



# Red Hat Advanced Cluster Management for Kubernetes 2.10

## About

About 2.10





## Legal Notice

Copyright © 2024 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

Learn about architecture and other specific information about Red Hat Advanced Cluster Management for Kubernetes.

---

## Table of Contents

<b>CHAPTER 1. WELCOME TO RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES</b>	<b>3</b>
1.1. MULTICLUSTER ARCHITECTURE	3
1.1.1. Hub cluster	4
1.1.2. Managed cluster	5
1.1.3. Cluster lifecycle	5
1.1.4. Application lifecycle	5
1.1.5. Governance	5
1.1.6. Observability	6
1.1.7. References	6
1.2. GLOSSARY OF TERMS	6
1.2.1. Relevant standardized glossaries	6
1.2.2. Red Hat Advanced Cluster Management for Kubernetes terms	6
1.2.2.1. Application lifecycle	6
1.2.2.2. Channel	6
1.2.2.3. Cluster lifecycle	6
1.2.2.4. Console	6
1.2.2.5. Deployable	6
1.2.2.6. Governance	7
1.2.2.7. Hosted cluster	7
1.2.2.8. Hosted cluster infrastructure	7
1.2.2.9. Hosted control plane	7
1.2.2.10. Hosted control plane infrastructure	7
1.2.2.11. Hosting service cluster	7
1.2.2.12. Hosted service cluster infrastructure	7
1.2.2.13. Hub cluster	7
1.2.2.14. klusterlet	7
1.2.2.15. Klusterlet add-on	7
1.2.2.16. Managed cluster	7
1.2.2.17. Placement binding	8
1.2.2.18. Placement policy	8
1.2.2.19. Subscriptions	8



# CHAPTER 1. WELCOME TO RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES

Kubernetes provides a platform for deploying and managing containers in a standard, consistent control plane. However, as application workloads move from development to production, they often require multiple fit-for-purpose Kubernetes clusters to support DevOps pipelines.

**Note:** Use of this Red Hat product requires licensing and subscription agreement.

Users, such as administrators and site reliability engineers, face challenges as they work across a range of environments, including multiple data centers, private clouds, and public clouds that run Kubernetes clusters. Red Hat Advanced Cluster Management for Kubernetes provides the tools and capabilities to address these common challenges.

Red Hat Advanced Cluster Management for Kubernetes provides end-to-end management visibility and control to manage your Kubernetes environment. Take control of your application modernization program with management capabilities for cluster creation, application lifecycle, and provide security and compliance for all of them across hybrid cloud environments. Clusters and applications are all visible and managed from a single console, with built-in security policies. Run your operations from anywhere that Red Hat OpenShift runs, and manage any Kubernetes cluster in your fleet.

The *Welcome page* from the Red Hat Advanced Cluster Management for Kubernetes console has a header that displays the *Applications* switcher to return to Red Hat OpenShift Container Platform and more. The tiles describe the main functions of the product and link to important console pages. For more information, see the [Console overview](#).

With Red Hat Advanced Cluster Management for Kubernetes:

- Work across a range of environments, including multiple data centers, private clouds and public clouds that run Kubernetes clusters.
- Easily create Kubernetes clusters and offer cluster lifecycle management in a single console.
- Enforce policies at the target clusters using Kubernetes-supported custom resource definitions.
- Deploy and maintain day-two operations of business applications distributed across your cluster landscape.

This guide assumes that users are familiar with Kubernetes concepts and terminology. For more information about Kubernetes concepts, see [Kubernetes Documentation](#).

Access the [Support matrix](#) to learn about hub cluster and managed cluster requirements and support.

