Red Hat Advanced Cluster Management for Kubernetes 2.5

Install

Read more about installing on connected and disconnected networks, requirements and recommendations for installation, multicluster advanced configurations, and instructions for upgrading and uninstalling.

Last Updated: 2023-07-13

Red Hat Advanced Cluster Management for Kubernetes 2.5 Install

Read more about installing on connected and disconnected networks, requirements and recommendations for installation, multicluster advanced configurations, and instructions for upgrading and uninstalling.

Legal Notice

Abstract

Read more about installing on connected and disconnected networks, requirements and recommendations for installation, multicluster advanced configurations, and instructions for upgrading and uninstalling.

Table of Contents

CHAPTER 1. INSTALLING

Learn how to install and uninstall Red Hat Advanced Cluster Management for Kubernetes. Before you install Red Hat Advanced Cluster Management for Kubernetes, review the required hardware and system configuration for each product. You can install the Red Hat Advanced Cluster Management for Kubernetes online on Linux with a supported version of Red Hat OpenShift Container Platform.

1. You must have a supported version of OpenShift Container Platform. For example, you can use Red Hat OpenShift Service on AWS, or Red Hat OpenShift Dedicated.

2. You must install the operator for Red Hat Advanced Cluster Management for Kubernetes from the catalog.

FIPS notice: If you do not specify your own ciphers in **spec.ingress.sslCiphers**, then the **multiclusterhub-operator** provides a default list of ciphers. For 2.3, this list includes two ciphers that are *not* FIPS approved. If you upgrade from a version 2.3.x or earlier and want FIPS compliance, remove the following two ciphers from the **multiclusterhub** resource: **ECDHE-ECDSA-CHACHA20-POLY1305** and **ECDHE-RSA-CHACHA20-POLY1305**.

Installing Red Hat Advanced Cluster Management for Kubernetes sets up a multi-node cluster production environment. You can install Red Hat Advanced Cluster Management for Kubernetes in either standard or high-availability configurations. View the following documentation for more information about the installation procedure:

- Requirements and recommendations
- Sizing your cluster
- Performance and scalability
- Installing while connected online
- Install on disconnected networks
- MultiClusterHub advanced configuration
- Network configuration
- Upgrading by using the operator
- Upgrading OpenShift Container Platform
- Uninstalling

1.1. REQUIREMENTS AND RECOMMENDATIONS

Before you install Red Hat Advanced Cluster Management for Kubernetes, review the following system configuration requirements and settings:

- Supported operating systems and platforms
- Supported browsers

1.1.1. Supported operating systems and platforms

To see recent information about hub cluster and managed cluster platforms, refer to the Red Hat Advanced Cluster Management 2.5 Support matrix.

1.1.2. Supported browsers

You can access the Red Hat Advanced Cluster Management console from Mozilla Firefox, Google Chrome, Microsoft Edge, and Safari. See the following versions that are tested and supported:

Platform	Supported browsers
Microsoft Windows	Microsoft Edge - 44 or later, Mozilla Firefox - 82.0 or later, Google Chrome - Version 86.0 and later
Linux	Mozilla Firefox - 82.0 and later, Google Chrome - Version 86.0 and later
macOS	Mozilla Firefox - 82.0 and later, Google Chrome - Version 86.0 and later, Safari - 14.0 and later

1.1.3. Network configuration

Configure your network settings to allow the connections in the following sections.

1.1.3.1. Hub cluster networking requirements

For the hub cluster networking requirements, see the following table:

Direction	Protocol	Connection	Port (if specified)
Outbound to managed cluster	HTTPS	Retrieval of logs dynamically from Search console for the pods of the managed cluster. This connection creates a route called **klusterlet-addon-workmgr** in the **open-cluster-management-agent-addon** namespace of the managed cluster. The host of the route is **<route name>-<namespace>.apps.<cluster domain>**.	443
Outbound to managed cluster	HTTPS	Kubernetes API server of the managed cluster that is provisioned during installation to install the Klusterlet	6443

Direction	Protocol	Connection	Port (if specified)
Outbound to the channel source	HTTPS	The channel source, including GitHub, Object Store, and Helm repository. This is only required when you are using Application lifecycle, OpenShift GitOps or ArgoCD to connect to these sources.	443
Inbound from the managed cluster	HTTPS	Managed cluster to push metrics and alerts (alerts are gathered only for managed clusters running OpenShift Container Platform version 4.8, or later)	443
Inbound from the managed cluster	HTTPS	Kube API Server of hub cluster being watched for changes from managed cluster	6443
Outbound to **ObjectStore**	HTTPS	Sends metric data of Observability for long term storage in the ObjectStore **and/or** when the Cluster Backup Operator is running.	443
Outbound to image repository	HTTPS	Access images for OpenShift Container Platform and Red Hat Advanced Cluster Management	443

1.1.3.2. Managed cluster networking requirements

Note: Registration Agent and **Work Agent** on the managed cluster do not support proxy settings because they communicate with **apiserver** on the hub cluster by establishing an mTLS connection, which cannot pass through the proxy.

For the managed cluster networking requirements, see the following table:

Direction	Protocol	Connection	Port (if specified)
Inbound from the hub cluster	HTTPS	Sending of logs dynamically for the pods of the managed cluster. This connection uses a service running on the managed cluster called - **klusterlet-addon-workmgr**	443
Inbound from the hub cluster	HTTPS	Kubernetes API server of the managed cluster that is provisioned during installation to install the Klusterlet	6443
Outbound to image repository	HTTPS	Access images for OpenShift Container Platform and Red Hat Advanced Cluster Management	443
Outbound to the hub cluster	HTTPS	Managed cluster to push metrics and alerts (alerts are gathered only for managed clusters running OpenShift Container Platform version 4.8, or later)	443
Outbound to the hub cluster	HTTPS	Watches the Kubernetes API server of the hub cluster for changes	6443
Outbound to the channel source	HTTPS	The managed cluster to the channel source, which includes GitHub, Object Store, and Helm repository. This is only required when you are using application lifecycle to connect to these sources.	443
Outbound to the hub cluster	HTTPS	For cluster-proxy add-on on the managed cluster to register.	443

1.2. PERFORMANCE AND SCALABILITY

Red Hat Advanced Cluster Management for Kubernetes is tested to determine certain scalability and performance data. The major areas that are tested are cluster scalability and search performance.

You can use this information to help you plan your environment.

Note: Data is based on the results from a lab environment at the time of testing. Your results might vary, depending on your environment, network speed, and changes to the product.

- Maximum number of managed clusters
- Search scalability
- Scaling for observability

1.2.1. Maximum number of managed clusters

The maximum number of clusters that Red Hat Advanced Cluster Management can manage varies based on several factors, including:

- Number of resources in the cluster, which depends on factors like the number of policies and applications that are deployed.
- Configuration of the hub cluster, such as how many pods are used for scaling.

The following table shows the configuration information for the clusters on the Amazon Web Services cloud platform that were used during this testing:

Node	Flavor	vCPU	RAM (GiB)	Disk type	Disk size (GiB)	Count	Region
Master	m5.2xlarge	8	32	gp2	100	3	us-east-1
Worker or Infrastructure	m5.2xlarge	8	32	gp2	100	3 or 5 nodes	us-east-1

For more information about infrastructure nodes, see, Installing the Red Hat Advanced Cluster Management hub cluster on infrastructure nodes. Also see Creating infrastructure machine sets.

1.2.2. Search scalability

The scalability of the Search component depends on the performance of the data store. The following variables are important when analyzing the search performance:

- Physical memory
- Write throughput (Cache recovery time)
- Query execution time

1.2.2.1. Physical memory

Search keeps the data in-memory to achieve fast response times. The memory required is proportional to the number of Kubernetes resources and their relationships in the cluster.

