# Red Hat Advanced Cluster Management for Kubernetes 2.0

## Release notes

Red Hat Advanced Cluster Management for Kubernetes Release notes

Last Updated: 2021-06-10

# Red Hat Advanced Cluster Management for Kubernetes 2.0 Release notes

Red Hat Advanced Cluster Management for Kubernetes Release notes

## Legal Notice



## Abstract

Red Hat Advanced Cluster Management for Kubernetes release notes, what's new and known issues

# Table of Contents

# CHAPTER 1. RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES RELEASE NOTES

- What's new in Red Hat Advanced Cluster Management for Kubernetes
- Fix pack updates
- Known issues and limitations
- Red Hat Advanced Cluster Management for Kubernetes considerations for GDPR readiness

## 1.1. WHAT'S NEW IN RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES

Red Hat Advanced Cluster Management for Kubernetes is now a generally available product. See what is available in version 2.0.

Red Hat Advanced Cluster Management for Kubernetes provides visibility of your entire Kubernetes domain with built-in governance, cluster lifecycle management, and application lifecycle management.

- Get an overview of Red Hat Advanced Cluster Management for Kubernetes from Welcome to Red Hat Advanced Cluster Management for Kubernetes.
- See the Multicluster architecture topic to learn more about major components of the product.
- The Getting started guide references common tasks that get you started, as well as the *Troubleshooting guide*.

### 1.1.1. Installation

With operator-based installation, you can install a Red Hat OpenShift Container Platform cluster on a configured cloud provider, such as Amazon Web Services, in less than 10 minutes. See Installing while connected online for more information.

### 1.1.2. Cluster management

- Create clusters on various Kubernetes service providers. You can provision and manage Red Hat OpenShift Container Platform clusters on selected Kubernetes cloud service providers. See Creating a cluster with Red Hat Advanced Cluster Management for Kubernetes for more information.
- Import existing Kubernetes clusters. Import your existing Kubernetes clusters that are hosted on popular cloud service providers, or on private clouds to manage your clusters conveniently in one place. See Importing a target managed cluster to the hub cluster for more information.
- Manage all of your Red Hat OpenShift Container Platform cluster upgrades in one interface. You can upgrade imported and provisioned Red Hat OpenShift Container Platform clusters either individually or in groups by using the console.

### 1.1.3. Application management

Deploy and maintain business applications distributed across your clusters. This is accomplished through subscription-based automation.

You can also view the complete picture of your applications and their resource statuses from the topology page in the console.

- Subscriptions are Kubernetes resources that serve as sets of definitions for identifying Kubernetes resources (in GitHub, Objectstores, or hub deployables) and Helm charts within channels by using annotations, labels, and versions.
- Application resources are used to group and view the components across your applications.
- Placement rules define where and how your applications are subscribed. Use placement rules to help you facilitate multicluster deployments.
- Channel resources define the source you subscribe to get your application components. (Git, Objectstore, Helm repository or templates (deployables) on the hub.

For more information, see Managing applications.

### 1.1.4. Security and compliance

Red Hat Advanced Cluster Management for Kubernetes supports several roles and uses Kubernetes authorization mechanisms. For more information, see Role-based access control.

Use the product governance framework to enhance the security for your managed clusters. With the Governance and risk dashboard, you can view and manage the number of security risks and policy violations in your clusters and applications.

Create custom policy controllers to report and validate the compliance of your policies on your cluster. Enable and manage the following policy controllers that are installed by default:

- Certificate policy controller
- Kubernetes configuration policy controller
- IAM policy controller

See Governance and risk to learn more about the dashboard and the policy framework.

As you create policies, use the policy element, **templates** to describe how your resource is defined. For more information about the policy elements, see Manage security policies.

## 1.2. ERRATA UPDATES

By default, Errata updates are automatically applied. See Upgrading by using the operator for more information.

**Important:**

- For reference, Errata links and GitHub numbers might be added to the content and used internally. Links that require access might not be available for the user.
- Red Hat OpenShift Container Platform 4.7 is not supported with 2.0.x Errata.

### 1.2.1. Errata 2.0.11

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.11 updates:

**Important:** Red Hat OpenShift Container Platform 4.5 is not supported with 2.0.11. You must run Red Hat OpenShift Container Platform version 4.6 to upgrade to Red Hat Advanced Cluster Management version 2.0.11. If you cannot upgrade your Red Hat OpenShift Container Platform version to 4.6, you can continue to use Red Hat Advanced Cluster Management version 2.0.10.

