Red Hat Advanced Cluster Management for Kubernetes 2.3

Release notes

Read more about Release notes for what's new, errata updates, known issues, deprecations and removals, and product considerations for GDPR and FIPS readiness.

Last Updated: 2022-08-31

Red Hat Advanced Cluster Management for Kubernetes 2.3 Release notes

Read more about Release notes for what's new, errata updates, known issues, deprecations and removals, and product considerations for GDPR and FIPS readiness.

Legal Notice

Abstract

Read more about Release notes for what's new, errata updates, known issues, deprecations and removals, and product considerations for GDPR and FIPS readiness.

Table of Contents

CHAPTER 1. RELEASE NOTES

The 2.1 version of Red Hat Advanced Cluster Management is *removed* and no longer supported. The documentation might remain available, but it is deprecated without any Errata or other updates available. Earlier versions of the documentation are also not supported.

- What's new in Red Hat Advanced Cluster Management for Kubernetes
- Errata updates
- Known issues and limitations
- Deprecations and removals
- 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 provides visibility of your entire Kubernetes domain with built-in governance, cluster lifecycle management, and application lifecycle management, along with observability. With this release, you can move towards managing clusters in more environments, GitOps integration for applications, and more.

Important: Some features and components are identified and released as Technology Preview.

Learn more about what is new this release:

- Get an overview of Red Hat Advanced Cluster Management for Kubernetes from Welcome to Red Hat Advanced Cluster Management for Kubernetes.
- The open source *Open Cluster Management* repository is ready for interaction, growth, and contributions from the open community. To get involved, see open-cluster-management.io. You can access the GitHub repository for more information, as well.
- 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*.
- Web console
  - Observability
- Clusters
- Applications
- Governance

1.1.1. Web console

- The changes to the side-bar navigation align with other products and offer a better user experience. From the navigation, you can access various product features. With the header, you can more easily access Red Hat OpenShift Container Platform, Search, the *Configure client* page, view the *About modal*, and more.

- **Technology Preview:** You can access Visual Web Terminal from the navigation.

1.1.1.1. Observability

- Red Hat Advanced Cluster Management observability service supports 7.4.2 of Grafana. See the Observability service section to learn more information.
- You can now configure the observability storage size per component. See Creating the MultiClusterObservability CR for more information.
- The API storage version for observability is now **v1beta2**. Using **v1beta2** for the version retrieves both **v1beta1** and **v1beta2** custom resource definitions.
- Add recording rules in the observability service to specify the new metrics, which come from the aggregate query expression. See Adding custom metrics for more information.
- You can now customize the **advanced** configuration in the **MultiClusterObservability** custom resource. See Adding *advanced* configuration for more information.
- You can now remove default metrics. For more information, see Removing default metrics.
- Receive information about potential problems in your connected clusters now with Red Hat Insights. For more information, see Observability with Red Hat Insights .
- You can now view the *etcd* dashboard from the Grafana console. See Viewing the etcd table.
- Identify the metrics that are coming from a Single Node OpenShift (SNO) cluster with the SNO label. See Viewing and exploring data for more information.
- You can now use Bring Your Own (BYO) observability certificate authority (CA) certificates. See Bring Your Own (BYO) observability certificate authority (CA) certificates for more information.
- You can now update the number of replicas for your observability pods. See Updating the *multiclusterobservability* CR replicas from the console for more information.
- Forward alerts from managed clusters to the **Alertmanager** in the Red Hat Advanced Cluster Management hub cluster. See Forwarding alerts for more information.
- Use the external API for metrics to be queried through the OpenShift Container Platform route, **rbac-query-proxy**. See Using the external metric query for more information.

1.1.2. Clusters

- You can now select your OpenShift Container Platform version 4.6 or later channel for your cluster upgrades in the Red Hat Advanced Cluster Management console. The channel selection informs you of available upgrades for your cluster. See Selecting a channel for more information.
- Updated the cluster creation process with the console with a more intuitive progression. See Creating a cluster for more information.
- You can now edit the HiveConfig resource directly, and the **MultiClusterHub** operator does not revert the changes. If the HiveConfig resource is deleted, the **MultiClusterHub** operator recreates it exactly as it was configured when the **MultiClusterHub** resource was first created.

- Your credentials now update automatically on the managed cluster when you update them on the hub cluster.

- Create an OpenShift Container Platform managed cluster on the Red Hat OpenStack Platform by using the Red Hat Advanced Cluster Management console. See Creating a cluster on Red Hat OpenStack Platform for more information.

- You can now import OpenShift Container Platform clusters for management that are hosted on IBM Power systems.

- Added information about managing bare metal assets using **BareMetalAsset** CRs and the Red Hat Advanced Cluster Management web console. See Creating and modifying bare metal assets for more information.

1.1.2.1. Clusters (Technology Preview)

The following capabilities are *Technology Preview* this release:

- You can test the ability to host the hub cluster on IBM Power systems.

- You can now group resources in a **ManagedClusterSet** to control the RBAC access permissions for managed clusters, cluster pools, cluster deployments, and cluster claims. See Creating and managing ManagedClusterSets (Technology Preview) for more information.

- Configure **AnsibleJob** Templates to initiate with the installation or upgrade of a managed cluster. See Configuring Ansible Tower tasks to run on managed clusters for more information.

- Create cluster pools to better manage your resources and have configured OpenShift Container Platform clusters available to claim when you need them. See Managing cluster pools for more information.

- You can hibernate certain OpenShift Container Platform managed clusters that are created by Red Hat Advanced Cluster Management to manage your resources with more flexibility. See Hibernating a created cluster for more information.

- You can now configure the Submariner networking service on VMware vSphere and Google Cloud Platform managed clusters. See Submariner networking service for more information.

- You can now deploy Submariner on your clusters by using the Red Hat Advanced Cluster Management console. See Deploying Submariner with the console for more information.

