Red Hat Advanced Cluster Management for Kubernetes 2.0

Manage cluster

Manage cluster

Last Updated: 2021-06-10

Red Hat Advanced Cluster Management for Kubernetes 2.0 Manage cluster

Manage cluster

Legal Notice

Abstract

Manage cluster in Red Hat Advanced Cluster Management for Kubernetes

Table of Contents

CHAPTER 1. MANAGING YOUR CLUSTERS WITH RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES 4

CHAPTER 2. SUPPORTED CLOUDS 5

2.1. SUPPORTED HUB CLUSTER PROVIDER 5

2.2. SUPPORTED MANAGED CLUSTER PROVIDERS 5

2.3. CONFIGURING KUBECTL 5

CHAPTER 3. RESIZING A CLUSTER 6

3.1. AMAZON WEB SERVICES 6

3.2. GOOGLE CLOUD PLATFORM 6

3.3. MICROSOFT AZURE 6

3.4. BARE METAL CLUSTER 7

3.5. IBM KUBERNETES SERVICE 7

CHAPTER 4. RELEASE IMAGES 8

4.1. SYNCHRONIZING AVAILABLE RELEASE IMAGES 8

4.1.1. Maintaining a custom list of release images when connected 10

4.1.2. Maintaining a custom list of release images while disconnected 11

CHAPTER 5. CREATING AND MODIFYING BARE METAL ASSETS 12

5.1. PREREQUISITES 12

5.2. CREATING A BARE METAL ASSET WITH THE CONSOLE 13

5.3. MODIFYING A BARE METAL ASSET 14

5.4. REMOVING A BARE METAL ASSET 14

CHAPTER 6. CREATING A PROVIDER CONNECTION 15

6.1. CREATING A PROVIDER CONNECTION FOR AMAZON WEB SERVICES 15

6.1.1. Prerequisites 15

6.1.2. Creating a provider connection by using the console 15

6.1.3. Deleting your provider connection 16

6.2. CREATING A PROVIDER CONNECTION FOR MICROSOFT AZURE 16

6.2.1. Prerequisites 17

6.2.2. Creating a provider connection by using the console 17

6.2.3. Deleting your provider connection 18

6.3. CREATING A PROVIDER CONNECTION FOR GOOGLE CLOUD PLATFORM 18

6.3.1. Prerequisites 19

6.3.2. Creating a provider connection by using the console 19

6.3.3. Deleting your provider connection 20

6.4. CREATING A PROVIDER CONNECTION FOR BARE METAL 20

6.4.1. Prerequisites 20

6.4.2. Creating a provider connection by using the console 22

6.4.3. Deleting your provider connection 23

CHAPTER 7. CREATING A CLUSTER WITH RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES 24

7.1. CREATING A CLUSTER ON AMAZON WEB SERVICES 24

7.1.1. Prerequisites 24

7.1.2. Creating your cluster with the Red Hat Advanced Cluster Management for Kubernetes console 24

7.1.3. Accessing your cluster 26

7.1.4. Removing a cluster from management 26

7.2. CREATING A CLUSTER ON MICROSOFT AZURE 27

7.2.1. Prerequisites 27

7.2.2. Creating your cluster with the Red Hat Advanced Cluster Management for Kubernetes console 27
7.2.3. Accessing your cluster 28
7.2.4. Removing a cluster from management 29
7.3. CREATING A CLUSTER ON GOOGLE CLOUD PLATFORM 29
7.3.1. Prerequisites 29
7.3.2. Creating your cluster with the Red Hat Advanced Cluster Management for Kubernetes console 30
7.3.3. Accessing your cluster 31
7.3.4. Removing a cluster from management 32
7.4. CREATING A CLUSTER ON BARE METAL 32
7.4.1. Prerequisites 32
7.4.2. Creating your cluster with the Red Hat Advanced Cluster Management for Kubernetes console 34
7.4.3. Accessing your cluster 35
7.4.4. Removing a cluster from management 35

CHAPTER 8. IMPORTING A TARGET MANAGED CLUSTER TO THE HUB CLUSTER **37**
8.1. IMPORTING AN EXISTING CLUSTER WITH THE CONSOLE 37
8.1.1. Prerequisites 37
8.1.2. Importing a cluster 37
8.1.2.1. YAML parameters and descriptions 39
8.1.3. Removing an imported cluster 40
8.2. IMPORTING A MANAGED CLUSTER WITH THE CLI 40
8.2.1. Prerequisites 40
8.2.2. Supported architecture 41
8.2.3. Prepare for import 41
8.2.4. Importing the klusterlet 42
8.3. MODIFYING THE KLUSTERLET ADDONS SETTINGS OF YOUR CLUSTER 43
8.3.1. Modify using the console on the hub cluster 43
8.3.2. Modify using the command line on the hub cluster 44

CHAPTER 9. UPGRADING YOUR CLUSTER **45**
9.1. UPGRADING DISCONNECTED CLUSTERS 45
9.1.1. Prerequisites 46
9.1.2. Prepare your disconnected mirror registry 46
9.1.3. Deploy the operator for OpenShift Update Service 46
9.1.4. Build the graph data init container 47
9.1.5. Configure certificate for the mirrored registry 48
9.1.6. Deploy the OpenShift Update Service instance 48
9.1.7. Deploy a policy to override the default registry (optional) 49
9.1.8. Deploy a policy to deploy a disconnected catalog source 51
9.1.9. Deploy a policy to change the managed cluster parameter 53
9.1.10. Viewing available upgrades 56
9.1.11. Upgrading the cluster 56

CHAPTER 1. MANAGING YOUR CLUSTERS WITH RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES

Learn how to create, import, and manage clusters across cloud providers by using both the Red Hat Advanced Cluster Management for Kubernetes console.

Learn how to manage clusters across cloud providers in the following topics:

- Supported clouds
- Resizing a cluster
- Creating a provider connection
- Creating a cluster with Red Hat Advanced Cluster Management for Kubernetes
- Importing a target managed cluster to the hub cluster
- Upgrading your cluster

CHAPTER 2. SUPPORTED CLOUDS

Learn about the cloud providers that are available with Red Hat Advanced Cluster Management for Kubernetes. Also, find the documented managed providers that are available.

- Supported hub cluster provider
- Supported managed cluster providers
- Configuring kubectl

Best practice: For managed cluster providers, use the latest version of Kubernetes.

2.1. SUPPORTED HUB CLUSTER PROVIDER

Red Hat OpenShift Container Platform 4.3.18 or later, 4.4.4 or later, and 4.5.2 or later are supported for the hub cluster.

- See OpenShift on Amazon Web Services.

2.2. SUPPORTED MANAGED CLUSTER PROVIDERS

Red Hat OpenShift Container Platform 3.11.200 or later, 4.3.18 or later, 4.4.4 or later, and 4.5.2 or later are supported for the managed clusters.

See the available managed cluster options and documentation:

- See OpenShift on Amazon Web Services.
- See Red Hat OpenShift on IBM Cloud .
- See Red Hat OpenShift Kubernetes Engine .
- See Getting started with IBM Cloud Kubernetes Service .
- See Google Kubernetes Engine .
- See Azure Kubernetes Service.
- See Amazon Elastic Container Service for Kubernetes .

2.3. CONFIGURING KUBECTL

From vendor documentation previously listed, you might need to learn how configure your **kubectl**. You must have **kubectl** installed when you import a managed cluster to a hub cluster. See Importing a target managed cluster to the hub cluster for details.

CHAPTER 3. RESIZING A CLUSTER

You can customize your managed cluster specifications, such as virtual machine sizes and number of nodes. See the following list of recommended settings for each available provider, but also see the documentation for more specific information:

3.1. AMAZON WEB SERVICES

You can change the number of nodes of a Red Hat OpenShift Container Platform cluster that was created in an Amazon Web Services environment by modifying the **MachineSet** parameters on the hub cluster.

Remember: Because Red Hat Advanced Cluster Mangement for Kubernetes uses Hive for OpenShift to determine the number of nodes in the cluster, you must change the **MachineSet** parameter to change the number of nodes. If you just remove or add a node without changing the **MachineSets** parameter, nodes are added or removed to match the current value of that parameter.

See Recommended cluster scaling practices and Manually scaling a MachineSet in the OpenShift Container Platform documentation that applies to your version.

Tip: If you created the cluster by using the Red Hat Advanced Cluster Management for Kubernetes console, then it is an OpenShift Container Platform cluster.

If you are changing the number of nodes of an Amazon EKS cluster that you imported, see Cluster autoscaler for information about scaling the cluster.

3.2. GOOGLE CLOUD PLATFORM

You can change the number of nodes of a Red Hat OpenShift Container Platform cluster that was created in an Google Cloud Platform environment by modifying the **MachineSet** parameters on the hub cluster.

Remember: Because Red Hat Advanced Cluster Mangement for Kubernetes uses Hive for OpenShift to determine the number of nodes in the cluster, you must change the **MachineSet** parameter to change the number of nodes. If you just remove or add a node without changing the **MachineSets** parameter, nodes are added or removed to match the current value of that parameter.

See Recommended cluster scaling practices and Manually scaling a MachineSet in the OpenShift Container Platform documentation that applies to your version for more information about scaling your cluster. **Tip:** If you created the cluster by using Red Hat Advanced Cluster Management, then it is an OpenShift Container Platform cluster.

If you are changing the number of nodes of a Google Kubernetes Engine cluster that you imported, see Resizing a cluster for information about scaling the cluster.

3.3. MICROSOFT AZURE

You can change the number of nodes of a Red Hat OpenShift Container Platform cluster that was created in a Microsoft Azure environment by modifying the **MachineSet** parameters on the hub cluster.

Remember: Because Red Hat Advanced Cluster Mangement for Kubernetes uses Hive for OpenShift to determine the number of nodes in the cluster, you must change the **MachineSet** parameter to change the number of nodes. If you just remove or add a node without changing the **MachineSets** parameter, nodes are added or removed to match the current value of that parameter.

See Recommended cluster scaling practices and Manually scaling a MachineSet in the OpenShift Container Platform documentation that applies to your version. **Tip:** If you created the cluster by using Red Hat Advanced Cluster Management for Kubernetes, then it is an OpenShift Container Platform cluster.

If you are changing the number of nodes of an Azure Kubernetes Services cluster that you imported, see Scaling a cluster for information about scaling the cluster.

3.4. BARE METAL CLUSTER

You can change the number of nodes of a Red Hat OpenShift Container Platform cluster that was created in a bare metal environment by modifying the **MachineSet** parameters on the hub cluster.

Remember: Because Red Hat Advanced Cluster Mangement for Kubernetes uses Hive for OpenShift to determine the number of nodes in the cluster, you must change the **MachineSet** parameter to change the number of nodes. If you just remove or add a node without changing the **MachineSets** parameter, nodes are added or removed to match the current value of that parameter.

See Recommended cluster scaling practices and Manually scaling a MachineSet in the OpenShift Container Platform documentation that applies to your version. **Tip:** If you created the cluster by using Red Hat Advanced Cluster Management for Kubernetes, then it is an OpenShift Container Platform cluster.

If you are changing the number of nodes of a bare metal cluster that you imported, see Installing a cluster on bare metal with network customizations for information about scaling the cluster.

Note: Bare metal clusters are only supported when the hub cluster is OpenShift Container Platform version 4.5, and later.