1. Updated Search code to use data from other fields as a result of Kubernetes **selfLink** removal, which impacted Search logic that depended on those fields. (GitHub 11904)

2. Fixed an issue that caused unexpected policies to display in the *Policy listing* page. (GitHub 11853)

3. Removed obsolete translated console content. (GitHub 12640)

### 1.2.2. Errata 2.0.10

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.10 updates:

1. Fixed the *Clusters* page to display the imported cluster distributor version. (GitHub 11776)

2. Updated Red Hat OpenShift release **ClusterImageSets** that are available when you create a new cluster. (GitHub 10928)

3. Fixed an issue with bare metal assets reporting incorrect status. (GitHub 10009)

### 1.2.3. Errata 2.0.9

The Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.9 resolved container upgrade requirements.

### 1.2.4. Errata 2.0.8

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.8 updates:

1. Fixed an issue with the hub cluster subscription crashing when a Helm subscription is created for subscribing resources from a private Helm repository channel, where only **spec.SecretRef** is defined. The private Helm repo channel secret must be defined in the same channel namespace. (Bugzilla 1925281)

2. Fixed an issue with the **cert-manager-webhook** pod failing to start with a permissions issue. The image was updated so there is no longer a dependency on specific user permissions. (GitHub 9913)

3. Fixed issue with pod crashing for a Helm subscription with a **Deployment** kind template that doesn't contain **spec.replicas**, or contains a **spec.replicas** value that is not an integer. ( Bugzilla 1921531)

### 1.2.5. Errata 2.0.7

The Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.7 resolved identified security CVEs.

### 1.2.6. Errata 2.0.6

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.6 updates:

1. Fixed an issue that cluster destroy on Google Cloud Platform was not cleaning up all Service Accounts. (GitHub 5948)
2. Fixed an issue that caused a temporary error on the create resources page after you detach a managed cluster. (GitHub 6299)
3. Fixed an issue that prevented the complete destroying or detaching of a Microsoft Azure managed cluster after the addition of the cluster failed. (GitHub 6353)
4. Fixed an issue that caused bare metal clusters to fail to upgrade to 2.1.0 due to memory errors. (GitHub 6898) (Bugzilla 1895799)
5. Corrected a PATH error when starting a new Visual Web Terminal session. (GitHub 6928)
6. Resolved an issue with the subscription **timewindow** function that sometimes prevented it from transitioning to and from **blocking** and **unblocking** at the scheduled times. (GitHub 7337)

### 1.2.7. Errata 2.0.5

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.5 updates:

1. Updated the certificate for security components. (GitHub 6368)
2. Added support for Red Hat OpenShift Container Platform 4.6.1. (GitHub 6545)
3. Added the ClusterImageSet resource for Red Hat OpenShift Container Platform version 4.6.1. (GitHub 6696)
4. Improved the application flow for policies. (1890827)

### 1.2.8. Errata 2.0.4

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.4 updates:

1. Increased the default memory for **search-operator** pod for upgrade. (1882748)
2. Provided a solution for the search pod collector to prevent crashes. (1883694)
3. Provided a solution for a problem with provisioned Bare Metal clusters remaining in **Pending import** state. (1860233)
4. Added viewer restrictions for **ManagedClusterAction** resource. (GitHub 5843)
5. Enhanced certificate refresh process for agents. (GitHub 4914)

### 1.2.9. Errata 2.0.3

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.3 updates:

1. Added upgrade and install improvements and fixes.
2. Resolved resource leaks in **open-cluster-management** that created system instability.
3. Improved bare metal workload messaging since worker nodes are not required.
4. Fixed bare metal provider connection edit function, along with other bare metal usability issues.

5. Resolved a webhook validation error that caused uninstall failure.
6. Fixed a Klusterlet search pods crash.
7. Added policy improvements.
8. In the Console, fixed the following inconsistencies and added the following improvements:
    a. Fixed instability in *Application overview* page applications list.
    b. Resolved *Governance and risk* page failing if a policy annotation is missing.
    c. Fixed *Topology* inconsistencies for policy violations.
    d. Fixed refresh settings on Policy violation pages.
    e. Fixed subscriptions that were propagated, but failing in the console.
    f. Added scroll to cloud providers list to show Bare metal option.
    g. Enabled DNS VIP field in bare metal cluster create console.

### 1.2.10. Errata 2.0.2

Errata 2.0.2 resolves a rare problem that caused some managed cluster imports to fail after upgrading from version 2.0.0 to version 2.0.1. You must upgrade to Errata 2.0.1 before upgrading to Errata 2.0.2.

### 1.2.11. Errata 2.0.1

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.0.1 updates.

1. The cluster import process was improved.
2. Upgraded the **oc** and **kubectl** CLIs to the latest versions for the Visual Web Terminal.
3. Administrator (**admin**) role access to the pod logs of managed clusters is fixed.
4. The product uninstallation process was improved.
5. Added a label for **Bare metal** to the Cloud field options list, on the *Importing a cluster* page.
6. The default **Network type** when you create a cluster is updated from OpenShiftSDN to OVNKubernetes.
7. Subscriptions support **kustomization.yaml** files that contains an inline patch where the patch content inside the file is a single string.
8. Improved how cloud providers manage sensitive data.
9. Removed DNS virtual IP parameter from the create cluster flow.
10. Overview page does not become blank when clusters are detached.

