OpenShift Dedicated 4

Support

Support for OpenShift Dedicated 4

Last Updated: 2021-10-11

OpenShift Dedicated 4 Support

Support for OpenShift Dedicated 4

Legal Notice

Abstract

This document details on how to get support for OpenShfit Dedicated

Table of Contents

CHAPTER 1. GETTING SUPPORT

1.1. 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 Insights in OpenShift Cluster Manager (OCM). Insights provides details about issues and, if available, information on how to solve a problem.

If you have a suggestion for improving this documentation or have found an error, please submit a Bugzilla report against the **OpenShift Container Platform** product for the **Documentation** component. Please provide specific details, such as the section name and OpenShift Dedicated version.

1.2. 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.

1.3. 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. In the main Red Hat Customer Portal 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
3. Click **Search**.
4. Select the **OpenShift Dedicated** product filter.

5. Select the **Knowledgebase** content type filter.

1.4. SUBMITTING A SUPPORT CASE

Prerequisites

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

Procedure

1. Log in to the Red Hat Customer Portal and select **SUPPORT CASES → Open a case**.
2. Select the appropriate category for your issue (such as **Defect / Bug**), product (**OpenShift Dedicated**), and product version (**4**, if this is not already autofilled).
3. 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**.
4. Enter a concise but descriptive problem summary and further details about the symptoms being experienced, as well as your expectations.
5. 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**.
6. Ensure that the account information presented is as expected, and if not, amend accordingly.
7. 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 the OpenShift Dedicated web console:
     i. Navigate to **Home → Dashboards → Overview**.
     ii. 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.
     i. From the toolbar, navigate to **(?) Help → Open Support Case**.
     ii. 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"}'
     ```
8. Complete the following questions where prompted and then click **Continue**:

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

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

1.5. ADDITIONAL RESOURCES

- For details about identifying issues with your cluster, see Using Insights to identify issues with your cluster.

CHAPTER 2. REMOTE HEALTH MONITORING WITH CONNECTED CLUSTERS

2.1. 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 cloud.redhat.com/openshift.

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 Bugzilla.
- **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 cloud.redhat.com/openshift 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.

2.1.1. 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.

2.1.1.1. Information collected by Telemetry

The following information is collected by Telemetry:

- The unique random identifier that is generated during an installation
- 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 name of the provider platform that OpenShift Dedicated is deployed on and the data center location
- 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
- The OpenShift Dedicated framework components installed in a cluster and their condition and status
- Usage information about components, features, and extensions
- Usage details about Technology Previews and unsupported configurations
- Information about degraded software
- Information about nodes that are marked as **NotReady**
- Events for all namespaces listed as "related objects" for a degraded Operator
- Configuration details that help Red Hat Support to provide beneficial support for customers. This includes node configuration at the cloud infrastructure level, host names, IP addresses, Kubernetes pod names, namespaces, and services.
- Information about the validity of certificates

Telemetry does not collect identifying information such as user names, 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.

2.1.2. 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 OpenShift Cluster Manager (OCM). If any issues have been identified, Insights provides further details and, if available, steps on how to solve a problem.

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

Red Hat uses all connected cluster information to:

- Proactively identify potential cluster issues and provide a solution and preventive actions in OpenShift Cluster Manager (OCM)
- Improve OpenShift Dedicated by providing aggregated and critical information to product and support teams
- Make OpenShift Dedicated more intuitive

Additional resources

- The Insights Operator is installed and enabled by default. If you need to opt out of remote health reporting, see Opting out of remote health reporting .

2.1.2.1. 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, such as Amazon Web Services, 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

- 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.

2.1.3. Understanding Telemetry and Insights Operator data flow

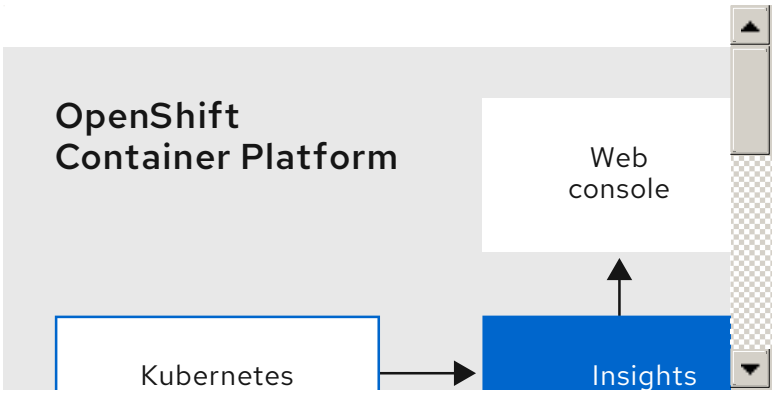
The Telemeter Client collects selected time series data from the Prometheus API. The time series data is uploaded to api.openshift.com every four minutes and thirty seconds for processing.

The Insights Operator gathers selected data from the Kubernetes API and the Prometheus API into an archive. The archive is uploaded to cloud.redhat.com every two hours for processing. The Insights Operator also downloads the latest Insights analysis from cloud.redhat.com. This is used to populate the **Insights 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



2.1.4. Additional details about how remote health monitoring data is used

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

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

Collection safeguards

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

Sharing

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

Third party service providers

Red Hat may engage certain service providers to assist in the collection and storage of the telemetry and configuration data.