Clusters	Kubernetes resources	Relationships	Observed size (with simulated data)
1 medium	5000	9500	50 Mi
5 medium	25,000	75,000	120 Mi
15 medium	75,000	20,0000	492 Mi
30 medium	150,000	450,000	1 Gi
50 medium	250,000	750,000	2 Gi

For more information on how you can change the amount of memory used for the search component, see Options to increase the redisgraph memory.

1.2.2.2. Write throughput (cache recovery time)

Most clusters in steady state generate a small number of resource updates. The highest rate of updates happen when the data in RedisGraph is cleared, which causes the remote collectors to synchronize their full state around the same time. When the datastore is cleared, recovery times are measured for a different number of managed clusters.

Clusters	Kubernetes resources	Relationships	Average recovery time from simulation
1 medium	5000	9500	less than 2 seconds
5 medium	25,000	75,000	less than 15 seconds
15 medium	75,000	200,000	2 minutes and 40 seconds
30 medium	150,000	450,000	5-8 minutes

Remember: Times might increase for clusters that have a slow network connection to the hub. The write throughput information that is previously stated is applicable only if **persistence** is disabled.

1.2.2.3. Query execution considerations

There are some things that can affect the time that it takes to run and return results from a query. Consider the following items when planning and configuring your environment:

- Searching for a keyword is not efficient.

If you search for **RedHat** and you manage a large number of clusters, it might take a longer time to receive search results.

- The first search takes longer than later searches because it takes additional time to gather user role-based access control rules.
- The length of time to complete a request is proportional to the number of namespaces and resources the user is authorized to access.
  Note: If you save and share a Search query with another user, returned results depend on access level for that user. For more information on role access, see Using RBAC to define and apply permissions in the OpenShift Container Platform documentation.
- The worst performance is observed for a request by a non-administrator user with access to all of the namespaces, or all of the managed clusters.

1.2.3. Scaling for observability

You need to plan your environment if you want to enable and use the observability service. The resource consumption later is for the OpenShift Container Platform project, where observability components are installed. Values that you plan to use are sums for all observability components.

Note: Data is based on the results from a lab environment at the time of testing. Your results might vary, depending on your environment, network speed, and changes to the product.

1.2.3.1. Sample observability environment

In the sample environment, hub clusters and managed clusters are located in Amazon Web Services cloud platform and have the following topology and configuration:

Node	Flavor	vCPU	RAM (GiB)	Disk type	Disk size (GiB)	Count	Region
Master node	m5.4xlarge	16	64	gp2	100	3	sa-east-1
Worker node	m5.4xlarge	16	64	gp2	100	3	sa-east-1

The observability deployment is configured for high availability environments. With a high availability environment, each Kubernetes deployment has two instances, and each StatefulSet has three instances.

During the sample test, a different number of managed clusters are simulated to push metrics, and each test lasts for 24 hours. See the following throughput:

1.2.3.2. Write throughput

Pods	Interval (minute)	Time series per min
400	1	83000

1.2.3.3. CPU usage (millicores)

CPU usage is stable during testing:

Size	CPU Usage
10 clusters	400
20 clusters	800

1.2.3.4. RSS and working set memory

View the following descriptions of the RSS and working set memory:

- **Memory usage RSS:** From the metrics **container_memory_rss** and remains stable during the test.
- **Memory usage working set:** From the metrics **container_memory_working_set_bytes**, increases along with the test.

The following results are from a 24-hour test:

Size	Memory usage RSS	Memory usage working set
10 clusters	9.84	4.93
20 clusters	13.10	8.76

1.2.3.5. Persistent volume for thanos-receive component

Important: Metrics are stored in **thanos-receive** until retention time (four days) is reached. Other components do not require as much volume as **thanos-receive** components.

Disk usage increases along with the test. Data represents disk usage after one day, so the final disk usage is multiplied by four.

See the following disk usage:

Size	Disk usage (GiB)
10 clusters	2
20 clusters	3

1.2.3.6. Network transfer

During tests, network transfer provides stability. See the sizes and network transfer values:

Size	Inbound network transfer	Outbound network transfer
10 clusters	6.55 MBs per second	5.80 MBs per second
20 clusters	13.08 MBs per second	10.9 MBs per second

1.2.3.7. Amazon Simple Storage Service (S3)

Total usage in Amazon Simple Storage Service (S3) increases. The metrics data is stored in S3 until default retention time (five days) is reached. See the following disk usages:

Size	Disk usage (GiB)
10 clusters	16.2
20 clusters	23.8

1.2.4. Sizing your hub cluster

Each Red Hat Advanced Cluster Management for Kubernetes cluster is unique and the following guidelines provide sample deployment sizes for you. Recommendations are classified by size and purpose. Red Hat Advanced Cluster Management applies the following dimensions for sizing and placement of supporting services:

- Availability Zones isolate potential fault domains across the cluster. Typical clusters should have nearly equivalent worker node capacity in three or more availability zones.
- vCPU reservations and limits, which establish vCPU capacity on a worker node to assign to a container. A vCPU is equivalent to a Kubernetes compute unit. For more information, see Kubernetes Meaning of CPU.
- Memory reservations and limits, which establish memory capacity on a worker node to assign to a container.
- Persistent data, which is managed by the product and stored in the etcd cluster that is used by Kubernetes.

Important: For OpenShift Container Platform, distribute the master nodes of the cluster across three (3) availability zones.

1.2.4.1. Product environment

Note: The following requirements are *not* minimum requirements.

Table 1.1. Product environment

Node type	Availability zones	etcd	Total reserved memory	Total reserved CPU

Master	3	3	Per OpenShift Container Platform sizing guidelines	Per OpenShift Container Platform sizing guidelines
Worker or infrastructure	3	1	12 GB	6

In addition to Red Hat Advanced Cluster Management, the OpenShift Container Platform cluster runs additional services to support cluster features. See Installing the Red Hat Advanced Cluster Management hub cluster on infrastructure nodes for more details.

1.2.4.1.1. OpenShift Container Platform on Amazon Web Services

See the Amazon Web Services information in the OpenShift Container Platform product documentation for more information.

Also learn more about machine types.

- Node count: 3
- Availability zones: 3
- Instance size: m5.xlarge
  - vCPU: 4
  - Memory: 16 GB
  - Storage size: 120 GB

1.2.4.1.2. OpenShift cluster on Google Cloud Platform

See the Google Cloud Platform product documentation for more information about quotas.

Also learn more about machine types.

- Node count: 3
- Availability zones: 3
- Instance size: N1-standard-4 (0.95–6.5 GB)
  - vCPU: 4
  - Memory: 15 GB
  - Storage size: 120 GB

1.2.4.1.3. OpenShift cluster on Microsoft Azure

See the following product documentation for more details.

- Node count: 3

- Availability zones: 3
- Instance size: Standard_D4_v3
  - vCPU: 4
  - Memory: 16 GB
  - Storage size: 120 GB

1.2.4.1.4. OpenShift cluster on VMware vSphere

See the following product documentation for more details.

- Node count: 3
- Availability zones: 3
- Instance size:
  - Memory: 16 GB
  - Storage size: 120 GB
  - vCPUs: 4
  - Cores per socket: 2

1.2.4.1.5. OpenShift Container Platform on IBM Z systems

See Installing a cluster on IBM Z systems in the OpenShift Container Platform documentation for more information.

- Node count: 3
- Availability zones: 3
- Instance size:
  - Memory: 16 GB
  - Storage size: 100 GB
  - vCPU: 10
    IBM Z systems provide the ability to configure simultaneous multithreading (SMT), which extends the number of vCPUs that can run on each core. If you configured SMT, One physical core (IFL) provides two logical cores (threads). The hypervisor can provide two or more vCPUs.

    One vCPU is equivalent to one physical core when simultaneous multithreading (SMT), or hyperthreading, is not enabled. When enabled, use the following formula to calculate the corresponding ratio: (threads per core × cores) × sockets = vCPUs.

    For more information about SMT, see Simultaneous multithreading.