3.5. IBM KUBERNETES SERVICE

If you are changing the number of nodes of an IBM Kubernetes Service cluster that you imported, see Adding worker nodes and zones to clusters for information about scaling the cluster.

Remember: Because Red Hat Advanced Cluster Mangement for Kubernetes uses Hive for OpenShift to determine the number of nodes in the cluster, you must change the **MachineSet** parameter to change the number of nodes. If you just remove or add a node without changing the **MachineSets** parameter, nodes are added or removed to match the current value of that parameter.

CHAPTER 4. RELEASE IMAGES

When you create a cluster on a provider by using the Red Hat Advanced Cluster Management for Kubernetes, you must specify a release image to use for the new cluster. The release image specifies which version of Red Hat OpenShift Container Platform is used to build the cluster.

The files that reference the release images are **yaml** files that are maintained in the **acm-hive-openshift-releases** GitHub repository. Red Hat Advanced Cluster Management for Kubernetes uses those files to create the list of the available release images in the console. The repository contains the **clusterImageSets** directory and the **subscription** directory, which are the directories that you use when working with the release images.

The **clusterImageSets** directory contains the following directories:

- Fast - Contains files that reference the latest two versions of the release images for each OpenShift Container Platform version that is supported
- Releases - Contains files that reference all of the release images for each OpenShift Container Platform version that is supported. **Note:** These releases have not all been tested and determined to be stable.
- Stable - Contains files that reference the latest two stable versions of the release images for each OpenShift Container Platform version that is supported. The release images in this folder are tested and verified.

The **subscription** directory contains files that specify where the list of release images is pulled from. The default release images for Red Hat Advanced Cluster Management are provided in a Quay.io directory. They are referenced by the files in the acm-hive-openshift-releases GitHub repository.

4.1. SYNCHRONIZING AVAILABLE RELEASE IMAGES

The release images are updated frequently, so you might want to synchronize the list of release images to ensure that you can select the latest available versions. The release images are available in the acm-hive-openshift-releases GitHub repository.

There are three levels of stability of the release images:

Table 4.1. Stability levels of release images

Category	Description
stable	Fully tested images that are confirmed to install and build clusters correctly.
fast	Partially tested, but likely less stable than a stable version.
candidate	Not tested, but the most current image. Might have some bugs.

Complete the following steps to refresh the list:

1. Clone the acm-hive-openshift-releases GitHub repository.

2. Connect to your Red Hat Advanced Cluster Management for Kubernetes hub cluster by entering the following command:

   ```
   oc apply -k subscription/
   ```

   After about one minute, the latest two **fast** entries are available.

3. To synchronize your list of **stable** release images after you have cloned the **acm-hive-openshift-releases** GitHub repository, enter the following command to update the **stable** images:

   ```
   make subscribe-stable
   ```

   Note: You can only run this **make** command when you are using the Linux or MacOS operating system. If you are using the Windows operating system, enter the following command to update the **stable** images:

   ```
   oc apply -f subscription-stable
   ```

   After running this command, the list of available **stable** release images updates with the currently available images in about one minute.

   - To synchronize and display the fast release images, enter the following command:

     ```
     make subscribe-fast
     ```

     Note: You can only run this **make** command when you are using the Linux or MacOS operating system. If you are using the Windows operating system, enter the following command to updated the **fast** images:

     ```
     oc apply -f subscription/subscription-fast.yaml
     ```

     After running the command, the list of available **stable** and **fast** release images updates with the currently available images in about one minute.

   - To synchronize and display the **candidate** release images, enter the following command:

     ```
     make subscribe-candidate
     ```

     Note: You can only run this **make** command when you are using the Linux or MacOS operating system. If you are using the Windows operating system, enter the following command to updated the **candidate** images:

     ```
     oc apply -f subscription/subscription-candidate.yaml
     ```

     After running the command, the list of available **stable**, **fast**, and **candidate** release images updates with the currently available images in about 1 minute.

4. View the list of currently available release images in the Red Hat Advanced Cluster Management console when you are creating a cluster.

5. You can unsubscribe from any of these channels to stop viewing the updates by entering a command in the following format:

```
oc delete -f subscription/subscription-stable
```

4.1.1. Maintaining a custom list of release images when connected

You might want to ensure that you use the same release image for all of your clusters. To simplify, you can create your own custom list of release images that are available when creating a cluster. Complete the following steps to manage your available release images:

1. Fork the acm-hive-openshift-releases GitHub repository.

2. Update the **./subscription/channel.yaml** file by changing the **spec: pathname** to access your the GitHub name for your forked repository, instead of **open-cluster-management**. This step specifies where the hub cluster retrieves the release images. Your updated content should look similar to the following example:

   ```
   spec:
     type: GitHub
     pathname: https://github.com/<forked_content>/acm-hive-openshift-releases.git
   ```

   Replace *forked_content* with the path to your forked repository.

3. Add the **yaml** files for the images that you want available when you create a cluster by using the Red Hat Advanced Cluster Management for Kubernetes console to the **./clusterImageSets/stable/** *or* **./clusterImageSets/fast/** directory. **Tip:** You can retrieve the available **yaml** files from the main repository by merging changes into your forked repository.

4. Commit and merge your changes to your forked repository.

5. To synchronize your list of stable release images after you have cloned the **acm-hive-openshift-releases** repository, enter the following command to update the stable images:

   ```
   make subscribe-stable
   ```

   Note: You can only run this **make** command when you are using the Linux or MacOS operating system. If you are using the Windows operating system, enter the following commands:

   ```
   oc apply -k subscription/
   oc delete -f subscription/subscription-fast.yaml
   oc apply -f subscription/subscription-stable.yaml
   ```

   After running this command, the list of available stable release images updates with the currently available images in about one minute.

6. By default, only the stable images are listed. To synchronize and display the fast release images, enter the following command:

   ```
   make subscribe-fast
   ```

   Note: You can only run this **make** command when you are using the Linux or MacOS operating system. If you are using the Windows operating system, enter the following commands:

   ```
   oc apply -k subscription/
   oc apply -f subscription/subscription-fast.yaml
   ```

After running this command, the list of available fast release images updates with the currently available images in about 1 minute.

7. By default, Red Hat Advanced Cluster Management pre-loads a few ClusterImageSets. Use the following commands to list what is available and remove the defaults, if desired.

```
oc get clusterImageSets
oc delete clusterImageSet <clusterImageSet_NAME>
```

8. View the list of currently available release images in the Red Hat Advanced Cluster Management console when you are creating a cluster.

4.1.2. Maintaining a custom list of release images while disconnected

In some cases, you need to maintain a custom list of release images when the hub cluster has no Internet connection. You can create your own custom list of release images that are available when creating a cluster. Complete the following steps to manage your available release images while disconnected:

1. While you are on a connected system, navigate to the acm-hive-openshift-releases GitHub repository.

2. Copy the **clusterImageSets** directory to a system that can access the disconnected Red Hat Advanced Cluster Management for Kubernetes hub cluster.

3. Add the **yaml** files for the images that you want available when you create a cluster by using the Red Hat Advanced Cluster Management for Kubernetes console by manually adding the **clusterImageSet** yamls.

4. Create **clusterImageSets** command:

   ```
   oc create -f <clusterImageSet_FILE>
   ```

   After running this command for each resource you want to add, the list of available release images will be available.

5. Alternately you can paste the image url directly in the the create cluster console in Red Hat Advanced Cluster Management. This will create new clusterImageSets if they do not exist.

6. View the list of currently available release images in the Red Hat Advanced Cluster Management console when you are creating a cluster.

CHAPTER 5. CREATING AND MODIFYING BARE METAL ASSETS

Important: The bare metal cluster function is a technology preview,and should not be used in production environments.

Bare metal assets are virtual or physical servers that are configured to run your cloud operations. Red Hat Advanced Cluster Management for Kubernetes connects to a bare metal asset that your administrator creates, and can create clusters on it.

You must create a bare metal asset in Red Hat Advanced Cluster Management for Kubernetes to create a cluster on it. Use the following procedure to create a bare metal asset that can host a cluster that is managed by Red Hat Advanced Cluster Management for Kubernetes.

Note: The bare metal features are only provided as a technology preview. The bare metal options are hidden by feature flags, by default. To view the bare metal options, you must enable the feature flags by completing the instructions in the *Prerequisites* section.

5.1. PREREQUISITES

You need the following prerequisites before creating a bare metal asset:

- A deployed Red Hat Advanced Cluster Management for Kubernetes hub cluster on OpenShift Container Platform version 4.5, or later.
- Access for your Red Hat Advanced Cluster Management for Kubernetes hub cluster to connect to the bare metal asset.
- A configured bare metal asset, and log in credentials with the required permissions to log in and manage it. **Note:** Login credentials for your bare metal asset include the following items for the asset that are provided by your administrator:
  - user name
  - password
  - Baseboard Management Controller Address
  - boot NIC MAC address
- Bare metal feature flags that are enabled to view the bare metal options. The bare metal options are hidden by feature flags by default. Complete the following steps to enable the feature flags:
  a. Start the Red Hat OpenShift Container Platform command line interface.
  b. Set the **featureFlags_baremetal** setting to **true** for the **console-header** container by entering the following command:

```
oc patch deploy console-header -n <namespace> -p '{"spec":{"template":{"spec":
{"containers":[{"name":"console-header","env": [{"name":
"featureFlags_baremetal","value":"true"}]}]}}}}'
```

  Replace <namespace> with your Red Hat Advanced Cluster Management project namespace.

After the update, your **consoleui** CRD should look like the following example:

```
spec:
  ...
  template:
    ...
    spec:
      ...
      containers:
      - env:                              # Search for env:
        - name: featureFlags_baremetal
          value: "true"
        ...
```

c. Set the **featureFlags_baremetal** value to **true** for the **hmc-ui** container:

```
oc patch -n <namespace> $(oc get deploy -o name | grep consoleui) -p '{"spec":
{"template":{"spec":{"containers":[{"name":"hcm-ui","env": [{"name":
"featureFlags_baremetal","value":"true"}]}]}}}}'
```

Replace <namespace> with your Red Hat Advanced Cluster Management project namespace.

Your update should look like the following example:

```
spec:
  ...
  template:
    ...
    spec:
      ...
      containers:
      - env:                              # Search for env:
        - name: featureFlags_baremetal
          value: "true"
        ...
```

d. Make sure the **console-chart-...-consoleui...** and **console-header-...** pods are running:

```
oc -n open-cluster-management get pods
```

e. When the pods are running again, log out of the Red Hat Advanced Cluster Management for Kubernetes console and log back in. The bare metal options are now included in the console.

5.2. CREATING A BARE METAL ASSET WITH THE CONSOLE

To create a bare metal asset using the Red Hat Advanced Cluster Management for Kubernetes console, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Bare metal assets**.

2. On the *Bare metal assets* page, Click **Create bare metal asset**

3. Enter a name for your asset that identifies it when you create a cluster.
   Tip: You can view the **yaml** content updates as you enter the information in the console by setting the *YAML* switch to **ON**.

4. Enter the namespace where you want to create the bare metal asset.
   Note: The bare metal asset, managed bare metal cluster, and its related secret must be in the same namespace.

   Users who have access to this namespace can associate this asset to the cluster when creating a cluster.

5. Enter the Baseboard Management Conroller address. This is the controller that enables communication with the host. The following protocols are supported:

   - IPMI, see IPMI 2.0 Specification for more information.

