Red Hat Advanced Cluster Management for Kubernetes 2.1

About

About

Last Updated: 2021-12-06

About

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

About Red Hat Advanced Cluster Management for Kubernetes, including architecture, components and quick start

Table of Contents

CHAPTER 1. WELCOME TO RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES 3
1.1. MULTICLUSTER ARCHITECTURE 4
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 and risk 5
1.2. COMPONENTS AND SERVICES 6
1.2.1. Default enabled services 6
1.2.2. Default disabled services 6
1.2.3. Default enabled components for hub cluster 6
1.2.4. Default enabled components for managed cluster 9
1.3. GETTING STARTED 9
1.3.1. Introduction 10
1.3.2. Install 10
1.3.3. Manage clusters 10
1.3.4. Manage applications 10
1.3.5. Manage security 11
1.4. GLOSSARY OF TERMS 11
1.4.1. Relevant standardized glossaries 11
1.4.2. Red Hat Advanced Cluster Management for Kubernetes terms 11
1.4.2.1. Application lifecycle 11
1.4.2.2. Channel 11
1.4.2.3. Cluster lifecycle 11
1.4.2.4. Console 11
1.4.2.5. Deployable 11
1.4.2.6. Governance and risk 11
1.4.2.7. Hub cluster 12
1.4.2.8. Managed cluster 12
1.4.2.9. Klusterlet 12
1.4.2.10. Klusterlet add-on 12
1.4.2.11. Placement policy 12
1.4.2.12. Placement rule 12
1.4.2.13. Subscriptions 12

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

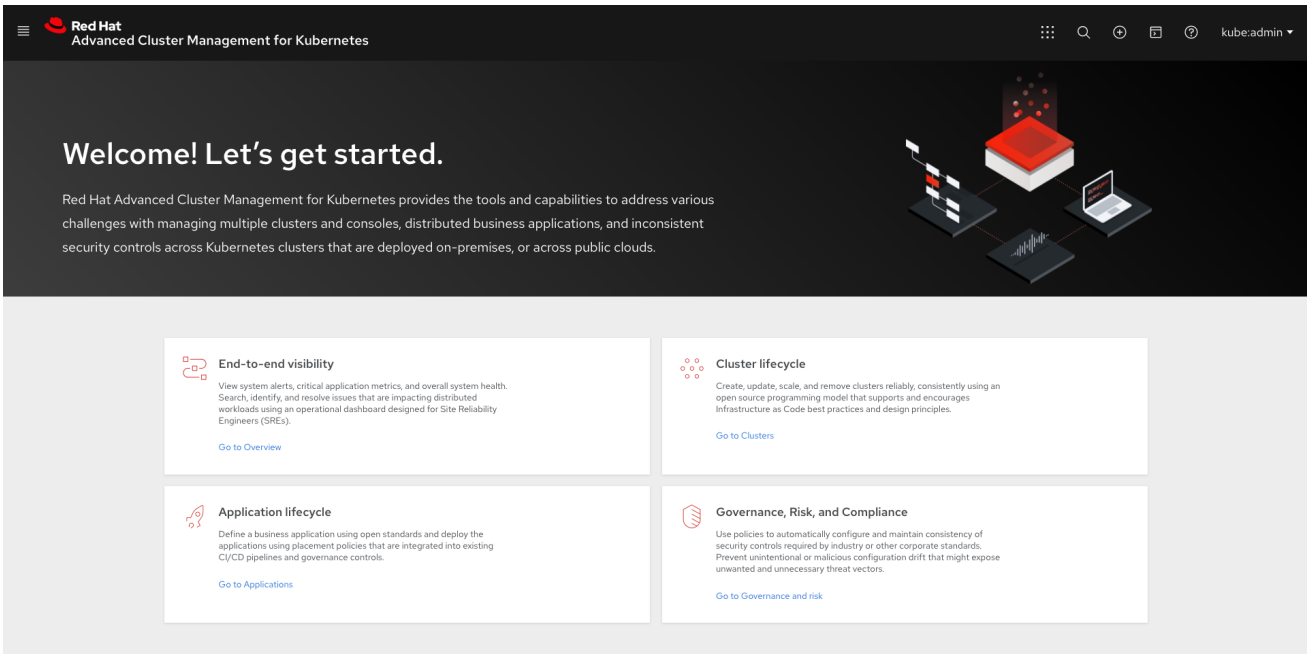
Kubernetes provides a platform for deploying and managing containers in a standard, consistent control plane. However, as application workloads move from development to production, they often require multiple fit-for-purpose Kubernetes clusters to support DevOps pipelines.