- You can use the **MachinePools** resource to scale your clusters by using the Red Hat Advanced Cluster Management or the command line. See Resizing a cluster for more information.

- Discover OpenShift Container Platform 4 clusters that are now available from OpenShift Cluster Manager. After discovering clusters, you can import your clusters to manage. See the Discovery service introduction (Technology Preview) for information.

1.1.3. Applications

- You are now directed to the *Applications* page after you select an application to view from the *Search* page. See Query ArgoCD applications for more information.

- Now if you deploy Argo applications on an OpenShift Container Platform cluster with Red Hat Advanced Cluster Management installed, you can visualize Argo applications in the *Applications* table and in the *Topology* view.

- From the *Overview* or the *Topology* overview, you can launch an Argo editor and manage your Argo application.

- Other improvements to the Application console include a *Commit hash* and *Tag*, which are specific to the Git repository channel type. Additionally, new reconcile inputs are added for both Git and Helm repository types.

- You can now select reconcile frequency options: high, medium, low, and off in channel configuration to avoid unnecessary resource reconciliations and prevent overload on subscription operator. See *Reconcile option* in Subscribing Git resources for more information.

- The *Repository reconcile rate* is added to the console with the default value set as **medium**. If auto-reconcile is disabled, the reconcile option is hidden because the resources will not either merge or replace what is currently reconciled.

- You can set up subscriptions to subscribe resources that are defined in the Amazon Simple Storage Service (Amazon S3) object storage service. See Managing apps with Object storage repositories for more information.

- From the console, you can view your **ApplicationSet**, which represents Argo applications that are generated from the **ApplicationSet** controller. For information about the Application console, see the Application console overview.

For other Application topics, see Managing applications.

1.1.4. Governance

- You can now add or include templates in configuration policies. See Support for templates in configuration policies for more information.

- Red Hat Advanced Cluster Management now uses Red Hat OpenShift Container Platform Service Serving Certificate. For more information, see Certificates.

- You can now create policy violation automations with Ansible Tower. For more information, see Configuring Ansible Tower for governance .

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

To see more release note topics, go to the Release notes.

1.2. ERRATA UPDATES

By default, errata updates are automatically applied when released. 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.

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**.

1.2.1. Errata 2.3.12

- Delivers updates to one or more of the product container images and security fixes.

1.2.2. Errata 2.3.11

- Resolves an issue that sometimes caused observability to fail because of an incorrect pod name after upgrading from version 2.2. (Bugzilla #2087277)
- Delivers updates to one or more of the product container images and security fixes.

1.2.3. Errata 2.3.10

- Delivers updates to one or more of the product container images and security fixes.

1.2.4. Errata 2.3.9

- Delivers updates to one or more of the product container images and security fixes.

1.2.5. Errata 2.3.8

- Delivers updates to one or more of the product container images and security fixes.

1.2.6. Errata 2.3.7

- Delivers updates to one or more of the product container images and security fixes.

1.2.7. Errata 2.3.6

- Fixes an error when upgrading that suggested that the information for the CRD in the previous version was not updated. (Bugzilla #2015663)
- Resolves the occasional issue of limited bandwidth on a recently upgraded managed cluster. (Bugzilla #2021128)
- Fixes an issue that prevented the add-ons of managed clusters from displaying in the Red Hat Advanced Cluster Management console after upgrading Red Hat Advanced Cluster Management from version 2.3.5 to 2.3.6. (Bugzilla #2050847)
- Delivers updates to one or more of the product container images.

1.2.8. Errata 2.3.5

- Delivers updates to one or more of the product container images.

1.2.9. Errata 2.3.4

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.3.4 updates:

- Fixes an issue that prevented Red Hat Advanced Cluster Management version 2.3 from working correctly with Red Hat OpenShift Container Platform version 4.9. Beginning with version 2.3.4, Red Hat Advanced Cluster Management is now supported when running on OpenShift Container Platform version 4.9. (Bugzilla #1984470)

- Fixed an issue that prevented the Resource Optimization dashboard from showing the data in the CPU usage panel on OpenShift Container Platform version 4.9 clusters. (Bugzilla #2021766)

- Delivers updates to one or more of the product container images.

1.2.10. Errata 2.3.3

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.3.3 updates:

- Fixes an issue that caused an application subscription to fail when proxy is enabled because the **NO_PROXY** setting that contains the source Git repository is not applied. (Bugzilla #2000951)

- Fixes an issue that caused VMware managed clusters to appear in the section labeled **Other** on the console *Overview* page, rather than in the **VMware** section. (Bugzilla #2004188)

- Fixes an issue that caused the memory requirements of the Grafana pods to continually grow with active clients against it, rather than stabilizing. (GitHub #13382)

- Fixes the Red Hat Advanced Cluster Management console link from the OpenShift Container Platform header bar when it is deployed on Red Hat OpenShift Kubernetes Service (ROKS). (GitHub #14353)

- Fixes an issue that sometimes caused the **management-ingress** pods to restart multiple times in the **ppc64le** environment. (GitHub #15729)

- Fixes an issue that prevented a user with the **cluster-manager-admin** role from creating, editing, or deleting policy automations. (GitHub #15750)

- Fixes an issue where an application deployed in Red Hat Advanced Cluster Management version 2.1 would not correctly display the topology after upgrading to Red Hat Advanced Cluster Management version 2.3. (GitHub #15765)

- Fixes issues with the multicluster observability operator that caused some managed clusters to be degraded after upgrading. (GitHub #15996, #16123)

- Delivers updates to one or more of the product container images.

1.2.11. Errata 2.3.2

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.3.2 updates:

- Fixes the links from the console to the credentials documentation. (GitHub #14993)

- Fixes the issue that prevented the multicluster observability operands from successfully upgrading. (Bugzilla #1993188)

- Delivers updates to one or more of the product container images.