   - iDRAC, see Support for Integrated Dell Remote Access Controller 9 (iDRAC9) for more information.

   - iRMC, see Data Sheet: FUJITSU Software ServerView Suite integrated Remote Management Controller - iRMC S5 for more information.

   - Redfish, see Redfish specification for more information.

6. Enter the user name and password for the bare metal asset.

7. Add the boot NIC MAC address for the bare metal asset. This is the MAC address of the host’s network-connected NIC that is used to provision the host on the bare metal asset.

You can continue with Creating a cluster on bare metal .

5.3. MODIFYING A BARE METAL ASSET

If you need to modify the settings for a bare metal asset, complete the following steps:

1. In the Red Hat Advanced Cluster Management for Kubernetes console navigation, select: **Automate infrastructure** > **Bare metal assets**.

2. Select the options menu for the asset that you want to modify in the table.

3. Select **Modify**.

5.4. REMOVING A BARE METAL ASSET

When a bare metal asset is no longer used for any of the clusters, you can remove it from the list of available bare metal assets. Removing unused assets both simplifies your list of available assets, and prevents the accidental selection of that asset.

To remove a bare metal asset, complete the following steps:

1. In the Red Hat Advanced Cluster Management for Kubernetes console navigation, select: **Automate infrastructure** > **Bare metal assets**.

2. Select the options menu for the asset that you want to remove in the table.

3. Select **Delete**.

CHAPTER 6. CREATING A PROVIDER CONNECTION

A *provider connection* is required to create a Red Hat OpenShift Container Platform cluster on a cloud service provider with Red Hat Advanced Cluster Management for Kubernetes.

The provider connection stores the access credentials and configuration information for a provider. Each provider account requires its own provider connection, as does each domain on a single provider.

The following files detail the information that is required for creating a connection document for each supported provider:

- Creating a provider connection for Amazon Web Services
- Creating a provider connection for Microsoft Azure
- Creating a provider connection for Google Cloud Platform
- Creating a provider connection for bare metal

6.1. CREATING A PROVIDER CONNECTION FOR AMAZON WEB SERVICES

You need a provider connection to use Red Hat Advanced Cluster Management for Kubernetes console to deploy and manage an OpenShift cluster on Amazon Web Services (AWS).

NOTE

This procedure must be done before you can create a cluster with Red Hat Advanced Cluster Management for Kubernetes.

6.1.1. Prerequisites

You must have the following prerequisites before creating a provider connection:

- A deployed Red Hat Advanced Cluster Management for Kubernetes hub cluster
- Internet access for your Red Hat Advanced Cluster Management for Kubernetes hub cluster so it can create the Kubernetes cluster on Amazon Web Services
- Amazon Web Services (AWS) login credentials, which include access key ID and secret access key. See Understanding and getting your security credentials .
- Account permissions that allow installing clusters on AWS. See Configuring an AWS account for instructions on how to configure.

6.1.2. Creating a provider connection by using the console

To create a provider connection from the Red Hat Advanced Cluster Management for Kubernetes console, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. On the *Clusters* page, select the *Provider connections* tab.
   Existing provider connections are displayed.

3. Select **Add a connection**.

4. Select **Amazon Web Services** as your provider.

5. Add a name for your provider connection.

6. Select a namespace for your provider connection from the list.

   TIP

   Create a namespace specifically to host your provider connections, both for convenience and added security.

7. You can optionally add a *Base DNS domain* for your provider connection. If you add the base DNS domain to the provider connection, it is automatically populated in the correct field when you create a cluster with this provider connection.

8. Add your *AWS Access Key ID* for your Amazon Web Services account. Log in to AWS to find the ID.

9. Add your *AWS Secret Access Key ID* .

10. Enter your *Red Hat OpenShift Pull Secret* . You can download your pull secret from Pull secret.

11. Add your *SSH Private Key* and *SSH Public Key*, which allows you to connect to the cluster. You can use an existing key pair, or create a new one with key generation program. See Generating an SSH private key and adding it to the agent for more information about how to generate a key.

12. Click **Create**. When you create the provider connection, it is added to the list of provider connections.

You can create a cluster that uses this provider connection by completing the steps in Creating a cluster on Amazon Web Services.

6.1.3. Deleting your provider connection

When you are no longer managing a cluster that is using a provider connection, delete the provider connection to protect the information in the provider connection.

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.

2. Select **Provider connections**.

3. Select the options menu beside the provider connection that you want to delete.

4. Select **Delete connection**.

6.2. CREATING A PROVIDER CONNECTION FOR MICROSOFT AZURE

You need a provider connection to use Red Hat Advanced Cluster Management for Kubernetes console to create and manage a Red Hat OpenShift Container Platform cluster on Microsoft Azure.

NOTE

This procedure is a prerequisite for creating a cluster with Red Hat Advanced Cluster Management for Kubernetes.

6.2.1. Prerequisites

You must have the following prerequisites before creating a provider connection:

- A deployed Red Hat Advanced Cluster Management for Kubernetes hub cluster
- Internet access for your Red Hat Advanced Cluster Management for Kubernetes hub cluster so that it can create the Kubernetes cluster on Azure
- Azure login credentials, which include your Base Domain Resource Group and Azure Service Principal JSON. See azure.microsoft.com.
- Account permissions that allow installing clusters on Azure. See How to configure Cloud Services and Configuring an Azure account for more information.

6.2.2. Creating a provider connection by using the console

To create a provider connection from the Red Hat Advanced Cluster Management for Kubernetes console, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. On the *Clusters* page, select the *Provider connections* tab.
   Existing provider connections are displayed.
3. Select **Add a connection**.
4. Select **Microsoft Azure** as your provider.
5. Add a name for your provider connection.
6. Select a namespace for your provider connection from the list.

   TIP

   You can create a namespace specifically to host your provider connections, both for convenience and added security.

7. You can optionally add a *Base DNS domain* for your provider connection. If you add the base DNS domain to the provider connection, it is automatically populated in the correct field when you create a cluster with this provider connection.
8. Add your *Base Domain Resource Group Name* for your Azure account. This entry is the resource name that you created with your Azure account. You can find your Base Domain Resource Group Name by selecting **Home** > **DNS Zones** in the Azure interface. Your Base Domain Resource Group name is in the *Resource Group* column of the entry that contains the Base DNS domain that applies to your account.
9. Add your *Client ID*. This value is generated as the **appId** property when you create a service principal with the following command:

   ■

```
az ad sp create-for-rbac --role Contributor --name <service_principal>
```

Replace *service_principal* with the name of your service principal.

10. Add your *Client Secret*. This value is generated as the **password** property when you create a service principal with the following command:

    ```
    az ad sp create-for-rbac --role Contributor --name <service_principal>
    ```

    Replace *service_principal* with the name of your service principal.

11. Add your *Subscription ID*. This value is the **id** property in the output of the following command:

    ```
    az account show
    ```

12. Add your *Tenant ID*. This value is the **tenantId** property in the output of the following command:

    ```
    az account show
    ```

13. Enter your *Red Hat OpenShift Pull Secret* . You can download your pull secret from Pull secret.

14. Add your *SSH Private Key* and *SSH Public Key* to use to connect to the cluster. You can use an existing key pair, or create a new pair using a key generation program. See Generating an SSH private key and adding it to the agent for more information about how to generate a key.

15. Click **Create**. When you create the provider connection, it is added to the list of provider connections.

You can create a cluster that uses this provider connection by completing the steps in Creating a cluster on Microsoft Azure.

6.2.3. Deleting your provider connection

When you are no longer managing a cluster that is using a provider connection, delete the provider connection to protect the information in the provider connection.

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.

2. Select **Provider connections**.

3. Select the options menu for the provider connection that you want to delete.

4. Select **Delete connection**.

6.3. CREATING A PROVIDER CONNECTION FOR GOOGLE CLOUD PLATFORM

You need a provider connection to use Red Hat Advanced Cluster Management for Kubernetes console to create and manage a Red Hat OpenShift Container Platform cluster on Google Cloud Platform (GCP).

NOTE

This procedure is a prerequisite for creating a cluster with Red Hat Advanced Cluster Management for Kubernetes.

6.3.1. Prerequisites

You must have the following prerequisites before creating a provider connection:

- A deployed Red Hat Advanced Cluster Management for Kubernetes hub cluster
- Internet access for your Red Hat Advanced Cluster Management for Kubernetes hub cluster so it can create the Kubernetes cluster on GCP
- GCP login credentials, which include user Google Cloud Platform Project ID and Google Cloud Platform service account JSON key. See Creating and managing projects.
- Account permissions that allow installing clusters on GCP. See Configuring a GCP project for instructions on how to configure an account.

6.3.2. Creating a provider connection by using the console

To create a provider connection from the Red Hat Advanced Cluster Management for Kubernetes console, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. On the Clusters page, select the *Provider connections* tab.
   Existing provider connections are displayed.
3. Select **Add a connection**.
4. Select **Google Cloud Platform** as your provider.
5. Add a name for your provider connection.
6. Select a namespace for your provider connection from the list.

   TIP

   Create a namespace specifically to host your provider connections, for both convenience and security.

7. You can optionally add a *Base DNS domain* for your provider connection. If you add the base DNS domain to the provider connection, it is automatically populated in the correct field when you create a cluster with this provider connection.
8. Add your *Google Cloud Platform Project ID* for your GCP account. Log in to GCP to retrieve your settings.
9. Add your *Google Cloud Platform service account JSON key* . Complete the following steps to create one with the correct permissions:
   a. In the GCP main menu, select **IAM & Admin**and start the **Service Accounts applet**
   b. Select **Create Service Account**

c. Provide the *Name*, *Service account ID*, and *Description* of your service account.

d. Select **Create** to create the service account.

e. Select a role of **Owner**, and click **Continue**.

f. Click **Create Key**

g. Select **JSON**, and click **Create**.

h. Save the resulting file to your computer.

i. Provide the contents for the *Google Cloud Platform service account JSON key* .

10. Enter your *Red Hat OpenShift Pull Secret*. You can download your pull secret from Pull secret.

11. Add your *SSH Private Key* and *SSH Public Key* so you can access the cluster. You can use an existing key pair, or create a new pair using a key generation program. See Generating an SSH private key and adding it to the agent for more information about how to generate a key.

12. Click **Create**. When you create the provider connection, it is added to the list of provider connections.

You can use this connection when you create a cluster by completing the steps in Creating a cluster on Google Cloud Platform.

6.3.3. Deleting your provider connection

When you are no longer managing a cluster that is using a provider connection, delete the provider connection to protect the information in the provider connection.

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. Select **Provider connections**.
3. Select the options menu beside the provider connection that you want to delete.
4. Select **Delete connection**.

6.4. CREATING A PROVIDER CONNECTION FOR BARE METAL

Important: The bare metal cluster function is a technology preview,and should not be used in production environments.

You need a provider connection to use Red Hat Advanced Cluster Management for Kubernetes console to deploy and manage a Red Hat OpenShift Container Platform cluster in a bare metal environment.

> NOTE
>
> The options for bare metal in the console are for technology preview only, and are hidden by feature flags by default. See the instructions for enabling the feature flags in the *Prerequisites* section.

6.4.1. Prerequisites

You need the following prerequisites before creating a provider connection:

- A Red Hat Advanced Cluster Management for Kubernetes hub cluster that is deployed. When managing bare metal clusters, you must have the hub cluster installed on Red Hat OpenShift Container Platform version 4.5, or later.

