



Red Hat Advanced Cluster Management for Kubernetes 2.3

Install

Install

Red Hat Advanced Cluster Management for Kubernetes 2.3 Install

Install

Legal Notice

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Installing instructions for Red Hat Advanced Cluster Management for Kubernetes

Table of Contents

CHAPTER 1. INSTALLING	3
1.1. REQUIREMENTS AND RECOMMENDATIONS	3
1.1.1. Supported operating systems and platforms	3
1.1.2. Supported browsers	4
1.1.3. Network configuration	4
1.1.3.1. Hub cluster networking requirements	4
1.1.3.2. Managed cluster networking requirements	5
1.1.3.3. Submariner networking requirements	5
1.1.3.4. Application deployment network requirements	6
1.2. PERFORMANCE AND SCALABILITY	6
1.2.1. Maximum number of managed clusters	6
1.2.2. Search scalability	7
1.2.2.1. Physical memory	7
1.2.2.2. Write throughput (cache recovery time)	8
1.2.2.3. Query execution considerations	8
1.2.3. Scaling for observability	8
1.2.3.1. Sample observability environment	9
1.2.3.2. Write throughput	9
1.2.3.3. CPU usage (millicores)	9
1.2.3.4. RSS and working set memory	9
1.2.3.5. Persistent volume for thanos-receive component	10
1.2.3.6. Network transfer	10
1.2.3.7. Amazon Simple Storage Service (S3)	10
1.2.4. Sizing your cluster	10
1.2.4.1. Product environment	11
1.2.4.1.1. OpenShift Container Platform on Amazon Web Services	11
1.2.4.1.2. OpenShift cluster on Google Cloud Platform	12
1.2.4.1.3. OpenShift cluster on Microsoft Azure	12
1.2.4.1.4. OpenShift cluster on VMware vSphere	12
1.2.4.1.5. OpenShift Container Platform on IBM Power systems (Technology Preview)	13
1.2.4.1.6. OpenShift cluster on bare metal	13
1.3. INSTALLING WHILE CONNECTED ONLINE	13
1.3.1. Prerequisites	13
1.3.2. Confirm your installation	14
1.3.3. Preparing to install the hub cluster on an infrastructure node	15
1.3.4. Installing from the OperatorHub	15
1.3.5. Installing from the CLI	18
1.4. INSTALL ON DISCONNECTED NETWORKS	20
1.4.1. Prerequisites	21
1.4.2. Confirm your OpenShift Container Platform installation	21
1.4.3. Installing in a disconnected environment	22
1.5. UPGRADING BY USING THE OPERATOR	23
1.6. UPGRADING OPENSIFT CONTAINER PLATFORM	25
1.7. UNINSTALLING	26
1.7.1. Prerequisite: Detach enabled services	26
1.7.2. Removing resources by using commands	27
1.7.3. Deleting the components by using the console	29

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.

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.

Important:

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**.

- [Requirements and recommendations](#)
- [Sizing your cluster](#)
- [Performance and scalability](#)
- [Installing while connected online](#)
- [Install on disconnected networks](#)
- [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](#)
- [Network configuration](#)

1.1.1. Supported operating systems and platforms

See the following table for supported operating systems:

Platform	Operating system	Red Hat OpenShift Container Platform version
Linux x86_64	Red Hat Enterprise Linux 7.6 (or later), or CoreOS	Refer to the Red Hat Advanced Cluster Management 2.3 Support matrix for the most current list of supported OpenShift Container Platform platforms.

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	Connection	Port (if specified)
Outbound	API of the cloud provider	
Outbound	Kubernetes API server of the provisioned managed cluster	6443
Outbound	The channel source, including GitHub, Object Store, and Helm repository. This is only required when you are using Application lifecycle to connect to these sources.	