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 data centers and 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.

See the following image of the *Welcome page* from the Red Hat Advanced Cluster Management for Kubernetes console. The header displays the *Applications* icon to return to OpenShift Container Platform, access to the Visual Web Terminal, and more. The tiles describe the main functions of the product and link to important console pages.



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.

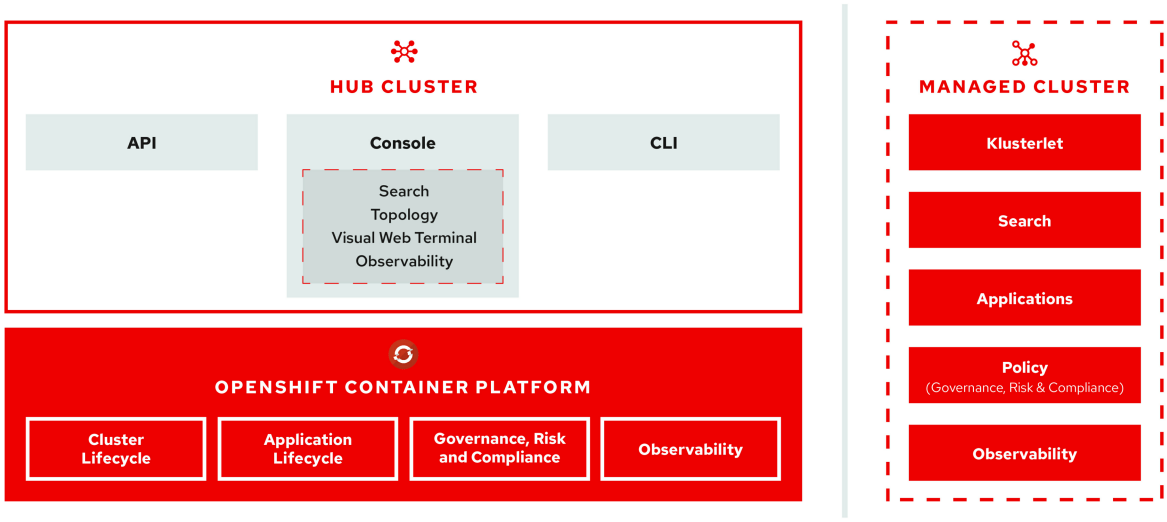
This guide assumes that users are familiar with Kubernetes concepts and terminology. For more information about Kubernetes concepts, see Kubernetes Documentation.

See the following documentation for information about the product:

- Multicluster architecture
- Components and services
- Installing

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.



Learn more about the following components for Red Hat Advanced Cluster Management for Kubernetes:

- Hub cluster
- Managed cluster
- Cluster lifecycle
- Application lifecycle
- Governance and risk

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 APIs such as the **rcm-api**, which handles API requests

related to cluster lifecycle management, which is defined later in this topic.

From the hub cluster, you can use the console to search resources across clusters, run commands with the Visual Web Terminal, and view your topology. Additionally, you can enable *observability* to monitor metrics from your managed clusters and cloud providers.

The hub cluster aggregates information from multiple clusters by using an asynchronous work request model. With a graph database, the hub cluster maintains the state of clusters and applications that run on it.

The hub cluster also uses **etcd**, a distributed key value store, to store the state of work requests and results from multiple clusters, and provides a set of REST APIs for the various functions that it supports.

1.1.2. Managed cluster

The *managed* cluster is the term that is used to define additional clusters with the *Klusterlet*, which is the agent that initiates a connection to the Red Hat Advanced Cluster Management for Kubernetes hub cluster. The managed cluster receives and applies requests, then returns the results.

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 for Kubernetes *cluster lifecycle* defines the process of creating, importing, and managing clusters across public and private clouds.

From the hub cluster console, you can view an aggregation of all cluster health statuses, or view individual health metrics. You can upgrade managed Red Hat OpenShift clusters individually or in bulk, as well as destroy any Red Hat OpenShift clusters that you created from your hub cluster.