- Internet access for your Red Hat Advanced Cluster Management for Kubernetes hub cluster so it can create the Kubernetes cluster on your bare metal server

- Your bare metal server login credentials, which include the libvirt URI, SSH Private Key, and a list of SSH known hosts; see Generating an SSH private key and adding it to the agent

- Account permissions that allow installing clusters on the bare metal infrastructure

- Bare metal feature flags that are enabled to view the bare metal options. The bare metal options are hidden by feature flags by default. Complete the following steps to enable the feature flags:

  a. Start the Red Hat OpenShift Container Platform command line interface.

  b. Set the **featureFlags_baremetal** setting to **true** for the **console-header** container by entering the following command:

```
oc patch deploy console-header -n <namespace> -p '{"spec":{"template":{"spec":
{"containers":[{"name":"console-header","env": [{"name":
"featureFlags_baremetal","value":"true"}]}]}}}}'
```

  Replace <namespace> with your Red Hat Advanced Cluster Management project namespace.

  After the update, your **consoleui** CRD should look like the following example:

```
spec:
  ...
  template:
    ...
    spec:
      ...
      containers:
      - env:                              # Search for env:
        - name: featureFlags_baremetal
          value: "true"
        ...
```

  c. Set the **featureFlags_baremetal** value to **true** for the **hmc-ui** container:

```
oc patch -n <namespace> $(oc get deploy -o name | grep consoleui) -p '{"spec":
{"template":{"spec":{"containers":[{"name":"hcm-ui","env": [{"name":
"featureFlags_baremetal","value":"true"}]}]}}}}'
```

  Replace <namespace> with your Red Hat Advanced Cluster Management project namespace.

  Your update should look like the following example:

```
spec:
  ...
```

```
template:
  ...
  spec:
    ...
    containers:
    - env:                              # Search for env:
      - name: featureFlags_baremetal
        value: "true"
      ...
```

d. Make sure the **console-chart-...-consoleui...** and **console-header-...** pods are running:

```
oc -n open-cluster-management get pods
```

e. When the pods are running again, log out of the Red Hat Advanced Cluster Management for Kubernetes console and log back in. The bare metal options are now included in the console.

6.4.2. Creating a provider connection by using the console

To create a provider connection from the Red Hat Advanced Cluster Management for Kubernetes console, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.

2. On the *Clusters* page, select the *Provider connections* tab.
   Existing provider connections are displayed.

3. Select **Add connection**.

4. Select **Bare metal** as your provider.

5. Add a name for your provider connection.

6. Select a namespace for your provider connection from the list.
   Tip: Create a namespace specifically to host your provider connections, both for convenience and added security.

7. You can optionally add a *Base DNS domain* for your provider connection. If you add the base DNS domain to the provider connection, it is automatically populated in the correct field when you create a cluster with this provider connection.

8. Add your *libvirt URI*. See Connection URIs for more information.

9. Enter your *Red Hat OpenShift Pull Secret* . You can download your pull secret from Pull secret.

10. Add your *SSH Private Key* and your *SSH Public Key* so you can access the cluster. You can use an existing key, or use a key generation program to create a new one. See Generating an SSH private key and adding it to the agent for more information about how to generate a key.

11. Add a list of your SSH known hosts.

12. For disconnected installations only: Complete the fields in the **Configuration for disconnected installation** subsection with the required information:

- *Image Registry Mirror*: This optional value contains the disconnected registry path. The path contains the hostname, port, and repository path to all of the installation images for disconnected installations. Example: **repository.com:5000/openshift/ocp-release**.
- *Bootstrap OS Image*: This value contains the URL to the image to use for the bootstrap machine.
- *Cluster OS Image*: This value contains the URL to the image to use for Red Hat OpenShift Container Platform cluster machines.
- *Additional Trust Bundle*: This value provides the contents of the certificate file that is required to access the mirror registry.

13. Click **Create**. When you create the provider connection, it is added to the list of provider connections.

You can create a cluster that uses this provider connection by completing the steps in Creating a cluster on bare metal.

6.4.3. Deleting your provider connection

When you are no longer managing a cluster that is using a provider connection, delete the provider connection to protect the information in the provider connection.

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. Select **Provider connections**.
3. Select the options menu beside the provider connection that you want to delete.
4. Select **Delete connection**.

CHAPTER 7. CREATING A CLUSTER WITH RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES

Learn how to create Red Hat OpenShift Container Platform clusters across cloud providers with Red Hat Advanced Cluster Management for Kubernetes.

- Creating a cluster on Amazon Web Services
- Creating a cluster on Google Cloud Platform
- Creating a cluster on Microsoft Azure
- Creating a cluster on bare metal (Requires Red Hat OpenShift Container Platform version 4.4, or later)

7.1. CREATING A CLUSTER ON AMAZON WEB SERVICES

You can use the Red Hat Advanced Cluster Management for Kubernetes console to create a Red Hat OpenShift Container Platform cluster on Amazon Web Services (AWS).

7.1.1. Prerequisites

You must have the following prerequisites before creating a cluster on AWS:

- A deployed Red Hat Advanced Cluster Management for Kubernetes hub cluster
- Internet access for your Red Hat Advanced Cluster Management for Kubernetes hub cluster so it can create the Kubernetes cluster on Amazon Web Services
- AWS provider connection. See Creating a provider connection for Amazon Web Services for more information.
- A configured domain in AWS. See Configuring an AWS account for instructions on how to configure a domain.
- Amazon Web Services (AWS) login credentials, which include user name, password, access key ID, and secret access key. See Understanding and Getting Your Security Credentials.
- A Red Hat OpenShift image pull secret. See Using image pull secrets.

Note: If you change your cloud provider access key, you must manually update the provisioned cluster access key. For more information, see the known issue, Automatic secret updates for provisioned clusters is not supported.

7.1.2. Creating your cluster with the Red Hat Advanced Cluster Management for Kubernetes console

To create clusters from the Red Hat Advanced Cluster Management for Kubernetes console, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. On the Clusters page, Click **Add Cluster**.
3. Select **Create a cluster**.

NOTE

This procedure is for creating a cluster. If you have an existing cluster that you want to import, see Importing a target managed cluster to the hub cluster for those steps.

4. Enter a name for your cluster. This name is used in the hostname of the cluster.

   TIP

   You can view the **yaml** content updates as you enter the information in the console by setting the *YAML* switch to **ON**.

5. Select **Amazon Web Services** for the infrastructure platform.

6. Specify a **Release image** that you want to use for the cluster. This identifies the version of the Red Hat OpenShift Container Platform image that is used to create the cluster. If the version that you want to use is available, you can select the image from the list of images. If the image that you want to use is not a standard image, you can enter the url to the image that you want to use. See Release images for more information about release images.

7. Select your provider connection from the available connections on the list. If you do not have one configured, or want to configure a new one, select **Add connection**. See Creating a provider connection for Amazon Web Services for more information about creating a provider connection.

8. Enter the base domain information that you configured for your AWS account. If there is already a base domain associated with the selected provider connection, that value is populated in that field. You can change the value by overwriting it. See Configuring an AWS account for more information. This name is used in the hostname of the cluster.

9. Add the *Labels* that you want to associate with your cluster. These labels help to identify the cluster and limit search results.

10. Configure the *Node pools* for your cluster.
    The node pools define the location and size of the nodes that are used for your cluster.

    The *Region* specifies where the nodes are located geographically. A closer region might provide faster performance, but a more distant region might be more distributed.

    - Master pool: There are three Master nodes that are created for your cluster in the master pool. The master nodes share the management of the cluster activity. You can select multiple zones within the region for a more distributed group of master nodes. You can change the type and size of your instance after it is created, but you can also specify it in this section. The default values are *mx4.xlarge - 4 vCPU, 16 GiB RAM - General Purpose* with 500 GiB of root storage.

    - Worker pools: You can create one or more worker nodes in a worker pool to run the container workloads for the cluster. They can be in a single worker pool, or distributed across multiple worker pools.

11. **Optional:** Configure the cluster networking options.

12. **Optional:** Configure a label for the cluster.

13. Click **Create**. You can view your cluster details after the create and import process is complete.

NOTE

You do not have to run the **kubectl** command that is provided with the cluster details to import the cluster. When you create the cluster, it is automatically configured under the management of Red Hat Advanced Cluster Management for Kubernetes.

7.1.3. Accessing your cluster

To access a cluster that is managed by Red Hat Advanced Cluster Management for Kubernetes, complete the following steps:

1. From the Red Hat Advanced Cluster Management for Kubernetes navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. Select the name of the cluster that you created or want to access. The cluster details are displayed.
3. Select **Reveal credentials** to view the user name and password for the cluster. Note these values to use when you log in to the cluster.
4. Select **Console URL** to link to the cluster.
5. Log in to the cluster by using the user ID and password that you found in step 3.
6. Select the *Actions* menu for the cluster that you want to access.
7. Select **Launch to cluster**.

TIP

If you already know the log in credentials, you can access the cluster by selecting the *Actions* menu for the cluster, and selecting **Launch to cluster**.

7.1.4. Removing a cluster from management

When you remove a Red Hat OpenShift Container Platform cluster from management that was created with Red Hat Advanced Cluster Management for Kubernetes, you can either *detach* it or *destroy* it.

Detaching a cluster removes it from management, but does not completely delete it. You can import it again, if you decide that you want to bring it back under management. This is only an option when the cluster is in a *Ready* state.

Destroying a cluster removes it from management and deletes the components of the cluster. This is permanent, and it cannot be brought back under management after deletion.

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. Select the option menu beside the cluster that you want to delete.
3. Select **Destroy cluster** or **Detach cluster**.

TIP

You can detach or destroy multiple clusters by selecting the check boxes of the clusters that you want to detach or destroy. Then select **Detach** or **Destroy**.

7.2. CREATING A CLUSTER ON MICROSOFT AZURE

You can use the Red Hat Advanced Cluster Management for Kubernetes console to deploy a Red Hat OpenShift Container Platform cluster on Microsoft Azure.

7.2.1. Prerequisites

You must have the following prerequisites before creating a cluster on Azure:

- A deployed Red Hat Advanced Cluster Management for Kubernetes hub cluster
- Internet access for your Red Hat Advanced Cluster Management for Kubernetes hub cluster so it can create the Kubernetes cluster on Azure
- Azure provider connection. See Creating a provider connection for Microsoft Azure for more information.
- A configured domain in Azure. See Configuring a custom domain name for an Azure cloud service for instructions on how to configure a domain.
- Azure login credentials, which include user name and password. See azure.microsoft.com.
- Azure service principals, which include **clientId**, **clientSecret**, and **tenantId**. See azure.microsoft.com.
- A Red Hat OpenShift image pull secret. See Using image pull secrets.

Note: If you change your cloud provider access key, you must manually update the provisioned cluster access key. For more information, see the known issue, Automatic secret updates for provisioned clusters is not supported.

7.2.2. Creating your cluster with the Red Hat Advanced Cluster Management for Kubernetes console

To create clusters from the Red Hat Advanced Cluster Management for Kubernetes console, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. On the *Clusters* page, Click **Add Cluster**.
3. Select **Create a cluster**.

   NOTE

   This procedure is for creating a cluster. If you have an existing cluster that you want to import, see Importing a target managed cluster to the hub cluster for those steps.