User control / enabling and disabling telemetry and configuration data collection

You may disable OpenShift Dedicated Telemetry and the Insights Operator by following the instructions in Opting out of remote health reporting .

2.2. SHOWING DATA COLLECTED BY REMOTE HEALTH MONITORING

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

2.2.1. Showing data collected by Telemetry

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

Prerequisites

- Install the OpenShift CLI (**oc**).
- You must log in to the cluster with a user that has either the **cluster-admin** role or the **cluster-monitoring-view** role.

Procedure

1. Find the URL for the Prometheus service that runs in the OpenShift Dedicated cluster:

```
$ oc get route prometheus-k8s -n openshift-monitoring -o jsonpath="{.spec.host}"
```

2. Navigate to the URL.
3. Enter this query in the **Expression** input box and press **Execute**:

```
{__name__=~"cluster:usage:.*|count:up0|count:up1|cluster_version|cluster_version_available_u
pdates|cluster_operator_up|cluster_operator_conditions|cluster_version_payload|cluster_install
er|cluster_infrastructure_provider|cluster_feature_set|instance:etcd_object_counts:sum|ALERT
S|code:apiserver_request_total:rate:sum|cluster:capacity_cpu_cores:sum|cluster:capacity_mem
ory_bytes:sum|cluster:cpu_usage_cores:sum|cluster:memory_usage_bytes:sum|openshift:cpu_
usage_cores:sum|openshift:memory_usage_bytes:sum|workload:cpu_usage_cores:sum|worklo
ad:memory_usage_bytes:sum|cluster:virt_platform_nodes:sum|cluster:node_instance_type_cou
nt:sum|cnv:vmi_status_running:count|node_role_os_version_machine:cpu_capacity_cores:sum
node_role_os_version_machine:cpu_capacity_sockets:sum|subscription_sync_total|csv_succee
ded|csv_abnormal|ceph_cluster_total_bytes|ceph_cluster_total_used_raw_bytes|ceph_health_s
tatus|job:ceph_osd_metadata:count|job:kube_pv:count|job:ceph_pools_iops:total|job:ceph_pool
s_iops_bytes:total|job:ceph_versions_running:count|job:noobaa_total_unhealthy_buckets:sum|j
b:noobaa_bucket_count:sum|job:noobaa_total_object_count:sum|noobaa_accounts_num|noob
aa_total_usage|console_url|cluster:network_attachment_definition_instances:max|cluster:netwo
rk_attachment_definition_enabled_instance_up:max|insightsclient_request_send_total|cam_app
_workload_migrations|cluster:apiserver_current_inflight_requests:sum:max_over_time:2m|clust
er:telemetry_selected_series:count",alertstate=~"firing|"}
```

This query replicates the request that Telemetry makes against a running OpenShift Dedicated cluster's Prometheus service and returns the full set of time series captured by Telemetry.

2.2.2. Showing data collected by the Insights Operator

You can review the data that is collected by the Insights Operator.

Prerequisites

- Access to the cluster as a user with the **cluster-admin** role.

Procedure

1. Find the name of the currently running pod for the Insights Operator:

```
$ INSIGHTS_OPERATOR_POD=$(oc get pods --namespace=openshift-insights -o custom-columns=:metadata.name --no-headers  --field-selector=status.phase=Running)
```

2. Copy the recent data archives collected by the Insights Operator:

```
$ oc cp openshift-insights/$INSIGHTS_OPERATOR_POD:/var/lib/insights-operator ./insights-data
```

The recent Insights Operator archives are now available in the **insights-data** directory.

2.3. OPTING OUT OF REMOTE HEALTH REPORTING

On OpenShift Dedicated, remote health reporting is always enabled. You cannot opt out.

2.4. USING INSIGHTS TO IDENTIFY ISSUES WITH YOUR CLUSTER

Insights repeatedly analyzes the data Insights Operator sends. Users of OpenShift Dedicated can display the report on the **Insights** tab of each cluster in OpenShift Cluster Manager (OCM).

2.4.1. Displaying potential issues with your cluster

This section describes how to display the Insights report in the OpenShift Cluster Manager (OCM).

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

Prerequisites

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

Procedure

1. Click the **Clusters** menu in the left pane.
2. Click the cluster's name to display the details of the cluster.
3. Open the **Insights** tab of the cluster.
   Depending on the result, the tab displays one of the following:
   - **Your cluster passed all health checks** if Insights did not identify any issues.

- A list of issues Insights has detected, prioritized by risk (low, moderate, important, and critical).
- **No health checks to display** if Insights has not yet analyzed the cluster. The analysis starts shortly after the cluster has been installed and connected to the internet.

4. If any issues are displayed on the tab, click the > icon in front of the entry for further details. Depending on the issue, the details can also contain a link to an Red Hat Knowledge Base article. For details and information on how to solve the problem, click **How to remediate this issue**.

2.4.2. Displaying the Insights status in the web console

Insights repeatedly analyzes your cluster and you can display the status of identified potential issues of your cluster in the OpenShift Dedicated web console. This status shows the number of issues in the different categories and, for further details, links to the reports in the OpenShift Cluster Manager (OCM).