1.2.12. Errata 2.3.1

Fixes issues from the 2.3 version of a few product images.

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. For your Red Hat OpenShift Container Platform cluster, see OpenShift Container Platform known issues .

- Installation known issues
- Web console known issues
  - Observability known issues
- Cluster management known issues
- Application management known issues
- Governance known issues

1.3.1. Installation known issues

1.3.1.1. Compatibility issues with OpenShift Container Platform version 4.9

The functional testing to determine the compatibility between OpenShift Container Platform version 4.9 and Red Hat Advanced Cluster Management version 2.3 is still in progress. Until the compatibility between them is verified, there might be compatibility issues when using Red Hat Advanced Cluster Management version 2.3 on OpenShift Container Platform version 4.9.

1.3.1.2. Upgrade from version 2.2.x to 2.3.1 upgrade fails to progress

When you upgrade your Red Hat Advanced Cluster Management from version 2.2.x to 2.3.1, the upgrade fails. The **Multiclusterhub** status displays: **failed to download chart from helm repo** in the component error messages. You may also see errors that reference a problem with **no endpoints available for service "ocm-webhook"**.

On your hub cluster, run the following command in the namespace where Red Hat Advanced Cluster Management is installed to restart deployments and upgrade to version 2.3.1:

```
oc delete deploy ocm-proxyserver ocm-controller ocm-webhook multiclusterhub-repo
```

Note: The errors resolve, but the reconciliation process might not start immediately. This can be accelerated by restarting the **multicluster-operators-standalone-subscription** in the same namespace that the product is installed.

1.3.1.3. Upgrade from version 2.3.0 to 2.3.1 fails with a ImagePullBackOff error

When you upgrade your Red Hat Advanced Cluster Management from version 2.3.0 to 2.3.1, the **klusterlet-addon-operator** pod in the **open-cluster-management-agent-addon** namespace on your managed clusters returns an **ImagePullBackOff** error.

Complete the following steps on your hub cluster to fix this issue and upgrade to version 2.3.1:

1. Run the following commands to delete the **ConfigMap** for the **MultiClusterHub** manifest:

   ```
   oc delete cm -n open-cluster-management mch-image-manifest-2.3.0
   ```

2. Run the following command to restart the pod for the controller:

```
oc delete po -n open-cluster-management -lapp=klusterlet-addon-controller-v2
```

3. If the Grafana pod on the **open-cluster-management-observability** namespace of the hub also returns an **ImagePullBackOff** error after upgrading from 2.3.0 to 2.3.1, run the following command to restart the pod so it uses the correct images:

   ```
   oc delete po -n open-cluster-management -lname=multicluster-observability-operator
   ```

You are running Red Hat Advanced Cluster Management version 2.3.1.

1.3.1.4. 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.

1.3.2. 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 version

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. Secret content is no longer displayed

For security reasons, search does not display the contents of secrets found on managed clusters. When you search for a secret from the console, you might receive the following error message:

```
Unable to load resource data - Check to make sure the cluster hosting this resource is online
```

1.3.2.7. Restrictions for storage size in searchcustomization

When you update the storage size in the **searchcustomization** CR, the PVC configuration does not change. If you need to update the storage size, update the PVC (***<storageclassname>-search-redisgraph-0***) with the following command:

```
oc edit pvc <storageclassname>-search-redisgraph-0
```

1.3.3. Observability known issues

1.3.3.1. Observability endpoint operator fails to pull image

The observability endpoint operator fails if you create a pull-secret to deploy to the MultiClusterObservability CustomResource (CR) and there is no pull-secret in the **open-cluster-management-observability** namespace. When you import a new cluster, or import a Hive cluster that is created with Red Hat Advanced Cluster Management, you need to manually create a pull-image secret on the managed cluster.

For more information, see Enabling observability.

1.3.3.2. There is no data from ROKS cluster

Red Hat Advanced Cluster Management observability does not display data from an ROKS cluster on some panels within built-in dashboards. This is because ROKS does not expose any API Server metrics from servers they manage. The following Grafana dashboards contain panels that do not support ROKS clusters: **Kubernetes/API server**, **Kubernetes/Compute Resources/Workload**, **Kubernetes/Compute Resources/Namespace(Workload)**

1.3.3.3. There is no etcd data from ROKS clusters

For ROKS clusters, Red Hat Advanced Cluster Management observability does not display data in the *etcd* panel of the dashboard.

1.3.3.4. High CPU usage by the search-collector pod

When search is disabled on a hub cluster that manages 1000 clusters, the **search-collector** pod CPU usage levels are higher than normal. For a four-day period, usage was approximately 2.148Mi of CPU. You can reduce memory usage by reducing the **search-collector** to **0** replicas.

1.3.3.5. Search pods fail to complete the TLS handshake due to invalid certificates

In some rare cases, the search pods are not automatically redeployed after certificates change. This causes a mismatch of certificates across the service pods, which causes the Transfer Layer Security (TLS) handshake to fail. To fix this problem, restart the search pods to reset the certificates.

1.3.3.6. Metrics are unavailable in the Grafana console

- Annotation query failed in the Grafana console:
  When you search for a specific annotation in the Grafana console, you might receive the following error message due to an expired token:

  "Annotation Query Failed"

  Refresh your browser and verify you are logged into your hub cluster.

- Error in *rbac-query-proxy* pod:
  Due to unauthorized access to the **managedcluster** resource, you might receive the following error when you query a cluster or project:

  no project or cluster found

  Check the role permissions and update appropriately. See Role-based access control for more information.

1.3.3.7. Clusters are degraded after upgrade to 2.3.2

When you upgrade to 2.3.2 where observability is enabled, some clusters are degraded because the observability add-on is not ready. Restart the **multicluster-observability-operator** pod by running the following command:

```
oc delete po multicluster-observability-operator -n open-cluster-management
```

The observability pod is recreated.