4. Enter a name for your cluster. This name is used in the hostname of the cluster.

   TIP

   You can view the **yaml** content updates as you enter the information in the console by setting the *YAML* switch to **ON**.

5. Select **Microsoft Azure** for the infrastructure platform.

6. Specify a **Release image** that you want to use for the cluster. This identifies the version of the Red Hat OpenShift Container Platform image that is used to create the cluster. If the version that you want to use is available, you can select the image from the list of images. If the image that you want to use is not a standard image, you can enter the url to the image that you want to use. See Release images for more information about release images.

7. Select your provider connection from the available connections on the list. If you do not have one configured, or want to configure a new one, select **Add connection** . See Creating a provider connection for Microsoft Azure for more information about creating a provider connection.

8. Enter the base domain information that you configured for your Azure account. If there is already a base domain associated with the selected provider connection, that value is populated in that field. You can change the value by overwriting it. See Configuring a custom domain name for an Azure cloud service for more information. This name is used in the hostname of the cluster.

9. Add the *Labels* that you want to associate with your cluster. These labels help to identify the cluster and limit search results.

10. Configure the *Node pools* for your cluster.
    The node pools define the location and size of the nodes that are used for your cluster.

    The *Region* specifies where the nodes are located geographically. A closer region might provide faster performance, but a more distant region might be more distributed.

    - Master pool: There are three Master nodes that are created for your cluster in the master pool. The master nodes share the management of the cluster activity. You can select multiple zones within the region for a more distributed group of master nodes. You can change the type and size of your instance after it is created, but you can also specify it in this section. The default values are *Standard_D2s_v3 - 2 vCPU, 8 GiB RAM - General Purpose* with 512 GiB of root storage.

    - Worker pools: You can create one or more worker nodes in a worker pool to run the container workloads for the cluster. They can be in a single worker pool, or distributed across multiple worker pools.

11. **Optional:** Configure the cluster networking options.

12. **Optional:** Configure a label for the cluster.

13. Click **Create**. You can view your cluster details after the create and import process is complete.

    NOTE

    You do not have to run the **kubectl** command that is provided with the cluster details to import the cluster. When you create the cluster, it is automatically configured under the management of Red Hat Advanced Cluster Management for Kubernetes.

7.2.3. Accessing your cluster

To access a cluster that is managed by Red Hat Advanced Cluster Management for Kubernetes, complete the following steps:

1. From the Red Hat Advanced Cluster Management for Kubernetes navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. Select the name of the cluster that you created or want to access. The cluster details are displayed.
3. Select **Reveal credentials** to view the user name and password for the cluster. Note these values to use when you log in to the cluster.
4. Select **Console URL** to link to the cluster.
5. Log in to the cluster by using the user ID and password that you found in step 3.
6. Select the *Actions* menu for the cluster that you want to access.
7. Select **Launch to cluster**.

TIP

If you already know the log in credentials, you can access the cluster by selecting the *Actions* menu for the cluster, and selecting **Launch to cluster**.

7.2.4. Removing a cluster from management

When you remove a Red Hat OpenShift Container Platform cluster from management that was created with Red Hat Advanced Cluster Management for Kubernetes, you can either *Detach* it or *Destroy* it.

Detaching a cluster removes it from management, but does not completely delete it. You can import it again, if you decide that you want to bring it back under management. This is only an option when the cluster is in a *Ready* state.

Destroying a cluster removes it from management and deletes the components of the cluster. This is permanent, and it cannot be brought back under management after deletion.

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. Select the option menu beside the cluster that you want to delete.
3. Select **Destroy cluster** or **Detach cluster**.

TIP

You can detach or destroy multiple clusters by selecting the check boxes of the clusters that you want to detach or destroy. Then select **Detach** or **Destroy**.

7.3. CREATING A CLUSTER ON GOOGLE CLOUD PLATFORM

Follow the procedure to create a Red Hat OpenShift Container Platform cluster on Google Cloud Platform (GCP). For more information about Google Cloud Platform, see Google Cloud Platform.

7.3.1. Prerequisites

You must have the following prerequisites before creating a cluster on GCP:

- A deployed Red Hat Advanced Cluster Management for Kubernetes hub cluster

- Internet access for your Red Hat Advanced Cluster Management for Kubernetes hub cluster so it can create the Kubernetes cluster on GCP
- GCP provider connection. See Creating a a provider connection for Google Cloud Platform for more information.
- A configured domain in GCP. See Setting up a custom domain for instructions on how to configure a domain.
- GCP login credentials, which include user name and password.
- A Red Hat OpenShift image pull secret. See Using image pull secrets.

Note: If you change your cloud provider access key, you must manually update the provisioned cluster access key. For more information, see the known issue, Automatic secret updates for provisioned clusters is not supported.

7.3.2. Creating your cluster with the Red Hat Advanced Cluster Management for Kubernetes console

To create clusters from the Red Hat Advanced Cluster Management for Kubernetes console, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. On the *Clusters* page, Click **Add Cluster**.
3. Select **Create a cluster**.

   NOTE

   This procedure is for creating a cluster. If you have an existing cluster that you want to import, see Importing a target managed cluster to the hub cluster for those steps.

4. Enter a name for your cluster. There are some restrictions that apply to naming your GCP cluster. These restrictions include not beginning the name with **goog** or containing a group of letters and numbers that resemble **google** anywhere in the name. See Bucket naming guidelines for the complete list of restrictions.
   This name is used in the hostname of the cluster.

   TIP

   You can view the **yaml** content updates as you enter the information in the console by setting the *YAML* switch to **ON**.

5. Select **Google Cloud** for the infrastructure platform.
6. Specify a **Release image** that you want to use for the cluster. This identifies the version of the Red Hat OpenShift Container Platform image that is used to create the cluster. If the version that you want to use is available, you can select the image from the list of images. If the image that you want to use is not a standard image, you can enter the url to the image that you want to use. See Release images for more information about release images.
7. Select your provider connection from the available connections on the list. If you do not have

one configured, or want to configure a new one, select **Add connection**. See Creating a provider connection for Google Cloud Platform for more information about creating a provider connection.

8. Enter the base domain information that you configured for your Google Cloud Platform account. If there is already a base domain associated with the selected provider connection, that value is populated in that field. You can change the value by overwriting it. See Setting up a custom domain for more information. This name is used in the hostname of the cluster.

9. Add the *Labels* that you want to associate with your cluster. These labels help to identify the cluster and limit search results.

10. Configure the *Node pools* for your cluster.
    The node pools define the location and size of the nodes that are used for your cluster.

    The *Region* specifies where the nodes are located geographically. A closer region might provide faster performance, but a more distant region might be more distributed.

    - Master pool: There are three Master nodes that are created for your cluster in the master pool. The master nodes share the management of the cluster activity. You can select multiple zones within the region for a more distributed group of master nodes. You can change the type and size of your instance after it is created, but you can also specify it in this section. The default values are *n1-standard-1 - n1-standard-1 1 vCPU - General Purpose* with 500 GiB of root storage.

    - Worker pools: You can create one or more worker nodes in a worker pool to run the container workloads for the cluster. They can be in a single worker pool, or distributed across multiple worker pools.

11. **Optional:** Configure the cluster networking options.

12. **Optional:** Configure a label for the cluster.

13. Click **Create**.

You can view your cluster details after the create and import process is complete.

+ NOTE: You do not have to run the **kubectl** command that is provided with the cluster details to import the cluster. When you create the cluster, it is automatically configured under the management of Red Hat Advanced Cluster Management for Kubernetes.

7.3.3. Accessing your cluster

To access a cluster that is managed by Red Hat Advanced Cluster Management for Kubernetes, complete the following steps:

1. From the Red Hat Advanced Cluster Management for Kubernetes navigation menu, navigate to **Automate infrastructure** > **Clusters**.

2. Select the name of the cluster that you created or want to access. The cluster details are displayed.

3. Select **Reveal credentials** to view the user name and password for the cluster. Note these values to use when you log in to the cluster.

4. Select **Console URL** to link to the cluster.

5. Log in to the cluster by using the user ID and password that you found in step 3.

6. Select the *Actions* menu for the cluster that you want to access.

7. Select **Launch to cluster**.

   TIP

   If you already know the log in credentials, you can access the cluster by selecting the *Actions* menu for the cluster, and selecting **Launch to cluster**.

7.3.4. Removing a cluster from management

When you remove a Red Hat OpenShift Container Platform cluster from management that was created with Red Hat Advanced Cluster Management for Kubernetes, you can either *detach* it or *destroy* it.

Detaching a cluster removes it from management, but does not completely delete it. You can import it again, if you decide that you want to bring it back under management. This is only an option when the cluster is in a *Ready* state.

Destroying a cluster removes it from management and deletes the components of the cluster. This is permanent, and it cannot be brought back under management after deletion.

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.

2. Select the option menu beside the cluster that you want to delete.

3. Select **Destroy cluster** or **Detach cluster**.

   TIP

   You can detach or destroy multiple clusters by selecting the check boxes of the clusters that you want to detach or destroy. Then select **Detach** or **Destroy**.

7.4. CREATING A CLUSTER ON BARE METAL

Important: The bare metal cluster function is a technology preview,and should not be used in production environments.

You can use the Red Hat Advanced Cluster Management for Kubernetes console to create a Red Hat OpenShift Container Platform cluster in a bare metal environment.

NOTE

The options for bare metal in the console are a technology preview only, and are hidden by a feature flag by default. See the instructions for enabling the feature flag in the *Prerequisites* section.

7.4.1. Prerequisites

You need the following prerequisites before creating a cluster in a bare metal environment:

- A deployed Red Hat Advanced Cluster Management for Kubernetes hub cluster on OpenShift Container Platform version 4.5, or later.

- Internet access for your Red Hat Advanced Cluster Management for Kubernetes hub cluster so it can create the Kubernetes cluster in the bare metal environment

- Bare metal provider connection; see Creating a provider connection for bare metal for more information

- Login credentials for your bare metal environment, which include user name, password, and Baseboard Management Controller Address

- A Red Hat OpenShift Container Platform image pull secret; see Using image pull secrets.
  Note: The bare metal asset, managed bare metal cluster, and its related secret must be in the same namespace.

- Bare metal feature flags that are enabled to view the bare metal options. The bare metal options are hidden by feature flags by default. Complete the following steps to enable the feature flags:

  a. Start the Red Hat OpenShift Container Platform command line interface.

  b. Set the **featureFlags_baremetal** setting to **true** for the **console-header** container by entering the following command:

  ```
  oc patch deploy console-header -n <namespace> -p '{"spec":{"template":{"spec":
  {"containers":[{"name":"console-header","env": [{"name":
  "featureFlags_baremetal","value":"true"}]}]}}}}'
  ```

  Replace <namespace> with your Red Hat Advanced Cluster Management project namespace.

  After the update, your **consoleui** CRD should look like the following example:

  ```
  spec:
    ...
    template:
      ...
      spec:
        ...
        containers:
        - env:                          # Search for env:
          - name: featureFlags_baremetal
            value: "true"
          ...
  ```

  c. Set the **featureFlags_baremetal** value to **true** for the **hmc-ui** container:

  ```
  oc patch -n <namespace> $(oc get deploy -o name | grep consoleui) -p '{"spec":
  {"template":{"spec":{"containers":[{"name":"hcm-ui","env": [{"name":
  "featureFlags_baremetal","value":"true"}]}]}}}}'
  ```

  Replace <namespace> with your Red Hat Advanced Cluster Management project namespace.