1.2.4.1.6. OpenShift Container Platform on IBM Power systems

See Installing a cluster on Power systems in the OpenShift Container Platform documentation for more information.

- Node count: 3
- Availability zones: 3
- Instance size:
  - Memory: 16 GB
  - Storage size: 120 GB
  - vCPU: 16
    IBM Power systems provide the ability to configure simultaneous multithreading (SMT), which extends the number of vCPUs that can run on each core. If you configured SMT, your SMT level determines how you satisfy the 16 vCPU requirement. The most common configurations are:
    - Two cores running on SMT-8 (the default configuration for systems that are running IBM PowerVM) provides the required 16 vCPUs.
    - Four cores running on SMT-4 provides the required 16 vCPUs.
      For more information about SMT, see Simultaneous multithreading.

1.2.4.1.7. OpenShift Container Platform cluster on bare metal assets

See the following product documentation for more details.

A Red Hat Advanced Cluster Management for Kubernetes hub cluster can be installed and supported on OpenShift Container Platform bare metal. The hub cluster can run on a compact bare metal topology, in which there are 3 schedulable control plane nodes, and 0 additional workers.

- Node count: 3
- Availability zones: 3
- Instance size:
  - Memory: 16 GB
  - Storage size: 120 GB
  - vCPUs: 4

1.2.4.1.8. Creating and managing single node OpenShift Container Platform clusters

See example requirements for creating and managing 2200 single node OpenShift Container Platform clusters. See the minimum requirements for using Red Hat Advanced Cluster Management to create single node OpenShift (SNO) clusters (230 and more provisioned at the same time), and manage those SNO clusters with a hub cluster:

- Master (schedulable)
  - Node count: 3
  - Memory: 289 GB (cluster max)

- Memory: 110 GB (single node max)
- CPU cluster max: 90
- CPU single node max: 44

Note: The CPU utilization values peaked while multiple clusters were created at the same time.

1.3. INSTALLING WHILE CONNECTED ONLINE

Red Hat Advanced Cluster Management for Kubernetes is installed through Operator Lifecycle Manager, which manages the installation, upgrade, and removal of the components that encompass the Red Hat Advanced Cluster Management hub cluster.

Before you get started, review the Requirements and recommendations section, then see the following documentation:

Required access: Cluster administrator. **OpenShift Container Platform Dedicated environment required access:** You must have **cluster-admin** permissions. By default **dedicated-admin** role does not have the required permissions to create namespaces in the OpenShift Container Platform Dedicated environment.

- By default, the hub cluster components are installed on worker nodes of your OpenShift Container Platform cluster without any additional configuration. You can install the hub cluster on worker nodes by using the OpenShift Container Platform OperatorHub web console interface, or by using the OpenShift Container Platform CLI.
- If you have configured your OpenShift Container Platform cluster with infrastructure nodes, you can install the hub cluster on those infrastructure nodes by using the OpenShift Container Platform CLI with additional resource parameters. See the *Installing the Red Hat Advanced Cluster Management hub cluster on infrastructure node* section for more details.
- If you plan to import Kubernetes clusters that were not created by OpenShift Container Platform or Red Hat Advanced Cluster Management, you need to configure an image pull secret.

For information on how to configure advanced configurations, see options in the MultiClusterHub advanced configuration section of the documentation.

- Prerequisites
- Confirm your OpenShift Container Platform installation
- Installing from the OperatorHub web console interface
- Installing from the OpenShift Container Platform CLI
- Installing the Red Hat Advanced Cluster Management hub cluster on infrastructure nodes

1.3.1. Prerequisites

Before you install Red Hat Advanced Cluster Management, see the following requirements:

- Your Red Hat OpenShift Container Platform cluster must have access to the Red Hat Advanced Cluster Management operator in the OperatorHub catalog from the OpenShift Container Platform console.

- You need access to the catalog.redhat.com.
- OpenShift Container Platform version 4.8, or later, must be deployed in your environment, and you must be logged into with the OpenShift Container Platform CLI. OpenShift Container Platform version 4.8, or later, must be deployed in your environment, and you must be logged into with the OpenShift Container Platform CLI. See the following install documentation for OpenShift Container Platform and change to earlier versions of needed: OpenShift Container Platform version 4.10
- Your OpenShift Container Platform command line interface (CLI) must be configured to run **oc** commands. See Getting started with the CLI for information about installing and configuring the OpenShift Container Platform CLI.
- Your OpenShift Container Platform permissions must allow you to create a namespace. Without a namespace, installation will fail.
- You must have an Internet connection to access the dependencies for the operator.
- **Important:** To install in a OpenShift Container Platform Dedicated environment, see the following requirements:
  - You must have the OpenShift Container Platform Dedicated environment configured and running.
  - You must have **cluster-admin** authority to the OpenShift Container Platform Dedicated environment where you are installing the hub cluster.
  - To import, you must use the **stable-2.0** channel of the klusterlet operator for 2.5.

1.3.2. Confirm your OpenShift Container Platform installation

You must have a supported OpenShift Container Platform version, including the registry and storage services, installed and working. For more information about installing OpenShift Container Platform, see the OpenShift Container Platform documentation.

1. Verify that a Red Hat Advanced Cluster Management hub cluster is not already installed on your OpenShift Container Platform cluster. Red Hat Advanced Cluster Management allows only one single Red Hat Advanced Cluster Management hub cluster installation on each OpenShift Container Platform cluster. Continue with the following steps if there is no Red Hat Advanced Cluster Management hub cluster installed:
2. To ensure that the OpenShift Container Platform cluster is set up correctly, access the OpenShift Container Platform web console with the following command:

   ```
   kubectl -n openshift-console get route
   ```

   See the following example output:

   ```
   openshift-console console console-openshift-console.apps.new-coral.purple-chesterfield.com
   console   https   reencrypt/Redirect     None
   ```

3. Open the URL in your browser and check the result. If the console URL displays **console-openshift-console.router.default.svc.cluster.local**, set the value for **openshift_master_default_subdomain** when you install OpenShift Container Platform. See the following example of a URL: **https://console-openshift-console.apps.new-coral.purple-chesterfield.com**.

You can proceed to install Red Hat Advanced Cluster Management from the console or the CLI. Both procedures are documented.

1.3.3. Installing from the OperatorHub web console interface

Best practice: From the *Administrator* view in your OpenShift Container Platform navigation, install the OperatorHub web console interface that is provided with OpenShift Container Platform.

1. Select **Operators** > **OperatorHub** to access the list of available operators, and select *Advanced Cluster Management for Kubernetes* operator.

2. On the *Operator subscription* page, select the options for your installation:

   - Namespace information:

     - The Red Hat Advanced Cluster Management hub cluster must be installed in its own namespace, or project.

     - By default, the OperatorHub console installation process creates a namespace titled **open-cluster-management**. **Best practice:** Continue to use the **open-cluster-management** namespace if it is available.

     - If there is already a namespace named **open-cluster-management**, choose a different namespace.

   - Channel: The channel that you select corresponds to the release that you are installing. When you select the channel, it installs the identified release, and establishes that the future Errata updates within that release are obtained.

   - Approval strategy for updates: The approval strategy identifies the human interaction that is required for applying updates to the channel or release to which you subscribed.

     - Select **Automatic** to ensure any updates within that release are automatically applied.

     - Select **Manual** to receive a notification when an update is available. If you have concerns about when the updates are applied, this might be best practice for you.

   Important: To upgrade to the next minor release, you must return to the *OperatorHub* page and select a new channel for the more current release.

3. Select **Install** to apply your changes and create the operator.

4. Create the *MultiClusterHub* custom resource.

   a. In the OpenShift Container Platform console navigation, select **Installed Operators** > **Advanced Cluster Management for Kubernetes**.

   b. Select the **MultiClusterHub** tab.

   c. Select **Create MultiClusterHub**.

   d. Update the default values in the YAML file. See options in the *MultiClusterHub advanced configuration* section of the documentation.