1.3.3.8. Observability stateful set uses wrong images in disconnected environment

In rare cases with disconnected environments, some pods of the observability **StatefulSet** are stuck in the following status, **ErrPullImage** because the pods cannot pull the images. The images defined in those pods are different from the ones defined in the related **StatefulSets**. To fix this problem, you need to delete the pods that use the wrong images. The pods restart automatically and should use the correct images.

1.3.3.9. Error ingesting out-of-order samples

Observability **receive** pods report the following error message:

```
Error on ingesting out-of-order samples
```

The error message means that the time series data sent by a managed cluster, during a metrics collection interval is older than the time series data it sent in the previous collection interval. When this problem happens, data is discarded by the Thanos receivers and this might create a gap in the data shown in Grafana dashboards. If the error is seen frequently, it is recommended to increase the metrics collection interval to a higher value. For example, you can increase the interval to 60 seconds.

The problem is only noticed when the time series interval is set to a lower value, such as 30 seconds. Note, this problem is not seen when the metrics collection interval is set to the default value of 300 seconds.

1.3.4. Cluster management known issues

1.3.4.1. Namespace of the managed cluster stuck in terminating when deleting a managed cluster

In some cases, when you delete a managed cluster, the namespace of the managed cluster might be stuck in a terminating state because its **manifestworks** resources are not deleted. Run the following command to remove the remaining **manifestworks** resources and remove the namespace:

```
kubectl -n <managed-cluster-namespace> get manifestworks | grep -v NAME | awk '{print $1}' | xargs
kubectl -n <managed-cluster-namespace> patch manifestworks -p '{"metadata":{"finalizers": []}}' --
type=merge
```

1.3.4.2. Invalid cluster deployment appears to be creating

When invalid information is provided when deploying a cluster, the uncreatable cluster status displays a **creating** status, even though it was not able to start. The invalid information, such as a region specified in the deployment that does not match the region specified within the **install-config** file, causes the following error to be added to the Hive provision pods:

```
provision failed, requirements not met
```

Red Hat Advanced Cluster Management version 2.3 continues to report a status of **creating** for the cluster, though the invalid information is preventing the creation of the cluster.

1.3.4.3. Cluster provisioning with Infrastructure Operator fails

When creating OpenShift Container Platform clusters using the Infrastructure Operator, the file name of the ISO image might be too long. The long image name causes the image provisioning and the cluster provisioning to fail. To determine if this is the problem, complete the following steps:

1. View the bare metal host information for the cluster that you are provisioning by running the following command:

   ```
   oc get bmh -n <cluster_provisioning_namespace>
   ```

2. Run the **describe** command to view the error information:

   ```
   oc describe bmh -n <cluster_provisioning_namespace> <bmh_name>
   ```

3. An error similar to the following example indicates that the length of the filename is the problem:

   ```
   Status:
     Error Count:    1
     Error Message:  Image provisioning failed: ... [Errno 36] File name too long ...
   ```

If this problem occurs, it is typically on the following versions of OpenShift Container Platform, because the infrastructure operator was not using image service:

- 4.8.17 and earlier

To avoid this error, upgrade your OpenShift Container Platform to version 4.8.18 or later.

1.3.4.4. Cluster provisioning on Google Cloud Platform fails

When you try to provision a cluster on Google Cloud Platform (GCP), it might fail with the following error:

```
Cluster initialization failed because one or more operators are not functioning properly.
The cluster should be accessible for troubleshooting as detailed in the documentation linked below,
https://docs.openshift.com/container-platform/latest/support/troubleshooting/troubleshooting-
```

```
installations.html
The 'wait-for install-complete' subcommand can then be used to continue the installation
```

You can work around this error by enabling the Network Security API on the GCP project, which allows your cluster installation to continue.

1.3.4.5. Local-cluster status offline after reimporting with a different name

When you accidentally try to reimport the cluster named **local-cluster** as a cluster with a different name, the status for **local-cluster** and for the reimported cluster display `offline.

To recover from this case, complete the following steps:

1. Run the followiung command on the hub cluster to edit the setting for self-management of the hub cluster temporarily:

   ```
   oc edit mch -n open-cluster-management multiclusterhub
   ```

2. Add the setting **spec.disableSelfManagement=true**.
3. Run the following command on the hub cluster to delete and redeploy the local-cluster:

   ```
   oc delete managedcluster local-cluster
   ```

4. Enter the following command to remove the **local-cluster** management setting:

   ```
   oc edit mch -n open-cluster-management multiclusterhub
   ```

5. Remove **spec.disableSelfManagement=true** that you previously added.

1.3.4.6. Cluster status is different in different views of the console after a failed Ansible cluster creation

When you specify an invalid Ansible job template name while attempting to create a cluster and it fails, the cluster shows a different status on different screens of the console. When you view the status by selecting: **Infrastructure** > **Clusters** > **Managed clusters**, it displays a **Failed** status. When you select: **Infrastructure** > **Clusters** > **Cluster sets** > **<your_cluster_set_name>** > **Managed clusters**, the status remains stuck in **Creating**. The correct status is **Failed** for this case. You can try creating the cluster again, and enter the correct Ansible template name.

1.3.4.7. The local-cluster might not be automatically reimported

Sometimes, after you detach a local-cluster, the local-cluster might not be automatically reimported. When this happens, the local-cluster shows a constant status of **Pending Import** in the Red Hat Advanced Cluster Management console.

To reimport the local-cluster, complete the following steps:

1. Delete the klusterlet deployment by running the following command:

   ```
   oc -n open-cluster-management-agent delete deployment klusterlet
   ```

2. Restart the **managedcluster-import-controller** by running the following command:

```
oc -n open-cluster-management get pods -l app=managedcluster-import-controller-v2 | awk
'NR>1{print $1}' | xargs oc -n open-cluster-management delete pods
```

1.3.4.8. The clusterdeployment of the managed cluster is stuck in terminating state

When you delete a managed cluster that was created with the Red Hat Advanced Cluster Management console, the **clusterdeployment** of the managed cluster might get stuck in a terminating state. To bypass this issue and delete this **clusterdeployment**, manually delete the **agentclusterinstall.agent-install.openshift.io/ai-deprovision** finalizer by editing the **agentclusterinstall** resource for the cluster.

