Red Hat OpenShift Service on AWS 4

Support

Red Hat OpenShift Service on AWS Support.

Last Updated: 2026-02-16

Red Hat OpenShift Service on AWS 4 Support

Red Hat OpenShift Service on AWS Support.

Legal Notice

Copyright © Red Hat.

Except as otherwise noted below, the text of and illustrations in this documentation are licensed by Red Hat under the Creative Commons Attribution–Share Alike 3.0 Unported license . If you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

Red Hat, the Red Hat logo, JBoss, Hibernate, and RHCE are trademarks or registered trademarks of Red Hat, LLC. or its subsidiaries in the United States and other countries.

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

XFS is a trademark or registered trademark of Hewlett Packard Enterprise Development LP or its subsidiaries in the United States and other countries.

The OpenStack ® Word Mark and OpenStack logo are trademarks or registered trademarks of the Linux Foundation, used under license.

All other trademarks are the property of their respective owners.

Abstract

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

Table of Contents

CHAPTER 1. SUPPORT OVERVIEW 5
1.1. GET SUPPORT 5
1.2. REMOTE HEALTH MONITORING ISSUES 5
1.3. GATHER DATA ABOUT YOUR CLUSTER 5
1.4. TROUBLESHOOTING ISSUES 6

CHAPTER 2. MANAGING YOUR CLUSTER RESOURCES 7
2.1. INTERACTING WITH YOUR CLUSTER RESOURCES 7

CHAPTER 3. APPROVED ACCESS 8
3.1. ENABLING APPROVED ACCESS FOR ROSA CLUSTERS BY SUBMITTING A SUPPORT CASE 8
3.2. REVIEWING AN ACCESS REQUEST FROM AN EMAIL NOTIFICATION 9
3.3. REVIEWING AN ACCESS REQUEST FROM THE HYBRID CLOUD CONSOLE 9

CHAPTER 4. GETTING SUPPORT 11
4.1. GETTING SUPPORT 11
4.2. ABOUT THE RED HAT KNOWLEDGEBASE 11
4.3. SEARCHING THE RED HAT KNOWLEDGEBASE 11
4.4. SUBMITTING A SUPPORT CASE 12
4.5. ADDITIONAL RESOURCES 13

CHAPTER 5. REMOTE HEALTH MONITORING WITH CONNECTED CLUSTERS 14
5.1. ABOUT REMOTE HEALTH MONITORING 14
5.1.1. Telemetry and Insights Operator benefits 14
5.1.2. About Telemetry 15
5.1.3. Additional resources 15
5.1.3.1. Information collected by Telemetry 15
5.1.3.1.1. System information 15
5.1.3.1.2. Sizing Information 15
5.1.3.1.3. Usage information 16
5.1.3.2. User Telemetry 16
5.1.4. Additional resources 16
5.1.5. About the Insights Operator 16
5.1.5.1. Information collected by the Insights Operator 17
5.1.6. Additional resources 17
5.1.7. Understanding Telemetry and Insights Operator data flow 17
5.1.7.1. Telemetry and Insights Operator data flow 18
5.1.8. Additional details about how remote health monitoring data is used 18
5.1.9. Collection safeguards 18
5.1.10. Sharing 18
5.1.11. Third parties 18
5.2. SHOWING DATA COLLECTED BY REMOTE HEALTH MONITORING 18
5.2.1. Showing data collected by Telemetry 19
5.3. USING RED HAT LIGHTSPEED TO IDENTIFY ISSUES WITH YOUR CLUSTER 22
5.3.1. About Red Hat Lightspeed Advisor for Red Hat OpenShift Service on AWS 22
5.3.2. Understanding Red Hat Lightspeed advisor service recommendations 23
5.3.3. Displaying potential issues with your cluster 23
5.3.4. Displaying all Red Hat Lightspeed advisor service recommendations 24
5.3.5. Advisor recommendation filters 24
5.3.5.1. Filtering Red Hat Lightspeed advisor service recommendations 25
5.3.5.2. Removing filters from Red Hat Lightspeed advisor service recommendations 25
5.3.6. Disabling Red Hat Lightspeed advisor service recommendations 26

5.3.7. Enabling a previously disabled Red Hat Lightspeed advisor service recommendation 26
5.3.8. About Red Hat Lightspeed advisor service recommendations for workloads 27
5.3.9. Displaying the Red Hat Lightspeed status in the web console 27
5.4. USING THE INSIGHTS OPERATOR 28
5.4.1. Additional resources 28
5.4.2. Understanding Insights Operator alerts 28
5.4.3. Obfuscating Deployment Validation Operator data 28

CHAPTER 6. GATHERING DATA ABOUT YOUR CLUSTER 30
6.1. ABOUT THE MUST-GATHER TOOL 30
6.1.1. Gathering data about your cluster for Red Hat Support 31
6.2. REDUCING THE SIZE OF MUST-GATHER OUTPUT 32
6.2.1. Gathering data for specific resources 32
6.2.2. Must-gather flags 33
6.2.3. Gathering data about specific features 35
6.3. ADDITIONAL RESOURCES 40
6.3.1. Gathering network logs 40
6.3.2. Changing the must-gather storage limit 41
6.4. ABOUT SUPPORT LOG GATHER 42
6.4.1. Installing Support Log Gather by using the web console 42
6.4.2. Installing Support Log Gather by using the CLI 43
6.4.3. Configuring a Support Log Gather instance 45
6.5. ADDITIONAL RESOURCES 47
6.5.1. Configuration parameters for MustGather custom resource 47
6.5.2. Uninstalling Support Log Gather 49
6.5.3. Removing Support Log Gather resources 50
6.6. OBTAINING YOUR CLUSTER ID 51
6.7. QUERYING CLUSTER NODE JOURNAL LOGS 51
6.8. NETWORK TRACE METHODS 51
6.9. COLLECTING A HOST NETWORK TRACE 52
6.10. COLLECTING A NETWORK TRACE FROM AN RED HAT OPENSHIFT SERVICE ON AWS NODE OR CONTAINER 54
6.11. PROVIDING DIAGNOSTIC DATA TO RED HAT SUPPORT 56
6.12. ABOUT TOOLBOX 57
6.12.1. Installing packages to a toolbox container 57
6.12.2. Starting an alternative image with toolbox 57

CHAPTER 7. SUMMARIZING CLUSTER SPECIFICATIONS 59
7.1. SUMMARIZING CLUSTER SPECIFICATIONS BY USING A CLUSTER VERSION OBJECT 59