      - The following example shows the default template. Confirm that **namespace** is your project namespace. See the sample:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
```

5. Select **Create** to initialize the custom resource. It can take up to 10 minutes for the Red Hat Advanced Cluster Management hub cluster to build and start.
   After the Red Hat Advanced Cluster Management hub cluster is created, the **MultiClusterHub** resource status displays *Running* from the *MultiClusterHub* tab of the Red Hat Advanced Cluster Management operator details. You can now access the console for the Red Hat Advanced Cluster Management hub cluster. See the following steps:

6. In the OpenShift Container Platform console navigation, select **Networking** > **Routes**.

7. View the URL for your Red Hat Advanced Cluster Management hub cluster in the list, and navigate to it to access the console.

1.3.4. Installing from the OpenShift Container Platform CLI

1. Create a Red Hat Advanced Cluster Management hub cluster namespace where the operator requirements are contained. Run the following command, where **namespace** is the name for your Red Hat Advanced Cluster Management hub cluster namespace. The value for **namespace** might be referred to as *Project* in the OpenShift Container Platform environment:

```
oc create namespace <namespace>
```

2. Switch your project namespace to the one that you created. Replace **namespace** with the name of the Red Hat Advanced Cluster Management hub cluster namespace that you created in step 1.

```
oc project <namespace>
```

3. Create a YAML file to configure an **OperatorGroup** resource. Each namespace can have only one operator group. Replace **default** with the name of your operator group. Replace **namespace** with the name of your project namespace. See the following sample:

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <default>
spec:
  targetNamespaces:
  - <namespace>
```

4. Run the following command to create the **OperatorGroup** resource. Replace **operator-group** with the name of the operator group YAML file that you created:

```
oc apply -f <path-to-file>/<operator-group>.yaml
```

5. Create a YAML file to configure an OpenShift Container Platform subscription. Your file is similar to the following sample:

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: acm-operator-subscription
spec:
  sourceNamespace: openshift-marketplace
  source: redhat-operators
  channel: release-2.5
  installPlanApproval: Automatic
  name: advanced-cluster-management
```

Note: For installing the Red Hat Advanced Cluster Management hub cluster on infrastructure nodes, the see the Operator Lifecycle Manager Subscription additional configuration section.

6. Run the following command to create the OpenShift Container Platform subscription. Replace **subscription** with the name of the subscription file that you created:

```
oc apply -f <path-to-file>/<subscription>.yaml
```

7. Create a YAML file to configure the **MultiClusterHub** custom resource. Your default template should look similar to the following example. Replace **namespace** with the name of your project namespace:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec: {}
```

Note: For installing the Red Hat Advanced Cluster Management hub cluster on infrastructure nodes, see the MultiClusterHub custom resource additional configuration section:

8. Run the following command to create the **MultiClusterHub** custom resource. Replace **custom-resource** with the name of your custom resource file:

```
oc apply -f <path-to-file>/<custom-resource>.yaml
```

If this step fails with the following error, the resources are still being created and applied. Run the command again in a few minutes when the resources are created:

```
error: unable to recognize "./mch.yaml": no matches for kind "MultiClusterHub" in version
"operator.open-cluster-management.io/v1"
```

9. Run the following command to get the custom resource. It can take up to 10 minutes for the **MultiClusterHub** custom resource status to display as **Running** in the **status.phase** field after you run the command:

```
oc get mch -o=jsonpath='{.items[0].status.phase}'
```

10. After the status is **Running**, view the list of routes to find your route:

```
oc get routes
```

If you are reinstalling Red Hat Advanced Cluster Management and the pods do not start, see Troubleshooting reinstallation failure for steps to work around this problem.

Notes:

- A **ServiceAccount** with a **ClusterRoleBinding** automatically gives cluster administrator privileges to Red Hat Advanced Cluster Management and to any user credentials with access to the namespace where you install Red Hat Advanced Cluster Management.
- The installation also creates a namespace called **local-cluster** that is reserved for the Red Hat Advanced Cluster Management hub cluster when it is managed by itself. There cannot be an existing namespace called **local-cluster**. For security reasons, do not release access to the **local-cluster** namespace to any user who does not already have **cluster-administrator** access.

1.3.5. Installing the Red Hat Advanced Cluster Management hub cluster on infrastructure nodes

An OpenShift Container Platform cluster can be configured to contain infrastructure nodes for running approved management components. Running components on infrastructure nodes avoids allocating OpenShift Container Platform subscription quota for the nodes that are running those management components.

After adding infrastructure nodes to your OpenShift Container Platform cluster, follow the Installing from the OpenShift Container Platform CLI instructions and add configurations to the Operator Lifecycle Manager subscription and **MultiClusterHub** custom resource.

1.3.5.1. Add infrastructure nodes to the OpenShift Container Platform cluster

Follow the procedures that are described in Creating infrastructure machine sets in the OpenShift Container Platform documentation. Infrastructure nodes are configured with a Kubernetes **taint** and **label** to keep non-management workloads from running on them.

To be compatible with the infrastructure node enablement provided by Red Hat Advanced Cluster Management, ensure your infrastructure nodes have the following **taint** and **label** applied:

```
metadata:
  labels:
    node-role.kubernetes.io/infra: ""
spec:
  taints:
  - effect: NoSchedule
    key: node-role.kubernetes.io/infra
```

1.3.5.2. Operator Lifecycle Manager Subscription additional configuration

Add the following additional configuration before applying the Operator Lifecycle Manager Subscription:

```
spec:
  config:
    nodeSelector:
      node-role.kubernetes.io/infra: ""
    tolerations:
```

```
  - key: node-role.kubernetes.io/infra
    effect: NoSchedule
    operator: Exists
```

1.3.5.3. MultiClusterHub custom resource additional configuration

Add the following additional configuration before applying the **MultiClusterHub** custom resource:

```
spec:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
```

1.4. INSTALL ON DISCONNECTED NETWORKS

You might need to install Red Hat Advanced Cluster Management for Kubernetes on Red Hat OpenShift Container Platform clusters that are not connected to the Internet. The procedure to install on a disconnected hub requires some of the same steps as the connected installation.

You must download copies of the packages to access them during the installation, rather than accessing them directly from the network during the installation.

Before you get started, review the Requirements and recommendations section, then see the following documentation:

- Prerequisites
- Confirm your OpenShift Container Platform installation
- Installing in a disconnected environment

1.4.1. Prerequisites

You must meet the following requirements before you install Red Hat Advanced Cluster Management for Kubernetes:

- Red Hat OpenShift Container Platform version 4.8 or later must be deployed in your environment, and you must be logged in with the command line interface (CLI).
- You need access to the catalog.redhat.com.
  Note: For managing bare metal clusters, you must have OpenShift Container Platform version 4.8 or later.

  See the OpenShift Container Platform version 4.10 , OpenShift Container Platform version 4.10 .
- Your Red Hat OpenShift Container Platform CLI must be version 4.8 or later, and configured to run **oc** commands. See Getting started with the CLI for information about installing and configuring the Red Hat OpenShift CLI.
- Your Red Hat OpenShift Container Platform permissions must allow you to create a namespace. Installation fails without a namespace.
- You must have a workstation with Internet connection to download the dependencies for the operator.

1.4.2. Confirm your OpenShift Container Platform installation

- You must have a supported OpenShift Container Platform version, including the registry and storage services, installed and working in your cluster. For information about OpenShift Container Platform version 4.10, see the OpenShift Container Platform Documentation .

- When and if you are connected, run the **kubectl -n openshift-console get route** command to access the OpenShift Container Platform web console. See the following example output:

```
openshift-console  console  console-openshift-console.apps.new-coral.purple-chesterfield.com  console  https  reencrypt/Redirect  None
```

  The console URL in this example is: **https:// console-openshift-console.apps.new-coral.purple-chesterfield.com**. Open the URL in your browser and check the result.