Direction	Connection	Port (if specified)
Outbound and inbound	The WorkManager service route on the managed cluster	443
Inbound	The kube API server of the hub cluster from the managed cluster	6443
Inbound	Post-commit hook from GitHub to the hub cluster. This setting is only required when you use certain application management functions.	6443

1.1.3.2. Managed cluster networking requirements

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

Direction	Connection	Port (if specified)
Outbound and inbound	Kubernetes API server of the hub cluster	6443
Outbound	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.	
Inbound	The WorkManager service endpoint from the hub cluster	443

1.1.3.3. Submariner networking requirements

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

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

Direction	Connection	Port (if specified)
Inbound	Each of the managed clusters	8080/TCP

1.1.3.4. Application deployment network requirements

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

See the [Red Hat Advanced Cluster Management for Kubernetes 2.3 Support matrix](#) for additional information.

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)/IO S	Count	Region
------	--------	------	-----------	-----------	----------------------	-------	--------

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

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

By default, the redisgraph pod (**search-redisgraph-0**) is deployed with a memory limit of 4 Gi. If you are managing larger clusters, you might need to increase this limit by editing the **redisgraph_resource.limit_memory** for the **searchoperator** in the hub cluster namespace. For example, you can update the limit to 8Gi with the following command:

```
oc patch searchoperator searchoperator --type='merge' -p '{"spec":{"redisgraph_resource":{"limit_memory":"8Gi"}}}'
```

When the change is made, delete the **search-redisgraph** StatefulSet for the new limit to take effect.

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)/IO S	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

Memory usage RSS: From the metrics `container_memory_rss` and keeps stability 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	4.83
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 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 three dimensions for sizing and placement of supporting services:

- Availability Zones, which 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. Reservations establish a *lower* bound of CPU or memory, and limits establish an *upper* bound.
- Persistent data, which is managed by the product and stored in the etcd cluster that is used by Kubernetes. **Best practice:** 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.

OpenShift Container Platform node role	Availability zones	Data stores	Total reserved memory (<i>lower</i> bound)	Total reserved CPU (<i>lower</i> bound)
Master	3	etcd x 3	Per OpenShift Container Platform sizing guidelines	Per OpenShift sizing guidelines
Worker	3	redisgraph/redis x 1	12 GB	6 CPU

In addition to Red Hat Advanced Cluster Management, the OpenShift Container Platform cluster runs additional services to support cluster features. The following node sizes (3 nodes of the types noted in the information that follows, distributed evenly across 3 availability zones) are recommended.

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.

- Self-managed hub cluster:
 - Cores per socket: 2
 - CPUs: 4
 - Memory: 16 GB
 - Storage size: 120 GB
- Managed cluster:
 - Cores per socket: 2
 - CPUs: 4
 - Memory: 16 GB
 - Storage size: 120 GB

1.2.4.1.5. OpenShift Container Platform on IBM Power systems (Technology Preview)

See [Installing a cluster on Power systems](#) in the OpenShift Container Platform documentation for more information.

- Node count: 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](#).

1.2.4.1.6. OpenShift cluster on bare metal

See the following [product documentation](#) for more details.

- CPUs: 6 (minimum)
- Memory: 16 GB (minimum)
- Storage size: 50 GB (minimum)

1.3. INSTALLING WHILE CONNECTED ONLINE

Red Hat Advanced Cluster Management for Kubernetes is installed using an operator that deploys all of the required components.

- [Prerequisites](#)
- [Confirm your installation](#)
- [Preparing to install the hub cluster on an infrastructure node](#)
- [Installing from the OperatorHub](#)
- [Installing from the CLI](#)

1.3.1. Prerequisites

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

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

- You need access to the catalog.redhat.com.
- OpenShift Container Platform version 4.6, or later, must be deployed in your environment, and you must be logged into with the CLI. See the following install doc for OpenShift Container Platform:
 - [OpenShift Container Platform version 4.8](#)
 - [OpenShift Container Platform version 4.7](#)
 - [OpenShift Container Platform version 4.6](#)
- 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.
- You must have an Internet connection to access the dependencies for the operator.
- To install in a OpenShift Container Platform Dedicated environment, see the following:
 - 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.
- **Important:**

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``.

