Red Hat OpenShift Service on AWS 4

Upgrading

Understanding upgrading options for Red Hat OpenShift Service on AWS

Last Updated: 2023-01-30

Red Hat OpenShift Service on AWS 4 Upgrading

Understanding upgrading options for Red Hat OpenShift Service on AWS

Legal Notice

Abstract

This document provides information about upgrading Red Hat OpenShift Service on AWS (ROSA) clusters.

Table of Contents

CHAPTER 1. PREPARING TO UPGRADE ROSA TO 4.9

Upgrading your Red Hat OpenShift Service on AWS clusters to OpenShift 4.9 requires you to evaluate and migrate your APIs as the latest version of Kubernetes has removed a significant number of APIs.

Before you can upgrade your Red Hat OpenShift Service on AWS clusters, you must update the required tools to the appropriate version.

1.1. REQUIREMENTS FOR UPGRADING TO OPENSHIFT 4.9

You must meet the following requirements before upgrading a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS) from version 4.8 to 4.9.

Prerequisites

- You have installed the latest AWS CLI on your installation host.
- You have installed 1.1.10 or later of the ROSA CLI on your installation host.
- You have installed version 4.9 or later of the OpenShift CLI (**oc**) on your workstation(s) as needed.
- You have the permissions required to update the AWS account-wide roles and policies.
- You have access to the cluster as a user with the **cluster-admin** role.
- You updated the AWS Identity and Access Management (IAM) account-wide roles and policies, including the Operator policies to version 4.9.

1.1.1. Administrator acknowledgment when upgrading to OpenShift 4.9

Red Hat OpenShift Service on AWS 4.9 uses Kubernetes 1.22, which removed a significant number of deprecated **v1beta1** APIs.

Red Hat OpenShift Service on AWS 4.8.14 introduced a requirement that an administrator must provide a manual acknowledgment before the cluster can be upgraded from Red Hat OpenShift Service on AWS 4.8 to 4.9. This is to help prevent issues after upgrading to Red Hat OpenShift Service on AWS 4.9, where APIs that have been removed are still in use by workloads, tools, or other components running on or interacting with the cluster. Administrators must evaluate their cluster for any APIs in use that will be removed and migrate the affected components to use the appropriate new API version. After this is done, the administrator can provide the administrator acknowledgment.

All Red Hat OpenShift Service on AWS 4.8 clusters require this administrator acknowledgment before they can be upgraded to Red Hat OpenShift Service on AWS 4.9.

1.1.2. Removed Kubernetes APIs

Red Hat OpenShift Service on AWS 4.9 uses Kubernetes 1.22, which removed the following deprecated **v1beta1** APIs. You must migrate manifests and API clients to use the **v1** API version. For more information about migrating removed APIs, see the Kubernetes documentation.

Table 1.1. v1beta1 APIs removed from Kubernetes 1.22

Resource	API	Notable changes
APIService	apiregistration.k8s.io/v1beta1	No
CertificateSigningRequest	certificates.k8s.io/v1beta1	Yes
ClusterRole	rbac.authorization.k8s.io/v1beta1	No
ClusterRoleBinding	rbac.authorization.k8s.io/v1beta1	No
CSIDriver	storage.k8s.io/v1beta1	No
CSINode	storage.k8s.io/v1beta1	No
CustomResourceDefinition	apiextensions.k8s.io/v1beta1	Yes
Ingress	extensions/v1beta1	Yes
Ingress	networking.k8s.io/v1beta1	Yes
IngressClass	networking.k8s.io/v1beta1	No
Lease	coordination.k8s.io/v1beta1	No
LocalSubjectAccessReview	authorization.k8s.io/v1beta1	Yes
MutatingWebhookConfiguration	admissionregistration.k8s.io/v1beta1	Yes
PriorityClass	scheduling.k8s.io/v1beta1	No
Role	rbac.authorization.k8s.io/v1beta1	No
RoleBinding	rbac.authorization.k8s.io/v1beta1	No
SelfSubjectAccessReview	authorization.k8s.io/v1beta1	Yes
StorageClass	storage.k8s.io/v1beta1	No
SubjectAccessReview	authorization.k8s.io/v1beta1	Yes
TokenReview	authentication.k8s.io/v1beta1	No
ValidatingWebhookConfiguration	admissionregistration.k8s.io/v1beta1	Yes
VolumeAttachment	storage.k8s.io/v1beta1	No

1.2. EVALUATING YOUR CLUSTER FOR REMOVED APIS

There are several methods to help administrators identify where APIs that will be removed are in use. However, Red Hat OpenShift Service on AWS cannot identify all instances, especially workloads that are idle or external tools that are used. It is the responsibility of the administrator to properly evaluate all workloads and other integrations for instances of removed APIs.