## 1.3. KNOWN ISSUES

Review the known issues for Red Hat Advanced Cluster Management for Kubernetes. The following list contains known issues for this release, or known issues that continued from the previous release.

- Installation known issues
- Web console known issues
- Cluster management known issues
- Application management known issues
- Security known issues

## 1.3.1. Installation known issues

### 1.3.1.1. OpenShift Container Platform cluster upgrade failed status

When an OpenShift Container Platform cluster is in the upgrade stage, the cluster pods are restarted and the cluster might remain in **upgrade failed** status for a variation of 1-5 minutes. This behavior is expected and resolves after a few minutes.

### 1.3.1.2. Certificate manager must not exist during an installation

Certificate manager must not exist on a cluster when you install Red Hat Advanced Cluster Management for Kubernetes.

When certificate manager already exists on the cluster, Red Hat Advanced Cluster Management for Kubernetes installation fails.

To resolve this issue, verify if the certificate manager is present in your cluster by running the following command:

```
kubectl get crd | grep certificates.certmanager
```

## 1.3.2. Web console known issues

### 1.3.2.1. Node discrepancy between Cluster page and search results

You might see a discrepancy between the nodes displayed on the *Cluster* page and the *Search* results.

### 1.3.2.2. LDAP user names are case-sensitive

LDAP user names are case-sensitive. You must use the name exactly the way it is configured in your LDAP directory.

### 1.3.2.3. Console features might not display in Firefox earlier versions

The product supports Mozilla Firefox 74.0 or the latest version that is available for Linux, macOS, and Windows. Upgrade to the latest version for the best console compatibility.

### 1.3.2.4. Unable to search using values with empty spaces

From the console and Visual Web Terminal, users are unable to search for values that contain an empty space.

#### 1.3.2.5. At logout user kubeadmin gets extra browser tab with blank page

When you are logged in as **kubeadmin** and you click the **Log out** option in the drop-down menu, the console returns to the login screen, but a browser tab opens with a **/logout** URL. The page is blank and you can close the tab without impact to your console.

#### 1.3.2.6. Create resource fails in the console

After you select the **Create resource** button on the *Welcome* page, you might receive an alert about an error that occurred when the resource was created.

To resolve this issue complete the following steps:

1. Clear the browser cache and cookies.
2. Log in to the Red Hat Advanced Cluster Management console.
3. Click **Create resource** to try again.

**Important**: The information that you entered is lost when you refresh the *Create resource* page.

### 1.3.3. Cluster management known issues

#### 1.3.3.1. Console might report managed cluster policy inconsistency

After a cluster is imported, log in to the imported cluster and make sure all pods that are deployed by the Klusterlet are running. Otherwise, you might see inconsistent data in the console.

For example, if a policy controller is not running, you might not get the same results of violations on the *Governance and risk* page and the *Cluster status*.

For instance, you might see 0 violations listed in the *Overview* status, but you might have 12 violations reported on the *Governance and risk* page.

In this case, inconsistency between the pages represents a disconnection between the **policy-controller-addon** on managed clusters and the policy controller on the hub cluster. Additionally, the managed cluster might not have enough resources to run all the Klusterlet components.

As a result, the policy was not propagated to managed cluster, or the violation was not reported back from managed clusters.

#### 1.3.3.2. Importing clusters might require two attempts

When you import a cluster that was previously managed and detached by a Red Hat Advanced Cluster Management hub cluster, the import process might fail the first time. The cluster status is **pending import**. Run the command again, and the import should be successful.

#### 1.3.3.3. Klusterlet runs on a detached cluster

If you detach an online cluster immediately after it was attached, the Klusterlet starts to run on the detached cluster before the **manifestwork** syncs. Removal of the managed cluster from the hub cluster does not uninstall the Klusterlet. Complete the following steps to fix the issue:

1. Download the **cleanup-managed-cluster** script from the **deploy** Git repository.
2. Run the **cleanup-managed-cluster.sh** script by entering the following command:

```
./cleanup-managed-cluster.sh
```

#### 1.3.3.4. Importing certain versions of IBM Red Hat OpenShift Kubernetes Service clusters is not supported

You cannot import IBM Red Hat OpenShift Kubernetes Service version 3.11 clusters. Later versions of IBM OpenShift Kubernetes Service are supported.