1.3.2. Confirm your installation

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

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

1.3.3. Preparing to install the hub cluster on an infrastructure node

By using **tolerations**, the Red Hat Advanced Cluster Management for Kubernetes hub cluster allows hub cluster components to be installed on an infrastructure node. To install the hub cluster on an infrastructure node, complete the following steps to prepare:

- Configure your infrastructure nodes as infrastructure machine sets according to the procedure in [Creating infrastructure machine sets](#) in the OpenShift Container Platform documentation. See the following example of the **toleration**:

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

- Add the following **nodeSelector** entry to the **MultclusterHub** resource *object* section:

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

- Complete the steps to finish installing the hub cluster.

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 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.

1.3.4. Installing from the OperatorHub

Best practice: Install by using the OperatorHub that is provided with OpenShift Container Platform.

Note: For OpenShift Container Platform Dedicated environment only, log in to your OpenShift Container Platform Dedicated environment with **cluster-admin** permissions.

- From the *Administrator* view in your OpenShift Container Platform navigation, select **Operators** > **OperatorHub** to access the list of available operators.
- Find and select the *Advanced Cluster Management for Kubernetes* operator.

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

- Namespace:
 - The 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: 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.

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

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

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

Important: These secrets are namespace-specific, so be sure to create a secret in the namespace where you installed Red Hat Advanced Cluster Management.

- a. Copy your OpenShift Container Platform pull secret from cloud.redhat.com/openshift/install/pull-secret by selecting **Copy pull secret**. You need the content of this pull secret in a step later in this procedure. Your OpenShift Container Platform pull secret is associated with your Red Hat Customer Portal ID and is the same across all Kubernetes providers.
- b. In the OpenShift Container Platform console navigation, select **Workloads > Secrets**.
- c. Select **Create > Image Pull Secret**.
- d. Enter a name for your secret.
- e. Select **Upload Configuration File** as the authentication type.
- f. In the *Configuration file* field, paste the pull secret that you copied from **cloud.redhat.com**.
- g. Select **Create** to create the secret.

6. 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, according to your needs.
 - The following example shows the default template if you did not create an image pull secret. Confirm that **namespace** is your project namespace:

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

- The following example is the default template if you created an image pull secret. Replace **secret** with the name of the pull secret that you created. Confirm that **namespace** is your project namespace.:

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

7. **Optional:** Disable hub self management, if necessary. By default, the hub cluster is automatically imported and managed by itself, like any other cluster. If you do not want the hub cluster to manage itself, then change the setting for **disableHubSelfManagement** from **false** to **true**. If the setting is not included in the YAML file that defines the custom resource, add it as shown in the example of the previous step.

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

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

8. Select **Create** to initialize the custom resource. It can take up to 10 minutes for the hub cluster to build and start.
After the hub cluster is created, the status for the operator is *Running* on the *Installed Operators* page.
9. Access the console for the hub cluster.

- a. In the OpenShift Container Platform console navigation, select **Networking > Routes**.
- b. View the URL for your hub cluster in the list, and navigate to it to access the console.

1.3.5. Installing from the CLI

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

1. Create a hub cluster namespace where the operator requirements are contained. Run the following command, where **namespace** is the name for your 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 hub cluster namespace that you created in step 1.

```
oc project <namespace>
```

3. 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. **Important:** These secrets are namespace-specific, so make sure that you are in the namespace that you created in step 1.
 - a. Download your OpenShift Container Platform pull secret file from cloud.redhat.com/openshift/install/pull-secret by selecting **Download pull secret**. Your OpenShift Container Platform pull secret is associated with your Red Hat Customer Portal ID, and is the same across all Kubernetes providers.

- b. 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.

4. Create an operator group. Each namespace can have only one operator group.
 - a. Create a YAML file that defines the operator group. Your file should look similar to the following example. Replace **default** with the name of your operator group. Replace **namespace** with the name of your project namespace:

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

```
spec:
  targetNamespaces:
  - <namespace>
```

- b. Apply the file that you created to define the operator group:

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

Replace **operator-group** with the name of the operator group YAML file that you created.