  If the console URL displays **console-openshift-console.router.default.svc.cluster.local**, set the value for **openshift_master_default_subdomain** when you install OpenShift Container Platform.

See Sizing your cluster to learn about setting up capacity for your hub cluster.

1.4.3. Installing in a disconnected environment

Important: You need to download the required images to a mirroring registry to install the operators in a disconnected environment. Without the download, you might receive **ImagePullBackOff** errors during your deployment.

Follow these steps to install Red Hat Advanced Cluster Management in a disconnected environment:

1. Create a mirror registry. If you do not already have a mirror registry, create one by completing the procedure in the Mirroring images for a disconnected installation topic of the Red Hat OpenShift Container Platform documentation.
   If you already have a mirror registry, you can configure and use your existing one.

   Note: Ensure you follow the steps in the OpenShift Container Platform documentation at Populating OperatorHub from mirrored Operator catalogs .

2. Mirror operator catalogs. Ensure that the operator catalogs are mirrored by following the procedure in Mirroring Operator catalogs for use with disconnected clusters .

Notes:

- If you are pruning packages from the existing Red Hat Operators index image, ensure that both the **advanced-cluster-management** and **multicluster-engine** packages are pruned. See Filtering a SQLite-based index image for more information.

- During the process for Generated manifests, you will generate a **catalogSource.yaml** file in the manifest directory. You will use this sample file when you configure the disconnected Operator Lifecycle Manager.

- For bare metal only, you need to provide the certificate information for the disconnected registry in your **install-config.yaml** file. To access the image in a protected disconnected registry, you must provide the certificate information so Red Hat Advanced Cluster Management can access the registry.

  a. Copy the certificate information from the registry.

b. Open the **install-config.yaml** file in an editor.

c. Find the entry for **additionalTrustBundle: |**.

d. Add the certificate information after the **additionalTrustBundle** line. The content result is similar to the following example:

```
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  certificate_content
  -----END CERTIFICATE-----
sshKey: >-
```

Important: Additional mirrors for disconnected image registries are needed if the following Governance policies are required:

- Container security operator policy: The images are located in the source **registry.redhat.io/quay**.
- Compliance operator policy: The images are located in the source **registry.redhat.io/compliance**.
- Gatekeeper operator policy: The images are located in the source **registry.redhat.io/rhacm2**.
  See the following example of mirrors lists for all three operators:

```
- mirrors:
  - <your_registry>/rhacm2
  source: registry.redhat.io/rhacm2
- mirrors:
  - <your_registry>/quay
  source: registry.redhat.io/quay
- mirrors:
  - <your_registry>/compliance
  source: registry.redhat.io/compliance
```

1. Save the **install-config.yaml** file.
2. Create a YAML file that contains the **ImageContentSourcePolicy** with the name **rhacm-policy.yaml**. **Note:** If you modify this on a running cluster, it causes a rolling restart of all nodes.

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: rhacm-repo
spec:
  repositoryDigestMirrors:
  - mirrors:
    - mirror.registry.com:5000/rhacm2
    source: registry.redhat.io/rhacm2
```

3. Apply the **ImageContentSourcePolicy** file by entering the following command:

```
oc apply -f rhacm-policy.yaml
```

4. Enable the disconnected Operator Lifecycle Manager Red Hat Operators and Community Operators. Red Hat Advanced Cluster Management is included in the Operator Lifecycle Manager Red Hat Operator catalog.

5. Configure the disconnected Operator Lifecycle Manager for the Red Hat Operator catalog. Follow the steps in the Using Operator Lifecycle Manager on restricted networks topic of the Red Hat OpenShift Container Platform documentation.

   - For the Adding a catalog source to a cluster step, use the **catalogSource.yaml** file that you created when mirroring the operator catalog.

   - If you use your own **catalogSource.yaml** file and the catalog source name is different from the expected **redhat-operator-index**, you will need to add the following annotation to the **MultiClusterHub** custom resource with your catalog source in place of **my-operator-catalog**.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  annotations:
    installer.open-cluster-management.io/mce-subscription-spec: '{"source": "my-operator-catalog"}'
```

Now that you have the image in the disconnected Operator Lifecycle Manager, continue to install Red Hat Advanced Cluster Management for Kubernetes from the Operator Lifecycle Manager catalog.

See Installing while connected online for the required steps, or return to the Installing overview.

1.5. MULTICLUSTERHUB ADVANCED CONFIGURATION

Red Hat Advanced Cluster Management for Kubernetes is installed using an operator that deploys all of the required components. Red Hat Advanced Cluster Management can be further configured during or after installation by adding one or more of the following attributes to the **MultiClusterHub** custom resource:

1.5.1. Custom Image Pull Secret

If you plan to import Kubernetes clusters that were not created by OpenShift Container Platform or Red Hat Advanced Cluster Management, generate a secret that contains your OpenShift Container Platform pull secret information to access the entitled content from the distribution registry.

The secret requirements for OpenShift Container Platform clusters are automatically resolved by OpenShift Container Platform and Red Hat Advanced Cluster Management, so you do not have to create the secret if you are not importing other types of Kubernetes clusters to be managed. Your OpenShift Container Platform pull secret is associated with your Red Hat Customer Portal ID, and is the same across all Kubernetes providers.

Important: These secrets are namespace-specific, so make sure that you are in the namespace that you use for your hub cluster.

1. Go to cloud.redhat.com/openshift/install/pull-secret to download the OpenShift Container Platform pull secret file.

2. Click **Download pull secret**

3. Run the following command to create your secret:

```
oc create secret generic <secret> -n <namespace> --from-file=.dockerconfigjson=<path-to-pull-secret> --type=kubernetes.io/dockerconfigjson
```

- Replace **secret** with the name of the secret that you want to create.
- Replace **namespace** with your project namespace, as the secrets are namespace-specific.
- Replace **path-to-pull-secret** with the path to your OpenShift Container Platform pull secret that you downloaded.

The following example displays the **spec.imagePullSecret** template to use if you want to use a custom pull secret. Replace secret with the name of your pull secret:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  imagePullSecret: <secret>
```

1.5.2. availabilityConfig

The Red Hat Advanced Cluster Management hub cluster has two availabilities: **High** and **Basic**. By default, the hub cluster has an availability of **High**, which gives hub cluster components a **replicaCount** of **2**. This provides better support in cases of failover but consumes more resources than the **Basic** availability, which gives components a **replicaCount** of **1**.

The following examples shows the **spec.availabilityConfig** template with **Basic** availability:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  availabilityConfig: "Basic"
```

1.5.3. nodeSelector

You can define a set of node selectors in the Red Hat Advanced Cluster Management hub cluster to install to specific nodes on your cluster. The following example shows **spec.nodeSelector** to assign Red Hat Advanced Cluster Management pods to nodes with the label **node-role.kubernetes.io/infra**:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
```

1.5.4. tolerations

You can define a list of tolerations to allow the Red Hat Advanced Cluster Management hub cluster to tolerate specific taints defined on the cluster.

The following example shows a **spec.tolerations** that matches a **node-role.kubernetes.io/infra** taint:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  tolerations:
  - key: node-role.kubernetes.io/infra
    effect: NoSchedule
    operator: Exists
```

The previous infra-node toleration is set on pods by default without specifying any tolerations in the configuration. Customizing tolerations in the configuration replaces this default.

1.5.5. disableHubSelfManagement

By default, the Red Hat Advanced Cluster Management hub cluster is automatically imported and managed by itself. This *managed* hub cluster is named, **local-cluster**.

If you do not want the Red Hat Advanced Cluster Management hub cluster to manage itself, you need to change the setting for **spec. disableHubSelfManagement** from **false** to **true**. If the setting is not included in the YAML file that defines the custom resource, you need to add it. The hub cluster can only be managed with this option.

Setting this option to **true** and attempting to manage the hub manually leads to unexpected behavior.

