apiserver-operator
- name: openshift-kube-controller-manager
- name: openshift-kube-controller-manager-operator
- name: openshift-kube-scheduler
- name: openshift-kube-scheduler-operator
- name: openshift-kube-storage-version-migrator
- name: openshift-kube-storage-version-migrator-operator
- name: openshift-machine-api
- name: openshift-machine-config-operator
- name: openshift-marketplace
- name: openshift-monitoring
- name: openshift-multus
- name: openshift-network-diagnostics
- name: openshift-network-operator
- name: openshift-nutanix-infra
- name: openshift-oauth-apiserver
- name: openshift-openstack-infra
- name: openshift-operator-lifecycle-manager
- name: openshift-operators
- name: openshift-ovirt-infra
```

```
    - name: openshift-sdn
    - name: openshift-ovn-kubernetes
    - name: openshift-platform-operators
    - name: openshift-route-controller-manager
    - name: openshift-service-ca
    - name: openshift-service-ca-operator
    - name: openshift-user-workload-monitoring
    - name: openshift-vsphere-infra
```

7.9.4. OpenShift Dedicated add-on namespaces

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

Example 7.3. 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
  integreatly-operator: redhat-rhmi-operator
  nvidia-gpu-addon: redhat-nvidia-gpu-addon
  integreatly-operator-internal: redhat-rhmi-operator
  rhoams: redhat-rhoam-operator
  ocs-converged: openshift-storage
  addon-operator: redhat-addon-operator
  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
```

7.9.5. OpenShift Dedicated validating webhooks

OpenShift Dedicated 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 webhooks evaluate each request and either accept or reject them. 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 7.4. 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": "clusterroles-validation",
    "rules": [
      {
        "operations": [
          "DELETE"
        ],
        "apiGroups": [
          "rbac.authorization.k8s.io"
        ],
        "apiVersions": [
          "v1"
```

```
        ],
        "resources": [
          "clusterroles"
        ],
        "scope": "Cluster"
      }
    ],
    "documentString": "Managed OpenShift Customers may not delete protected ClusterRoles
including cluster-admin, view, edit, admin, specific system roles (system:admin, system:node,
system:node-proxier, system:kube-scheduler, system:kube-controller-manager), and backplane-*
roles"
  },
  {
    "webhookName": "customresourcedefinitions-validation",
    "rules": [
      {
        "operations": [
          "CREATE",
          "UPDATE",
          "DELETE"
        ],
        "apiGroups": [
          "apiextensions.k8s.io"
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "customresourcedefinitions"
        ],
        "scope": "Cluster"
      }
    ],
    "documentString": "Managed OpenShift Customers may not change
CustomResourceDefinitions managed by Red Hat."
  },
  {
    "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 that would conflict with system registries (e.g. quay.io, registry.redhat.io, registry.access.redhat.com, etc). For more details, see https://docs.openshift.com/"
  },
  {
    "webhookName": "ingress-config-validation",
    "rules": [
      {
        "operations": [
          "CREATE",
```

```
        "UPDATE",
        "DELETE"
      ],
      "apiGroups": [
        "config.openshift.io"
      ],
      "apiVersions": [
        "*"
      ],
      "resources": [
        "ingresses"
      ],
      "scope": "Cluster"
    }
  ],
  "documentString": "Managed OpenShift customers may not modify ingress config resources because it can can degrade cluster operators and can interfere with OpenShift SRE monitoring."
 },
 {
  "webhookName": "ingresscontroller-validation",
  "rules": [
   {
    "operations": [
     "CREATE",
     "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 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/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": "network-operator-validation",
    "rules": [
      {
        "operations": [
          "UPDATE"
        ],
        "apiGroups": [
          "operator.openshift.io"
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "network",
          "networks"
        ],
        "scope": "Cluster"
      }
    ],
    "documentString": "Managed OpenShift customers may not modify critical fields in the network.operator CRD (such as spec.migration.networkType) because it can disrupt Cluster Network Operator operations and CNI migrations. Even cluster-admin users are blocked from modifying these critical fields."
  },
  {
    "webhookName": "networkpolicies-validation",
    "rules": [
      {
        "operations": [
          "CREATE",
          "UPDATE",
          "DELETE"
        ],
        "apiGroups": [
          "networking.k8s.io"
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
```

```
          "networkpolicies"
        ],
        "scope": "Namespaced"
      }
    ],
    "documentString": "Managed OpenShift Customers may not create NetworkPolicies in
namespaces managed by Red Hat."
  },
  {
    "webhookName": "node-validation-osd",
    "rules": [
      {
        "operations": [
          "CREATE",
          "UPDATE",
          "DELETE"
        ],
        "apiGroups": [
          ""
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "nodes",
          "nodes/*"
        ],
        "scope": "*"
      }
    ],
    "documentString": "Managed OpenShift customers may not alter Node objects."
  },
  {
    "webhookName": "pod-validation",
    "rules": [
      {
        "operations": [
          "*"
        ],
        "apiGroups": [
          "v1"
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "pods"
        ],
        "scope": "Namespaced"
      }
    ],
    "documentString": "Managed OpenShift Customers may use tolerations on Pods that could
cause those Pods to be scheduled on infra or master nodes."
  },
  {
    "webhookName": "podimagespec-mutation",
```