1.2.1. Reviewing alerts to identify uses of removed APIs

The **APIRemovedInNextReleaseInUse** alert tells you that there are removed APIs in use on your cluster. If this alert is firing in your cluster, review the alert; take action to clear the alert by migrating manifests and API clients to use the new API version. You can use the **APIRequestCount** API to get more information about which APIs are in use and which workloads are using removed APIs.

1.2.2. Using APIRequestCount to identify uses of removed APIs

You can use the **APIRequestCount** API to track API requests and review if any of them are using one of the removed APIs.

Prerequisites

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

Procedure

- Run the following command and examine the **REMOVEDINRELEASE** column of the output to identify the removed APIs that are currently in use:

```
$ oc get apirequestcounts
```

Example output

```
NAME                                      REMOVEDINRELEASE   REQUESTSINCURRENTHOUR
REQUESTSINLAST24H
cloudcredentials.v1.operator.openshift.io                32                  111
ingresses.v1.networking.k8s.io                           28                  110
ingresses.v1beta1.extensions             1.22            16                  66
ingresses.v1beta1.networking.k8s.io      1.22            0                   1
installplans.v1alpha1.operators.coreos.com               93                  167
...
```

NOTE

You can safely ignore the following entries that appear in the results:

- **system:serviceaccount:kube-system:generic-garbage-collector** appears in the results because it walks through all registered APIs searching for resources to remove.
- **system:kube-controller-manager** appears in the results because it walks through all resources to count them while enforcing quotas.

You can also use **-o jsonpath** to filter the results:

```
$ oc get apirequestcounts -o jsonpath='{range .items[?(@.status.removedInRelease!="")]}
{.status.removedInRelease}{"\t"}{.metadata.name}{"\n"}{end}'
```

Example output

```
1.22    certificatesigningrequests.v1beta1.certificates.k8s.io
1.22    ingresses.v1beta1.extensions
1.22    ingresses.v1beta1.networking.k8s.io
```

1.2.3. Using APIRequestCount to identify which workloads are using the removed APIs

You can examine the **APIRequestCount** resource for a given API version to help identify which workloads are using the API.

Prerequisites

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

Procedure

- Run the following command and examine the **username** and **userAgent** fields to help identify the workloads that are using the API:

```
$ oc get apirequestcounts <resource>.<version>.<group> -o yaml
```

For example:

```
$ oc get apirequestcounts ingresses.v1beta1.networking.k8s.io -o yaml
```

You can also use **-o jsonpath** to extract the **username** values from an **APIRequestCount** resource:

```
$ oc get apirequestcounts ingresses.v1beta1.networking.k8s.io -o jsonpath='{range
..username}{$}{"\n"}{end}' | sort | uniq
```

Example output

```
user1
user2
app:serviceaccount:delta
```

1.3. MIGRATING INSTANCES OF REMOVED APIS

For information on how to migrate removed Kubernetes APIs, see the Deprecated API Migration Guide in the Kubernetes documentation.

CHAPTER 2. UPGRADING ROSA CLUSTERS WITH STS

2.1. LIFE CYCLE POLICIES AND PLANNING

To plan an upgrade, review the Red Hat OpenShift Service on AWS update life cycle . The life cycle page includes release definitions, support and upgrade requirements, installation policy information and life cycle dates.

Upgrades are manually initiated or automatically scheduled. Red Hat Site Reliability Engineers (SREs) monitor upgrade progress and remedy any issues encountered.

2.2. UPGRADING A ROSA CLUSTER THAT USES STS

There are two methods to upgrade Red Hat OpenShift Service on AWS (ROSA) clusters that uses the AWS Security Token Service (STS):

- Individual upgrades through the **rosa** CLI
- Individual upgrades through the OpenShift Cluster Manager console

NOTE

For steps to upgrade a ROSA cluster that does not use the AWS Security Token Service (STS), see Upgrading ROSA clusters.