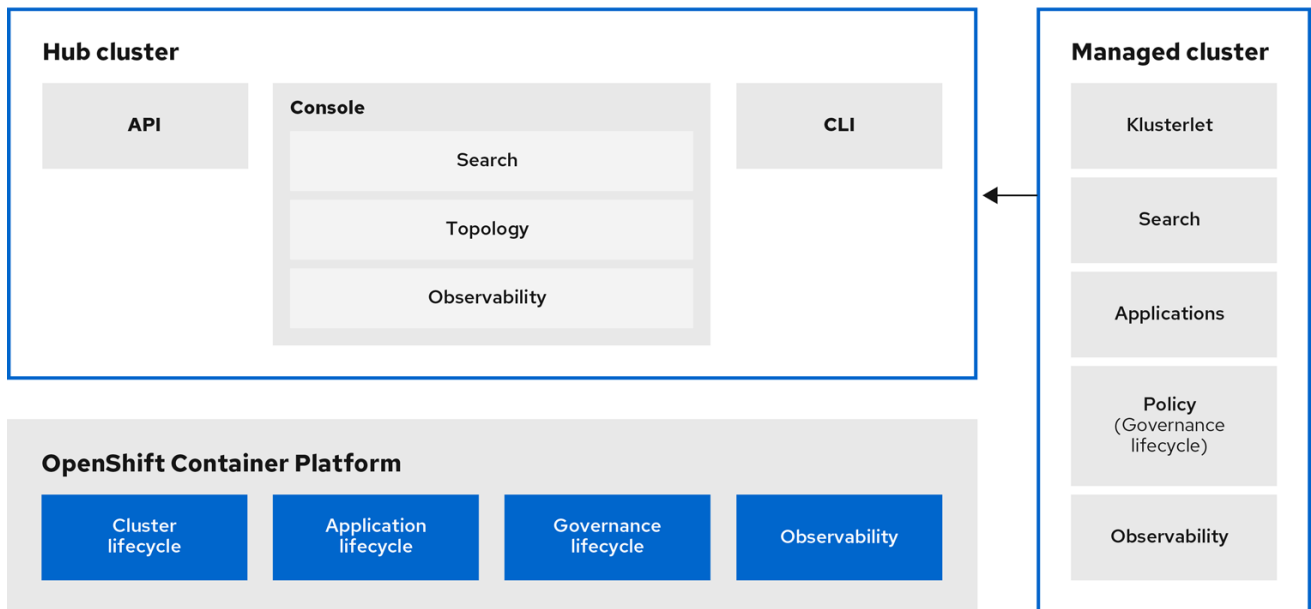
See the following documentation for information about the product:

- [Multicluster architecture](#)
- [Glossary of terms](#)

## 1.1. MULTICLUSTER ARCHITECTURE

Red Hat Advanced Cluster Management for Kubernetes consists of several multicluster components, which are used to access and manage your clusters. Learn more about the architecture in the following sections, then follow the links to more detailed documentation.

Access the [Support matrix](#) to learn about hub cluster and managed cluster requirements and support.



186\_RHACM\_1221

See the following high-level multicluster terms and components:

- [Hub cluster](#)
- [Managed cluster](#)
- [Cluster lifecycle](#)
- [Application lifecycle](#)
- [Governance](#)
- [Observability](#)
- [References](#)

### 1.1.1. Hub cluster

The *hub* cluster is the common term that is used to define the central controller that runs in a Red Hat Advanced Cluster Management for Kubernetes cluster. From the hub cluster, you can access the console and product components, as well as the Red Hat Advanced Cluster Management APIs. You can also use the console to search resources across clusters and view your topology.

Additionally, you can enable *observability* on your hub cluster to monitor metrics from your managed clusters across your cloud providers.

The Red Hat Advanced Cluster Management hub cluster uses the **MultiClusterHub** operator to manage, upgrade, and install hub cluster components and runs in the **open-cluster-management** namespace. The hub cluster aggregates information from multiple clusters by using an asynchronous work request model and search collectors. The hub cluster maintains the state of clusters and applications that run on it.

The *local cluster* is the term used to define a hub cluster that is also a managed cluster, discussed in the following sections.



## 1.1.2. Managed cluster

The *managed* cluster is the term that is used to define additional clusters that are managed by the hub cluster. The connection between the two is completed by using the *klusterlet*, which is the agent that is installed on the managed cluster. The managed cluster receives and applies requests from the hub cluster and enables it to service cluster lifecycle, application lifecycle, governance, and observability on the managed cluster.

For example, managed clusters send metrics to the hub cluster if the observability service is enabled. See [Observing environments](#) to receive metrics and optimize the health of all managed clusters.

## 1.1.3. Cluster lifecycle

Red Hat Advanced Cluster Management *cluster lifecycle* defines the process of creating, importing, managing, and destroying Kubernetes clusters across various infrastructure cloud providers, private clouds, and on-premises data centers.

The cluster lifecycle function is provided by the multicluster engine for Kubernetes operator, which is installed automatically with Red Hat Advanced Cluster Management. See [Cluster lifecycle introduction](#) for general information about the cluster lifecycle function.

From the hub cluster console, you can view an aggregation of all cluster health statuses, or view individual health metrics of many Kubernetes clusters. Additionally, you can upgrade managed OpenShift Container Platform clusters individually or in bulk, as well as destroy any OpenShift Container Platform clusters that you created using your hub cluster. From the console, you can also hibernate, resume, and detach clusters.

## 1.1.4. Application lifecycle

Red Hat Advanced Cluster Management *Application lifecycle* defines the processes that are used to manage application resources on your managed clusters. A multicluster application allows you to deploy resources on multiple managed clusters, as well as maintain full control of Kubernetes resource updates for all aspects of the application with high availability.

A multicluster application uses the Kubernetes specification, but provides additional automation of the deployment and lifecycle management of resources. Ansible Automation Platform jobs allow you to automate tasks. You can also set up a continuous GitOps environment to automate application consistency across clusters in development, staging, and production environments.