CHAPTER 8. TROUBLESHOOTING 61
8.1. REVIEW YOUR CLUSTER NOTIFICATIONS 61
8.1.1. Viewing cluster notifications using the Red Hat Hybrid Cloud Console 61
8.2. TROUBLESHOOTING RED HAT OPENSHIFT SERVICE ON AWS CLUSTER INSTALLATIONS 61
8.2.1. Installation troubleshooting 61
8.2.2. Verifying installation of Red Hat OpenShift Service on AWS clusters 62
8.2.3. Additional resources 62
8.2.4. Troubleshooting Red Hat OpenShift Service on AWS installation error codes 63
8.2.5. Additional resources 65
8.2.6. Troubleshooting access to Red Hat Hybrid Cloud Console 65
8.2.7. Additional resources 65
8.2.8. Verifying access to Red Hat OpenShift Service on AWS web console for Red Hat OpenShift Service on AWS cluster in ready state 65
8.2.9. Additional resources 65

8.2.10. Verifying access to Red Hat Hybrid Cloud Console for private Red Hat OpenShift Service on AWS clusters 65
8.2.11. Additional resources 66
8.3. TROUBLESHOOTING NETWORKING 66
8.3.1. Connectivity issues on clusters with private Network Load Balancers 66
8.4. VERIFYING NODE HEALTH 66
8.4.1. Reviewing node status, resource usage, and configuration 66
8.5. TROUBLESHOOTING OPERATOR ISSUES 67
8.5.1. Operator subscription condition types 67
8.5.2. Viewing Operator subscription status by using the CLI 68
8.5.3. Viewing Operator catalog source status by using the CLI 69
8.5.4. Additional resources 71
8.5.5. Querying Operator pod status 71
8.6. INVESTIGATING POD ISSUES 71
8.6.1. Understanding pod error states 72
8.6.2. Reviewing pod status 73
8.6.3. Inspecting pod and container logs 74
8.6.4. Accessing running pods 75
8.6.5. Starting debug pods with root access 76
8.6.6. Copying files to and from pods and containers 77
8.7. TROUBLESHOOTING THE SOURCE-TO-IMAGE PROCESS 77
8.7.1. Strategies for Source-to-Image troubleshooting 77
8.7.2. Gathering Source-to-Image diagnostic data 78
8.7.3. Gathering application diagnostic data to investigate application failures 79
8.7.4. Additional resources 81
8.8. TROUBLESHOOTING STORAGE ISSUES 81
8.8.1. Resolving multi-attach errors 82
8.9. INVESTIGATING MONITORING ISSUES 82
8.9.1. Investigating why user-defined project metrics are unavailable 83
8.9.2. Determining why Prometheus is consuming a lot of disk space 86
8.10. DIAGNOSING OPENSHIFT CLI (OC) ISSUES 88
8.10.1. Understanding OpenShift CLI (oc) log levels 88
8.10.2. Specifying OpenShift CLI (oc) log levels 88
8.11. TROUBLESHOOTING EXPIRED TOKENS 89
8.11.1. Troubleshooting expired offline access tokens 89
8.12. TROUBLESHOOTING IAM ROLES 89
8.12.1. Resolving issues with ocm-roles and user-role IAM resources 90
8.12.1.1. Creating an ocm-role IAM role 90
8.12.2. Additional resources 92
8.12.2.1. Creating a user-role IAM role 92
8.12.3. Additional resources 93
8.12.3.1. Associating your AWS account with IAM roles 93
8.12.3.2. Associating multiple AWS accounts with your Red Hat organization 94
8.13. TROUBLESHOOTING RED HAT OPENSHIFT SERVICE ON AWS CLUSTER DEPLOYMENTS 95
8.13.1. Obtaining information about a failed cluster 95
8.13.2. Troubleshooting cluster creation with an osdCcsAdmin error 95
8.13.3. Creating the Elastic Load Balancing (ELB) service-linked role 96
8.13.4. Repairing a cluster that cannot be deleted 96

CHAPTER 1. SUPPORT OVERVIEW

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

1.1. GET SUPPORT

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

1.2. REMOTE HEALTH MONITORING ISSUES

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

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

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 Red Hat OpenShift Service on AWS 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 Red Hat OpenShift Service on AWS component issues:

- Node issues: A cluster administrator can verify and troubleshoot node-related issues by reviewing the status, resource usage, and configuration of a node. You can query the following:
  - Kubelet's status on a node.
  - Cluster node journal logs.
- Operator issues: A cluster administrator can do the following to resolve Operator issues:
  - Verify Operator subscription status.
  - Check Operator pod health.
  - Gather Operator logs.
- Pod issues: A cluster administrator can troubleshoot pod-related issues by reviewing the status of a pod and completing the following:
  - Review pod and container logs.
  - Start debug pods with root access.
- 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 Red Hat OpenShift Service on AWS. Operators apply these configuration settings across the cluster.