Prerequisites

- Your cluster is registered in the OpenShift Cluster Manager (OCM).
- 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 **Insights** on the **Status** card.
   The pop-up window lists potential issues grouped by priority. Click the individual categories or **View all in OpenShift Dedicated** to display further details.

CHAPTER 3. 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

3.1. 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, such as:

- Resource definitions
- Audit logs
- Service logs

You can specify one or more images when you run the command by including the **--image** argument. When you specify an image, the tool collects data related to that feature or product.

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

3.2. 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.

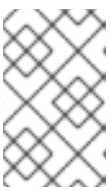
Prerequisites

- Access to the cluster as a user with the **cluster-admin** role.
- The OpenShift CLI (**oc**) installed.

Procedure

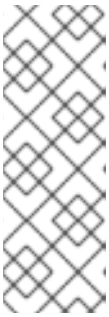
1. Navigate to the directory where you want to store the **must-gather** data.
2. Run the **oc adm must-gather** command:

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



NOTE

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. Contact Red Hat Support for the recommended resources to gather.



NOTE

If your cluster is using a restricted network, you must take additional steps. If your mirror registry has a trusted CA, you must first add the trusted CA to the cluster. For all clusters on restricted networks, you must import the default **must-gather** image as an image stream before you use the **oc adm must-gather** command.

```
$ oc import-image is/must-gather -n openshift
```

3. Create a compressed file from the **must-gather** directory that was just created in your working directory. For example, on a computer that uses a Linux operating system, run the following command:

```
$ tar cvaf must-gather.tar.gz must-gather.local.5421342344627712289/ 1
```

1 Make sure to replace **must-gather-local.5421342344627712289/** with the actual directory name.

4. Attach the compressed file to your support case on the Red Hat Customer Portal .

3.3. 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 3.1. Supported must-gather images

Image	Purpose
registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel8:v2.6.0	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-rhel7	Data collection for Red Hat OpenShift Service Mesh.
registry.redhat.io/rhcam-1-2/openshift-migration-must-gather-rhel8	Data collection for migration-related information.
registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.7	Data collection for Red Hat OpenShift Container Storage.
registry.redhat.io/openshift4/ose-cluster-logging-operator	Data collection for OpenShift Logging.

NOTE

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

Prerequisites

- Access to the cluster as a user with the **cluster-admin** role.
- The OpenShift CLI (**oc**) 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. 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-rhel8:v2.6.0 2
```

1. The default OpenShift Container Platform **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 3.1. 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. Create a compressed file from the **must-gather** directory that was just created in your working directory. For example, on a computer that uses a Linux operating system, run the following command:

```
$ tar cvaf must-gather.tar.gz must-gather.local.5421342344627712289/ 1
```

1 Make sure to replace **must-gather-local.5421342344627712289/** with the actual directory name.

4. Attach the compressed file to your support case on the Red Hat Customer Portal .

3.4. 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 Container Platform web console. You can also manually obtain your cluster ID by using the web console or the OpenShift CLI (**oc**).

Prerequisites

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

Procedure

- To open a support case and have your cluster ID autofilled using the web console:
  a. From the toolbar, navigate to **(?) Help → Open Support Case**.
  b. The **Cluster ID** value is autofilled.
- To manually obtain your cluster ID using the web console:
  a. Navigate to **Home → Dashboards → 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"}'
```

3.5. ABOUT SOSREPORT

sosreport is a tool that collects configuration details, system information, and diagnostic data from Red Hat Enterprise Linux (RHEL) and Red Hat Enterprise Linux CoreOS (RHCOS) systems. **sosreport** provides a standardized way to collect diagnostic information relating to a node, which can then be provided to Red Hat Support for issue diagnosis.

In some support interactions, Red Hat Support may ask you to collect a **sosreport** archive for a specific OpenShift Container Platform node. For example, it might sometimes be necessary to review system logs or other node-specific data that is not included within the output of **oc adm must-gather**.

3.6. GENERATING A SOSREPORT ARCHIVE FOR AN OPENSHIFT CONTAINER PLATFORM CLUSTER NODE

The recommended way to generate a **sosreport** for an OpenShift Container Platform 4 cluster node is through a debug pod.

Prerequisites

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

NOTE

OpenShift Container Platform 4 cluster nodes running Red Hat Enterprise Linux CoreOS (RHCOS) are immutable and rely on Operators to apply cluster changes. Accessing cluster nodes using SSH is not recommended and nodes will be tainted as *accessed*. However, if the OpenShift Container Platform 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>** instead.

4. Start a **toolbox** container, which includes the required binaries and plug-ins to run **sosreport**:

```
# toolbox
```

NOTE

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

5. Collect a **sosreport** archive.

   a. Run the **sosreport** command and enable the **crio.all** and **crio.logs** CRI-O container engine **sosreport** plug-ins:

```
# sosreport -k crio.all=on -k crio.logs=on (1)
```

   (1) **-k** enables you to define **sosreport** plug-in parameters outside of the defaults.

   b. Press **Enter** when prompted, to continue.

   c. Provide the Red Hat Support case ID. **sosreport** adds the ID to the archive's file name.

   d. The **sosreport** output provides the archive's location and checksum. The following sample output references support case ID **01234567**:

```
Your sosreport has been generated and saved in:
  /host/var/tmp/sosreport-my-cluster-node-01234567-2020-05-28-eyjknxt.tar.xz (1)

The checksum is: 382ffc167510fd71b4f12a4f40b97a4e
```

   (1) The **sosreport** archive's file path is outside of the **chroot** environment because the toolbox container mounts the host's root directory at **/host**.