1.3.4.9. Cannot delete managed cluster namespace manually

You cannot delete the namespace of a managed cluster manually. The managed cluster namespace is automatically deleted after the managed cluster is detached. If you delete the managed cluster namespace manually before the managed cluster is detached, the managed cluster shows a continuous terminating status after you delete the managed cluster. To delete this terminating managed cluster, manually remove the finalizers from the managed cluster that you detached.

1.3.4.10. Cannot create clusters across architectures

You cannot create a managed cluster on a different architecture than the architecture of the hub cluster without creating a release image (**ClusterImageSet**) that contains files for both architectures. For example, you cannot create an **x86_64** cluster from a **ppc64le** hub cluster. The cluster creation fails because the OpenShift Container Platform release registry does not provide a multi-architecture image manifest.

To work around this issue, complete the following steps:

1. From the OpenShift Container Platform release registry, create a manifest list that includes both **x86_64** and **ppc64le** release images.

   a. Pull the manifest lists for both architectures from the Quay repository:

   ```
   $ podman pull quay.io/openshift-release-dev/ocp-release:4.8.1-x86_64
   $ podman pull quay.io/openshift-release-dev/ocp-release:4.8.1-ppc64le
   ```

   b. Log in to your private repository where you maintain your images:

   ```
   $ podman login <private-repo>
   ```

   Replace **private-repo** with the path to your repository.

   c. Add the release image manifest to your private repository by running the following commands:

   ```
   $ podman push quay.io/openshift-release-dev/ocp-release:4.8.1-x86_64 <private-
   repo>/ocp-release:4.8.1-x86_64
   $ podman push quay.io/openshift-release-dev/ocp-release:4.8.1-ppc64le <private-
   repo>/ocp-release:4.8.1-ppc64le
   ```

   Replace **private-repo** with the path to your repository.

   d. Create a manifest for the new information:

```
$ podman manifest create mymanifest
```

e. Add references to both release images to the manifest list:

```
$ podman manifest add mymanifest <private-repo>/ocp-release:4.8.1-x86_64
$ podman manifest add mymanifest <private-repo>/ocp-release:4.8.1-ppc64le
```

Replace **private-repo** with the path to your repository.

f. Merge the list in your manifest list with the existing manifest:

```
$ podman manifest push mymanifest docker://<private-repo>/ocp-release:4.8.1
```

Replace **private-repo** with the path to your repository.

2. On the hub cluster, create a release image that references the manifest in your repository.

   a. Create a **YAML** file that contains information that is similar to the following example:

```
apiVersion: hive.openshift.io/v1
kind: ClusterImageSet
metadata:
  labels:
    channel: fast
    visible: "true"
  name: img4.8.1-appsub
spec:
  releaseImage: <private-repo>/ocp-release:4.8.1
```

Replace **private-repo** with the path to your repository.

   b. Run the following command on your hub cluster to apply the changes:

```
oc apply -f <file-name>.yaml
```

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

3. Select the new release image when you create your OpenShift Container Platform cluster.

The creation process uses the merged release images to create the cluster.

1.3.4.11. Cannot reassign a cluster to cluster set by changing label

You cannot reassign a cluster or cluster set from one cluster set to another by updating the label for the cluster to the new cluster set. To move a cluster or cluster set to another one, remove it from the cluster set by using the Red Hat Advanced Cluster Management console. After you remove it from the cluster set, add it to the new cluster set by using the console.

1.3.4.12. Cannot use Ansible Tower integration with an IBM Power hub cluster

You cannot use the Ansible Tower integration when the Red Hat Advanced Cluster Management for Kubernetes hub cluster is running on IBM Power because the Ansible Automation Platform Resource Operator does not provide **ppc64le** images.

1.3.4.13. Cannot change credentials on clusters after upgrading to version 2.3

After you upgrade Red Hat Advanced Cluster Management to version 2.3, you cannot change the credential secret for any of the managed clusters that were created and managed by Red Hat Advanced Cluster Management before the upgrade.

1.3.4.14. Cannot create bare metal managed clusters on OpenShift Container Platform version 4.8

You cannot create bare metal managed clusters by using the Red Hat Advanced Cluster Management hub cluster when the hub cluster is hosted on OpenShift Container Platform version 4.8.

1.3.4.15. Create resource drop-down error

When you detach a managed cluster, the *Create resources* page might temporarily break and display the following error:

```
Error occurred while retrieving clusters info. Not found.
```

Wait until the namespace automatically gets removed, which takes 5-10 minutes after you detach the cluster. Or, if the namespace is stuck in a terminating state, you need to manually delete the namespace. Return to the page to see if the error resolved.

1.3.4.16. Hub cluster and managed clusters clock not synced

Hub cluster and manage cluster time might become out-of-sync, displaying in the console **unknown** and eventually **available** within a few minutes. Ensure that the Red Hat OpenShift Container Platform hub cluster time is configured correctly. See Customizing nodes.

1.3.4.17. Importing certain versions of IBM OpenShift Container Platform Kubernetes Service clusters is not supported

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

1.3.4.18. Detaching OpenShift Container Platform 3.11 does not remove the *open-cluster-management-agent*

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

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

1.3.4.19. 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. This is required when your credentials expire on the cloud provider where the managed cluster is hosted and you try delete the managed cluster. If something like this occurs, 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)
  You can identify this issue by a repeating log error message that reads, **Invalid JWT Signature** when you attempt to destroy the cluster. If your log contains this message, obtain a new Google Cloud Provider service account JSON key and enter the following command:

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

  Replace ***CLUSTER-NAME*** with the name of your cluster.