  Your update should look like the following example:

  ```
  spec:
  ```

```
...
template:
  ...
  spec:
    ...
    containers:
    - env:                              # Search for env:
      - name: featureFlags_baremetal
        value: "true"
      ...
```

d. Make sure the **console-chart-...-consoleui...** and **console-header-...** pods are running:

```
oc -n open-cluster-management get pods
```

e. When the pods are running again, log out of the Red Hat Advanced Cluster Management for Kubernetes console and log back in. The bare metal options are now included in the console.

Note: If you change your cloud provider access key, you must manually update the provisioned cluster access key. For more information, see the known issue, Automatic secret updates for provisioned clusters is not supported.

7.4.2. Creating your cluster with the Red Hat Advanced Cluster Management for Kubernetes console

To create clusters from the Red Hat Advanced Cluster Management for Kubernetes console, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. On the Clusters page, Click **Add Cluster**.
3. Select **Create a cluster**.
   Note: This procedure is for creating a cluster. If you have an existing cluster that you want to import, see Importing a target managed cluster to the hub cluster for those steps.
4. Enter a name for your cluster. This name is used in the hostname of the cluster.
   Tip: You can view the **yaml** content updates as you enter the information in the console by setting the *YAML* switch to **ON**.
5. Select **Bare Metal** for the infrastructure platform.
6. Specify a **Release image** that you want to use for the cluster. This identifies the version of the Red Hat OpenShift Container Platform image that is used to create the cluster. If the version that you want to use is available, you can select the image from the list of images. If the image that you want to use is not a standard image, you can enter the url to the image that you want to use. See Release images for more information about release images.
7. Select your provider connection from the available connections on the list. If you do not have one configured, or want to configure a new one, select **Add provider**. See Creating a provider connection for bare metal for more information about creating a provider connection.
8. Enter the base domain information that you configured in your bare metal environment. If there is already a base domain associated with the selected provider connection, that value is

populated in that field. You can change the value by overwriting it. This name is used in the hostname of the cluster.

9. Select your hosts from the list of hosts that are associated with your provider connection. Select a minimum of three assets that are on the same bridge networks as the hypervisor.
10. **Optional:** Configure the cluster networking options.
11. **Optional:** Configure a label for the cluster.
12. **Optional:** Update the advanced settings, if you want to change the setting for including a configmap.
13. Click **Create**. You can view your cluster details after the create and import process is complete.
    Note: You do not have to run the **kubectl** command that is provided with the cluster details to import the cluster. When you create the cluster, it is automatically configured under the management of Red Hat Advanced Cluster Management for Kubernetes.

7.4.3. Accessing your cluster

To access a cluster that is managed by Red Hat Advanced Cluster Management for Kubernetes, complete the following steps:

1. From the Red Hat Advanced Cluster Management for Kubernetes navigation menu, navigate to **Automate infrastructure** > **Clusters**.
2. Select the name of the cluster that you created or want to access. The cluster details are displayed.
3. Select **Reveal credentials** to view the user name and password for the cluster. Note these values to use when you log in to the cluster.
4. Select **Console URL** to link to the cluster.
5. Log in to the cluster by using the user ID and password that you found in step 3.
6. Select the *Actions* menu for the cluster that you want to access.
7. Select **Launch to cluster**.
   Tip: If you already know the log in credentials, you can access the cluster by selecting the *Actions* menu for the cluster, and selecting **Launch to cluster**.

7.4.4. Removing a cluster from management

When you remove a Red Hat OpenShift Container Platform cluster from management that was created with Red Hat Advanced Cluster Management for Kubernetes, you can either *detach* it or *destroy* it.

Detaching a cluster removes it from management, but does not completely delete it. You can import it again, if you decide that you want to bring it back under management. This is only an option when the cluster is in a *Ready* state.

Destroying a cluster removes it from management and deletes the components of the cluster. This is permanent, and it cannot be brought back under management after deletion.

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**.

2. Select the option menu beside the cluster that you want to delete.

3. Select **Destroy cluster** or **Detach cluster**.
   Tip: You can detach or destroy multiple clusters by selecting the check boxes of the clusters that you want to detach or destroy. Then select **Detach** or **Destroy**.

CHAPTER 8. IMPORTING A TARGET MANAGED CLUSTER TO THE HUB CLUSTER

You can import clusters from different Kubernetes cloud providers. After you import, the targeted cluster becomes a managed cluster for the Red Hat Advanced Cluster Management for Kubernetes hub cluster. Unless otherwise specified, complete the import tasks anywhere where you can access the hub cluster and the targeted managed cluster.

> **NOTE**
>
> A hub cluster cannot manage *any* other hub cluster; you must import an existing cluster.

Choose from the following instructions to set up your managed cluster, either from the console or from the CLI:

Required user type or access level: Cluster administrator

- Importing an existing cluster with the console
- Importing a managed cluster with the CLI
- Modifying the klusterlet addons settings of your cluster

8.1. IMPORTING AN EXISTING CLUSTER WITH THE CONSOLE

After you install Red Hat Advanced Cluster Management for Kubernetes, you are ready to import a cluster to manage. You can import from both the console and the CLI. Follow this procedure to import from the console. You need your terminal for authentication during this procedure.

- Prerequisites
- Importing a cluster
- Removing a cluster

8.1.1. Prerequisites

- You need a Red Hat Advanced Cluster Management for Kubernetes hub cluster that is deployed. If you are importing bare metal clusters, you must have the hub cluster installed on Red Hat OpenShift Container Platform version 4.4, or later.
- You need a cluster that you want to manage and Internet connectivity.
- Install **kubectl**. To install **kubectl**, see *Install and Set Up kubectl* in the Kubernetes documentation.
- You need the **base64** command line tool.

Required user type or access level: Cluster administrator

8.1.2. Importing a cluster

You can import existing clusters from the Red Hat Advanced Cluster Management for Kubernetes console for each of the available cloud providers.

NOTE

A hub cluster cannot manage *any* other hub cluster; you must import an existing cluster.

1. From the navigation menu, hover over **Automate infrastructure** and click **Clusters**.

2. Click **Add cluster**.

3. Click **Import an existing cluster**.

4. Provide a cluster name. By default, the namespace is set to the same value as your cluster name. **Best practice:** Leave the namespace value and do not edit.

5. **Optional:** Click to expand **Edit cluster import YAML file**and modify the endpoint configuration.
   See Table 1. YAML file parameters and descriptions for details about each parameter.

6. **Optional**: After you import, you can add labels by clicking **Configure advanced parameters** and use these labels to search.

7. **Optional**: Configure the **MANAGED CLUSTER URLS**. By configuring the **MANAGED CLUSTER URLS**, the URLs display in the table when you run the **oc get managedcluster** command.

   a. If it is not already on, turn on the **YAML** content using the switch in the web console so you can view the content.

   b. Add the **manageClusterClientConfigs** section to the **ManagedCluster** spec in the **import.yaml** file, as shown in the following example:

   ```
   apiVersion: cluster.open-cluster-management.io/v1
   kind: ManagedCluster
   metadata:
     labels:
       cloud: auto-detect
    vendor: auto-detect
    name: cluster-test
     name: cluster-test
   spec:
     hubAcceptsClient: true
     managedClusterClientConfigs:
     - url: https://multicloud-console.apps.new-managed.dev.redhat.com
   ---
   apiVersion: agent.open-cluster-management.io/v1
   ...
   ```

   Replace the URL value is the external access URL address of the managed cluster.

8. Click **Generate Command** to retrieve the command to deploy the **open-cluster-management-agent-addon**.

9. From the *Import an existing cluster* window, hover and click the **Copy command** icon to copy the import command and the token that you are provided. You must click the **Copy** icon to receive the accurate copy. **Important:** The command contains pull secret information that is copied to each of the imported clusters. Anyone who can access the imported clusters can also view the pull secret information. Consider creating a secondary pull secret at

https://cloud.redhat.com/ or by creating a service account so your personal credentials are not compromised. See Using image pull secrets or Understanding and creating service accounts for more information.

10. From your terminal, authenticate to your managed cluster. Configure your **kubectl** for your targeted managed cluster.
    See Supported clouds to learn how to configure your **kubectl**.

11. To deploy the **open-cluster-management-agent-addon** to the managed cluster, run the command that you generated and copied from *step 8*.

12. Click **View cluster** to view the *Overview* page and a summary of your cluster.

Note You can continue to import more clusters. Click **Import another** to repeat the process.

8.1.2.1. YAML parameters and descriptions

Table 1: The following table lists the parameters and descriptions that are available in the YAML file:

Parameter	Description	Default value
clusterLabels	Provide cluster labels; you can add labels to your file	none
clusterLabels.cloud	The provider label for your cluster	auto-detect
clusterLabels.vendor	The Kubernetes vendor label for your cluster	auto-detect
clusterLabels.environment	The environment label for your cluster	none
clusterLabels.region	The region where your cluster is set up	none
applicationManager.enabled	Enables multicluster manager application deployment, deploys subscription controller and deployable controller	true
searchCollector.enabled	Enables search collection and indexing	true
policyController.enabled	Enable the Governance and risk dashboard policy feature	true, updateInterval: 15
certPolicyController.enabled	Monitors certificate expiration based on distributed policies	true
iamPolicyController	Monitors identity controls based on distributed policies	true

Parameter	Description	Default value
serviceRegistry.enabled	Service registry that is used to discover services that are deployed by Application Deployable among managed clusters.	false
serviceRegistry.dnsSuffix	The suffix of the registry DNS name, which is added to the end of the target clusters dns domain name.	mcm.svc
serviceRegistry.plugins	Comma-separated list of enabled plugins. Supported plugins: **kube-service**, **kube-ingress**, and **istio**.	kube-service
version	Version of **open-cluster-management-agent-addon**	2.0

8.1.3. Removing an imported cluster

Complete the following procedure to remove an imported cluster and the **open-cluster-management-agent-addon** that was created on the managed cluster.

1. From the *Clusters* page, find your imported cluster in the table.

2. Click **Options** > **Detach cluster** to remove your cluster from management.

8.2. IMPORTING A MANAGED CLUSTER WITH THE CLI

After you install Red Hat Advanced Cluster Management for Kubernetes, you are ready to import a cluster to manage. You can import from both the console and the CLI. Follow this procedure to import from the CLI.

- Prerequisites

- Supported architecture

- Importing the klusterlet

NOTE

A hub cluster cannot manage another hub cluster.

8.2.1. Prerequisites

- You need a Red Hat Advanced Cluster Management for Kubernetes hub cluster that is deployed. If you are importing bare metal clusters, you must have the hub cluster installed on Red Hat OpenShift Container Platform version 4.4, or later. **Important:** The bare metal function is a technology preview, and should not be used in production environments.

- You need a separate cluster that you want to manage and Internet connectivity.
- You need the Red Hat OpenShift Container Platform CLI version 4.3, or later, to run **oc** commands. See Getting started with the CLI for information about installing and configuring the Red Hat OpenShift CLI, **oc**.
- You need to install the Kubernetes CLI, **kubectl**. To install **kubectl**, see *Install and Set Up kubectl* in the Kubernetes documentation.

> NOTE
>
> Download the installation file for CLI tools from the console.