2.1. INTERACTING WITH YOUR CLUSTER RESOURCES

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

Prerequisites

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

Procedure

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

```
$ oc 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. APPROVED ACCESS

Red Hat Site Reliability Engineering (SRE) typically does not require elevated access to systems as part of normal operations to manage and support Red Hat OpenShift Service on AWS clusters. Elevated access gives SRE the access levels of a cluster-admin role. See cluster roles for more information.

In the unlikely event that SRE needs elevated access to systems, you can use the *Approved Access* interface to review and *approve* or *deny* access to these systems.

Elevated access requests to clusters on Red Hat OpenShift Service on AWS clusters and the corresponding cloud accounts can be created by SRE either in response to a customer-initiated support ticket or in response to alerts received by SRE as part of the standard incident response process.

When *Approved Access* is enabled and an SRE creates an access request, *cluster owners* receive an email notification informing them of a new access request. The email notification contains a link allowing the cluster owner to quickly approve or deny the access request. You must respond in a timely manner otherwise there is a risk to your SLA for Red Hat OpenShift Service on AWS.

- If customers require additional users that are not the cluster owner to receive the email, they can add notification cluster contacts.
- Pending access requests are available in the Hybrid Cloud Console on the clusters list or **Access Requests** tab on the cluster overview for the specific cluster.

NOTE

Denying an access request requires you to complete the **Justification** field. In this case, SRE can not directly act on the resources related to the incident. Customers can still use the **Customer Support** to help investigate and resolve any issues.

3.1. ENABLING APPROVED ACCESS FOR ROSA CLUSTERS BY SUBMITTING A SUPPORT CASE

Red Hat OpenShift Service on AWS *Approved Access* is not enabled by default. To enable *Approved Access* for your Red Hat OpenShift Service on AWS clusters, you should create a support ticket.

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 **Configuration** category and click **Continue**.
4. Enter the following information:
   a. In the **Product** field, select **Red Hat OpenShift Service on AWS Hosted control planes**
   b. In the **Problem statement** field, enter **Enable ROSA Access Protection**
   c. Click **See more options**.

5. Select **OpenShift Cluster ID** from the drop-down list.
6. Fill the remaining mandatory fields in the form:
   a. What are you experiencing? What are you expecting to happen?
      i. Fill with **Approved Access**.
   b. Define the value or impact to you or the business.
      i. Fill with **Approved Access**.
   c. Click **Continue**.
7. Select **Severity** as **4(Low)** and click **Continue**.
8. Preview the case details and click **Submit**.

3.2. REVIEWING AN ACCESS REQUEST FROM AN EMAIL NOTIFICATION

Cluster owners will receive an email notification when Red Hat Site Reliability Engineering (SRE) request access to their cluster with a link to review the request in the Hybrid Cloud Console.

Procedure

1. Click the link within the email to bring you to the Hybrid Cloud Console.
2. In the **Access Request Details** dialog, click **Approve** or **Deny** under **Decision**.

NOTE

Denying an access request requires you to complete the **Justification** field. In this case, SRE can not directly act on the resources related to the incident. Customers can still use the **Customer Support** to help investigate and resolve any issues.

3. Click **Save**.

3.3. REVIEWING AN ACCESS REQUEST FROM THE HYBRID CLOUD CONSOLE

Review access requests for your Red Hat OpenShift Service on AWS clusters from the Hybrid Cloud Console.

Procedure

1. Navigate to OpenShift Cluster Manager and select **Cluster List**.
2. Click the cluster name to review the **Access Request**.
3. Select the **Access Requests** tab to list all **states**.
4. Select **Open** under **Actions** for the **Pending** state.

5. In the **Access Request Details** dialog, click **Approve** or **Deny** under **Decision**.

> **NOTE**
>
> Denying an access request requires you to complete the **Justification** field. In this case, SRE can not directly act on the resources related to the incident. Customers can still use the **Customer Support** to help investigate and resolve any issues.

6. Click **Save**.

CHAPTER 4. GETTING SUPPORT

You can get support for Red Hat OpenShift Service on AWS by searching the knowledge base, submitting a support case, and using remote health monitoring tools.

4.1. GETTING SUPPORT

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

From the Customer Portal, you can:

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

To identify issues with your cluster, you can use 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 Red Hat OpenShift Service on AWS version.

4.2. 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.

4.3. SEARCHING THE RED HAT KNOWLEDGEBASE

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

Prerequisites

- You have a Red Hat Customer Portal account.

Procedure

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:
    - Red Hat OpenShift Service on AWS components (such as **etcd**)

- Related procedure (such as **installation**)
- Warnings, error messages, and other outputs related to explicit failures

4. Click the **Enter** key.
5. Optional: Select the **Red Hat OpenShift Service on AWS**product filter.
6. Optional: Select the **Documentation** content type filter.

4.4. SUBMITTING A SUPPORT CASE

Submit a support case to Red Hat Support to get help with issues you encounter with Red Hat OpenShift Service on AWS.

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 **Red Hat OpenShift Service on AWS**from the **Product** drop-down menu.
5. Review the list of suggested Red Hat Knowledgebase solutions for a potential match against the problem that is being reported. If the suggested articles do not address the issue, click **Continue**.
6. Review the updated list of suggested Red Hat Knowledgebase solutions for a potential match against the problem that is being reported. The list is refined as you provide more information during the case creation process. If the suggested articles do not address the issue, click **Continue**.
7. Ensure that the account information presented is as expected, and if not, amend accordingly.
8. Check that the autofilled Red Hat OpenShift Service on AWS Cluster ID is correct. If it is not, manually obtain your cluster ID.

- To manually obtain your cluster ID using the Red Hat OpenShift Service on AWS web console:
  a. Navigate to **Home → Overview**.
  b. Find the value in the **Cluster ID** field of the **Details** section.
- Alternatively, it is possible to open a new support case through the Red Hat OpenShift Service on AWS web console and have your cluster ID autofilled.
  a. From the toolbar, navigate to **(?) Help → Open Support Case**.
  b. The **Cluster ID** value is autofilled.
- To obtain your cluster ID using the OpenShift CLI (**oc**), run the following command:

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

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

4.5. ADDITIONAL RESOURCES

- Using Red Hat Lightspeed to identify issues with your cluster

CHAPTER 5. REMOTE HEALTH MONITORING WITH CONNECTED CLUSTERS

5.1. ABOUT REMOTE HEALTH MONITORING

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

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

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

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

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

5.1.1. Telemetry and Insights Operator benefits

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

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

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

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

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

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

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

5.1.2. About Telemetry

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

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

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

5.1.3. Additional resources

- Red Hat OpenShift Service on AWS upgrade documentation

5.1.3.1. Information collected by Telemetry

The following information is collected by Telemetry:

5.1.3.1.1. System information

- Version information, including the Red Hat OpenShift Service on AWS cluster version and installed update details that are used to determine update version availability
- Update information, including the number of updates available per cluster, the channel and image repository used for an update, update progress information, and the number of errors that occur in an update
- The unique random identifier that is generated during an installation
- Configuration details that help Red Hat Support to provide beneficial support for customers, including node configuration at the cloud infrastructure level, hostnames, IP addresses, Kubernetes pod names, namespaces, and services
- The Red Hat OpenShift Service on AWS framework components installed in a cluster and their condition and status
- Events for all namespaces listed as "related objects" for a degraded Operator
- Information about degraded software
- Information about the validity of certificates
- The name of the provider platform that Red Hat OpenShift Service on AWS is deployed on and the data center location

5.1.3.1.2. Sizing Information

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

5.1.3.1.3. 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.

5.1.3.2. 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.

5.1.4. Additional resources

- Showing data collected by Telemetry
- Upstream cluster-monitoring-operator source code

5.1.5. About the Insights Operator

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

Users of Red Hat OpenShift Service on AWS can display the report of each cluster in the 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 Red Hat OpenShift Service on AWS by providing aggregated and critical information to product and support teams
- Make Red Hat OpenShift Service on AWS more intuitive

5.1.5.1. Information collected by the Insights Operator

The following information is collected by the Insights Operator:

- General information about your cluster and its components to identify issues that are specific to your Red Hat OpenShift Service on AWS version and environment.
- Configuration files, such as the image registry configuration, of your cluster to determine incorrect settings and issues that are specific to parameters you set.
- Errors that occur in the cluster components.
- Progress information of running updates, and the status of any component upgrades.
- Details of the platform that Red Hat OpenShift Service on AWS is deployed on and the region that the cluster is located in
- Cluster workload information transformed into discreet Secure Hash Algorithm (SHA) values, which allows Red Hat to assess workloads for security and version vulnerabilities without disclosing sensitive details.
- Workload information about the operating system and runtime environment, including runtime kinds, names, and version. This data gives Red Hat a better understanding of how you use Red Hat OpenShift Service on AWS containers so that we can proactively help you make investment decisions to drive optimal utilization.
- If an Operator reports an issue, information is collected about core Red Hat OpenShift Service on AWS pods in the **openshift-*** and **kube-*** projects. This includes state, resource, security context, volume information, and more.

5.1.6. Additional resources

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

5.1.7. 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 Red Hat OpenShift Service on AWS web console.

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

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

5.1.7.1. Telemetry and Insights Operator data flow

5.1.8. 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.

5.1.9. Collection safeguards

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

5.1.10. 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.

5.1.11. Third parties

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

5.2. SHOWING DATA COLLECTED BY REMOTE HEALTH MONITORING

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

5.2.1. 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 Red Hat OpenShift Service on AWS 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:provisione
r: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",quant
ile="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",quan
tile="0.99"}' \
--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"}'
```

5.3. 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 Red Hat OpenShift Service on AWS can display the results in the Advisor service on Red Hat Hybrid Cloud Console.