  Replace the path to the file **$HOME/.gcp/osServiceAccount.json** with the path to the file that contains your new Google Cloud Provider service account JSON key.

- Microsoft Azure

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

- VMware vSphere

  ```
  oc patch secret {CLUSTER-NAME}-vsphere-creds -n {CLUSTER-NAME} --type json -
  p='[{"op": "add", "path": "/stringData", "value":{"username": "{YOUR-NEW-VMware-
  username}","password":"{YOUR-NEW-VMware-password}"} }]'
  ```

1.3.4.20. Cannot run management ingress as non-root user

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

1.3.4.21. 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.

1.3.4.22. Process to destroy a cluster does not complete

When you destroy a managed cluster, the status continues to display **Destroying** after one hour, and the cluster is not destroyed. To resolve this issue complete the following steps:

1. Manually ensure that there are no orphaned resources on your cloud, and that all of the provider resources that are associated with the managed cluster are cleaned up.

2. Open the **ClusterDeployment** information for the managed cluster that is being removed by entering the following command:

   ```
   oc edit clusterdeployment/<mycluster> -n <namespace>
   ```

   Replace ***mycluster*** with the name of the managed cluster that you are destroying.

Replace ***namespace*** with the namespace of the managed cluster.

3. Remove the **hive.openshift.io/deprovision** finalizer to forcefully stop the process that is trying to clean up the cluster resources in the cloud.

4. Save your changes and verify that **ClusterDeployment** is gone.

5. Manually remove the namespace of the managed cluster by running the following command:

   ```
   oc delete ns <namespace>
   ```

   Replace ***namespace*** with the namespace of the managed cluster.

1.3.4.23. Cannot upgrade OpenShift Container Platform managed clusters on OpenShift Container Platform Dedicated with the console

You cannot use the Red Hat Advanced Cluster Management console to upgrade OpenShift Container Platform managed clusters that are in the OpenShift Container Platform Dedicated environment.

1.3.4.24. Work manager add-on search details

The search details page for a certain resource on a certain managed cluster might fail. You must ensure that the work-manager add-on in the managed cluster is in **Available** status before you can search.

1.3.4.25. Argo CD is not supported with IBM Power hub cluster

The Argo CD integration with Red Hat Advanced Cluster Management does not work on a Red Hat Advanced Cluster Management hub cluster that is running on IBM Power because there are no available **ppc64le** images.

1.3.4.26. Non-Red Hat OpenShift Container Platform managed clusters must have LoadBalancer enabled

Both Red Hat OpenShift Container Platform and non-OpenShift Container Platform clusters support the pod log feature, however non-OpenShift Container Platform clusters require **LoadBalancer** to be enabled to use the feature. Complete the following steps to enable **LoadBalancer**:

1. Cloud providers have different **LoadBalancer** configurations. Visit your cloud provider documentation for more information.

2. Verify if **LoadBalancer** is enabled on your Red Hat Advanced Cluster Management by checking the **loggingEndpoint** in the status of **managedClusterInfo**.

3. Run the following command to check if the **loggingEndpoint.IP** or **loggingEndpoint.Host** has a valid IP address or host name:

   ```
   oc get managedclusterinfo <clusterName> -n <clusterNamespace> -o json | jq -r
   '.status.loggingEndpoint'
   ```

For more information about the **LoadBalancer** types, see the *Service* page in the Kubernetes documentation.

1.3.5. Application management known issues

1.3.5.1. Application search undefined error Application table

When you click on the **Search application by row** action button on the *Applications table*, you are directed to the *Search* page with results that might differ from the same preset filters you can receive from the *Search* resource from the *Application details* page.

1.3.5.2. Application search undefined error Argo CD

When clicking on the link **Search all related applications** from the application details page for an Argo CD application, the search page returns invalid preset filters.

You can remove the directory filter in the search console to resolve this issue.

1.3.5.3. No branch information during application creation for proxy

When you create a Red Hat Advanced Cluster Management application by using the **Create application** editor, you might receive the following error for Git repositories when your hub cluster is behind a proxy:

The connection to the Git repository failed. Cannot get branches.

You can go to your Git repository URL instead to get the branch information. The application deploys properly when the branch is entered.

1.3.5.4. Application Argo search undefined error

The cluster node search link for an Argo application might return **name:undefined**.

When you click the **Launch resource in search** link from the cluster node details of an Argo application, the search filter might contain **name:undefined**.

Replace the **undefined** value with the cluster name in the cluster node details to resolve this error.

1.3.5.5. Application topology clusters with multiple subscriptions not grouped properly

A cluster might not group properly in the *Application topology* if the cluster is using multiple subscriptions.

When you deploy an application with multiple subscriptions, you might see that the *All subscriptions* view does not group the cluster nodes properly.

For instance, when you deploy an application with multiple subscriptions containing a mixed combination of *Helm* and *Git* repositories, the *All subscriptions* view does not display statuses correctly for the resources within the Helm subscription.

View the topology from the individual subscription views instead to display the correct cluster node grouping information.

1.3.5.6. Application topology subscription switch

The Application topology can fail when you switch between application subscriptions by using the subscriptions drop-down menu.

To resolve, attempt to switch to another subscription, or refresh the browser to see a refresh of the topology display.

1.3.5.7. Application Ansible hook stand-alone mode

Ansible hook stand-alone mode is not supported. To deploy Ansible hook on the hub cluster with a subscription, you might use the following subscription YAML:

```
apiVersion: apps.open-cluster-management.io/v1
kind: Subscription
metadata:
  name: sub-rhacm-gitops-demo
  namespace: hello-openshift
annotations:
  apps.open-cluster-management.io/github-path: myapp
  apps.open-cluster-management.io/github-branch: master
spec:
  hooksecretref:
      name: toweraccess
  channel: rhacm-gitops-demo/ch-rhacm-gitops-demo
  placement:
     local: true
```

However, this configuration might never create the Ansible instance, since the **spec.placement.local:true** has the subscription running on **standalone** mode. You need to create the subscription in hub mode.