The following example shows the default template to use if you want to disable the hub cluster self-management feature. Replace **namespace** with the name of your project:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  disableHubSelfManagement: true
```

1.5.6. disableUpdateClusterImageSets

If want to ensure that you use the same release image for all of your clusters, you can create your own custom list of release images that are available when creating a cluster.

See the following instructions in Maintaining a custom list of release images when connected to manage your available release images and to set the **spec.disableUpdateClusterImageSets** attribute, which stops the custom image list from being overwritten.

The following example shows the default template that disables updates to the cluster image set. Replace **namespace** with the name of your project:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  disableUpdateClusterImageSets: true
```

1.5.7. customCAConfigmap

By default, Red Hat OpenShift Container Platform uses the Ingress Operator to create an internal CA.

The following example shows the default template used to provide a customized OpenShift Container Platform default ingress CA certificate to Red Hat Advanced Cluster Management. Replace **namespace** with the name of your project. Replace the **spec.customCAConfigmap** value with the name of your **ConfigMap**:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  customCAConfigmap: <configmap>
```

1.5.8. sslCiphers

By default, the Red Hat Advanced Cluster Management hub cluster includes the full list of supported SSL ciphers.

The following example shows the default **spec.ingress.sslCiphers** template that is used to list **sslCiphers** for the management ingress. Replace **namespace** with the name of your project:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  ingress:
    sslCiphers:
    - "ECDHE-ECDSA-AES128-GCM-SHA256"
    - "ECDHE-RSA-AES128-GCM-SHA256"
```

1.5.9. ClusterProxyAddon (Technology Preview)

The following example shows the default **spec.overrides** template that is used to enable **cluster-proxy-addon**. Replace **namespace** with the name of your project:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
```

```
  namespace: <namespace>
spec:
  overrides:
    components:
    - name: cluster-proxy-addon
      enabled: true
```

Alternatively, you can run the following command. Replace **namespace** with the name of your project:

```
oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p='[{"op": "add", "path":
"/spec/overrides/components/-","value":{"name":"cluster-proxy-addon","enabled":true}}]'
```

Use of the **enableClusterProxyAddon** field is no longer supported, replaced by the above.

1.5.10. ClusterBackup

The **enableClusterBackup** field is no longer supported and is replaced by this component.

The following example shows the **spec.overrides** default template used to enable **ClusterBackup**. Replace **namespace** with the name of your project:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  overrides:
    components:
    - name: cluster-backup
      enabled: true
```

Alternatively, you can run the following command. Replace **namespace** with the name of your project.

```
oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p='[{"op": "add", "path":
"/spec/overrides/components/-","value":{"name":"cluster-backup","enabled":true}}]'
```

1.5.11. ManagedServiceAccount add-on (Technology Preview)

The following example shows the **spec.overrides** default template used to enable **ManagedServiceAccount**. Replace **namespace** with the name of your project:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  overrides:
    components:
    - name: managedserviceaccount-preview
      enabled: true
```

Alternatively, you can run the following command. Replace **namespace** with the name of your project.

```
oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p='[{"op": "add", "path":
"/spec/overrides/components/-","value":{"name":"managedserviceaccount-preview","enabled":true}}]'
```

1.5.12. Hypershift add-on (Technology Preview)

The following example shows the **spec.overrides** default template used to enable **Hypershift**. Replace **namespace** with the name of your project:

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  overrides:
    components:
    - name: hypershift-preview
      enabled: true
```

Alternatively, you can run the following command. Replace **namespace** with the name of your project.

```
oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p='[{"op": "add", "path":
"/spec/overrides/components/-","value":{"name":"hypershift-preview","enabled":true}}]'
```

1.6. NETWORK CONFIGURATION

You can refer to the configuration for your hub cluster and managed cluster network, as well as additional networking information:

- Hub cluster network configuration table
- Managed cluster network configuration table
- Additional networking requirements for infrastructure operator table
- Submariner networking requirements table
- Additional networking requirements for Hive table
- Application deployment network requirements table
- Namespace connection network requirements table

1.6.1. Hub cluster network configuration table

See the hub cluster network requirements in the following table:

Direction	Protocol	Connection	Port (if specified)	Source address	Destination address
Outbound to the managed cluster	HTTPS	Retrieval of logs dynamically from Search console for the pods of the managed cluster, uses the **klusterlet-addon-workmgr** service that is running on the managed cluster	443	None	IP address to access managed cluster route
Outbound to the managed cluster	HTTPS	Kubernetes API server of the managed cluster that is provisioned during installation to install the klusterlet	6443	None	IP of Kubernetes managed cluster API server
Outbound to the channel source	HTTPS	The channel source, including GitHub, Object Store, and Helm repository, which is only required when you are using Application lifecycle, OpenShift GitOps, or ArgoCD to connect	443	None	IP of the channel source

Direction	Protocol	Connection	Port (if specified)	Source address	Destination address
Inbound from the managed cluster	HTTPS	Managed cluster to push metrics and alerts that are gathered only for managed clusters that are running OpenShift Container Platform version 4.8, or later	443	None	IP address to hub cluster access route
Inbound from the managed cluster	HTTPS	Kubernetes API Server of hub cluster that is watched for changes from the managed cluster	6443	None	IP address of hub cluster Kubernetes API Server
Outbound to the ObjectStore	HTTPS	Sends Observability metric data for long term storage when the Cluster Backup Operator is running	443	None	IP address of ObjectStore
Outbound to the image repository	HTTPS	Access images for OpenShift Container Platform and Red Hat Advanced Cluster Management	443	None	IP address of image repository

1.6.2. Managed cluster network configuration table

Note: Registration Agent and **Work Agent** on the managed cluster do not support proxy settings because they communicate with **apiserver** on the hub cluster by establishing an mTLS connection, which cannot pass through the proxy.

See the managed cluster network requirements in the following table:

Direction	Protocol	Connection	Port (if specified)	Source address	Destination address
Inbound from the hub cluster	HTTPS	Sending of logs dynamically from Search console for the pods of the managed cluster, uses the **klusterlet-addon-workmgr** service that is running on the managed cluster	443	None	IP address to access managed cluster route
Inbound from the hub cluster	HTTPS	Kubernetes API server of the managed cluster that is provisioned during installation to install the klusterlet	6443	None	IP of Kubernetes managed cluster API server
Outbound to the image repository	HTTPS	Access images for OpenShift Container Platform and Red Hat Advanced Cluster Management	443	None	IP address of image repository
Outbound to the hub cluster	HTTPS	Managed cluster to push metrics and alerts that are gathered only for managed clusters that are running OpenShift Container Platform version 4.8, or later	443	None	IP address to hub cluster access route

Direction	Protocol	Connection	Port (if specified)	Source address	Destination address
Outbound to the hub cluster	HTTPS	Watches the Kubernetes API server of the hub cluster for changes	6443	None	IP address of hub cluster Kubernetes API Server
Outbound to the channel source	HTTPS	The channel source, including GitHub, Object Store, and Helm repository, which is only required when you are using Application lifecycle, OpenShift GitOps, or ArgoCD to connect	443	None	IP of the channel source

1.6.3. Additional networking requirements for infrastructure operator table

When you are installing bare metal managed clusters with the Infrastructure Operator, see the following table for the additional networking requirements:

Direction	Protocol	Connection	Port (if specified)
Hub cluster outbound to the ISO/rootfs image repository	HTTPS (HTTP in a disconnected environment	Used to create an ISO image on the Red Hat Advanced Cluster Management hub	443 (80 in disconnected environments)
Hub cluster outbound to BMC interface at single node OpenShift Container Platform managed cluster	HTTPS (HTTP in disconnected environment)	Boot the OpenShift Container Platform cluster	443
Outbound from the OpenShift Container Platform managed cluster to the hub cluster	HTTPS	Reports hardware information using the **assistedService** route	443