5.3.1. About Red Hat Lightspeed Advisor for Red Hat OpenShift Service on AWS

You can use the Red Hat Lightspeed advisor service to assess and monitor the health of your Red Hat OpenShift Service on AWS clusters. Whether you are concerned about individual clusters, or with your whole infrastructure, it is important to be aware of 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

5.3.2. 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

5.3.3. 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.

5.3.4. 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.

5.3.5. 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.

5.3.5.1. Filtering Red Hat Lightspeed advisor service recommendations

As an Red Hat OpenShift Service on AWS 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.

5.3.5.2. 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.

Procedure

- 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.

5.3.6. 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**.

5.3.7. 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**.

5.3.8. 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

5.3.9. 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 Red Hat OpenShift Service on AWS web console. This status shows the number of issues in the different categories and, for further details, links to the reports in OpenShift Cluster Manager.

Prerequisites

- Your cluster is registered in OpenShift Cluster Manager.
- Remote health reporting is enabled, which is the default.

- You are logged in to the Red Hat OpenShift Service on AWS web console.

Procedure

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

5.4. USING THE INSIGHTS OPERATOR

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

5.4.1. Additional resources

- Using Red Hat Lightspeed to identify issues with your cluster

5.4.2. Understanding Insights Operator alerts

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

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

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

Table 5.1. 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.

5.4.3. 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 Red Hat OpenShift Service on AWS. 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 Red Hat OpenShift Service on AWS 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 6. GATHERING DATA ABOUT YOUR CLUSTER

When opening a support case, it is helpful to provide debugging information about your cluster to Red Hat Support. You can use tools such as **must-gather**, **sosreport**, and cluster node journal logs to collect diagnostic data.

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

6.1. 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:v4.21.0
  ```

- 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:v4.21.0
```

6.1.1. 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.
- The Red Hat OpenShift Service on AWS CLI (**oc**) is installed.

Procedure

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

NOTE

If your cluster is in a disconnected environment, 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 in disconnected environments, you must import the default **must-gather** image as an image stream.

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

2. Run the **oc adm must-gather** command:

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

IMPORTANT

If you are in a disconnected environment, use the **--image** flag as part of must-gather and point to the payload image.

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>
```

where:

<must_gather_local_dir>

Replace 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.

6.2. REDUCING THE SIZE OF MUST-GATHER OUTPUT

The **oc adm must-gather** command collects comprehensive cluster information. However, a full data collection can result in a large file that is difficult to upload and analyze and could result in timeouts.

To manage the output size and target your data collection for more effective troubleshooting, you can pass specific flags to the underlying **gather** script or scope the collection to particular resources.

6.2.1. Gathering data for specific resources

Instead of collecting data for the entire cluster, you can direct the **must-gather** tool to inspect a specific resource. This method is highly effective for isolating issues within a single project, Operator, or application.

The **must-gather** tool uses **oc adm inspect** internally. You can specify what to inspect by passing the **inspect** command and its arguments after the **--** separator.

Procedure

- To gather data for a specific namespace, such as **my-project**, run the following command:

```
$ oc adm must-gather --dest-dir=my-project-must-gather -- oc adm inspect ns/my-project
```

- This command collects all standard resources within the **my-project** namespace, including logs from pods in that namespace, but excludes cluster-scoped resources.
- To gather data related to a specific Cluster Operator, such as **openshift-apiserver**, run the following command:

```
$ oc adm must-gather --dest-dir=apiserver-must-gather -- oc adm inspect
clusteroperator/openshift-apiserver
```

- To exclude logs entirely and significantly reduce the size of the **must-gather** archive, add a double dash (**--**) after **oc adm must-gather** command and add the **--no-logs** argument:

```
$ oc adm must-gather -- /usr/bin/gather --no-logs
```

6.2.2. Must-gather flags

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

Table 6.1. Red Hat OpenShift Service on AWS 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, Red Hat OpenShift Service on AWS's default **must-gather** image is used.

Flag	Example command	Description
--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.
--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.

6.2.3. 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 6.2. Supported must-gather images

Image	Purpose
registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel9:v4.21.0	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-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/odf4/odf-must-gather-rhel9:v<installed_version_ODF>	Data collection for Red Hat OpenShift Data Foundation.
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/openshift4/ose-local-storage-mustgather-rhel9:v<installed_version_LSO>	Data collection for Local Storage Operator.
registry.redhat.io/openshift-sandboxed-containers/osc-must-gather-rhel8:v<installed_version_sandboxed_containers>	Data collection for OpenShift sandboxed containers.

Image	Purpose
registry.redhat.io/workload-availability/node-healthcheck-must-gather-rhel8:v<installed_version_NHC>	Data collection for the Red Hat Workload Availability Operators, including the Self Node Remediation (SNR) Operator, the Fence Agents Remediation (FAR) Operator, the Machine Deletion Remediation (MDR) Operator, the Node Health Check (NHC) Operator, and the Node Maintenance Operator (NMO). Use this image if your NHC Operator version is **earlier than 0.9.0**. For more information, see the "Gathering data" section for the specific Operator in Remediation, fencing, and maintenance (Workload Availability for Red Hat OpenShift documentation).
registry.redhat.io/workload-availability/node-healthcheck-must-gather-rhel9:v<installed_version_NHC>	Data collection for the Red Hat Workload Availability Operators, including the Self Node Remediation (SNR) Operator, the Fence Agents Remediation (FAR) Operator, the Machine Deletion Remediation (MDR) Operator, the Node Health Check (NHC) Operator, and the Node Maintenance Operator (NMO). Use this image if your NHC Operator version is **0.9.0. or later**. For more information, see the "Gathering data" section for the specific Operator in Remediation, fencing, and maintenance (Workload Availability for Red Hat OpenShift documentation).
registry.redhat.io/numaresources/numaresources-must-gather-rhel9:v<installed-version-nro>	Data collection for the NUMA Resources Operator (NRO).
registry.redhat.io/openshift4/ptp-must-gather-rhel8:v<installed-version-ptp>	Data collection for the PTP 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.
registry.redhat.io/lvms4/lvms-must-gather-rhel9:v<installed_version_LVMS>	Data collection for the LVM Operator.

Image	Purpose
registry.redhat.io/compliance/openshift-compliance-must-gather-rhel8:<digest-version>	Data collection for the Compliance Operator.

NOTE

To determine the latest version for an Red Hat OpenShift Service on AWS 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 Red Hat OpenShift Service on AWS 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 information on gathering data about the Custom Metrics Autoscaler, see the Additional resources section that follows.

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 \
  --image=registry.redhat.io/container-native-virtualization/cnv-must-gather-rhel9:v4.21.0
```