2.2.1. Upgrading with the rosa CLI

You can upgrade a Red Hat OpenShift Service on AWS cluster that uses the AWS Security Token Service (STS) manually by using the **rosa** CLI.

This method schedules the cluster for an immediate upgrade, if a more recent version is available.

Prerequisites

- You have installed and configured the latest ROSA CLI on your installation host.
- If you are upgrading your cluster from 4.7 to 4.8, you have upgraded the AWS Identity and Access Management (IAM) account-wide roles and policies to version 4.8. You have also updated the **cloudcredential.openshift.io/upgradeable-to** annotation in the **CloudCredential** custom resource.

Procedure

1. To verify the current version of your cluster, enter the following command:

```
$ rosa describe cluster --cluster=<cluster_name|cluster_id> 1
```

1 Replace **<cluster_name|cluster_id>** with the cluster name or the ID of the cluster.

2. To verify that an upgrade is available, enter the following command:

```
$ rosa list upgrade --cluster=<cluster_name|cluster_id>
```

The command returns a list of versions to which the cluster can be upgraded, including a recommended version.

3. To upgrade a cluster to the latest available version, enter the following command:

```
$ rosa upgrade cluster --cluster=<cluster_name|cluster_id>
```

The cluster is scheduled for an immediate upgrade. This action can take an hour or longer, depending on your workload configuration, such as pod disruption budgets.

You will receive an email when the upgrade is complete. You can also check the status by running **rosa describe cluster** again from the **rosa** CLI or view the status in OpenShift Cluster Manager console.

2.2.2. Scheduling individual upgrades through the OpenShift Cluster Manager console

You can schedule upgrades for a Red Hat OpenShift Service on AWS cluster that uses the AWS Security Token Service (STS) manually one time by using OpenShift Cluster Manager console.

Prerequisites

- If you are upgrading your cluster from 4.7 to 4.8, you have upgraded the AWS Identity and Access Management (IAM) account-wide roles and policies to version 4.8. You have also updated the **cloudcredential.openshift.io/upgradeable-to** annotation in the **CloudCredential** custom resource. For more information, see *Preparing an upgrade from 4.7 to 4.8* .

Procedure

1. Log in to OpenShift Cluster Manager Hybrid Cloud Console .
2. Select a cluster to upgrade.
3. Click the **Settings** tab.
4. In the **Update strategy** pane, select **Individual Updates**.
5. Select the version you want to upgrade your cluster to. Recommended cluster upgrades appear in the UI.
6. If you select an update version that requires approval, provide an administrator's acknowledgment and click **Approve and continue**.
   For information about administrator acknowledgment, see Administrator acknowledgment when upgrading to OpenShift 4.9.
7. In the **Node draining** pane, select a grace period interval from the list. The grace period enables the nodes to gracefully drain before forcing the pod eviction. The default is **1 hour**.
8. In the **Update strategy** pane, click **Save** to apply your update strategy.
9. In the **Update status** pane, review the **Update available** information and click **Update**.

NOTE

The **Update** button is enabled only when an upgrade is available.

10. In the **Select version** dialog, choose a target upgrade version and click **Next**.

11. In the **Schedule update** dialog, schedule your cluster upgrade.

    - To upgrade within an hour, select **Update now** and click **Next**.

    - To upgrade at a later time, select **Schedule a different time** and set a time and date for your upgrade. Click **Next** to proceed to the confirmation dialog.

12. After reviewing the version and schedule summary, select **Confirm update**.
    The cluster is scheduled for an upgrade to the target version. This action can take an hour or longer, depending on the selected upgrade schedule and your workload configuration, such as pod disruption budgets.

    The status is displayed in the **Update status** pane.

CHAPTER 3. UPGRADING ROSA CLUSTERS

3.1. LIFE CYCLE POLICIES AND PLANNING

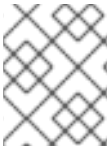
To plan an upgrade, review the Red Hat OpenShift Service on AWS update life cycle . The life cycle page includes release definitions, support and upgrade requirements, installation policy information and life cycle dates.

Upgrades are manually initiated or automatically scheduled. Red Hat Site Reliability Engineers (SREs) monitor upgrade progress and remedy any issues encountered.