#### 1.3.3.5. Detaching OpenShift Container Platform 3.11 does not remove the *open-cluster-manangement-agent*

When you detach managed clusters on OpenShift Container Platform 3.11, the **open-cluster-management-agent** namesapce is not automatically deleted. Manually remove the namespace by running the following command:

```
oc delete ns open-cluster-management-agent
```

#### 1.3.3.6. Automatic secret updates for provisioned clusters is not supported

When you change your cloud provider access key, the provisioned cluster access key is not updated in the namespace. Run the following command for your cloud provider to update the access key:

- Amazon Web Services (AWS)

```
oc patch secret {CLUSTER-NAME}-aws-creds -n {CLUSTER-NAME} --type json -p='[{"op": "add", "path": "/stringData", "value":{"aws_access_key_id": "{YOUR-NEW-ACCESS-KEY-ID}","aws_secret_access_key":"{YOUR-NEW-aws_secret_access_key}"} }]'
```

- Google Cloud Platform (GCP)

```
oc set data secret/{CLUSTER-NAME}-gcp-creds -n {CLUSTER-NAME} --from-file=osServiceAccount.json=$HOME/.gcp/osServiceAccount.json
```

- Microsoft Azure

```
oc set data secret/{CLUSTER-NAME}-azure-creds -n {CLUSTER-NAME} --from-file=osServiceAccount.json=$HOME/.azure/osServiceAccount.json
```

#### 1.3.3.7. Resources remain after you detach an offline managed cluster

When you detach a managed cluster that is in an offline state, there are some resources that cannot be removed from managed cluster. Complete the following steps to remove the additional resources:

1. Make sure you have the **oc** command line interface configured.
2. Make sure you have **KUBECONFIG** configured on your managed cluster.
   If you run **oc get ns | grep open-cluster-management-agent** you should see two namespaces:

```
open-cluster-management-agent         Active   10m
open-cluster-management-agent-addon   Active   10m
```

3. Download the **cleanup-managed-cluster** script from the **deploy** Git repository.
4. Run the **cleanup-managed-cluster.sh** script by entering the following command:

```
./cleanup-managed-cluster.sh
```

5. Run the following command to ensure that both namespaces are removed:

```
oc get ns | grep open-cluster-management-agent
```

#### 1.3.3.8. Cannot run management ingress as non-root user

You must be logged in as **root** to run the **management-ingress** service.

#### 1.3.3.9. Node information from the managed cluster cannot be viewed in search

Search maps RBAC for resources in the hub cluster. Depending on user RBAC settings for Red Hat Advanced Cluster Management, users might not see node data from the managed cluster. Results from search might be different from what is displayed on the *Nodes* page for a cluster.

### 1.3.4. Application management known issues

#### 1.3.4.1. YAML manifest cannot create multiple resoures

The **managedclusteraction** doesn't support multiple resources. You cannot apply the YAML manifest with multiple resource from console create resources features.

#### 1.3.4.2. Console pipeline cards might display different data

Search results for your pipeline return an accurate number of resources, but that number might be different in the pipeline card because the card displays resources not yet used by an application.

For instance, after you search for **kind:channel**, you might see you have 10 channels, but the pipeline card on the console might represent only 5 channels that are used.

#### 1.3.4.3. Namespace channel subscription remains in failed state

When you subscribe to a namespace channel and the subscription remains in **FAILED** state after you fixed other associated resources such as channel, secret, configmap, or placement rule, the namespace subscription is not continuously reconciled.

To force the subscription reconcile again to get out of **FAILED** state, complete the following steps:

1. Log in to your hub cluster.
2. Manually add a label to the subscription using the following command:

```
oc label subscriptions.apps.open-cluster-management.io the_subscription_name reconcile=true
```

#### 1.3.4.4. Deployable resources in a namespace channel

You need to manually create deployable resources within the channel namespace.

To create deployable resources correctly, add the following two labels that are required in the deployable to the subscription controller that identifies which deployable resources are added:

```
labels:
  apps.open-cluster-management.io/channel: <channel name>
  apps.open-cluster-management.io/channel-type: Namespace
```

Don't specify template namespace in each deployable **spec.template.metadata.namespace**.

For the namespace type channel and subscription, all the deployable templates are deployed to the subscription namespace on managed clusters. As a result, those deployable templates that are defined outside of the subscription namespace are skipped.

See Creating and managing channels for more information.

#### 1.3.4.5. Edit role for application error

A user performing in an **Editor** role should only have **read** or **update** authority on an application, but erroneously editor can also **create** and **delete** an application. Red Hat OpenShift Operator Lifecycle Manager default settings change the setting for the product. To workaround the issue, see the following procedure:

1. Run **oc edit clusterrole applications.app.k8s.io-v1beta1-edit -o yaml** to open the application edit cluster role.
2. Remove **create** and **delete** from the verbs list.
3. Save the change.