Direction	Protocol	Connection	Port (if specified)
Outbound from the OpenShift Container Platform managed cluster to the ISO/rootfs image repository	HTTP	Downloads the rootfs image	80

1.6.4. Submariner networking requirements table

Clusters that are using Submariner require three open ports. The following table shows which ports you might use:

Direction	Protocol	Connection	Port (if specified)
Outbound and inbound	UDP	Each of the managed clusters	4800
Outbound and inbound	UDP	Each of the managed clusters	4500, 500, and any other ports that are used for IPSec traffic on the gateway nodes
Inbound	TCP	Each of the managed clusters	8080

1.6.5. Additional networking requirements for Hive table

When you are installing bare metal managed clusters with the Hive Operator, which includes using Central Infrastructure Management, you must configure a layer 2 or layer 3 port connection between the hub cluster and the **libvirt** provisioning host. This connection to the provisioning host is required during the creation of a base metal cluster with Hive. See the following table for more information:

Direction	Protocol	Connection	Port (if specified)
Hub cluster outbound and inbound to the **libvirt** provisioning host	IP	Connects the hub cluster, where the Hive operator is installed, to the **libvirt** provisioning host that serves as a bootstrap when creating the bare metal cluster	

Note: These requirements only apply when installing, and are not required when upgrading clusters that were installed with Infrastructure Operator.

1.6.6. Application deployment network requirements table

In general, the application deployment communication is one way from a managed cluster to the hub

cluster. The connection uses **kubeconfig**, which is configured by the agent on the managed cluster. The application deployment on the managed cluster needs to access the following namespaces on the hub cluster:

- The namespace of the channel resource
- The namespace of the managed cluster

1.6.7. Namespace connection network requirements table

- Application lifecycle connections:
  - The namespace **open-cluster-management** needs to access the console API on port 4000.
  - The namespace **open-cluster-management** needs to expose the Application UI on port 3001.
- Application lifecycle backend components (pods):
  On the hub cluster, all of the application lifecycle pods are installed in the **open-cluster-management** namespace, including the following pods:
  - multicluster-operators-hub-subscription
  - multicluster-operators-standalone-subscription
  - multicluster-operators-channel
  - multicluster-operators-application
  - multicluster-integrations
    As a result of these pods being in the **open-cluster-management** namespace:
  - The namespace **open-cluster-management** needs to access the Kube API on port 6443.

  On the managed cluster, only the **klusterlet-addon-appmgr** application lifecycle pod is installed in the **open-cluster-management-agent-addon** namespace:
  - The namespace **open-cluster-management-agent-addon** needs to access the Kube API on port 6443.
- Governance and risk:
  On the hub cluster, the following access is required:
  - The namespace **open-cluster-management** needs to access the Kube API on port 6443.
  - The namespace **open-cluster-management** needs to access the OpenShift DNS on port 5353.

  On the managed cluster, the following access is required:
  - The namespace **open-cluster-management-addon** needs to access the Kube API on port 6443.

See the Red Hat Advanced Cluster Management for Kubernetes 2.5 Support matrix for additional information.

1.7. UPGRADING BY USING THE OPERATOR

You control your Red Hat Advanced Cluster Management for Kubernetes upgrades by using the operator subscription settings in the Red Hat OpenShift Container Platform console. When you initially deploy Red Hat Advanced Cluster Management by using the operator, you make the following selections:

- **Channel:** Corresponds to the version of the product that you are installing. The initial channel setting is often the most current channel that was available at the time of installation.
- **Approval:** Specifies whether approval is required for updates within the channel, or if they are done automatically.
  - If set to **Automatic**, then minor release updates in the selected channel are deployed without administrator intervention.
  - If set to **Manual**, then each update to the minor release within the channel requires an administrator to approve the update.

You also use these settings when you upgrade Red Hat Advanced Cluster Management by using the operator.

Required access: OpenShift Container Platform administrator

Complete the following steps to upgrade your operator:

Important: You cannot revert back to an earlier version after upgrading to a later version in the channel selection. You must uninstall the operator and reinstall it with the earlier version to use a previous version.

1. Log in to your OpenShift Container Platform operator hub.
2. In the OpenShift Container Platform navigation, select **Operators** > **Installed operators**.
3. Select the **Red Hat Advanced Cluster Management for Kubernetes**operator.
4. Select the *Subscription* tab to edit the subscription settings.
5. Ensure that the *Upgrade Status* is labeled *Up to date*. This status indicates that the operator is at the latest level that is available in the selected channel. If the *Upgrade Status* indicates that there is an upgrade pending, complete the following steps to update it to the latest minor release that is available in the channel:
   a. Click the **Manual** setting in the *Approval* field to edit the value.
   b. Select **Automatic** to enable automatic updates.
   c. Select **Save** to commit your change.
   d. Wait for the automatic updates to be applied to the operator. The updates automatically add the required updates to the latest version in the selected channel. When all of the updated updates are complete, the *Upgrade Status* field indicates **Up to date**.
   Tip: It can take up to 10 minutes for the **MultiClusterHub** custom resource to finish upgrading. You can check whether the upgrade is still in process by entering the following command:

```
oc get mch
```

While it is upgrading, the **Status** field shows **Updating**. After upgrading is complete, the **Status** field shows **Running**.

6. Now that the *Upgrade Status* is **Up to date**, click the value in the *Channel* field to edit it.

7. Select the channel for the next available feature release. **Deprecated: release-2.4** and **release-2.3** channels do not receive updates. To import, you must use the **stable-2.0** channel of the klusterlet operator for 2.5. You cannot skip channels when upgrading. For example, you cannot skip versions 2.2.z through 2.4.

8. Select **Save** to save your changes.

9. Wait for the automatic upgrade to complete. After the upgrade to the next feature release completes, the updates to the latest patch releases within the channel are deployed.

10. If you have to upgrade to a later feature release, repeat steps 7-9 until your operator is at the latest level of the desired channel. Make sure that all of the patch releases are deployed for your final channel.

11. Optional: You can set your *Approval* setting to **Manual**, if you want your future updates within the channel to require manual approvals.

Red Hat Advanced Cluster Management is running at the latest version of the selected channel.

For more information about upgrading your operator, see Operators in the OpenShift Container Platform documentation.

1.7.1. Managing cluster pools with an upgrade

If you are Managing cluster pools (Technology Preview), you need further configuration to stop automatic management of these cluster pools after upgrade.

Set **cluster.open-cluster-management.io/createmanagedcluster: "false"** in the **ClusterClaim** metadata.annotations.

All existing cluster claims are automatically imported when the product is upgraded unless you change this setting.

1.8. UPGRADING OPENSHIFT CONTAINER PLATFORM

You can upgrade the version of Red Hat OpenShift Container Platform that hosts your Red Hat Advanced Cluster Management for Kubernetes hub cluster. Back up your data before initiating any cluster-wide upgrade.

During the upgrade of the OpenShift Container Platform version, the Red Hat Advanced Cluster Management web console might show brief periods when pages or data are unavailable. Indicators can include HTTP 500 (Internal Server Error), HTTP 504 (Gateway Timeout Error), or errors that data that was previously available is not available. This is a normal part of the upgrade, and no data is lost when this occurs. The availability is eventually restored.

The search index is also rebuilt during this upgrade, so any queries that are submitted during the upgrade might be incomplete.

The following table contains some noted observations from an upgrade from OpenShift Container Platform version 4.4.3 to 4.4.10:

Table 1.2. Table Observations from an OpenShift Container Platform upgrade from version 4.3.3 to 4.4.10.