3.2. UPGRADING A ROSA CLUSTER

There are three methods to upgrade Red Hat OpenShift Service on AWS (ROSA) clusters:

- Individual upgrades through the **rosa** CLI
- Individual upgrades through the OpenShift Cluster Manager Hybrid Cloud Console console
- Recurring upgrades through the OpenShift Cluster Manager Hybrid Cloud Console console



NOTE

For steps to upgrade a ROSA cluster that uses the AWS Security Token Service (STS), see Upgrading ROSA clusters with STS.

3.2.1. Upgrading with the rosa CLI

You can upgrade a Red Hat OpenShift Service on AWS cluster manually by using the **rosa** CLI.

This method schedules the cluster for an immediate upgrade, if a more recent version is available.

Prerequisites

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

Procedure

1. To verify the current version of your cluster, enter the following command:

```
$ rosa describe cluster --cluster=<cluster_name|cluster_id> 1
```

1 Replace **<cluster_name|cluster_id>** with the cluster name or the ID of the cluster.

2. To verify that an upgrade is available, enter the following command:

```
$ rosa list upgrade --cluster=<cluster_name|cluster_id>
```

The command returns a list of versions to which the cluster can be upgraded, including a recommended version.

3. To upgrade a cluster to the latest available version, enter the following command:

```
$ rosa upgrade cluster --cluster=<cluster_name|cluster_id>
```

The cluster is scheduled for an immediate upgrade. This action can take an hour or longer, depending on your workload configuration, such as pod disruption budgets.

You will receive an email when the upgrade is complete. You can also check the status by running **rosa describe cluster** again from the **rosa** CLI or view the status in OpenShift Cluster Manager console.

3.2.2. Scheduling individual upgrades through the OpenShift Cluster Manager console

You can schedule upgrades for a Red Hat OpenShift Service on AWS cluster manually one time by using OpenShift Cluster Manager console.

Procedure

1. Log in to OpenShift Cluster Manager Hybrid Cloud Console .
2. Select a cluster to upgrade.
3. Click the **Settings** tab.
4. In the **Update strategy** pane, select **Individual Updates**.
5. Select the version you want to upgrade your cluster to. Recommended cluster upgrades appear in the UI.
6. If you select an update version that requires approval, provide an administrator's acknowledgment and click **Approve and continue**.
   For information about administrator acknowledgment, see Administrator acknowledgment when upgrading to OpenShift 4.9.
7. In the **Node draining** pane, select a grace period interval from the list. The grace period enables the nodes to gracefully drain before forcing the pod eviction. The default is **1 hour**.
8. In the **Update strategy** pane, click **Save** to apply your update strategy.
9. In the **Update status** pane, review the **Update available** information and click **Update**.

NOTE

The **Update** button is enabled only when an upgrade is available.

10. In the **Select version** dialog, choose a target upgrade version and click **Next**.
11. In the **Schedule update** dialog, schedule your cluster upgrade.
    - To upgrade within an hour, select **Update now** and click **Next**.
    - To upgrade at a later time, select **Schedule a different time** and set a time and date for your upgrade. Click **Next** to proceed to the confirmation dialog.
12. After reviewing the version and schedule summary, select **Confirm update**.

The cluster is scheduled for an upgrade to the target version. This action can take an hour or longer, depending on the selected upgrade schedule and your workload configuration, such as pod disruption budgets.

The status is displayed in the **Update status** pane.

3.2.3. Scheduling recurring upgrades for your cluster

You can schedule recurring, automatic upgrades for z-stream patch versions for your Red Hat OpenShift Service on AWS cluster through the OpenShift Cluster Manager console.

Procedure

1. Log in to OpenShift Cluster Manager Hybrid Cloud Console .
2. Select a cluster to upgrade.
3. Click the **Settings** tab.
4. In the **Update strategy** pane, select **Recurring updates**.
5. Select a preferred day of the week and start time for the upgrade, when updates are available.
6. Provide an administrator's acknowledgment and click **Approve and continue**. OpenShift Cluster Manager does not start scheduled y-stream updates for minor versions without receiving an administrator's acknowledgment.
   For information about administrator acknowledgment, see Administrator acknowledgment when upgrading to OpenShift 4.9.
7. In the **Node draining** pane, select a grace period interval from the list. The grace period enables the nodes to gracefully drain before forcing the pod eviction. The default is **1 hour**.
8. In the **Update strategy** pane, click **Save** to apply your update strategy.
   When upgrades are available, they are automatically applied to the cluster on the preferred day of the week and start time.