See [Managing applications](#) for more application topics.

## 1.1.5. Governance

*Governance* enables you to define policies that either enforce security compliance, or inform you of changes that violate the configured compliance requirements for your environment. Using dynamic policy templates, you can manage the policies and compliance requirements across all of your management clusters from a central interface.

For more information, see the [Risk and compliance](#) introduction. Additionally, learn about access requirements from the [Role-based access control](#) documentation.

After you configure a Red Hat Advanced Cluster Management hub cluster and a managed cluster, you can view and create policies with the Red Hat Advanced Cluster Management policy framework. You can visit the [policy-collection open community](#) to see what policies community members created and contributed, as well as contribute your own policies for others to use.

## 1.1.6. Observability

The *Observability* component collects and reports the status and health of the OpenShift Container Platform version 4.x or later, managed clusters to the hub cluster, which are visible from the Grafana dashboard. You can create custom alerts to inform you of problems with your managed clusters. Because it requires configured persistent storage, Observability must be enabled after the Red Hat Advanced Cluster Management installation.

For more information about Observability, see [Observing environments introduction](#).

## 1.1.7. References

- Learn more about the release from the [Release notes](#).
- See the product [Installing and upgrading](#) section to prepare your cluster and get configuration information.
- See [Cluster lifecycle overview](#) for more information about the operator that provides the cluster lifecycle features.

## 1.2. GLOSSARY OF TERMS

Red Hat Advanced Cluster Management for Kubernetes consists of several multicluster components that are defined in the following sections. Additionally, some common Kubernetes terms are used within the product. Terms are listed alphabetically.

### 1.2.1. Relevant standardized glossaries

- [Kubernetes glossary](#)

### 1.2.2. Red Hat Advanced Cluster Management for Kubernetes terms

#### 1.2.2.1. Application lifecycle

The processes that are used to manage application resources on your managed clusters. A multicluster application uses a Kubernetes specification, but with additional automation of the deployment and lifecycle management of resources to individual clusters.

#### 1.2.2.2. Channel

A custom resource definition that points to repositories where Kubernetes resources are stored, such as Git repositories, Helm chart repositories, ObjectStore repositories, or namespaces templates on the hub cluster. Channels support multiple subscriptions from multiple targets.

#### 1.2.2.3. Cluster lifecycle

Defines the process of creating, importing, and managing clusters across public and private clouds.

#### 1.2.2.4. Console

The graphical user interface for Red Hat Advanced Cluster Management for Kubernetes.

#### 1.2.2.5. Deployable

A resource that retrieves the output of a build, packages the output with configuration properties, and installs the package in a pre-defined location so that it can be tested or run.

### 1.2.2.6. Governance

The Red Hat Advanced Cluster Management for Kubernetes processes used to manage security and compliance.

### 1.2.2.7. Hosted cluster

An OpenShift Container Platform API endpoint that is managed by HyperShift.

### 1.2.2.8. Hosted cluster infrastructure

Resources that exist in the customer cloud account, including network, compute, storage, and so on.

### 1.2.2.9. Hosted control plane

An OpenShift Container Platform control plane that is running on the hosting service cluster, which is exposed by the API endpoint of a hosted cluster. The component parts of a control plane include **etcd**, **apiserver**, **kube-controller-manager**, **vpn**, and other components.

### 1.2.2.10. Hosted control plane infrastructure

Resources on the management cluster or external cloud provider that are prerequisites to running hosted control plane processes.

### 1.2.2.11. Hosting service cluster

An OpenShift Container Platform cluster that hosts the HyperShift operator and zero-to-many hosted clusters.

### 1.2.2.12. Hosted service cluster infrastructure

Resources of the hosting service cluster, including network, compute, storage, and so on.

### 1.2.2.13. Hub cluster

The central controller that runs in a Red Hat Advanced Cluster Management for Kubernetes cluster. From the hub cluster, you can access the console and components found on that console, as well as APIs.

### 1.2.2.14. klusterlet

The agent that contains two controllers on the managed cluster that initiates a connection to the Red Hat Advanced Cluster Management for Kubernetes hub cluster.

### 1.2.2.15. Klusterlet add-on

Specialized controller on the Klusterlet that provides additional management capability.

### 1.2.2.16. Managed cluster

Created and imported clusters are managed by the klusterlet agent and its add-ons, which initiates a connection to the Red Hat Advanced Cluster Management for Kubernetes hub cluster.

#### **1.2.2.17. Placement binding**

A resource that binds a placement to a policy.

#### **1.2.2.18. Placement policy**

A policy that defines where the application components are deployed and how many replicas there are.

#### **1.2.2.19. Subscriptions**

A resource that identifies the Kubernetes resources within channels (resource repositories), then places the Kubernetes resource on the target clusters.