8.2.2. Supported architecture

- Linux
- macOS

8.2.3. Prepare for import

1. Log in to your *hub cluster*. Run the following command:

   ```
   oc login
   ```

2. Run the following command on the hub cluster to create the namespace. **Note:** The cluster name that is defined in **<cluster_name>** is also used as the cluster namespace in the **.yaml** file file and commands:

   ```
   oc new-project ${CLUSTER_NAME}
   oc label namespace ${CLUSTER_NAME} cluster.open-cluster-
   management.io/managedCluster=${CLUSTER_NAME}
   ```

3. Edit the example ManagedCluster with the following sample of YAML:

   ```
   apiVersion: cluster.open-cluster-management.io/v1
   kind: ManagedCluster
   metadata:
     name: <cluster_name>
   spec:
     hubAcceptsClient: true
   ```

4. Save the file as **managed-cluster.yaml**.

5. Apply the YAML file with the following command:

   ```
   oc apply -f managed-cluster.yaml
   ```

6. Create the klusterlet addon configuration file. Enter the following example YAML:

   ```
   apiVersion: agent.open-cluster-management.io/v1
   kind: KlusterletAddonConfig
   metadata:
   ```

```
  name: <cluster_name>
  namespace: <cluster_name>
spec:
  clusterName: <cluster_name>
  clusterNamespace: <cluster_name>
  applicationManager:
    enabled: true
  certPolicyController:
    enabled: true
  clusterLabels:
    cloud: auto-detect
    vendor: auto-detect
  iamPolicyController:
    enabled: true
  policyController:
    enabled: true
  searchCollector:
    enabled: true
  version: 2.0.0
```

7. Save the file as **klusterlet-addon-config.yaml**.

8. Apply the YAML. Run the following command:

```
oc apply -f klusterlet-addon-config.yaml
```

The ManagedCluster-Import-Controller will generate a secret named **${CLUSTER_NAME}-import**. The **${CLUSTER_NAME}-import** secret contains the **import.yaml** that the user applies to a managed cluster to install klusterlet.

8.2.4. Importing the klusterlet

IMPORTANT

The import command contains pull secret information that is copied to each of the imported clusters. Anyone who can access the imported clusters can also view the pull secret information.

1. Obtain the **klusterlet-crd.yaml** that was generated by the managed cluster import controller. Run the following command:

```
oc get secret ${CLUSTER_NAME}-import -n ${CLUSTER_NAME} -o jsonpath=
{.data.crds\\.yaml} | base64 --decode > klusterlet-crd.yaml
```

2. Obtain the **import.yaml** that was generated by the managed cluster import controller. Run the following command:

```
oc get secret ${CLUSTER_NAME}-import -n ${CLUSTER_NAME} -o jsonpath=
{.data.import\\.yaml} | base64 --decode > import.yaml
```

3. Log in to your target *managed* cluster.

4. Apply the **klusterlet-crd.yaml** that was generated in step 1. Run the following command:

```
kubectl apply -f klusterlet-crd.yaml
```

5. Apply the **import.yaml** file that was generated in step 2. Run the following command:

   ```
   kubectl apply -f import.yaml
   ```

6. Validate the pod status on the target managed cluster. Run the following command:

   ```
   kubectl get pod -n open-cluster-management-agent
   ```

7. Validate **JOINED** and **AVAILABLE** status for your imported cluster. Run the following command from the *hub* cluster:

   ```
   kubectl get managedcluster -n ${CLUSTER_NAME}
   ```

8. Addons will be installed after the managed cluster is **AVAILABLE**. Validate the pod status of addons on the target managed cluster. Run the following command:

   ```
   kubectl get pod -n open-cluster-management-agent-addon
   ```

8.3. MODIFYING THE KLUSTERLET ADDONS SETTINGS OF YOUR CLUSTER

You can modify the settings of **klusterlet addon** to change your configuration using the hub cluster.

The **klusterlet addon** controller manages the functions that are enabled and disabled according to the settings in the **klusterletaddonconfigs.agent.open-cluster-management.io** Kubernetes resource.

The following settings can be updated in the **klusterletaddonconfigs.agent.open-cluster-management.io** Kubernetes resource:

Setting name	Value
applicationmanager	**true** or **false**
policyController	**true** or **false**
searchCollector	**true** or **false**
certPolicyController	**true** or **false**
iamPolicyController	**true** or **false**

8.3.1. Modify using the console on the hub cluster

You can modify the settings of the **klusterletaddonconfigs.agent.open-cluster-management.io** resource by using the hub cluster. Complete the following steps to change the settings:

1. Authenticate into the Red Hat Advanced Cluster Management for Kubernetes console of the hub cluster.

2. From the main menu of the hub cluster console, select **Search**.

3. In the search parameters, enter the following value: **kind:klusterletaddonconfigs**

4. Select the endpoint resource that you want to update.

5. Find the **spec** section and select **Edit** to edit the content.

6. Modify your settings.

7. Select **Save** to apply your changes.

8.3.2. Modify using the command line on the hub cluster

You must have access to the <cluster-name> namespace to modify your settings by using the hub cluster. Complete the following steps:

1. Authenticate into the hub cluster.

2. Enter the following command to edit the resource:

```
kubectl edit klusterletaddonconfigs.agent.open-cluster-management.io <cluster-name> -n
<cluster-name>
```

3. Find the **spec** section.

4. Modify your settings, as necessary.

CHAPTER 9. UPGRADING YOUR CLUSTER

After you create clusters that you want to manage with Red Hat Advanced Cluster Management for Kubernetes, you can use the Red Hat Advanced Cluster Management console to upgrade those clusters to the latest minor version that is available in the version channel that the managed cluster uses.

To upgrade to a major version, you must verify that you meet all of the prerequisites for upgrading to that version. You must update the version channel on the managed cluster before you can upgrade the cluster with the console. After you update the version channel on the managed cluster, the Red Hat Advanced Cluster Management console displays the latest versions that are available for the upgrade.

Note: You cannot upgrade Red Hat OpenShift Kubernetes Service clusters with the Red Hat Advanced Cluster Management for Kubernetes console.

This method of upgrading only works for Red Hat OpenShift Container Platform clusters that are in a *Ready* state.

To upgrade your cluster, complete the following steps:

1. From the navigation menu, navigate to **Automate infrastructure** > **Clusters**. If an upgrade is available, it is shown in the *Distribution version* column.

2. Select the clusters that you want to upgrade. **Note:** A cluster must be in *Ready* state, and must be an OpenShift Container Platform cluster to be upgraded with the console.

3. Select **Upgrade**.

4. Select the new version of each cluster.

5. Select **Upgrade**.

9.1. UPGRADING DISCONNECTED CLUSTERS

You can use Red Hat OpenShift Update Service with Red Hat Advanced Cluster Management for Kubernetes to upgrade your clusters in a disconnected environment.

Important: Red Hat OpenShift Update Service is a Red Hat OpenShift Container Platform Operator that is provided as a technical preview with OpenShift Container Platform 4.4. It is not intended for use in a production environment.

In some cases, security concerns prevent clusters from being connected directly to the Internet. This makes it difficult to know when upgrades are available, and how to process those upgrades. Configuring OpenShift Update Service can help.

OpenShift Update Service is a separate operator and operand that monitors the available versions of your managed clusters in a disconnected environment, and makes them available for upgrading your clusters in a disconnected environment. After OpenShift Update Service is configured, it can perform the following actions:

1. Monitor when upgrades are available for your disconnected clusters.

2. Identify which updates are mirrored to your local site for upgrading by using the graph data file.

3. Notify you that an upgrade is available for your cluster by using the Red Hat Advanced Cluster Management console.

9.1.1. Prerequisites

You must have the following prerequisites before you can use OpenShift Update Service to upgrade your disconnected clusters:

- A deployed Red Hat Advanced Cluster Management hub cluster that is running on Red Hat OpenShift Container Platform version 4.5, or later with restricted OLM configured. See Using Operator Lifecycle Manager on restricted networks for details about how to configure restricted OLM.
  Tip: Make a note of the catalog source image when you configure restricted OLM.

- An OpenShift Container Platform cluster that is managed by the Red Hat Advanced Cluster Management hub cluster

- Access credentials to a local repository where you can mirror the cluster images. See Creating a mirror registry for installation in a restricted network for more information about how to create this repository.
  Note: The image for the current version of the cluster that you upgrade must always be available as one of the mirrored images. If an upgrade fails, the cluster reverts back to the version of the cluster at the time that the upgrade was attempted.

9.1.2. Prepare your disconnected mirror registry

You must mirror both the image that you want to upgrade to and the current image that you are upgrading from to your local mirror registry. Complete the following steps to mirror the images:

1. Create a script file that contains content that resembles the following example:

   ```
   UPSTREAM_REGISTRY=quay.io
   PRODUCT_REPO=openshift-release-dev
   RELEASE_NAME=ocp-release
   OCP_RELEASE=4.5.2-x86_64
   LOCAL_REGISTRY=$(hostname):5000
   LOCAL_SECRET_JSON=/path/to/pull/secret

   oc adm -a ${LOCAL_SECRET_JSON} release mirror \
   --
   from=${UPSTREAM_REGISTRY}/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELE
   ASE} \
   --to=${LOCAL_REGISTRY}/ocp4 \
   --to-release-image=${LOCAL_REGISTRY}/ocp4/release:${OCP_RELEASE}
   ```

   Replace **/path/to/pull/secret** with the path to your OpenShift Container Platform pull secret.

2. Run the script to mirror the images, configure settings, and separate the release images from the release content.

Tip: You can use the output of the last line of this script when you create your **ImageContentSourcePolicy**.

9.1.3. Deploy the operator for OpenShift Update Service

To deploy the operator for OpenShift Update Service in your OpenShift Container Platform environment, complete the following steps:

1. On the hub cluster, access the OpenShift Container Platform operator hub.

2. Deploy the operator by selecting **Red Hat OpenShift Update Service Operator**. Update the default values, if necessary. The deployment of the operator creates a new project named **openshift-cincinnati**.

3. Wait for the installation of the operator to finish.
   Tip: You can check the status of the installation by entering the **oc get pods** command on your OpenShift Container Platform command line. Verify that the operator is in the **running** state.

9.1.4. Build the graph data init container

OpenShift Update Service uses graph data information to determine the available upgrades. In a connected environment, OpenShift Update Service pulls the graph data information for available upgrades directly from the Cincinnati graph data GitHub repository . Because you are configuring a disconnected environment, you must make the graph data available in a local repository by using an **init container**. Complete the following steps to create a graph data **init container**:

1. Clone the *graph data* Git repository by entering the following command:

   ```
   git clone https://github.com/openshift/cincinnati-graph-data
   ```

2. Create a file that contains the information for your graph data **init**. You can find this sample Dockerfile in the **cincinnati-operator** GitHub repository. The contents of the file is shown in the following sample:

   ```
   FROM registry.access.redhat.com/ubi8/ubi:8.1

   RUN curl -L -o cincinnati-graph-data.tar.gz https://github.com/openshift/cincinnati-graph-
   data/archive/master.tar.gz

   RUN mkdir -p /var/lib/cincinnati/graph-data/

   CMD exec /bin/bash -c "tar xvzf cincinnati-graph-data.tar.gz -C /var/lib/
   cincinnati/graph-data/ --strip-components=1"
   ```

   In this example:

   - The **FROM** value is the external registry where OpenShift Update Service finds the images.
   - The **RUN** commands create the directory and package the upgrade files.
   - The **CMD** command copies the package file to the local repository and extracts the files for an upgrade.