See Managing your clusters to learn about managing clusters, which is part of *Cluster lifecycle*.

1.1.4. Application lifecycle

Red Hat Advanced Cluster Management for Kubernetes *Application lifecycle* defines 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.

A multicluster application allows you to deploy resources on multiple clusters, while maintaining easy-to-reconcile service routes, as well as full control of Kubernetes resource updates for all aspects of the application.

See Managing applications for more application topics.

1.1.5. Governance and risk

Governance and risk is the term used to define the processes that are used to manage security and compliance from a central interface page. After you configure a Red Hat Advanced Cluster Management for Kubernetes hub cluster and a managed cluster, you can view and create policies with the Red Hat Advanced Cluster Management policy framework.

For more information about Governance and risk, see the Security introduction. Additionally, learn about access requirements from the Role-based access control documentation.

See the product Installing section to prepare your cluster and get configuration information.

See the Components and services topic to learn more about what is installed with the product.

1.2. COMPONENTS AND SERVICES

See the following tables for information about components and services that are enabled or disabled on the hub cluster and the managed cluster. See what is required and if it applies, the default value that is required.

- Default enabled services
- Default disabled services
- Default enabled components for hub cluster
- Default enabled components for managed cluster

1.2.1. Default enabled services

Service	Description	Required
cert-manager	This service manages the lifecycle of certificates.	Yes
cert-manager-webhook	This service extends the Kubernetes API server so the certificate manager resources can be dynamically validated.	Yes
configmap-watcher	This service can be used to restart pods when a dependent config map is updated.	Yes
management-ingress	Unifies all management services behind a network ingress controller with predictable annotations and TLS protection. This service should not be accessed directly.	Yes

1.2.2. Default disabled services

Service	Description	Required
multicluster-observability-operator	Observability and alert management service	No

1.2.3. Default enabled components for hub cluster

Service	Description	Required
application-ui	The web console for managing the lifecycle of applications.	Yes
console-api	Provide the backend API for the console-ui service.	Yes
console-ui	Provide a view of the resources available in the cluster with support for creation, modification, or removal.	Yes
console-header	Provides a set of services available in the header of the web console.	Yes
cluster-manager	Operator for installing cluster manager on a hub cluster.	Yes
cluster-manager-registration-controller	Service that handles cluster registration on hub cluster.	Yes
cluster-manager-registration-webhook	This service extends the Kubernetes API server to validate and mutate registration resource.	Yes
cluster-manager-work-webhook	This service extends the Kubernetes API server to validate and mutate work resource.	Yes
grc-ui	The web console for Governance and Risk management in Red Hat Advanced Cluster Management for Kubernetes.	Yes
grc-ui-api	The API service for managing Governance and Risk in Red Hat Advanced Cluster Management for Kubernetes.	Yes
grc-policy-propogator	Process events and requests to Red Hat Advanced Cluster Management for Kubernetes resources.	Yes
hive	Provides cluster provisioning and lifecycle management.	Yes

Service	Description	Required
klusterlet-addon-controller	Controller that creates, updates, and deletes klusterlet-addons on the managed cluster.	Yes
kui-web-terminal	Provides the Visual Web Terminal.	Yes
managedcluster-import-controller	Controller used for cluster lifecycle management.	Yes
multicluster-operators	Manages and reconciles subscriptions.	Yes
multiclusterhub-operator	Operator for installing Red Hat Advanced Cluster Management for Kubernetes on a hub cluster.	Yes
multiclusterhub-repo	Service for hosting internal-facing Helm repository, pre-populated with required component Helm charts.	Yes
ocm-controller	Controller provides cluster information, role, inventory related resource on hub cluster.	Yes
ocm-proxyserver	Server that is used for cluster status and proxy log request to managed clusters.	Yes
ocm-webhook	Service that extends the Kubernetes API server to validate and mutate API resource.	Yes
redisgraph-tls	Cache for search data.	Yes
search-aggregator	Receives and indexes data from **search-collector** in managed clusters	Yes
search-api	Provides the API for the search service.	Yes
search-collector	Provides the capability to search for resources using the console and Visual Web Terminal.	Yes
topology	Provides the user interface for the Topology dashboard.	Yes

Service	Description	Required
topology-api	Provides the backend services for the topology-ui service.	Yes

1.2.4. Default enabled components for managed cluster

The following table shows the services that are installed on the managed cluster:

Service	Description	Required
applicationManager	Processes events and other requests to managed resources.	Yes
certPolicyController	Monitors certificate expiration based on distributed policies.	Yes
iamPolicyController	Monitors identity controls based on distributed policies	Yes
imageRegistry	quay.io/open-cluster-management	Yes
klusterlet	Operator for installing Klusterlet on managed cluster.	Yes
klusterlet-addon-operator	Handles the deployment of internal Helm chart components.	Yes
klusterlet-addon-workmgr	Component that handles cluster info, managedClusterAction, and managedClusterView.	Yes
klusterlet-registration-agent	Agent that is used for managed cluster registration.	Yes
klusterlet-work-agent	Enables work, runs logic on managed cluster.	Yes
policyController	Distributes configured policies and monitors Kubernetes-based policies.	Yes
searchCollector	Collects cluster data to be indexed by search components on the hub cluster.	Yes

1.3. GETTING STARTED

1.3.1. Introduction

See the product architecture at Multicluster architecture. Review the Components and services topics to learn more about what is installed with Red Hat Advanced Cluster Management for Kubernetes when you accept the license and subscription agreement.

After you learn about the hub cluster and managed cluster architecture, learn about the Supported clouds, which lists the cloud provider cluster options.

The hub cluster is a Red Hat OpenShift cluster version 4.4, 4.5, or 4.6 and can run on any supported Red Hat OpenShift Container Platform infrastructure.

The Glossary of terms defines common terms for the product.

If you experience problems, see the Troubleshooting guide to learn about the **must-gather** command and see documented troubleshooting tasks that might help resolve issues.

1.3.2. Install

1. Before you install Red Hat Advanced Cluster Management for Kubernetes, review the system configuration requirements and settings at Requirements and recommendations. Get information about required operating systems and supported browsers. For instance, you want to ensure that you have a supported Red Hat OpenShift Container Platform version so that you can set up your hub cluster.

2. You also need to ensure that your hub cluster has the appropriate capacity. To prepare your hub cluster, see Preparing your hub cluster for installation .

3. With a supported version of OpenShift Container Platform installed and running on your hub cluster, you can proceed with Installing while connected online .

After installation, review the Web console guide to learn how to access your console and what features are available in the console.

1.3.3. Manage clusters

You are now ready to create and import clusters. From your hub cluster, you can create clusters from other Kubernetes services to manage, and you can view cluster information.

1. See Creating a cluster to learn about the types of managed clusters you can create. When you create a managed cluster, the new managed cluster imports automatically.

2. If you have a cluster that you want to import manually, you can view Importing a target managed cluster to the hub cluster to learn how to import a managed cluster.

3. When you no longer need to manage a cluster, you can detach that cluster from the *Cluster page*.

1.3.4. Manage applications

You can start managing applications on any created and imported managed clusters. The types of resources that you can create are applications, channels, subscriptions, and placement rules.

1. Learn more about the resources and how to create and manage them at Managing applications. Add or edit your **.yaml** to create your resources.

2. View and edit your resources from the *Applications Dashboard*.

1.3.5. Manage security

You can also manage security and compliance across your created and imported managed clusters.

1. Create a policy using the *policy* templates. See the Policy overview for details about how to create a policy with a **.yaml** template.
2. From the *Policies* page, you can view a summary of cluster and policy violations.
3. View your policies from the *Governance and risk* page in the console. You can also view policy details from the cluster *Overview*.

1.4. 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.

1.4.1. Relevant standardized glossaries

- Kubernetes glossary

1.4.2. Red Hat Advanced Cluster Management for Kubernetes terms

1.4.2.1. 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.

1.4.2.2. 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.

1.4.2.3. Cluster lifecycle

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

1.4.2.4. Console

The graphical user interface for Red Hat Advanced Cluster Management.

1.4.2.5. 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.

1.4.2.6. Governance and risk

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

1.4.2.7. 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.

1.4.2.8. 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.

1.4.2.9. 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.

1.4.2.10. Klusterlet add-on

Specialized controller on the Klusterlet that provides additional management capability.

1.4.2.11. Placement policy

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

1.4.2.12. Placement rule

A rule that defines the target clusters where subscriptions are delivered. For instance, verify the cluster name, resource annotations, or resource label(s).

1.4.2.13. Subscriptions

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