Elapsed time of upgrade process (minutes:seconds)	Observed change	Duration
03:40	Governance console experiences HTTP 500	Service restored within 20 seconds
05:30	AppUI experiences HTTP 504 Gateway Timeout	Service restored within 60 seconds
06:05	Cluster and Search console experience HTTP 504 Gateway Timeout	Service restored within 20 seconds
07:00	Cluster and Search console experience HTTP 504 Gateway Timeout	Service restored within 20 seconds
07:10	Topology and Cluster console Display error messages within the page	Service restored within 20 seconds
07:35	HTTP 500 for most console pages	Service restored within 60 seconds
08:30	Service restored for all pages	

1.9. UNINSTALLING

When you uninstall Red Hat Advanced Cluster Management for Kubernetes, you see two different levels of the uninstall process: A *custom resource removal* and a *complete operator uninstall*. The uninstall process can take up to 20 minutes.

- The first level is the custom resource removal, which is the most basic type of uninstall that removes the custom resource of the **MultiClusterHub** instance, but leaves other required operator resources. This level of uninstall is helpful if you plan to reinstall using the same settings and components.
- The second level is a more complete uninstall that removes most operator components, excluding components such as custom resource definitions. When you continue with this step, it removes all of the components and subscriptions that were not removed with the custom resource removal. After this uninstall, you must reinstall the operator before reinstalling the custom resource.

1.9.1. Prerequisite: Detach enabled services

Before you uninstall the Red Hat Advanced Cluster Management hub cluster, you must detach all of the clusters that are managed by that hub cluster. To resolve errors, detach all clusters that are still managed by the hub cluster, then try to uninstall again.

- If you use Discovery, you might see the following error when you attempt uninstall:

  ```
  Cannot delete MultiClusterHub resource because DiscoveryConfig resource(s) exist
  ```

  To disable Discovery, complete the following steps:

  - From the console, navigate to the **Discovered Clusters** table and click **Disable cluster discovery**. Confirm that you want to remove the service.
  - You can also use the terminal. Run the following command to disable Discover:

  ```
  $ oc delete discoveryconfigs --all --all-namespaces
  ```

- If you have managed clusters attached, you might see the following message. **Note:** This does not include the **local-cluster**, which is your self-managed hub cluster:

  ```
  Cannot delete MultiClusterHub resource because ManagedCluster resource(s) exist
  ```

  For more information about detaching clusters, see the *Removing a cluster from management* section by selecting the information for your provider in Creating a cluster.

- If you have Bare metal assets, you might see the following:

  ```
  Cannot delete MultiClusterHub resource because BareMetalAssets resource(s) exist
  ```

  For more information about removing the bare metal assets, see Removing a bare metal asset.

- If you have Observability, you might see the following:

  ```
  Cannot delete MultiClusterHub resource because MultiClusterObservability resource(s) exist
  ```

  - To disable and remove the **MultiClusterObservability** using the terminal, see the following procedure:

    a. Log in to your hub cluster.

    b. Delete the **MultiClusterObservability** custom resource by entering the following command:

    ```
    oc delete mco observability
    ```

  - To remove **MultiClusterObservability** custom resource with the console, see the following procedure:

    a. If the **MultiClusterObservability** custom resource is installed, select the tab for *MultiClusterObservability*.

    b. Select the *Options* menu for the **MultiClusterObservability** custom resource.

    c. Select **Delete MultiClusterObservability**.
    When you delete the resource, the pods in the **open-cluster-management-observability** namespace on Red Hat Advanced Cluster Management hub cluster, and the pods in **open-cluster-management-addon-observability** namespace on all managed clusters are removed.

Note: Your object storage is not affected after you remove the observability service.

1.9.2. Removing resources by using commands

1. If you have not already. ensure that your OpenShift Container Platform CLI is configured to run **oc** commands. See Getting started with the OpenShift CLI in the OpenShift Container Platform documentation for more information about how to configure the **oc** commands.

2. Change to your project namespace by entering the following command. Replace *namespace* with the name of your project namespace:

```
oc project <namespace>
```

3. Enter the following command to remove the **MultiClusterHub** custom resource:

```
oc delete multiclusterhub --all
```

   You can view the progress by entering the following command:

```
oc get mch -o yaml
```

4. Remove any potential remaining artifacts by running the clean-up script. Run this clean-up script if you plan to reinstall with an older version of Red Hat Advanced Cluster Management on the same cluster.

   a. Install the Helm CLI binary version 3.2.0, or later, by following the instructions at Installing Helm.

   b. Copy the following script into a file:

```
#!/bin/bash
ACM_NAMESPACE=<namespace>
oc delete mch --all -n $ACM_NAMESPACE
helm ls --namespace $ACM_NAMESPACE | cut -f 1 | tail -n +2 | xargs -n 1 helm delete --namespace $ACM_NAMESPACE
oc delete apiservice v1beta2.webhook.certmanager.k8s.io v1.admission.cluster.open-cluster-management.io v1.admission.work.open-cluster-management.io
oc delete clusterimageset --all
oc delete configmap -n $ACM_NAMESPACE cert-manager-controller cert-manager-cainjector-leader-election cert-manager-cainjector-leader-election-core
oc delete consolelink acm-console-link
oc delete crd klusterletaddonconfigs.agent.open-cluster-management.io placementbindings.policy.open-cluster-management.io policies.policy.open-cluster-management.io userpreferences.console.open-cluster-management.io searchservices.search.acm.com discoveredclusters.discovery.open-cluster-management.io discoveryconfigs.discovery.open-cluster-management.io
oc delete mutatingwebhookconfiguration cert-manager-webhook cert-manager-webhook-v1alpha1 ocm-mutating-webhook managedclustermutators.admission.cluster.open-cluster-management.io
oc delete oauthclient multicloudingress
oc delete rolebinding -n kube-system cert-manager-webhook-webhook-authentication-reader
oc delete scc kui-proxy-scc
oc delete validatingwebhookconfiguration cert-manager-webhook cert-manager-
```

```
webhook-v1alpha1 channels.apps.open.cluster.management.webhook.validator
application-webhook-validator multiclusterhub-operator-validating-webhook ocm-
validating-webhook
```

Replace **<namespace>** in the script with the name of the namespace where Red Hat Advanced Cluster Management was installed. Ensure that you specify the correct namespace, as the namespace is cleaned out and deleted.

c. Run the script to remove any possible artifacts that remain from the previous installation. If there are no remaining artifacts, a message is returned that no resources were found.
Note: If you plan to reinstall the same Red Hat Advanced Cluster Management version, you can skip the next steps in this procedure and reinstall the custom resource. Proceed for a complete operator uninstall.

5. Enter the following commands to delete the Red Hat Advanced Cluster Management **ClusterServiceVersion** and **Subscription** in the namespace it is installed in:

```
❯ oc get csv
NAME                                 DISPLAY                                VERSION   REPLACES   PHASE
advanced-cluster-management.v2.4.0   Advanced Cluster Management for Kubernetes   2.4.0
Succeeded

❯ oc delete clusterserviceversion advanced-cluster-management.v2.4.0

❯ oc get sub
NAME                         PACKAGE                      SOURCE                CHANNEL
acm-operator-subscription    advanced-cluster-management   acm-custom-registry   release-2.5

❯ oc delete sub acm-operator-subscription
```

Note: The name of the subscription and version of the CSV might differ.

1.9.3. Deleting the components by using the console

When you use the Red Hat OpenShift Container Platform console to uninstall, you remove the operator. Complete the following steps to uninstall by using the console:

1. In the OpenShift Container Platform console navigation, select **Operators** > **Installed Operators** > **Advanced Cluster Manager for Kubernetes**.

2. Remove the **MultiClusterHub** custom resource.

   a. Select the tab for *Multiclusterhub*.

   b. Select the *Options* menu for the MultiClusterHub custom resource.

   c. Select **Delete MultiClusterHub**.

3. Run the clean-up script according to the procedure in Removing a MultiClusterHub instance by using commands.
Tip: If you plan to reinstall the same Red Hat Advanced Cluster Management version, you can skip the rest of the steps in this procedure and reinstall the custom resource.

4. Navigate to **Installed Operators**.

5. Remove the *Red Hat Advanced Cluster Management* operator by selecting the *Options* menu and selecting **Uninstall operator**.