1. Create a placement rule that deploys to **local-cluster**. See the following sample:

```
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: <towhichcluster>
  namespace: hello-openshift
spec:
  clusterSelector:
    matchLabels:
      local-cluster: "true" #this points to your hub cluster
```

2. Reference that placement rule in your subscription. See the following:

```
apiVersion: apps.open-cluster-management.io/v1
kind: Subscription
metadata:
  name: sub-rhacm-gitops-demo
  namespace: hello-openshift
annotations:
  apps.open-cluster-management.io/github-path: myapp
  apps.open-cluster-management.io/github-branch: master
spec:
  hooksecretref:
      name: toweraccess
  channel: rhacm-gitops-demo/ch-rhacm-gitops-demo
  placement:
     placementRef:
        name: <towhichcluster>
        kind: PlacementRule
```

After applying both, you should see the Ansible instance created in your hub cluster.

1.3.5.8. Application Deploy on local cluster limitation

If you select **Deploy on local cluster** when you create or edit an application, the application Topology does not display correctly. **Deploy on local cluster** is the option to deploy resources on your hub cluster so that you can manage it as the **local cluster**, but this is not best practice for this release.

To resolve the issue, see the following procedure:

1. Deselect the **Deploy on local cluster** option in the console.
2. Select the **Deploy application resources only on clusters matching specified labels** option.
3. Create the following label: **local-cluster : 'true'**.

1.3.5.9. 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
```

1.3.5.10. 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. OpenShift Container Platform 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-v1beta2-edit -o yaml** to open the application edit cluster role.
2. Remove **create** and **delete** from the verbs list.
3. Save the change.

1.3.5.11. 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. OpenShift Container Platform 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.

1.3.5.12. 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.

1.3.5.13. Subscription operator does not create an SCC

Learn about Red Hat 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
```

1.3.5.14. Application channels require 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**. Ensure that you create your channel in a unique namespace. All channels need an individual namespace, except GitHub channels, which can share a namespace with another GitHub channel.

1.3.5.15. Ansible Automation Platform (early access) 2.0.0 job fail

When the Ansible Automation Platform (early access) 2.0.0 is installed, **AnsibleJobs** fails to run. If you want to submit prehook and posthook **AnsibleJobs** through Red Hat Advanced Cluster Management, use the Ansible Automation Platform Resource Operator 0.1.1.

1.3.5.16. Application name requirements

An application name cannot exceed 37 characters. The application deployment displays the following error if the characters exceed this amount.

```
status:
  phase: PropagationFailed
  reason: 'Deployable.apps.open-cluster-management.io "_long_lengthy_name_" is invalid:
metadata.labels: Invalid value: "_long_lengthy_name_": must be no more than 63 characters/n'
```

1.3.5.17. Application console tables

See the following limitations to various *Application* tables in the console:

- From the *Applications* table on the *Overview* page and the *Subscriptions* table on the *Advanced configuration* page, the *Clusters* column displays a count of clusters where application resources are deployed. Since applications are defined by resources on the local cluster, the local cluster is included in the search results, whether actual application resources are deployed on the local cluster or not.

- From the *Advanced configuration* table for *Subscriptions*, the *Applications* column displays the total number of applications that use that subscription, but if the subscription deploys child applications, those are included in the search result, as well.

- From the *Advanced configuration* table for *Channels*, the *Subscriptions* column displays the total number of subscriptions on the local cluster that use that channel, but this does not include subscriptions that are deployed by other subscriptions, which are included in the search result.

1.3.6. Governance known issues

1.3.6.1. Ansible Automation jobs continue to run hourly even though no new policy violations started the automation

In OpenShift Container Platform 4.8 the TTL Controller for Finished Resources is enabled by default, which means jobs are removed hourly. This job cleanup causes the Ansible Automation Platform Resource Operator to rerun the associated automation. The automation runs again with the existing details in the **AnsibleJob** resource that was created by the policy framework. The details provided might include previously identified violations, which can mistakenly appear as a repeated violation. You can disable the controller that cleans up the jobs to prevent these duplicate violations. To disable the controller that cleans up the jobs, complete the following steps:

1. Run the following command to edit the **kubeapiservers.operator.openshift.io** resource:

   ```
   oc edit kubeapiservers.operator.openshift.io cluster
   ```

2. Find the **unsupportedConfigOverrides** section.

3. Update the **unsupportedConfigOverrides** section to contain content that resembles the following example, which disables the job cleanup feature:

   ```
   unsupportedConfigOverrides:
     apiServerArguments:
       feature-gates:
       - TTLAfterFinished=false
   ```

4. Run the following command to edit the **kubecontrollermanager** resource:

   ```
   oc edit kubecontrollermanager cluster
   ```

5. Complete steps 2 and 3 to update the same section in the **kubecontrollermanager** resource.

1.3.6.2. PlacementRule matchExpression not removed with new matchLabel

When you update a policy **PlacementRule** from **matchExpressions** to **matchLabels**, the old **matchExpression** is not removed.

See the following examples where **matchExpressions** in the first sample is changed to **matchLabels** in the second sample, but **matchExpressions** is not removed.

```
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-etcdencryption
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      - {key: environment, operator: In, values: ["test"]}
```

```
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions: []
    matchLabel: {}
```

1.3.6.3. IAM policy controller does not consider group users

When the number of users with permissions to a given **ClusterRole** is determined, the IAM policy controller only checks for Kubernetes **User** resources and does not consider users in Kubernetes **Group** resources.