You can use the **must-gather** tool with additional arguments to gather data that is specifically related to OpenShift Logging and the Red Hat OpenShift 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 must-gather output for OpenShift Logging

```
├── cluster-logging
│   ├── clo
│   │   ├── cluster-logging-operator-74dd5994f-6ttgt
│   │   ├── clusterlogforwarder_cr
│   │   ├── cr
│   │   ├── csv
│   │   ├── deployment
│   │   └── logforwarding_cr
│   ├── collector
│   │   ├── fluentd-2tr64
│   ├── 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
│   │   │   ├── elasticsearch-im-app-1596020400-gm6nl.1626341a296c16a1.yaml
│   │   │   ├── elasticsearch-im-audit-1596020400-9l9n4.1626341a2af81bbd.yaml
│   │   │   ├── elasticsearch-im-infra-1596020400-v98tk.1626341a2d821069.yaml
│   │   │   ├── elasticsearch-im-app-1596020400-cc5vc.1626341a3019b238.yaml
│   │   │   ├── elasticsearch-im-audit-1596020400-s8d5s.1626341a31f7b315.yaml
│   │   │   ├── elasticsearch-im-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
│   │   ├── elasticsearch-cdm-lp8l38m0-1-794d6dd989-4jxms
│   │   ├── elasticsearch-im-app-1596030300-bpgcx
│   │   │   ├── elasticsearch-im-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 \
 --image=quay.io/kubevirt/must-gather
```

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>
```

where:

<must_gather_local_dir>

Replace 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.

6.3. ADDITIONAL RESOURCES

- Red Hat OpenShift Service on AWS update life cycle

6.3.1. 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>
```

   Replace the **<must_gather_local_dir>** placeholder 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.

6.3.2. 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.

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>
```

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

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

6.4. 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.

6.4.1. 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 Red Hat OpenShift Service on AWS web console.

Procedure

1. Log in to the Red Hat OpenShift Service on AWS 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.

6.4.2. 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.v4.21.0 support log gather 4.21.0             Succeeded
```

6.4.3. 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.

6.5. ADDITIONAL RESOURCES

- Understanding and creating service accounts

6.5.1. 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.

6.5.2. 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 Red Hat OpenShift Service on AWS web console.
- The Support Log Gather is installed.

Procedure

1. Log in to the Red Hat OpenShift Service on AWS 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**.

6.5.3. 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 Red Hat OpenShift Service on AWS web console.

Procedure

1. Log in to the Red Hat OpenShift Service on AWS 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**.

6.6. 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 Red Hat OpenShift Service on AWS 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 **cluster-admin** role.
- You have access to the web console or the OpenShift CLI (**oc**) installed.

Procedure

- 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"}'
```

6.7. 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.
- You have installed the OpenShift CLI (**oc**).

Procedure

- Query **kubelet journald** unit logs from Red Hat OpenShift Service on AWS cluster nodes. The following example queries worker nodes only:

```
$ oc adm node-logs --role=worker -u kubelet
```

kubelet

Replace as appropriate to query other unit logs.

6.8. NETWORK TRACE METHODS

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

Red Hat OpenShift Service on AWS supports two ways of performing a network trace. Review the following table and choose the method that meets your needs.

Table 6.3. 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.
Collecting a network trace from an Red Hat OpenShift Service on AWS 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.

6.9. 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 Red Hat OpenShift Service on AWS as a user with the **cluster-admin** role.
- 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
```

where:

--dest-dir /tmp/captures

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.

--source-dir '/tmp/tcpdump/'

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.

--image registry.redhat.io/openshift4/network-tools-rhel8:latest

The **--image** argument specifies a container image that includes the **tcpdump** command.

--node-selector 'node-role.kubernetes.io/worker'

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.

--host-network=true

The **--host-network=true** argument is required so that the packet captures are performed on the network interfaces of the node.

--timeout 30s

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.

-i any

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
 │    └─── registry-redhat-io-openshift4-network-tools-rhel8-sha256-bca...
 │         └─── 2022-01-13T19:31:31.pcap
 ├─── ip-10-0-201-178-ec2-internal
```

```
|     └─── registry-redhat-io-openshift4-network-tools-rhel8-sha256-bca...
|       └─── 2022-01-13T19:31:30.pcap
├─── ip-...
└─── timestamp
```

where:

ip-10-0-192-217-ec2-internal, ip-10-0-201-178-ec2-internal

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.

6.10. COLLECTING A NETWORK TRACE FROM AN RED HAT OPENSHIFT SERVICE ON AWS NODE OR CONTAINER

When investigating potential network-related Red Hat OpenShift Service on AWS issues, Red Hat Support might request a network packet trace from a specific Red Hat OpenShift Service on AWS cluster node or from a specific container. The recommended method to capture a network trace in Red Hat OpenShift Service on AWS 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 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
```

where:

/host/var/tmp/my-cluster-node_$(date +%d_%m_%Y-%H_%M_%S-%Z).pcap

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
```

   where:

   /host/var/tmp/my-cluster-node-my-container_$(date +%d_%m_%Y-%H_%M_%S-%Z).pcap

   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
```

where:

/host/var/tmp/my-tcpdump-capture-file.pcap

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.

6.11. PROVIDING DIAGNOSTIC DATA TO RED HAT SUPPORT

When investigating Red Hat OpenShift Service on AWS 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.
- 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 Red Hat OpenShift Service on AWS 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
```

where:

/host/var/tmp/my-diagnostic-data.tar.gz

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.

6.12. 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.

6.12.1. 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>
```

6.12.2. 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 Red Hat OpenShift Service on AWS 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
IMAGE=fedora/fedora:latest
TOOLBOX_NAME=toolbox-fedora-latest
```

where:

REGISTRY=quay.io

Optional: Specify an alternative container registry.

IMAGE=fedora/fedora:latest

Specify an alternative image to start.

TOOLBOX_NAME=toolbox-fedora-latest

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 7. SUMMARIZING CLUSTER SPECIFICATIONS

You can summarize your cluster specifications by querying the **clusterversion** resource to view cluster version information and component status.

7.1. SUMMARIZING CLUSTER SPECIFICATIONS BY USING A CLUSTER VERSION OBJECT

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

Prerequisites

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

Procedure

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