#### 1.3.4.6. Edit role for placement rule error

A user performing in an **Editor** role should only have **read** or **update** authority on an placement rule, but erroneously editor can also **create** and **delete**, as well. Red Hat OpenShift Operator Lifecycle Manager default settings change the setting for the product. To workaround the issue, see the following procedure:

1. Run **oc edit clusterrole placementrules.apps.open-cluster-management.io-v1-edit** to open the application edit cluster role.
2. Remove **create** and **delete** from the verbs list.
3. Save the change.

#### 1.3.4.7. Application not deployed after an updated placement rule

If applications are not deploying after an update to a placement rule, verify that the **klusterlet-addon-appmgr** pod is running. The **klusterlet-addon-appmgr** is the subscription container that needs to run on endpoint clusters.

You can run `oc get pods -n open-cluster-management-agent-addon ` to verify.

You can also search for **kind:pod cluster:yourcluster** in the console and see if the **klusterlet-addon-appmgr** is running.

If you cannot verify, attempt to import the cluster again and verify again.

#### 1.3.4.8. Subscription operator does not create an SCC

Learn about OpenShift Container Platform SCC at Managing Security Context Constraints (SCC) , which is an additional configuration required on the managed cluster.

Different deployments have different security context and different service accounts. The subscription operator cannot create an SCC automatically. Administrators control permissions for pods. A Security Context Constraints (SCC) CR is required to enable appropriate permissions for the relative service accounts to create pods in the non-default namespace:

To manually create an SCC CR in your namespace, complete the following:

1. Find the service account that is defined in the deployments. For example, see the following **nginx** deployments:

```
nginx-ingress-52edb
nginx-ingress-52edb-backend
```

2. Create an SCC CR in your namespace to assign the required permissions to the service account or accounts. See the following example where **kind: SecurityContextConstraints** is added:

```
apiVersion: security.openshift.io/v1
defaultAddCapabilities:
kind: SecurityContextConstraints
metadata:
  name: ingress-nginx
  namespace: ns-sub-1
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities:
fsGroup:
  type: RunAsAny
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
users:
- system:serviceaccount:my-operator:nginx-ingress-52edb
- system:serviceaccount:my-operator:nginx-ingress-52edb-backend
```

#### 1.3.4.9. Application channels in unique namespaces

Creating more than one channel in the same namespace can cause errors with the hub cluster. For instance, namespace **charts-v1** is used by the installer as a Helm type channel, so do not create any additional channels in **charts-v1**.

It is best practice to create each channel in a unique namespace. However, a Git channel can share a namespace with another type of channel including Git, Helm, Kubernetes Namespace, and Object store.

### 1.3.5. Security known issues

#### 1.3.5.1. Internal error 500 during login to the console

When Red Hat Advanced Cluster Management for Kubernetes is installed and the OpenShift Container Platform is customized with a custom ingress certificate, a **500 Internal Error** message appears. You are unable to access the console because the OpenShift Container Platform certificate is not included in the Red Hat Advanced Cluster Management for Kubernetes management ingress. Add the OpenShift Container Platform certificate by completing the following steps:

1. Create a ConfigMap that includes the certificate authority used to sign the new certificate. Your ConfigMap must be identical to the one you created in the **openshift-config** namespace. Run the following command:

```
oc create configmap custom-ca \
    --from-file=ca-bundle.crt=</path/to/example-ca.crt> \
    -n open-cluster-management
```

2. Edit your **multiclusterhub** YAML file by running the following command:

```
oc edit multiclusterhub multiclusterhub
```

   a. Update the **spec** section by editing the parameter value for **customCAConfigmap**. The parameter might resemble the following content:

```
customCAConfigmap: custom-ca
```

After you complete the steps, wait a few minutes for the changes to propagate to the charts and log in again. The OpenShift Container Platform certificate is added.

#### 1.3.5.2. Cluster name is not listed in the policy detail panel

All cluster violations from specific policies are listed in the policy detail panel. If a user does not have role access to a cluster, the cluster name is not visible. The cluster name is displayed with the following symbol: **-**

#### 1.3.5.3. Empty status in policies

The policies that are applied to the cluster are considered **NonCompliant** when clusters are not running. When you view violation details, the **status** parameter is empty.

#### 1.3.5.4. Placement rule and policy binding empty

After creating or modifying a policy, the placement rule and the policy binding might be empty in the policy details of the Red Hat Advanced Cluster Management console. This is generally because the policy is disabled, or there was some other updates made to the policy. Ensure that the settings are set correctly for the policy in the YAML view.