6. Provide the **sosreport** archive to Red Hat Support for analysis, using one of the following methods.

   - Upload the file to an existing Red Hat support case directly from an OpenShift Container Platform cluster.

     a. From within the toolbox container, run **redhat-support-tool** to attach the archive directly to an existing Red Hat support case. This example uses support case ID

01234567:

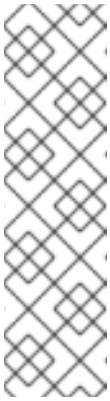
```
# redhat-support-tool addattachment -c 01234567 /host/var/tmp/my-sosreport.tar.xz
1
```

1 The toolbox container mounts the host's root directory at **/host**. Reference the absolute path from the toolbox container's root directory, including **/host/**, when specifying files to upload through the **redhat-support-tool** command.

- 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/sosreport-my-cluster-node-01234567-2020-05-28-eyjknxt.tar.xz' > /tmp/sosreport-my-cluster-node-01234567-2020-05-28-eyjknxt.tar.xz 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.



NOTE

OpenShift Container Platform 4 cluster nodes running Red Hat Enterprise Linux CoreOS (RHCOS) are immutable and rely on Operators to apply cluster changes. Transferring a **sosreport** archive from a cluster node by using **scp** is not recommended and nodes will be tainted as *accessed*. However, if the OpenShift Container Platform 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 copy a **sosreport** archive from a node by running **scp core@<node>.<cluster_name>.<base_domain>:<file_path> <local_path>**.

  b. Navigate to an existing support case within https://access.redhat.com/support/cases/.
  c. Select **Attach files** and follow the prompts to upload the file.

3.7. QUERYING BOOTSTRAP NODE JOURNAL LOGS

If you experience bootstrap-related issues, you can gather **bootkube.service journald** unit logs and container logs from the bootstrap node.

Prerequisites

- You have SSH access to your bootstrap node.
- You have the fully qualified domain name of the bootstrap node.

Procedure

1. Query **bootkube.service journald** unit logs from a bootstrap node during OpenShift Container Platform installation. Replace **<bootstrap_fqdn>** with the bootstrap node's fully qualified domain name:

```
$ ssh core<bootstrap_fqdn> journalctl -b -f -u bootkube.service
```

NOTE

The **bootkube.service** log on the bootstrap node outputs etcd **connection refused** errors, indicating that the bootstrap server is unable to connect to etcd on master nodes. After etcd has started on each master node and the nodes have joined the cluster, the errors should stop.

2. Collect logs from the bootstrap node containers using **podman** on the bootstrap node. Replace **<bootstrap_fqdn>** with the bootstrap node's fully qualified domain name:

```
$ ssh core@<bootstrap_fqdn> 'for pod in $(sudo podman ps -a -q); do sudo podman logs $pod; done'
```

3.8. 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.
- Your API service is still functional.
- You have installed the OpenShift CLI (**oc**).
- You have SSH access to your hosts.

Procedure

1. Query **kubelet journald** unit logs from OpenShift Container Platform cluster nodes. The following example queries master nodes only:

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

1 Replace **kubelet** 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 master nodes:

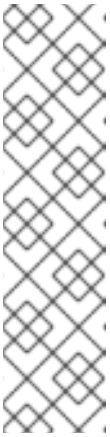
```
$ oc adm node-logs --role=master --path=openshift-apiserver
```

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

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

c. If the API is not functional, review the logs on each node using SSH instead. The following example tails **/var/log/openshift-apiserver/audit.log**:

```
$ ssh core@<master-node>.<cluster_name>.<base_domain> sudo tail -f
/var/log/openshift-apiserver/audit.log
```



NOTE

OpenShift Container Platform 4 cluster nodes running Red Hat Enterprise Linux CoreOS (RHCOS) are immutable and rely on Operators to apply cluster changes. Accessing cluster nodes using SSH is not recommended and nodes will be tainted as *accessed*. 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 Container Platform 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>**.

3.9. COLLECTING A NETWORK TRACE FROM AN OPENSHIFT CONTAINER PLATFORM NODE OR CONTAINER

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

Prerequisites

- You have access to the cluster as a user with the **cluster-admin** role.
- You have installed the OpenShift CLI (**oc**).
- You have a Red Hat standard or premium Subscription.
- You have a Red Hat Customer Portal account.
- You have an existing Red Hat Support case ID.
- You have SSH access to your hosts.