```
$ oc get clusterversion
```

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 8. TROUBLESHOOTING

8.1. REVIEW YOUR CLUSTER NOTIFICATIONS

When you are trying to resolve a problem with your cluster, your cluster notifications are a good source of information.

Cluster notifications are messages about the status, health, or performance of your cluster. They are also the primary way that Red Hat Site Reliability Engineering (SRE) communicates with you about cluster health and resolving problems with your cluster.

8.1.1. Viewing cluster notifications using the Red Hat Hybrid Cloud Console

Cluster notifications provide important information about the health of your cluster. You can view notifications that have been sent to your cluster in the **Cluster history** tab on the Red Hat Hybrid Cloud Console.

Prerequisites

- You are logged in to the Hybrid Cloud Console.

Procedure

1. Navigate to the Clusters page of the Hybrid Cloud Console.
2. Click the name of your cluster to go to the cluster details page.
3. Click the **Cluster history** tab.
   Cluster notifications appear under the Cluster history heading.
4. Optional: Filter for relevant cluster notifications
   Use the filter controls to hide cluster notifications that are not relevant to you, so that you can focus on your area of expertise or on resolving a critical issue. You can filter notifications based on text in the notification description, severity level, notification type, when the notification was received, and which system or person triggered the notification.

8.2. TROUBLESHOOTING RED HAT OPENSHIFT SERVICE ON AWS CLUSTER INSTALLATIONS

For help with the installation of Red Hat OpenShift Service on AWS clusters, see the following sections.

8.2.1. Installation troubleshooting

This procedure describes how to troubleshoot installation issues for Red Hat OpenShift Service on AWS clusters.

Procedure

- Inspect install or uninstall logs:
  - To display install logs, run the following command, replacing **<cluster_name>** with the name of your cluster:

```
$ rosa logs install --cluster=<cluster_name>
```

- To watch the logs, include the **--watch** flag:

```
$ rosa logs install --cluster=<cluster_name> --watch
```

- To display uninstall logs, run the following command, replacing **<cluster_name>** with the name of your cluster:

```
$ rosa logs uninstall --cluster=<cluster_name>
```

- To watch the logs, include the **--watch** flag:

```
$ rosa logs uninstall --cluster=<cluster_name> --watch
```

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

```
$ rosa verify quota
```

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

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

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

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

8.2.2. Verifying installation of Red Hat OpenShift Service on AWS clusters

If the Red Hat OpenShift Service on AWS cluster is in the installing state for over 30 minutes and has not become ready, ensure the AWS account environment is prepared for the required cluster configurations. If the AWS account environment is prepared for the required cluster configurations correctly, try to delete and recreate the cluster. If the problem persists, contact support.

Procedure

- Verify the AWS account environment is prepared for the required cluster configurations.
- If the AWS account environment is prepared correctly, try to delete and recreate the cluster.
- If the problem persists, contact support.

8.2.3. Additional resources

- AWS prerequisites for Red Hat OpenShift Service on AWS

8.2.4. Troubleshooting Red Hat OpenShift Service on AWS installation error codes

The following table lists Red Hat OpenShift Service on AWS installation error codes and what you can do to troubleshoot these errors.

Table 8.1. Red Hat OpenShift Service on AWS installation error codes

Error code	Description	Resolution
OCM3999	Unknown error.	Check the cluster installation logs for more details, or delete this cluster and retry cluster installation. If this issue persists, contact support by logging in to the **Customer Support** page.
OCM5001	Red Hat OpenShift Service on AWS cluster provision has failed.	Check the cluster installation logs for more details, or delete this cluster and retry cluster installation. If this issue persists, contact support by logging in to the **Customer Support** page.
OCM5002	The maximum resource tag size of 25 has been exceeded.	Check the cluster information to determine if you can remove any unnecessary tags you have specified and retry cluster installation.
OCM5003	Unable to establish an AWS client to provision the cluster.	You must create several role resources on your AWS account to create and manage a Red Hat OpenShift Service on AWS cluster. Ensure that your provided AWS credentials are correct and retry cluster installation. For more information about Red Hat OpenShift Service on AWS IAM role resources, see *ROSA IAM role resources* in the *Additional resources* section.

Error code	Description	Resolution
OCM5004	Unable to establish a cross-account AWS client to provision the cluster.	You must create several role resources on your AWS account to create and manage a Red Hat OpenShift Service on AWS cluster. Ensure that your provided AWS credentials are correct and retry cluster installation. For more information about Red Hat OpenShift Service on AWS IAM role resources, see *ROSA IAM role resources* in the *Additional resources* section.
OCM5005	Failed to retrieve AWS subnets defined for the cluster.	Review the provided subnet IDs and retry cluster installation.
OCM5006	You must configure at least one private AWS subnet for the cluster.	Review the provided subnet IDs and retry cluster installation.
OCM5007	Unable to create AWS STS prerequisites for the cluster.	Verify that account and operator roles have been created and are correct. For more information, see *AWS STS and ROSA with HCP explained* in the *Additional resources* section.
OCM5008	The provided cluster flavour is incorrect.	Verify that the provided name or ID is correct when you are using the flavour parameter and retry cluster creation.
OCM5009	The cluster version could not be found.	Ensure that the configured version ID matches a valid Red Hat OpenShift Service on AWS version.
OCM5010	Failed to tag subnets for the cluster.	Confirm that the AWS permissions and the subnet configurations are correct. You must tag at least one private subnet and, if applicable, one public subnet.
OCM5011	Cluster installation has failed due to unavailable capacity in the selected region.	Try your cluster installation in another region or retry cluster installation.

8.2.5. Additional resources

- AWS STS and Red Hat OpenShift Service on AWS explained

8.2.6. Troubleshooting access to Red Hat Hybrid Cloud Console

In Red Hat OpenShift Service on AWS clusters, the Red Hat OpenShift Service on AWS OAuth server is hosted in the Red Hat service's AWS account while the web console service is published using the cluster's default ingress controller in the cluster's AWS account. If you can log in to your cluster using the OpenShift CLI (oc) but cannot access the Red Hat OpenShift Service on AWS web console, verify the following criteria are met:

Procedure

- Verify the console workloads are running.
- Verify the default ingress controller's load balancer is active.
- Verify you are accessing the console from a machine that has network connectivity to the cluster's VPC network.

8.2.7. Additional resources

- Red Hat OpenShift Service on AWS OAuth server
- Web Console Overview

8.2.8. Verifying access to Red Hat OpenShift Service on AWS web console for Red Hat OpenShift Service on AWS cluster in ready state

Red Hat OpenShift Service on AWS clusters return a **ready** status when the control plane hosted in the Red Hat OpenShift Service on AWS service account becomes ready. Cluster console workloads are deployed on the cluster's worker nodes. The Red Hat OpenShift Service on AWS web console will not be available and accessible until the worker nodes have joined the cluster and console workloads are running.

Procedure

- If your Red Hat OpenShift Service on AWS cluster is ready but you are unable to access the Red Hat OpenShift Service on AWS web console for the cluster, wait for the worker nodes to join the cluster and retry accessing the console.
  You can either log in to the Red Hat OpenShift Service on AWS cluster or use the **rosa describe machinepool** command in the **rosa** CLI watch the nodes.

8.2.9. Additional resources

- **rosa describe machinepool**

8.2.10. Verifying access to Red Hat Hybrid Cloud Console for private Red Hat OpenShift Service on AWS clusters

The console of the private cluster is private by default. During cluster installation, the default Ingress Controller managed by OpenShift's Ingress Operator is configured with an internal AWS Network Load Balancer (NLB).

Procedure

- If your private Red Hat OpenShift Service on AWS cluster shows a **ready** status but you cannot access the Red Hat OpenShift Service on AWS web console for the cluster, try accessing the cluster console from either within the cluster VPC or from a network that is connected to the VPC.

8.2.11. Additional resources

- AWS VPC Documentation

8.3. TROUBLESHOOTING NETWORKING

This document describes how to troubleshoot networking errors.

8.3.1. Connectivity issues on clusters with private Network Load Balancers

Red Hat OpenShift Service on AWS clusters created with version 4 deploy AWS Network Load Balancers (NLB) by default for the **default** ingress controller. In the case of a private NLB, the NLB's client IP address preservation might cause connections to be dropped where the source and destination are the same host. See the AWS's documentation about how to Troubleshoot your Network Load Balancer. This IP address preservation has the implication that any customer workloads cohabitating on the same node with the router pods, may not be able send traffic to the private NLB fronting the ingress controller router.

Procedure

- To mitigate this impact, reschedule your workloads onto nodes separate from those where the router pods are scheduled. Alternatively, rely on the internal pod and service networks for accessing other workloads co-located within the same cluster.

8.4. VERIFYING NODE HEALTH

You can verify and troubleshoot node-related issues by reviewing the status, resource usage, and configuration of a node.

8.4.1. 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
```