5. Apply the subscription.

- a. Create a YAML file that defines the subscription. Your file should look similar to the following example:

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: acm-operator-subscription
spec:
  sourceNamespace: openshift-marketplace
  source: redhat-operators
  channel: release-2.3
  installPlanApproval: Automatic
  name: advanced-cluster-management
```

- b. Include the following if you are installing on infra nodes:

```
spec:
  config:
    nodeSelector:
      node-role.kubernetes.io/infra: ""
    tolerations:
    - key: node-role.kubernetes.io/infra
      effect: NoSchedule
      operator: Exists
```

- c. Run the following command. Replace **subscription** with the name of the subscription file that you created:

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

6. Apply the MultiClusterHub custom resource.

- a. Create a YAML file that defines the custom resource.

- Your default template should look similar to the following example. Replace **namespace** with the name of your project namespace. If you did not create a pull secret, it will not appear. If you did, replace **secret** with the name of your pull secret for this example:

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

```
namespace: <namespace>
spec:
  imagePullSecret: <secret>
```

- b. **Optional:** If the installer-managed **acm-hive-openshift-releases** subscription is enabled, you can disable the subscription by setting the value of **disableUpdateClusterImageSets** to **true**.
- c. **Optional:** Disable hub self management, if necessary. By default, the hub cluster is automatically imported and managed by itself, like any other cluster. If you do not want the hub cluster to manage itself, then change the setting for **disableHubSelfManagement** from **false** to **true**.

Your default template should look similar to the following example, if you created a pull secret and are enabling the **disableHubSelfManagement** feature. Replace **namespace** with the name of your project namespace. 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>
  disableHubSelfManagement: true
```

- d. Apply the custom resource with the following command. Replace **custom-resource** with the name of your custom resource file:

```
oc apply -f <path-to-path><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"
```

7. 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 following command:

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

8. 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.

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.

- [Prerequisites](#)
- [Confirm your OpenShift Container Platform installation](#)
- [Preparing to install the hub cluster on an infrastructure node](#)

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 4.6 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 4.6 or later.

See the [OpenShift Container Platform version 4.8](#), [OpenShift Container Platform version 4.6](#).

- Your Red Hat OpenShift Container Platform CLI must be version 4.6 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.
- 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 4.8, see [OpenShift Container Platform Documentation](#).
- When and if you are connected, you can ensure that the OpenShift Container Platform cluster is set up correctly. Access the OpenShift Container Platform web console. 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.
2. **Note:** 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 resulting content should look similar to the following example:

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

3. **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
```

4. Save the **install-config.yaml** file.
5. 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
```

6. Apply the ImageContentSourcePolicy file by entering the following command:

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

7. 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.
8. Configure the disconnected Operator Lifecycle Manager for the Red Hat Operator catalog. Follow the steps in the [https://access.redhat.com/documentation/en-us/openshift_container_platform/4.8/html/operators/administrator-tasks#Operator Lifecycle Manager-restricted-networks](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.8/html/operators/administrator-tasks#Operator_Lifecycle_Manager-restricted-networks)[Using Operator Lifecycle Manager on restricted networks] topic of the Red Hat OpenShift Container Platform documentation.
9. 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.

1.5. 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 the **Manual** setting is selected, then each update to the minor release within the channel requires an administrator to approve the update.

You also use those 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:

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. You cannot skip channels when upgrading.

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.

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.

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**.

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.

1.6. UPGRADING OPENSIFT 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 1.1. 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

Elapsed time of upgrade process (minutes:seconds)	Observed change	Duration
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	

1.7. UNINSTALLING

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

- The custom resource removal 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.

1.7.1. 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 avoid 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:

- Log in to your hub cluster.
- 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:

- If the **MultiClusterObservability** custom resource is installed, select the tab for *MultiClusterObservability*.
- Select the *Options* menu for the **MultiClusterObservability** custom resource.

- 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.

1.7.2. 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.

Note: If **cert-manager** was manually installed in your cluster, be sure to review the commands in the clean-up script and remove commands that are specific to **cert-manager**.

- 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
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 command to remove all of the related components and subscriptions:

```
oc delete subs --all
```


6. Enter the following command to delete the **ClusterServiceVersion**:

```
oc delete clusterserviceversion --all
```

1.7.3. 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**.