1.3.6.4. Unable to log out

When you use an external identity provider to log in to Red Hat Advanced Cluster Management, you might not be able to log out of Red Hat Advanced Cluster Management. This occurs when you use Red Hat Advanced Cluster Management, installed with IBM Cloud and Keycloak as the identity providers.

You must log out of the external identity provider before you attempt to log out of Red Hat Advanced Cluster Management.

1.3.6.5. Administrator cluster manager unable to create automation policy

A user with a cluster-wide role binding to **open-cluster-management:cluster-manager-admin** is unable to create automation policies. To fix this issue, you must manually add the role to the automation policy.

Create or update a cluster role (**ClusterRole**) to add rules to the **cluster-manager-admin** role, for the **Ansible** resource. Your YAML might resemeble the following file:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: add-ansible-rules
  labels:
    rbac.authorization.k8s.io/aggregate-to-ocm-cluster-manager-admin: "true"
rules:
- apiGroups: ["tower.ansible.com"]
  resources: ["ansiblejobs"]
  verbs: ["create","get", "list", "watch", "update", "delete", "deletecollection", "patch"]
```

1.3.6.6. Gatekeeper operator installation fails

When you install the gatekeeper operator on Red Hat OpenShift Container Platform version 4.9, the installation fails. Before you upgrade OpenShift Container Platform to version 4.9.0., you must upgrade the gatekeeper operator to version 0.2.0. See Upgrading gatekeeper and the gatekeeper operator for more information.

1.3.6.7. Configuration policy listed complaint when namespace is stuck in *Terminating* state

When you have a configuration policy that is configured with **mustnothave** for the **complianceType** parameter and **enforce** for the **remediationAction** parameter, the policy is listed as compliant after a deletion request is made to the Kubernetes API. Therefore, the Kubernetes object can be stuck in a **Terminating** state while the policy is listed as compliant.

1.4. DEPRECATIONS AND REMOVALS

Learn when parts of the product are deprecated or removed from Red Hat Advanced Cluster Management for Kubernetes. Consider the alternative actions in the *Recommended action* and details, which display in the tables for the current release and for two prior releases.

Important:

- The 2.1 version of Red Hat Advanced Cluster Management is *removed* and no longer supported. The documentation might remain available, but it is deprecated without any Errata or other updates available. Earlier versions of the documentation are also not supported.
- Upgrading to the most recent version of Red Hat Advanced Cluster Management is best practice.

1.4.1. API deprecations and removals

Red Hat Advanced Cluster Management follows the Kubernetes deprecation guidelines for APIs. See the Kubernetes Deprecation Policy for more details about that policy.

Red Hat Advanced Cluster Management APIs are only deprecated or removed outside of the following timelines:

- All **V1** APIs are generally available and supported for 12 months or three releases, whichever is greater. V1 APIs are not removed, but can be deprecated outside of that time limit.
- All **beta** APIs are generally available for nine months or three releases, whichever is greater. Beta APIs are not removed outside of that time limit.
- All **alpha** APIs are not required to be supported, but might be listed as deprecated or removed if it benefits users.

1.4.2. Red Hat Advanced Cluster Management deprecations

A *deprecated* component, feature, or service is supported, but no longer recommended for use and might become obsolete in future releases. Consider the alternative actions in the *Recommended action* and details that are provided in the following table:

Product or category	Affected item	Version	Recommended action	More details and links
Applications	**HelmRepo** channel specification: usage of **insecureSkipVerify: "true"** is no longer inside the **configMapRef**	2.2	Use **insecureSkipVerify: "true"** in the channel without the **configMapRef**	See the YAML sample for the change.
Installer	Hive settings in **operator.open-cluster-management.io_multiclusterhubs_crd.yaml**	2.2	Install, then edit **hiveconfig** directly with the **oc edit hiveconfig hive** command	None
klusterlet operator	**release-2.3** channel does not receive updates	2.3 and later	To import and manage Red Hat OpenShift dedicated clusters, you must upgrade.	See Upgrading by using the operator.

Product or category	Affected item	Version	Recommended action	More details and links
Installer	Separate cert-manager settings in **operator.open-cluster-management.io_multiclusterhubs_crd.yaml**	2.3	None	None
Governance	Custom policy controller	2.3	None	None

1.4.3. Removals

A *removed* item is typically function that was deprecated in previous releases and is no longer available in the product. You must use alternatives for the removed function. Consider the alternative actions in the *Recommended action* and details that are provided in the following table:

Product or category	Affected item	Version	Recommended action	More details and links
Observability Topology	Topology access from *Observe environments* removed completely	2.2	None	Application topology is located in *Application management* and no longer in the *Observability console*.
Applications	Channel type: Namespace, removed completely	2.2	None	None
Applications	Single ArgoCD import mode, secrets imported to one ArgoCD server on the hub cluster	2.3	You can import cluster secrets into multiple ArgoCD servers	None

Product or category	Affected item	Version	Recommended action	More details and links
Applications	ArgoCD cluster integration: **spec.applicationManager.argocdCluster**	2.3	Create a GitOps cluster and placement custom resource to register managed clusters.	Configuring GitOps on managed clusters
Governance	cert-manager internal certificate management	2.3	No action is required	None

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

1.5.1. 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.

1.5.2. Table of Contents

- Data Deletion
- Capability for Restricting Use of Personal Data
- Appendix

1.5.3. GDPR

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

1.5.3.1. 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

1.5.3.2. Read more about GDPR

- EU GDPR Information Portal
- Red Hat GDPR website

1.5.4. 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.

1.5.5. 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.

1.5.5.1. 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.

1.5.5.2. 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 .

1.5.6. 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.

1.5.7. 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 ETCD 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.

1.5.8. 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.

1.5.8.1. 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.

1.5.8.2. 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.

1.5.8.3. 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.

1.5.8.4. Pod Security

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

1.5.9. 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.

1.5.10. 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.

1.5.11. 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.

1.5.12. 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.