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
```

NOTE

OpenShift Container Platform 4 cluster nodes running Red Hat Enterprise Linux CoreOS (RHCOS) are immutable and rely on Operators to apply cluster changes. Accessing cluster nodes using SSH is not recommended and nodes will be tainted as *accessed*. However, if the OpenShift Container Platform 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>** instead.

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 plug-ins 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.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 directly from an OpenShift Container Platform cluster.

a. From within the toolbox container, run **redhat-support-tool** to attach the file directly to an existing Red Hat Support case. This example uses support case ID **01234567**:

```
# redhat-support-tool addattachment -c 01234567 /host/var/tmp/my-tcpdump-capture-file.pcap 1
```

1 The toolbox container mounts the host's root directory at **/host**. Reference the absolute path from the toolbox container's root directory, including **/host/**, when specifying files to upload through the **redhat-support-tool** command.

- 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.

NOTE

OpenShift Container Platform 4 cluster nodes running Red Hat Enterprise Linux CoreOS (RHCOS) are immutable and rely on Operators to apply cluster changes. Transferring a **tcpdump** capture file from a cluster node by using **scp** is not recommended and nodes will be tainted as *accessed*. However, if the OpenShift Container Platform 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 copy a **tcpdump** capture file from a node by running **scp core@<node>.<cluster_name>.<base_domain>:<file_path> <local_path>**.

b. Navigate to an existing support case within https://access.redhat.com/support/cases/.

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

3.10. PROVIDING DIAGNOSTIC DATA TO RED HAT SUPPORT

When investigating OpenShift Container Platform 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, or from an OpenShift Container Platform cluster directly by using the **redhat-support-tool** command.

Prerequisites

- You have access to the cluster as a user with the **cluster-admin** role.
- You have SSH access to your hosts.
- You have installed the OpenShift CLI (**oc**).
- You have a Red Hat standard or premium Subscription.
- You have a Red Hat Customer Portal account.
- You have an existing Red Hat Support case ID.

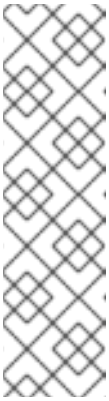
Procedure

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

1. Concatenate a diagnostic file contained on an OpenShift Container Platform 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.



NOTE

OpenShift Container Platform 4 cluster nodes running Red Hat Enterprise Linux CoreOS (RHCOS) are immutable and rely on Operators to apply cluster changes. Transferring files from a cluster node by using **scp** is not recommended and nodes will be tainted as *accessed*. However, if the OpenShift Container Platform 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 copy diagnostic files from a node by running **scp core@<node>.<cluster_name>.<base_domain>:<file_path> <local_path>**.

2. Navigate to an existing support case within https://access.redhat.com/support/cases/.
3. Select **Attach files** and follow the prompts to upload the file.

- Upload diagnostic data to an existing Red Hat support case directly from an OpenShift Container Platform cluster.

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
```

NOTE

OpenShift Container Platform 4 cluster nodes running Red Hat Enterprise Linux CoreOS (RHCOS) are immutable and rely on Operators to apply cluster changes. Accessing cluster nodes using SSH is not recommended and nodes will be tainted as *accessed*. However, if the OpenShift Container Platform 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>** instead.

4. Start a **toolbox** container, which includes the required binaries to run **redhat-support-tool**:

```
# toolbox
```

NOTE

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

a. Run **redhat-support-tool** to attach a file from the debug pod directly to an existing Red Hat Support case. This example uses support case ID '01234567' and example file path **/host/var/tmp/my-diagnostic-data.tar.gz**:

```
# redhat-support-tool addattachment -c 01234567 /host/var/tmp/my-diagnostic-data.tar.gz (1)
```

1 The toolbox container mounts the host's root directory at **/host**. Reference the absolute path from the toolbox container's root directory, including **/host/**, when

3.11. 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 plug-ins that are needed to run commands such as **sosreport** and **redhat-support-tool**.

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.

Installing packages to a toolbox container

By default, running the **toolbox** command starts a container with the **registry.redhat.io/rhel8/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.

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>
```

Starting an alternative image with toolbox

By default, running the **toolbox** command starts a container with the **registry.redhat.io/rhel8/support-tools:latest** image. You can start an alternative image by creating a **.toolboxrc** file and specifying the image to run.

Prerequisites

- You have accessed a node with the **oc debug node/<node_name>** command.

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. Create a **.toolboxrc** file in the home directory for the root user ID:

```
# vi ~/.toolboxrc
```

```
REGISTRY=quay.io                <.>
IMAGE=fedora/fedora:33-x86_64   <.>
TOOLBOX_NAME=toolbox-fedora-33  <.>
```

<.> Optional: Specify an alternative container registry. <.> Specify an alternative image to start. <.> Optional: Specify an alternative name for the toolbox container.

3. Start a toolbox container with the alternative image:

```
# toolbox
```

NOTE

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

CHAPTER 4. SUMMARIZING CLUSTER SPECIFICATIONS

4.1. SUMMARIZING CLUSTER SPECIFICATIONS THROUGH CLUSTERVERSION

You can obtain a summary of OpenShift Container Platform cluster specifications by querying the **clusterversion** resource.

Prerequisites

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

Procedure

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

```
$ oc get clusterversion
```

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

```
$ oc describe clusterversion
```

CHAPTER 5. OPENSHIFT CONTAINER PLATFORM MANAGED RESOURCES

5.1. OVERVIEW

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

5.2. HIVE MANAGED RESOURCES

The following list displays the OpenShift Container Platform 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 Container Platform clusters. Changes to OpenShift Container Platform resources should be made through OCM so that OCM and Hive are synchronized. Contact ocm-feedback@redhat.com if OCM does not support modifying the resources in question.

Example 5.1. List of Hive managed resources