8.5. TROUBLESHOOTING OPERATOR ISSUES

A cluster administrator can do the following to resolve Operator issues: verify Operator subscription status, check Operator pod health, and gather Operator logs.

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

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

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

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

8.5.1. Operator subscription condition types

Subscriptions can report the following condition types:

Table 8.2. 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 Red Hat OpenShift Service on AWS cluster Operators are managed by the Cluster Version Operator (CVO) and they do not have a **Subscription** object. Application Operators are managed by Operator Lifecycle Manager (OLM) and they have a **Subscription** object.

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

8.5.3. 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.

8.5.4. Additional resources

- States of Connectivity (gRPC documentation)

8.5.5. 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>
```

8.6. INVESTIGATING POD ISSUES

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

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

The first thing to check when pod issues arise is the pod's status. If an explicit pod failure has occurred, observe the pod's error state to identify specific image, container, or pod network issues. Focus diagnostic data collection according to the error state. Review pod event messages, as well as pod and

container log information. Diagnose issues dynamically by accessing running Pods on the command line, or start a debug pod with root access based on a problematic pod's deployment configuration.

8.6.1. 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 8.3. 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.
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.

Pod error state	Description
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.

8.6.2. 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
```

8.6.3. Inspecting pod and container logs

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

Prerequisites

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

Procedure

1. Query logs for a specific pod:

```
$ 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>
```

8.6.4. 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>
```

where:

<pod_name>

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>
```

where:

<pod_name> <host_port>:<pod_port>

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

8.6.5. 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.

8.6.6. 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>
```

-c <container_name> refers to the desired container in a pod. If you do not specify a container with the **-c** option, then the first container in a pod is selected.

2. Copy a file from a pod:

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

-c <container_name> refers to the desired container in a pod. If you do not specify a container with the **-c** option, then the first container in a pod is selected.

NOTE

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

8.7. TROUBLESHOOTING THE SOURCE-TO-IMAGE PROCESS

A cluster administrator can observe the S2I stages to determine where in the S2I process a failure occurred and gather diagnostic data to resolve Source-to-Image issues.

8.7.1. 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.

8.7.2. 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 **cluster-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
   ```

   Use the **-w** flag 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>
```

8.7.3. 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 **cluster-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.

5. If diagnostic binaries are not available within a container, you can run a host's diagnostic binaries within a container's namespace by using **nsenter**. The following example runs **ip ad** within a container's namespace, using the host`s **ip** binary.

a. 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
```

b. 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**
>
> Red Hat OpenShift Service on AWS 4 cluster nodes running Red Hat Enterprise Linux CoreOS (RHCOS) are immutable and rely on Operators to apply cluster changes. Accessing cluster nodes by using SSH is not recommended. However, if the Red Hat OpenShift Service on AWS API is not available, or the kubelet is not properly functioning on the target node, **oc** operations will be impacted. In such situations, it is possible to access nodes using **ssh core@<node>.<cluster_name>.<base_domain>** instead.

c. Determine the target container ID:

```
# crictl ps
```

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

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

e. Run **ip ad** within the container's namespace, using the host's **ip** binary. This example uses **31150** as the container's process ID. 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 **ip ad** command is run in the container's namespace from the host:

```
# nsenter -n -t 31150 -- ip ad
```

> **NOTE**
>
> Running a host's diagnostic binaries within a container's namespace is only possible if you are using a privileged container such as a debug node.

8.7.4. Additional resources

- Source-to-Image (S2I) build

8.8. TROUBLESHOOTING 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 resolve multi-attach

storage issues by enabling multiple attachments using RWX volumes or recovering/deleting the failed node when using an RWO volume.

8.8.1. 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.

8.9. INVESTIGATING MONITORING ISSUES

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

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.

8.9.1. 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 **cluster-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 Red Hat OpenShift Service on AWS web console UI.

   a. Log in to the Red Hat OpenShift Service on AWS 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.

8.9.2. 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 **cluster-admin** cluster role.
- You have installed the OpenShift CLI (**oc**).

Procedure

1. In the Red Hat OpenShift Service on AWS 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 Red Hat OpenShift Service on AWS project**, create a Red Hat support case on the Red Hat Customer Portal .
4. Review the TSDB status using the Prometheus HTTP API by following these steps when logged in as a cluster administrator:
   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},
...
```

8.10. DIAGNOSING OPENSHIFT CLI (oc) ISSUES

You can investigate OpenShift CLI (**oc**) issues by increasing the log level to get more detailed diagnostic information.

8.10.1. Understanding OpenShift CLI (oc) log levels

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

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

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

Table 8.4. 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.

8.10.2. Specifying OpenShift CLI (oc) log levels

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

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

Prerequisites

- Install the OpenShift CLI (**oc**).

Procedure

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