#### 1.3.5.5. Recovering *cert-manager* after removing the helm release

If you remove the **cert-manager** and the **cert-manager-webhook-helmreleases**, the Helm releases are triggered to automatically redeploy the charts and generate a new certificate. The new certificate must be synced to the other helm charts that create other Red Hat Advanced Cluster Management components. To recover the certificate components from the hub cluster, complete the following steps:

1. Remove the helm release for **cert-manager** by running the following commands:

```
oc delete helmrelease cert-manager-5ffd5
oc delete helmrelease cert-manager-webhook-5ca82
```

2. Verify that the helm release is recreated and the pods are running.
3. Make sure the certificate is generated by running the following command:

```
oc get certificates.certmanager.k8s.io
```

You might receive the following respoonse:

```
(base) ➜  cert-manager git:(master) ✗ oc get certificates.certmanager.k8s.io
NAME                      READY   SECRET                    AGE   EXPIRATION
multicloud-ca-cert        True    multicloud-ca-cert        61m   2025-09-27T17:10:47Z
```

4. Update the other components with this certificate, by downloading and running **generate-update-issuer-cert-manifest.sh** script.
5. Verify that all of the secrets from **oc get certificates.certmanager.k8s.io** have the ready state **True**.

## 1.4. RED HAT ADVANCED CLUSTER MANAGEMENT FOR KUBERNETES PLATFORM CONSIDERATIONS FOR GDPR READINESS

### 1.4.1. Notice

This document is intended to help you in your preparations for General Data Protection Regulation (GDPR) readiness. It provides information about features of the Red Hat Advanced Cluster Management for Kubernetes platform that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness. This information is not an exhaustive list, due to the many ways that clients can choose and configure features, and the large variety of ways that the product can be used in itself and with third-party clusters and systems.

**Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.**

**The products, services, and other capabilities described herein are not suitable for all client situations and may have restricted availability. Red Hat does not provide legal, accounting, or auditing advice or represent or warrant that its services or products will ensure that clients are in compliance with any law or regulation.**

### 1.4.2. Table of Contents

- Data Collection
- Data Storage
- Data Access
- Data Processing
- Data Deletion
- Capability for Restricting Use of Personal Data
- Appendix

### 1.4.3. GDPR

General Data Protection Regulation (GDPR) has been adopted by the European Union ("EU") and applies from May 25, 2018.

#### 1.4.3.1. Why is GDPR important?

GDPR establishes a stronger data protection regulatory framework for processing personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

#### 1.4.3.2. Read more about GDPR

- EU GDPR Information Portal
- Red Hat GDPR website

### 1.4.4. Product Configuration for GDPR

The following sections describe aspects of data management within the Red Hat Advanced Cluster Management for Kubernetes platform and provide information on capabilities to help clients with GDPR requirements.

### 1.4.5. Data Life Cycle

Red Hat Advanced Cluster Management for Kubernetes is an application platform for developing and managing on-premises, containerized applications. It is an integrated environment for managing containers that includes the container orchestrator Kubernetes, cluster lifecycle, application lifecycle, and security frameworks (governance, risk, and compliance).

As such, the Red Hat Advanced Cluster Management for Kubernetes platform deals primarily with technical data that is related to the configuration and management of the platform, some of which

might be subject to GDPR. The Red Hat Advanced Cluster Management for Kubernetes platform also deals with information about users who manage the platform. This data will be described throughout this document for the awareness of clients responsible for meeting GDPR requirements.

This data is persisted on the platform on local or remote file systems as configuration files or in databases. Applications that are developed to run on the Red Hat Advanced Cluster Management for Kubernetes platform might deal with other forms of personal data subject to GDPR. The mechanisms that are used to protect and manage platform data are also available to applications that run on the platform. Additional mechanisms might be required to manage and protect personal data that is collected by applications run on the Red Hat Advanced Cluster Management for Kubernetes platform.

To best understand the Red Hat Advanced Cluster Management for Kubernetes platform and its data flows, you must understand how Kubernetes, Docker, and the Operator work. These open source components are fundamental to the Red Hat Advanced Cluster Management for Kubernetes platform. You use Kubernetes deployments to place instances of applications, which are built into Operators that reference Docker images. The Operator contain the details about your application, and the Docker images contain all the software packages that your applications need to run.

#### 1.4.5.1. What types of data flow through Red Hat Advanced Cluster Management for Kubernetes platform

As a platform, Red Hat Advanced Cluster Management for Kubernetes deals with several categories of technical data that could be considered as personal data, such as an administrator user ID and password, service user IDs and passwords, IP addresses, and Kubernetes node names. The Red Hat Advanced Cluster Management for Kubernetes platform also deals with information about users who manage the platform. Applications that run on the platform might introduce other categories of personal data unknown to the platform.

Information on how this technical data is collected/created, stored, accessed, secured, logged, and deleted is described in later sections of this document.

#### 1.4.5.2. Personal data used for online contact

Customers can submit online comments/feedback/requests for information about in a variety of ways, primarily:

- The public Slack community if there is a Slack channel
- The public comments or tickets on the product documentation
- The public conversations in a technical community

Typically, only the client name and email address are used, to enable personal replies for the subject of the contact, and the use of personal data conforms to the Red Hat Online Privacy Statement .