```
Resources:
  ConfigMap:
  - 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: osd-rebalance-infra-nodes
  - namespace: openshift-monitoring
    name: sre-dns-latency-exporter-code
  - namespace: openshift-monitoring
    name: sre-ebs-iops-reporter-code
  - namespace: openshift-monitoring
    name: sre-stuck-ebs-vols-code
  - namespace: openshift-security
    name: osd-audit-policy
  - namespace: openshift-validation-webhook
    name: webhook-cert
  Endpoints:
  - namespace: openshift-monitoring
    name: sre-dns-latency-exporter
  - namespace: openshift-monitoring
    name: sre-ebs-iops-reporter
  - namespace: openshift-monitoring
    name: sre-stuck-ebs-vols
  - namespace: openshift-monitoring
    name: token-refresher
  - namespace: openshift-validation-webhook
    name: validation-webhook
  Namespace:
  - name: dedicated-admin
```

```
    - name: openshift-aqua
    - name: openshift-backplane
    - name: openshift-backplane-cee
    - name: openshift-backplane-managed-scripts
    - name: openshift-backplane-srep
    - name: openshift-build-test
    - name: openshift-cloud-ingress-operator
    - name: openshift-codeready-workspaces
    - name: openshift-compliance
    - name: openshift-container-security-operator
    - name: openshift-custom-domains-operator
    - name: openshift-customer-monitoring
    - name: openshift-logging
    - name: openshift-managed-upgrade-operator
    - name: openshift-must-gather-operator
    - name: openshift-operators-redhat
    - name: openshift-osd-metrics
    - name: openshift-rbac-permissions
    - name: openshift-route-monitor-operator
    - name: openshift-security
    - name: openshift-splunk-forwarder-operator
    - name: openshift-sre-pruning
    - name: openshift-sre-sshd
    - name: openshift-strimzi
    - name: openshift-validation-webhook
    - name: openshift-velero
    - name: openshift-monitoring
  ReplicationController:
  - namespace: openshift-monitoring
    name: sre-ebs-iops-reporter-1
  - namespace: openshift-monitoring
    name: sre-stuck-ebs-vols-1
  ServiceAccount:
  - namespace: openshift-backplane-managed-scripts
    name: osd-backplane
  - namespace: openshift-backplane-srep
    name: osd-delete-ownerrefs-serviceaccounts
  - namespace: openshift-backplane
    name: osd-delete-backplane-serviceaccounts
  - namespace: openshift-build-test
    name: sre-build-test
  - namespace: openshift-logging
    name: osd-delete-ownerrefs-bz1906584
  - namespace: openshift-marketplace
    name: osd-patch-subscription-source
  - 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-sre-pruning
```

```
  name: sre-pruner-sa
- namespace: openshift-validation-webhook
  name: validation-webhook
- namespace: openshift-velero
  name: velero
- namespace: openshift-backplane-srep
  name: UNIQUE_BACKPLANE_SERVICEACCOUNT_ID
Service:
- namespace: openshift-monitoring
  name: sre-dns-latency-exporter
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter
- namespace: openshift-monitoring
  name: sre-stuck-ebs-vols
- namespace: openshift-monitoring
  name: token-refresher
- namespace: openshift-validation-webhook
  name: validation-webhook
ValidatingWebhookConfiguration:
- name: sre-hiveownership-validation
- name: sre-namespace-validation
- name: sre-pod-validation
- name: sre-regular-user-validation
DaemonSet:
- namespace: openshift-monitoring
  name: sre-dns-latency-exporter
- namespace: openshift-security
  name: audit-exporter
- namespace: openshift-validation-webhook
  name: validation-webhook
Deployment:
- namespace: openshift-monitoring
  name: token-refresher
DeploymentConfig:
- namespace: openshift-monitoring
  name: sre-ebs-iops-reporter
- namespace: openshift-monitoring
  name: sre-stuck-ebs-vols
ClusterRoleBinding:
- name: aqua-scanner-binding
- name: backplane-cee-cluster-rolebinding
- name: backplane-cee-readers
- name: backplane-cluster-admin
- name: backplane-impersonate-cluster-admin
- name: cloud-ingress-operator
- name: configure-alertmanager-operator-prom
- name: dedicated-admins-cluster
- name: dedicated-admins-registry-cas-cluster
- name: openshift-backplane-managed-scripts-reader
- name: osd-cluster-ready
- name: osd-delete-backplane-script-resources
- name: osd-delete-ownerrefs-bz1906584
- 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-build-test
- name: sre-pruner-buildsdeploys-pruning
- name: velero
- name: webhook-validation
ClusterRole:
- name: backplane-cee-cluster-readers
- name: backplane-cee-readers-cluster
- name: backplane-impersonate-cluster-admin
- name: backplane-srep-admins-cluster
- name: backplane-srep-admins-project
- name: backplane-srep-readers-cluster
- name: cloud-ingress-operator
- 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: 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-bz1906584
- 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: pcap-dedicated-admins
- name: splunk-forwarder-operator
- name: sre-allow-read-machine-info
- name: sre-build-test
- name: sre-pruner-buildsdeploys-cr
- name: webhook-validation-cr
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-build-test
  name: sre-build-test
- namespace: openshift-logging
  name: osd-delete-ownerrefs-bz1906584
- namespace: openshift-marketplace
  name: osd-patch-subscription-source
- namespace: openshift-monitoring
  name: osd-rebalance-infra-nodes
```