```
$ 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...
```

8.11. TROUBLESHOOTING EXPIRED TOKENS

You can troubleshoot expired offline access tokens that prevent access to your Red Hat OpenShift Service on AWS cluster.

8.11.1. Troubleshooting expired offline access tokens

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

Example output

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

Procedure

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

8.12. TROUBLESHOOTING IAM ROLES

You can troubleshoot IAM role issues that prevent proper access to your Red Hat OpenShift Service on AWS cluster resources.

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

You may receive an error when trying to create a cluster using the ROSA command-line interface (CLI) (**rosa**). This error means that the **user-role** IAM role is not linked to your AWS account. The most likely cause of this error is that another user in your Red Hat organization created the **ocm-role** IAM role. Your **user-role** IAM role needs to be created.

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

Procedure

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

```
$ rosa list ocm-role
```

Example output

```
I: Fetching ocm roles
ROLE NAME                          ROLE ARN                                    LINKED  ADMIN
ManagedOpenShift-OCM-Role-1158  arn:aws:iam::2066:role/ManagedOpenShift-OCM-Role-
1158   No      No
```

```
$ rosa list user-role
```

Example output

```
I: Fetching user roles
ROLE NAME                              ROLE ARN                                    LINKED
ManagedOpenShift-User.osdocs-Role  arn:aws:iam::2066:role/ManagedOpenShift-
User.osdocs-Role  Yes
```

With the results of these commands, you can create and link the missing IAM resources.

8.12.1.1. Creating an ocm-role IAM role

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

Prerequisites

- You have an AWS account.
- You have Red Hat Organization Administrator privileges in the OpenShift Cluster Manager organization.
- You have the permissions required to install AWS account-wide roles.

- You have installed and configured the latest ROSA CLI, **rosa**, on your installation host.

Procedure

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

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

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

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

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

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

where:

Role prefix

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

Enable admin capabilities for the OCM role (optional)

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

> **NOTE**
>
> You do not see this prompt if you used the **--admin** option.

Permissions boundary ARN (optional)

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

Role Path (optional)

Specify an IAM path for the user name.

Role creation mode

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

Create the 'ManagedOpenShift-OCM-Role-182' role?

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

OCM Role ARN

Confirm that you want to associate your IAM role with your OpenShift Cluster Manager.

Link the 'arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182' role with organization '<AWS ARN>'?

Links the created role with your AWS organization.

8.12.2. Additional resources

- Methods of account-wide role creation

8.12.2.1. Creating a user-role IAM role

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

Prerequisites

- You have an AWS account.
- You have installed and configured the latest ROSA CLI, **rosa**, on your installation host.

Procedure

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

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

This command allows you to create the role by specifying specific attributes. The following example output shows the "auto mode" selected, which lets the ROSA CLI (**rosa**) to create your Operator roles and policies. See "Understanding the auto and manual deployment modes" for more information. The following example shows what your creation flow may look like.

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

where:

Role prefix

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

Permissions boundary ARN (optional)

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

Role Path (optional)

Specify an IAM path for the user name.

Role creation mode

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

Create the 'ManagedOpenShift-User.osdocs-Role' role?

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

Link the 'arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role' role with account '1AGE'?

Links the created role with your AWS organization.

IMPORTANT

If you unlink or delete your **user-role** IAM role before deleting your cluster, an error prevents you from deleting your cluster. You must create or relink this role to proceed with the deletion process.

8.12.3. Additional resources

- Methods of account-wide role creation

8.12.3.1. Associating your AWS account with IAM roles

You can associate or link your AWS account with existing IAM roles by using the ROSA command-line interface (CLI) (**rosa**).

Prerequisites

- You have an AWS account.
- You have the permissions required to install AWS account-wide roles. See the "Additional resources" of this section for more information.
- You have installed and configured the latest AWS CLI (**aws**) and ROSA CLI on your installation host.
- You have created the **ocm-role** and **user-role** IAM roles, but have not yet linked them to your AWS account. You can check whether your IAM roles are already linked by running the following commands:

```
$ rosa list ocm-role
```

```
$ rosa list user-role
```

If **Yes** is displayed in the **Linked** column for both roles, you have already linked the roles to an AWS account.

Procedure

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

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

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

For example:

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

2. In the ROSA CLI, link your **user-role** resource to your Red Hat user account by using your Amazon Resource Name (ARN):

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

For example:

```
I: Linking User role
? Link the 'arn:aws:iam::<ARN>:role/ManagedOpenShift-User-Role-125' role with
organization '<AWS ID>'? Yes
I: Successfully linked role-arn 'arn:aws:iam::<ARN>:role/ManagedOpenShift-User-Role-125'
with organization account '<AWS ID>'
```

8.12.3.2. Associating multiple AWS accounts with your Red Hat organization

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

With this capability, you can create clusters on different AWS profiles according to characteristics that make sense for your business, for example, by using one AWS profile for each region to create region-bound environments.

Prerequisites

- You have an AWS account.
- You are using OpenShift Cluster Manager to create clusters.

- You have the permissions required to install AWS account-wide roles.
- You have installed and configured the latest AWS CLI (**aws**) and ROSA command-line interface (CLI) (**rosa**) on your installation host.
- You have created the **ocm-role** and **user-role** IAM roles for Red Hat OpenShift Service on AWS.

Procedure

- To specify an AWS account profile when creating an OpenShift Cluster Manager role:

```
$ rosa create --profile <aws_profile> ocm-role
```

- To specify an AWS account profile when creating a user role:

```
$ rosa create --profile <aws_profile> user-role
```

- To specify an AWS account profile when creating the account roles:

```
$ rosa create --profile <aws_profile> account-roles
```

NOTE

If you do not specify a profile, the default AWS profile and its associated AWS region are used.

8.13. TROUBLESHOOTING RED HAT OPENSHIFT SERVICE ON AWS CLUSTER DEPLOYMENTS

This document describes how to troubleshoot cluster deployment errors.

8.13.1. Obtaining information about a failed cluster

If a cluster deployment fails, the cluster is put into an "error" state.

Procedure

- Run the following command to get more information:

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

8.13.2. Troubleshooting cluster creation with an osdCcsAdmin error

If a cluster creation action fails, you might receive the following error message.

Example output

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

Procedure

1. Delete the stack:

```
$ rosa init --delete
```

2. Reinitialize your account:

```
$ rosa init
```

8.13.3. Creating the Elastic Load Balancing (ELB) service-linked role

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

Example output

```
Error: Error creating network Load Balancer: AccessDenied: User:
arn:aws:sts::xxxxxxxxxxxx:assumed-role/ManagedOpenShift-Installer-Role/xxxxxxxxxxxxxxxxxx is
not authorized to perform: iam:CreateServiceLinkedRole on resource:
arn:aws:iam::xxxxxxxxxxxx:role/aws-service-
role/elasticloadbalancing.amazonaws.com/AWSServiceRoleForElasticLoadBalancing"
```

Procedure

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

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

NOTE

This command only needs to be executed once per account.

8.13.4. Repairing a cluster that cannot be deleted

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

Example output

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

Operation ID: b0572d6e-fe54-499b-8c97-46bf6890011c
```

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

Example output

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

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

Procedure

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

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

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

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