### 1.4.6. Data Collection

The Red Hat Advanced Cluster Management for Kubernetes platform does not collect sensitive personal data. It does create and manage technical data, such as an administrator user ID and password, service user IDs and passwords, IP addresses, and Kubernetes node names, which might be considered personal data. The Red Hat Advanced Cluster Management for Kubernetes platform also deals with information about users who manage the platform. All such information is only accessible by the system administrator through a management console with role-based access control or by the system administrator though login to a Red Hat Advanced Cluster Management for Kubernetes platform node.

Applications that run on the Red Hat Advanced Cluster Management for Kubernetes platform might collect personal data.

When you assess the use of the Red Hat Advanced Cluster Management for Kubernetes platform running containerized applications and your need to meet the requirements of GDPR, you must consider the types of personal data that are collected by the application and aspects of how that data is managed, such as:

- How is the data protected as it flows to and from the application? Is the data encrypted in transit?
- How is the data stored by the application? Is the data encrypted at rest?
- How are credentials that are used to access the application collected and stored?
- How are credentials that are used by the application to access data sources collected and stored?
- How is data collected by the application removed as needed?

This is not a definitive list of the types of data that are collected by the Red Hat Advanced Cluster Management for Kubernetes platform. It is provided as an example for consideration. If you have any questions about the types of data, contact Red Hat.

### 1.4.7. Data storage

The Red Hat Advanced Cluster Management for Kubernetes platform persists technical data that is related to configuration and management of the platform in stateful stores on local or remote file systems as configuration files or in databases. Consideration must be given to securing all data at rest. The Red Hat Advanced Cluster Management for Kubernetes platform supports encryption of data at rest in stateful stores that use **dm-crypt**.

The following items highlight the areas where data is stored, which you might want to consider for GDPR.

- **Platform Configuration Data:** The Red Hat Advanced Cluster Management for Kubernetes platform configuration can be customized by updating a configuration YAML file with properties for general settings, Kubernetes, logs, network, Docker, and other settings. This data is used as input to the Red Hat Advanced Cluster Management for Kubernetes platform installer for deploying one or more nodes. The properties also include an administrator user ID and password that are used for bootstrap.
- **Kubernetes Configuration Data:** Kubernetes cluster state data is stored in a distributed key-value store, **etcd**.
- **User Authentication Data, including User IDs and passwords:** User ID and password management are handled through a client enterprise LDAP directory. Users and groups that are defined in LDAP can be added to Red Hat Advanced Cluster Management for Kubernetes platform teams and assigned access roles. Red Hat Advanced Cluster Management for Kubernetes platform stores the email address and user ID from LDAP, but does not store the password. Red Hat Advanced Cluster Management for Kubernetes platform stores the group name and upon login, caches the available groups to which a user belongs. Group membership is not persisted in any long-term way. Securing user and group data at rest in the enterprise LDAP must be considered. Red Hat Advanced Cluster Management for Kubernetes platform also includes an authentication service, Open ID Connect (OIDC) that interacts with the enterprise directory and maintains access tokens. This service uses MongoDB as a backing store.

- **Service authentication data, including user IDs and passwords:** Credentials that are used by Red Hat Advanced Cluster Management for Kubernetes platform components for inter-component access are defined as Kubernetes Secrets. All Kubernetes resource definitions are persisted in the **etcd** key-value data store. Initial credentials values are defined in the platform configuration data as Kubernetes Secret configuration YAML files. For more information, see Managing secrets.

### 1.4.8. Data access

Red Hat Advanced Cluster Management for Kubernetes platform data can be accessed through the following defined set of product interfaces.

- Web user interface (the console)
- Kubernetes **kubectl** CLI
- Red Hat Advanced Cluster Management for Kubernetes CLI
- oc CLI

These interfaces are designed to allow you to make administrative changes to your Red Hat Advanced Cluster Management for Kubernetes cluster. Administration access to Red Hat Advanced Cluster Management for Kubernetes can be secured and involves three logical, ordered stages when a request is made: authentication, role-mapping, and authorization.

#### 1.4.8.1. Authentication

The Red Hat Advanced Cluster Management for Kubernetes platform authentication manager accepts user credentials from the console and forwards the credentials to the backend OIDC provider, which validates the user credentials against the enterprise directory. The OIDC provider then returns an authentication cookie (**auth-cookie**) with the content of a JSON Web Token ( **JWT**) to the authentication manager. The JWT token persists information such as the user ID and email address, in addition to group membership at the time of the authentication request. This authentication cookie is then sent back to the console. The cookie is refreshed during the session. It is valid for 12 hours after you sign out of the console or close your web browser.