```
- namespace: openshift-sre-pruning
  name: builds-pruner
- namespace: openshift-sre-pruning
  name: deployments-pruner
- namespace: openshift-sre-pruning
  name: services-pruner
- namespace: openshift-sre-pruning
  name: sre-idp-pruner
Job:
- namespace: openshift-monitoring
  name: osd-cluster-ready
APIScheme:
- namespace: openshift-cloud-ingress-operator
  name: rh-api
PublishingStrategy:
- namespace: openshift-cloud-ingress-operator
  name: publishingstrategy
ScanSettingBinding:
- namespace: openshift-compliance
  name: fedramp-moderate
ScanSetting:
- namespace: openshift-compliance
  name: osd
MachineHealthCheck:
- namespace: openshift-machine-api
  name: srep-infra-healthcheck
- namespace: openshift-machine-api
  name: srep-worker-healthcheck
MachineSet:
- namespace: openshift-machine-api
  name: blrm-9-17-5qbgh-infra-us-east-1a
- namespace: openshift-machine-api
  name: blrm-9-17-5qbgh-worker-us-east-1a
KubeletConfig:
- name: custom-kubelet
SubjectPermission:
- namespace: openshift-rbac-permissions
  name: backplane-srep
- 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-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
ClusterUrlMonitor:
- namespace: openshift-route-monitor-operator
  name: api
```

```
RouteMonitor:
- namespace: openshift-route-monitor-operator
  name: console
NetworkPolicy:
- namespace: openshift-monitoring
  name: token-refresher
CatalogSource:
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator-registry
- namespace: openshift-compliance
  name: compliance-operator-registry
- namespace: openshift-custom-domains-operator
  name: custom-domains-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-osd-metrics
  name: osd-metrics-exporter-registry
- namespace: openshift-rbac-permissions
  name: rbac-permissions-operator-registry
- namespace: openshift-route-monitor-operator
  name: route-monitor-operator-registry
- namespace: openshift-splunk-forwarder-operator
  name: splunk-forwarder-operator-catalog
- namespace: openshift-velero
  name: managed-velero-operator-registry
OperatorGroup:
- namespace: openshift-aqua
  name: openshift-aqua
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator
- namespace: openshift-codeready-workspaces
  name: openshift-codeready-workspaces
- namespace: openshift-compliance
  name: compliance-operator
- namespace: openshift-container-security-operator
  name: container-security-operator
- namespace: openshift-custom-domains-operator
  name: custom-domains-operator
- namespace: openshift-customer-monitoring
  name: openshift-customer-monitoring
- 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-osd-metrics
  name: osd-metrics-exporter
- namespace: openshift-rbac-permissions
  name: rbac-permissions-operator
- namespace: openshift-route-monitor-operator
  name: route-monitor-operator
```

```
- namespace: openshift-splunk-forwarder-operator
  name: splunk-forwarder-operator-og
- namespace: openshift-strimzi
  name: openshift-strimzi
- namespace: openshift-velero
  name: managed-velero-operator
Subscription:
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator
- namespace: openshift-compliance
  name: compliance-operator-sub
- namespace: openshift-container-security-operator
  name: container-security-operator
- namespace: openshift-custom-domains-operator
  name: custom-domains-operator
- 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-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-osd-metrics
  name: osd-metrics-exporter
- namespace: openshift-compliance
  name: compliance-operator
- namespace: openshift-must-gather-operator
  name: must-gather-operator
- namespace: openshift-rbac-permissions
  name: rbac-permissions-operator
- namespace: openshift-route-monitor-operator
  name: route-monitor-operator
- namespace: openshift-managed-upgrade-operator
  name: managed-upgrade-operator
- namespace: openshift-velero
  name: managed-velero-operator
- namespace: openshift-custom-domains-operator
  name: custom-domains-operator
- namespace: openshift-cloud-ingress-operator
  name: cloud-ingress-operator
- namespace: openshift-monitoring
  name: configure-alertmanager-operator
Status:
- {}
```

```
Project:
- name: dedicated-admin
- name: openshift-aqua
- name: openshift-backplane
- name: openshift-backplane-cee
- name: openshift-backplane-managed-scripts
- name: openshift-backplane-srep
- name: openshift-build-test
- name: openshift-cloud-ingress-operator
- name: openshift-codeready-workspaces
- name: openshift-compliance
- name: openshift-container-security-operator
- name: openshift-custom-domains-operator
- name: openshift-customer-monitoring
- name: openshift-logging
- name: openshift-managed-upgrade-operator
- name: openshift-must-gather-operator
- name: openshift-operators-redhat
- name: openshift-osd-metrics
- name: openshift-rbac-permissions
- name: openshift-route-monitor-operator
- name: openshift-security
- name: openshift-splunk-forwarder-operator
- name: openshift-sre-pruning
- name: openshift-sre-sshd
- name: openshift-strimzi
- name: openshift-validation-webhook
- name: openshift-velero
ClusterResourceQuota:
- name: loadbalancer-quota
- name: persistent-volume-quota
SecurityContextConstraints:
- name: pcap-dedicated-admins
- name: splunkforwarder
SplunkForwarder:
- namespace: openshift-security
  name: splunkforwarder
Group:
- name: dedicated-admins
User:
- name: backplane-cluster-admin
```

5.3. OPENSHIFT CONTAINER PLATFORM ADD-ON NAMESPACES

OpenShift Container Platform add-ons are services available for installation after cluster installation. These additional services include AWS CloudWatch, Red Hat CodeReady Workspaces, 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 5.2. List of add-on managed namespaces