```
    "rules": [
     {
      "operations": [
       "CREATE"
      ],
      "apiGroups": [
       ""
      ],
      "apiVersions": [
       "v1"
      ],
      "resources": [
       "pods"
      ],
      "scope": "Namespaced"
     }
    ],
    "documentString": "OpenShift debugging tools on Managed OpenShift clusters must be
available even if internal image registry is removed."
   },
   {
    "webhookName": "prometheusrule-validation",
    "rules": [
     {
      "operations": [
       "CREATE",
       "UPDATE",
       "DELETE"
      ],
      "apiGroups": [
       "monitoring.coreos.com"
      ],
      "apiVersions": [
       "*"
      ],
      "resources": [
       "prometheusrules"
      ],
      "scope": "Namespaced"
     }
    ],
    "documentString": "Managed OpenShift Customers may not create PrometheusRule in
namespaces managed by Red Hat."
   },
   {
    "webhookName": "regular-user-validation",
    "rules": [
     {
      "operations": [
       "*"
      ],
      "apiGroups": [
       "cloudcredential.openshift.io",
       "machine.openshift.io",
       "admissionregistration.k8s.io",
       "addons.managed.openshift.io",
```

```
      "cloudingress.managed.openshift.io",
      "managed.openshift.io",
      "ocmagent.managed.openshift.io",
      "splunkforwarder.managed.openshift.io",
      "upgrade.managed.openshift.io"
    ],
    "apiVersions": [
      "*"
    ],
    "resources": [
      "*/*"
    ],
    "scope": "*"
  },
  {
    "operations": [
      "*"
    ],
    "apiGroups": [
      "autoscaling.openshift.io"
    ],
    "apiVersions": [
      "*"
    ],
    "resources": [
      "clusterautoscalers",
      "machineautoscalers"
    ],
    "scope": "*"
  },
  {
    "operations": [
      "*"
    ],
    "apiGroups": [
      "config.openshift.io"
    ],
    "apiVersions": [
      "*"
    ],
    "resources": [
      "clusterversions",
      "clusterversions/status",
      "schedulers",
      "apiservers",
      "proxies"
    ],
    "scope": "*"
  },
  {
    "operations": [
      "CREATE",
      "UPDATE",
      "DELETE"
    ],
    "apiGroups": [
```

```
      ""
    ],
    "apiVersions": [
      "*"
    ],
    "resources": [
      "configmaps"
    ],
    "scope": "*"
  },
  {
    "operations": [
      "*"
    ],
    "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": [
          "network.openshift.io"
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "netnamespaces",
          "netnamespaces/*"
        ],
        "scope": "*"
      }
    ],
    "documentString": "Managed OpenShift customers may not manage any objects in the
following APIGroups [splunkforwarder.managed.openshift.io autoscaling.openshift.io
ocmagent.managed.openshift.io upgrade.managed.openshift.io config.openshift.io
machineconfiguration.openshift.io operator.openshift.io network.openshift.io
cloudcredential.openshift.io machine.openshift.io admissionregistration.k8s.io
addons.managed.openshift.io cloudingress.managed.openshift.io managed.openshift.io], nor may
Managed OpenShift customers alter the APIServer, KubeAPIServer, OpenShiftAPIServer,
ClusterVersion, Proxy or SubjectPermission objects."
  },
  {
    "webhookName": "scc-validation",
    "rules": [
      {
        "operations": [
          "UPDATE",
          "DELETE"
        ],
        "apiGroups": [
          "security.openshift.io"
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "securitycontextconstraints"
        ],
        "scope": "Cluster"
      }
    ],
    "documentString": "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]"
  },
  {
    "webhookName": "sdn-migration-validation",
    "rules": [
      {
```

```
        "operations": [
          "UPDATE"
        ],
        "apiGroups": [
          "config.openshift.io"
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "networks"
        ],
        "scope": "Cluster"
      }
    ],
    "documentString": "Managed OpenShift customers may not modify the network config type
because it can can degrade cluster operators and can interfere with OpenShift SRE monitoring."
  },
  {
    "webhookName": "service-mutation",
    "rules": [
      {
        "operations": [
          "CREATE",
          "UPDATE"
        ],
        "apiGroups": [
          ""
        ],
        "apiVersions": [
          "v1"
        ],
        "resources": [
          "services"
        ],
        "scope": "Namespaced"
      }
    ],
    "documentString": "LoadBalancer-type services on Managed OpenShift clusters must contain
an additional annotation for managed policy compliance."
  },
  {
    "webhookName": "serviceaccount-validation",
    "rules": [
      {
        "operations": [
          "DELETE"
        ],
        "apiGroups": [
          ""
        ],
        "apiVersions": [
          "v1"
        ],
        "resources": [
          "serviceaccounts"
```

```
        ],
        "scope": "Namespaced"
      }
    ],
    "documentString": "Managed OpenShift Customers may not delete the service accounts under
the managed namespaces。"
  },
  {
    "webhookName": "techpreviewnoupgrade-validation",
    "rules": [
      {
        "operations": [
          "CREATE",
          "UPDATE"
        ],
        "apiGroups": [
          "config.openshift.io"
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "featuregates"
        ],
        "scope": "Cluster"
      }
    ],
    "documentString": "Managed OpenShift Customers may not use TechPreviewNoUpgrade
FeatureGate that could prevent any future ability to do a y-stream upgrade to their clusters."
  }
]
```