For all subsequent authentication requests made from the console, the front-end NGINX server decodes the available authentication cookie in the request and validates the request by calling the authentication manager.

The Red Hat Advanced Cluster Management for Kubernetes platform CLI requires the user to provide credentials to log in.

The **kubectl** and **oc** CLI also requires credentials to access the cluster. These credentials can be obtained from the management console and expire after 12 hours. Access through service accounts is supported.

#### 1.4.8.2. Role Mapping

Red Hat Advanced Cluster Management for Kubernetes platform supports role-based access control (RBAC). In the role mapping stage, the user name that is provided in the authentication stage is mapped to a user or group role. The roles are used when authorizing which administrative activities can be carried out by the authenticated user.

#### 1.4.8.3. Authorization

Red Hat Advanced Cluster Management for Kubernetes platform roles control access to cluster configuration actions, to catalog and Helm resources, and to Kubernetes resources. Several IAM (Identity and Access Management) roles are provided, including Cluster Administrator, Administrator, Operator, Editor, Viewer. A role is assigned to users or user groups when you add them to a team. Team access to resources can be controlled by namespace.

#### 1.4.8.4. Pod Security

Pod security policies are used to set up cluster-level control over what a pod can do or what it can access.

### 1.4.9. Data Processing

Users of Red Hat Advanced Cluster Management for Kubernetes can control the way that technical data that is related to configuration and management is processed and secured through system configuration.

**Role-based access control** (RBAC) controls what data and functions can be accessed by users.

**Data-in-transit** is protected by using **TLS**. **HTTPS** (**TLS** underlying) is used for secure data transfer between user client and back end services. Users can specify the root certificate to use during installation.

**Data-at-rest** protection is supported by using **dm-crypt** to encrypt data.

These same platform mechanisms that are used to manage and secure Red Hat Advanced Cluster Management for Kubernetes platform technical data can be used to manage and secure personal data for user-developed or user-provided applications. Clients can develop their own capabilities to implement further controls.

### 1.4.10. Data Deletion

Red Hat Advanced Cluster Management for Kubernetes platform provides commands, application programming interfaces (APIs), and user interface actions to delete data that is created or collected by the product. These functions enable users to delete technical data, such as service user IDs and passwords, IP addresses, Kubernetes node names, or any other platform configuration data, as well as information about users who manage the platform.

Areas of Red Hat Advanced Cluster Management for Kubernetes platform to consider for support of data deletion:

- All technical data that is related to platform configuration can be deleted through the management console or the Kubernetes **kubectl** API.

Areas of Red Hat Advanced Cluster Management for Kubernetes platform to consider for support of account data deletion:

- All technical data that is related to platform configuration can be deleted through the Red Hat Advanced Cluster Management for Kubernetes or the Kubernetes **kubectl** API.

Function to remove user ID and password data that is managed through an enterprise LDAP directory would be provided by the LDAP product used with Red Hat Advanced Cluster Management for Kubernetes platform.

### 1.4.11. Capability for Restricting Use of Personal Data

Using the facilities summarized in this document, Red Hat Advanced Cluster Management for Kubernetes platform enables an end user to restrict usage of any technical data within the platform that is considered personal data.

Under GDPR, users have rights to access, modify, and restrict processing. Refer to other sections of this document to control the following:

- Right to access
    - Red Hat Advanced Cluster Management for Kubernetes platform administrators can use Red Hat Advanced Cluster Management for Kubernetes platform features to provide individuals access to their data.
    - Red Hat Advanced Cluster Management for Kubernetes platform administrators can use Red Hat Advanced Cluster Management for Kubernetes platform features to provide individuals information about what data Red Hat Advanced Cluster Management for Kubernetes platform holds about the individual.
- Right to modify
    - Red Hat Advanced Cluster Management for Kubernetes platform administrators can use Red Hat Advanced Cluster Management for Kubernetes platform features to allow an individual to modify or correct their data.
    - Red Hat Advanced Cluster Management for Kubernetes platform administrators can use Red Hat Advanced Cluster Management for Kubernetes platform features to correct an individual's data for them.
- Right to restrict processing
    - Red Hat Advanced Cluster Management for Kubernetes platform administrators can use Red Hat Advanced Cluster Management for Kubernetes platform features to stop processing an individual's data.

### 1.4.12. Appendix

As a platform, Red Hat Advanced Cluster Management for Kubernetes deals with several categories of technical data that could be considered as personal data, such as an administrator user ID and password, service user IDs and passwords, IP addresses, and Kubernetes node names. Red Hat Advanced Cluster Management for Kubernetes platform also deals with information about users who manage the platform. Applications that run on the platform might introduce other categories of personal data that are unknown to the platform.

This appendix includes details on data that is logged by the platform services.