```
addon-namespaces:
  ocs-converged-dev: openshift-storage
```

```
managed-api-service-internal: redhat-rhoami-operator
codeready-workspaces-operator: codeready-workspaces-operator
managed-odh: redhat-ods-operator
codeready-workspaces-operator-qe: codeready-workspaces-operator-qe
integreatly-operator: redhat-rhmi-operator
gpu-operator: redhat-gpu-operator
integreatly-operator-internal: redhat-rhmi-operator
rhosak-qe: redhat-managed-kafka-operator-qe
rhoams: redhat-rhoam-operator
ocs-converged: openshift-storage
addon-operator: redhat-addon-operator
rhosak: redhat-managed-kafka-operator
kas-fleetshard-operator-qe: redhat-kas-fleetshard-operator-qe
prow-operator: prow
cluster-logging-operator: openshift-logging
acm-operator: acm
dba-operator: addon-dba-operator
reference-addon: redhat-reference-addon
ocm-addon-test-operator: redhat-ocm-addon-test-operator
kas-fleetshard-operator: redhat-kas-fleetshard-operator
```

5.4. OPENSHIFT CONTAINER PLATFORM VALIDATING WEBHOOKS

OpenShift Container Platform 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. Upon request the webhooks accept or reject the request. 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 5.3. List of validating webhooks

```
[
  {
    "webhookName": "clusterlogging-validation",
    "rules": [
      {
        "operations": [
          "CREATE",
          "UPDATE"
        ],
        "apiGroups": [
          "logging.openshift.io"
        ],
        "apiVersions": [
          "v1"
        ],
        "resources": [
          "clusterloggings"
        ],
        "scope": "Namespaced"
      }
    ],
    "documentString": "Managed OpenShift Customers may set log retention outside the allowed
```

```
range of 0-7 days"
  },
  {
    "webhookName": "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": "namespace-validation",
    "rules": [
      {
        "operations": [
          "CREATE",
          "UPDATE",
          "DELETE"
        ],
        "apiGroups": [
          ""
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "namespaces"
        ],
        "scope": "Cluster"
      }
    ],
    "documentString": "Managed OpenShift Customers may not modify privileged namespaces
identified by this regular expression (^kube.*|^openshift.*|^default$|^redhat.*) because customer
workloads should be placed in customer-created namespaces. Customers may not create
namespaces identified by this regular expression (^com$|^io$|^in$) because it could interfere with
critical DNS resolution. Additionally, customers may not set or change the values of these
```

```
Namespace labels [managed.openshift.io/storage-pv-quota-exempt managed.openshift.io/service-lb-quota-exempt]."
  },
  {
    "webhookName": "pod-validation",
    "rules": [
      {
        "operations": [
          "*"
        ],
        "apiGroups": [
          "v1"
        ],
        "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": "regular-user-validation",
    "rules": [
      {
        "operations": [
          "*"
        ],
        "apiGroups": [
          "autoscaling.openshift.io",
          "cloudcredential.openshift.io",
          "machine.openshift.io",
          "admissionregistration.k8s.io",
          "cloudingress.managed.openshift.io",
          "managed.openshift.io",
          "splunkforwarder.managed.openshift.io",
          "upgrade.managed.openshift.io"
        ],
        "apiVersions": [
          "*"
        ],
        "resources": [
          "*/*"
        ],
        "scope": "*"
      },
      {
        "operations": [
          "*"
        ],
        "apiGroups": [
          "config.openshift.io"
```

```
    ],
    "apiVersions": [
      "*"
    ],
    "resources": [
      "clusterversions",
      "clusterversions/status",
      "schedulers",
      "apiservers"
    ],
    "scope": "*"
  },
  {
    "operations": [
      "*"
    ],
    "apiGroups": [
      "operator.openshift.io"
    ],
    "apiVersions": [
      "*"
    ],
    "resources": [
      "kubeapiservers",
      "openshiftapiservers"
    ],
    "scope": "*"
  },
  {
    "operations": [
      "*"
    ],
    "apiGroups": [
      ""
    ],
    "apiVersions": [
      "*"
    ],
    "resources": [
      "nodes",
      "nodes/*"
    ],
    "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 [autoscaling.openshift.io machine.openshift.io
splunkforwarder.managed.openshift.io upgrade.managed.openshift.io config.openshift.io
operator.openshift.io cloudcredential.openshift.io admissionregistration.k8s.io
cloudingress.managed.openshift.io managed.openshift.io network.openshift.io], nor may Managed
OpenShift customers alter the APIServer, KubeAPIServer, OpenShiftAPIServer, ClusterVersion,
Node or SubjectPermission objects."
  }
]
```