3. Run the following commands to build the **graph data init container**:

   ```
   podman build -f <path_to_Dockerfile> -t
   ${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-container:latest
   podman push ${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-
   container:latest --authfile=/path/to/pull_secret.json
   ```

   Replace *path_to_Dockerfile* with the path to the file that you created in the previous step.

   Replace *${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-container* with the path to your local graph data init container.

Replace */path/to/pull_secret* with the path to your pull secret file.

Note: You can also replace **podman** in the commands with **docker**, if you don't have **podman** installed.

9.1.5. Configure certificate for the mirrored registry

If you are using a secure external container registry to store your mirrored OpenShift Container Platform release images, OpenShift Update Service requires access to this registry to build an upgrade graph. Complete the following steps to configure your CA certificate to work with the OpenShift Update Service pod:

1. Find the OpenShift Container Platform external registry API, which is located in **image.config.openshift.io**. This is where the external registry CA certificate is stored. See Image Registry Operator in OpenShift Container Platform in the OpenShift Container Platform documentation for more information.

2. Create a ConfigMap in the **openshift-config** namespace.

3. Add your CA certificate under the key **cincinnati-registry**. OpenShift Update Service uses this setting to locate your certificate:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: trusted-ca
data:
  cincinnati-registry: |
    -----BEGIN CERTIFICATE-----
    ...
    -----END CERTIFICATE-----
```

4. Edit the **cluster** resource in the **image.config.openshift.io** API to set the **additionalTrustedCA** field to the name of the ConfigMap that you created.

```
oc patch image.config.openshift.io cluster -p '{"spec":{"additionalTrustedCA":
{"name":"trusted-ca"}}}' --type merge
```

Replace *trusted-ca* with the path to your new ConfigMap.

The OpenShift Update Service Operator watches the **image.config.openshift.io** API and the ConfigMap you created in the **openshift-config** namespace for changes, then restart the deployment if the CA cert has changed.

9.1.6. Deploy the OpenShift Update Service instance

When you finish deploying the OpenShift Update Service instance on your hub cluster, this instance is located where the images for the cluster upgrades are mirrored and made available to the disconnected managed cluster. Complete the following steps to deploy the instance:

1. If you do not want to use the default namespace of the operator, which is **openshift-cincinnati**, create a namespace for your OpenShift Update Service instance:

    a. In the OpenShift Container Platform hub cluster console navigation menu, select **Administration** > **Namespaces**.

   b. Select **Create Namespace**.

   c. Add the name of your namespace, and any other information for your namespace.

   d. Select **Create** to create the namespace.

2. In the *Installed Operators* section of the OpenShift Container Platform console, select **Red Hat OpenShift Update Service Operator**.

3. Select **Create Instance** in the menu.

4. Paste the contents from your OpenShift Update Service instance. Your YAML instance might resemble the following manifest:

   ```
   apiVersion: cincinnati.openshift.io/v1beta1
   kind: Cincinnati
   metadata:
     name: openshift-update-service-instance
     namespace: openshift-cincinnati
   spec:
     registry: <registry_host_name>:<port>
     replicas: 1
     repository: ${LOCAL_REGISTRY}/ocp4/release
     graphDataImage: '<host_name>:<port>/cincinnati-graph-data-container'
   ```

   Replace the **spec.registry** value with the path to your local disconnected registry for your images.

   Replace the **spec.graphDataImage** value with the path to your graph data init container. **Tip:** This is the same value that you used when you ran the **podman push** command to push your graph data init container.

5. Select **Create** to create the instance.

6. From the hub cluster CLI, enter the **oc get pods** command to view the status of the instance creation. It might take a while, but the process is complete when the result of the command shows that the instance and the operator are running.

9.1.7. Deploy a policy to override the default registry (optional)

Note: The steps in this section only apply if you have mirrored your releases into your mirrored registry.

OpenShift Container Platform has a default image registry value that specifies where it finds the upgrade packages. In a disconnected environment, you can create a policy to replace that value with the path to your local image registry where you mirrored your release images.

For these steps, the policy is named *ImageContentSourcePolicy*. Complete the following steps to create the policy:

1. Log in to the OpenShift Container Platform environment of your hub cluster.

2. In the OpenShift Container Platform navigation, select **Administration** > **Custom Resource Definitions**.

3. Select the *Instances* tab.

4. Select the name of the *ImageContentSourcePolicy* that you created when you set up your disconnected OLM to view the contents.

5. Select the *YAML* tab to view the content in **YAML** format.

6. Copy the entire contents of the ImageContentSourcePolicy.

7. From the Red Hat Advanced Cluster Management console, select **Govern risk** > **Create policy**.

8. Set the **YAML** switch to *On* to view the YAML version of the policy.

9. Delete all of the content in the **YAML** code.

10. Paste the following **YAML** content into the window to create a custom policy:

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
        spec:
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: v1
                kind: Pod
                metadata:
                  name: sample-nginx-pod
                  namespace: default
                status:
                  phase: Running
          remediationAction: inform
          severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
```

```
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      []  # selects all clusters if not specified
```

11. Replace the content inside the **objectDefinition** section of the template with content that is similar to the following content to add the settings for your ImageContentSourcePolicy:

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: ImageContentSourcePolicy
spec:
  repositoryDigestMirrors:
  - mirrors:
    - <path-to-local-mirror>
    source: registry.redhat.io
```

- Replace *path-to-local-mirror* with the path to your local mirror repository.

- Tip: You can find your path to your local mirror by entering the **oc adm release mirror** command.

12. Select the box for **Enforce if supported**.

13. Select **Create** to create the policy.

9.1.8. Deploy a policy to deploy a disconnected catalog source

Push the *Catalogsource* policy to the managed cluster to change the default location from a connected location to your disconnected local registry.

1. In the Red Hat Advanced Cluster Management console, select **Automate infrastructure** > **Clusters**.

2. Find the managed cluster to receive the policy in the list of clusters.

3. Note the value of the **name** label for the managed cluster. The label format is **name=managed-cluster-name**. This value is used when pushing the policy.

4. In the Red Hat Advanced Cluster Management console menu, select **Govern risk** > **Create policy**.

5. Set the **YAML** switch to *On* to view the YAML version of the policy.

6. Delete all of the content in the **YAML** code.

7. Paste the following **YAML** content into the window to create a custom policy:

8. Paste the following **YAML** content into the window to create a custom policy:

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
        spec:
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: v1
                kind: Pod
                metadata:
                  name: sample-nginx-pod
                  namespace: default
                status:
                  phase: Running
          remediationAction: inform
          severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
```

```
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      []  # selects all clusters if not specified
```

9. Add the following content to the policy:

   ```
   apiVersion: config.openshift.io/vi
   kind: OperatorHub
   metadata:
    name: cluster
   spec:
    disableAllDefaultSources: true
   ```

10. Add the following content:

    ```
    apiVersion: operators.coreos.com/v1alpha1
    kind: CatalogSource
    metadata:
      name: my-operator-catalog
      namespace: openshift-marketplace
    spec:
      sourceType: grpc
      image: <registry_host_name>:<port>/olm/redhat-operators:v1
      displayName: My Operator Catalog
      publisher: grpc
    ```

    Replace the value of *spec.image* with the path to your local restricted catalog source image.

11. In the Red Hat Advanced Cluster Management console navigation, select **Automate infrastructure** > **Clusters** to check the status of the managed cluster. When the policy is applied, the cluster status is **ready**.

9.1.9. Deploy a policy to change the managed cluster parameter

Push the *ClusterVersion* policy to the managed cluster to change the default location where it retrieves its upgrades.

1. From the managed cluster, confirm that the *ClusterVersion* upstream parameter is currently the default public OpenShift Update Service operand by entering the following command:

   ```
   oc get clusterversion -o yaml
   ```

   The returned content might resemble the following content:

   ```
   apiVersion: v1
   items:
   - apiVersion: config.openshift.io/v1
   ```

```
  kind: ClusterVersion
[..]
  spec:
    channel: stable-4.4
    upstream: https://api.openshift.com/api/upgrades_info/v1/graph
```

2. From the hub cluster, identify the route URL to the OpenShift Update Service operand by entering the following command: **oc get routes**.
   Tip: Note this value for later steps.

3. In the hub cluster Red Hat Advanced Cluster Management console menu, select **Govern risk** > **Create a policy**.

4. Set the **YAML** switch to *On* to view the YAML version of the policy.

5. Delete all of the content in the **YAML** code.

6. Paste the following **YAML** content into the window to create a custom policy:

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
        spec:
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: v1
                kind: Pod
                metadata:
                  name: sample-nginx-pod
                  namespace: default
                status:
                  phase: Running
          remediationAction: inform
          severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
```

```
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      []  # selects all clusters if not specified
```

7. Add the following content to **policy.spec** in the *policy* section:

   ```
   apiVersion: config.openshift.io/v1
     kind: ClusterVersion
     metadata:
       name: version
     spec:
       channel: stable-4.4
       upstream: https://example-cincinnati-policy-engine-uri/api/upgrades_info/v1/graph
   ```

   Replace the value of *spec.upstream* with the path to your hub cluster OpenShift Update Service operand.

   Tip: You can complete the following steps to determine the path to the operand:

   a. Run the **oc get get routes -A** command on the hub cluster.

   b. Find the route to **cincinnati**. + The path to the operand is the value in the **HOST/PORT** field.

8. In the managed cluster CLI, confirm that the upstream parameter in the **ClusterVersion** is updated with the local hub cluster OpenShift Update Service URL by entering:

   ```
   oc get clusterversion -o yaml
   ```

   Verify that the results resemble the following content:

   ```
   apiVersion: v1
   items:
   - apiVersion: config.openshift.io/v1
     kind: ClusterVersion
   [..]
   ```

```
spec:
  channel: stable-4.4
  upstream: https://<hub-cincinnati-uri>/api/upgrades_info/v1/graph
```

9.1.10. Viewing available upgrades

You can view a list of available upgrades for your managed cluster by completing the following steps:

1. Log in to your Red Hat Advanced Cluster Management console.
2. In the navigation menu, select **Automate Infrastructure** > **Clusters**.
3. Select a cluster that is in the *Ready* state.
4. From the **Actions** menu, select **Upgrade cluster**.
5. Verify that the optional upgrade paths are available.
   Note: No available upgrade versions are shown if the current version is not mirrored into the local image repository.

9.1.11. Upgrading the cluster

After configuring the disconnected registry, Red Hat Advanced Cluster Management and OpenShift Update Service use the disconnected registry to determine if upgrades are available. If no available upgrades are displayed, make sure that you have the release image of the current level of the cluster and at least one later level mirrored in the local repository. If the release image for the current version of the cluster is not available, no upgrades are available.

Complete the following steps to upgrade:

1. In the Red Hat Advanced Cluster Management console, select **Automate infrastructure** > **Clusters**.
2. Find the cluster that you want to determine if there is an available upgrade.
3. If there is an upgrade available, the **Distribution version** column for the cluster indicates that there is an upgrade available.
4. Select the *Options* menu for the cluster, and select **Upgrade cluster**.
5. Select the target version for the upgrade, and select **Upgrade**.

The managed cluster is updated to the selected version.