Red Hat Advanced Cluster Management for Kubernetes 2.4

Release notes

Last Updated: 2022-01-27

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
- FIPS 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.

1.1.1.1. Observability

- Add recording rules in the observability service to calculate the Kubernetes API server service-level indicator (SLI) and service-level objective (SLO). For more details, see Creating custom rules.
- You can now view an aggregate of clusters within a fleet for the service-level overview for the Kubernetes API server from the Grafana dashboard. See Viewing the cluster fleet service-level overview for the Kubernetes API server dashboard.
- View the error budget and time remaining within a seven or 30-day period from the Grafana dashboard, where the service can utilize downtime in the cluster service-level overview for the Kubernetes API server. See Viewing the cluster service-level overview for the Kubernetes API server dashboard.
- Use Red Hat Advanced Cluster Management with Grafana 8.1.3 to view metrics. See Observing environments for more information.
- View the cluster service-level overview for the Kubernetes API server from the Grafana dashboard. See Viewing the cluster service-level overview for the Kubernetes API server dashboard.

1.1.2. Clusters

- You can now host OpenShift Container Platform hub clusters on the following platforms, as well as manage clusters that are provisioned on these platforms:
  - IBM Z and LinuxONE
  - IBM Power systems
- Create and manage a Microsoft Azure Government cluster using the Red Hat Advanced Cluster Management console. See Creating a credential for Microsoft Azure for more information.
- Create infrastructure environments to manage your on-premises clusters. See Creating an infrastructure environment for more information.
- Create on-premises clusters in an infrastructure environment with the Red Hat Advanced Cluster Management console. See Creating a cluster in an on-premises environment for more information.
- Select your managed clusters based on the status of the add-ons. See Add-on status for more information.
- You can now configure the proxy information for your cluster with the Red Hat Advanced Cluster Management console when you create a cluster. See the creation topic of the cluster that you are creating in Creating a cluster for more information.

Technology Preview:

- Use the cluster backup and restore operator to schedule back up and restore for your cluster resources. See Cluster backup and restore operator (Technology Preview) for more information.
- Deploy multiple SNO clusters using zero touch provisioning. See Deploying distributed units at scale in a disconnected environment in the OpenShift Container Platform documentation for more information.

- Replicate persistent volume using VolSync, which enables you to create copies of the data on a persistent volume. See Replicating persistent volumes with VolSync for more information.

1.1.3. Applications

Choose a type of application to create from the Application console **Create application** drop-down menu. You have the option to create a Git repository, Helm repository, or Object Storage repository Subscription. Subscriptions are Kubernetes resources within channels (source repositories).

Technology Preview: You can now also choose to create an Argo CD ApplicationSet from the same drop-down menu so that you can manage Argo CD Applications across a larger amount of clusters.

With the YAML Editor on, you can see the file update as you create or edit your applications as you modify the fields.

You can specify the secondary channel when you create an application, or after. When you create the secondary channel, the application uses the secondary when the primary fails.

As a subscription administrator, you can create an allow and deny list. In the same role, you can also now deploy all application resources into the subscription namespace. See Application advanced configuration for more information about subscription administrator tasks and other advanced configuration topics.

For other Application topics, see Managing applications.

1.1.4. Governance

- You can use new columns that are displayed from the *Governance* page. Use the *Source* column from the console to identify policies that are deployed using GitOps. Use the *Status* column to verify the enablement of a policy. For more information, see Manage security policies.
- You can now use Red Hat Advanced Cluster Management clusters that are FIPS ready. For more details, see FIPS readiness.
- Use the integration of Red Hat Insights and governance to send alerts for governance violations. You can also identify which component sent the violation. See Managing insight PolicyReports for more information.
- Customize your dashboard with new filtering options that support bulk actions. See Customize the Governance page for more information.
- Use the **policy_governance_info** metric to view trends and analyze any policy failures. See Governance metric for more details.

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 2.4, this list includes two ciphers that are *not* FIPS approved. If you upgrade from a version 2.4.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 2.4.1

- Filters out all of the policies that are sourced by governance and risk from the **PolicyReport** data shown on the managed cluster details view, as the count for those policies is displayed elsewhere on that page. (GitHub #17438)
- Removes the optional sample **imagepullsecret** parameter in the Operatorhub Installation console to prevent accidental use of a pull secret that does not exist on their cluster. (GitHub #17884)
- Fixes an issue that caused a hibernating cluster in a cluster pool to hang in **Detaching** status when you destroy the cluster and the managed cluster status does not become **Unknown**. After upgrading to ACM 2.4.1, all clusters that were hung in a **Detaching** status, along with their namespaces, automatically terminate and resume expected behavior. No manual intervention should be required to clean up these clusters or their namespaces on the hub cluster. (GitHub #18249)
- Removes the required step to enable the feature gate to add a central infrastructure management (CIM) environment.
- Fixes an issue that prevented the validation errors caused by a failed **ClusterDeployment** from surfacing in the **ClusterPool** console. (Bugzilla #1995380)
- Delivers updates to one or more of the product container 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 .

- Documentation known issues
- Installation known issues
- Web console known issues
  - Observability known issues
- Cluster management known issues
- Application management known issues
- Governance known issues
- Backup and restore known issues

1.3.1. Documentation known issues

1.3.1.1. Documentation links in the Customer Portal might link to a higher-level section

In some cases, the internal links to other sections of the Red Hat Advanced Cluster Management documentation in the Customer Portal do not link directly to the named section. In some instances, the links resolve to the highest-level section.

If this happens, you can either find the specified section manually or complete the following steps to resolve:

1. Copy the link that is not resolving to the correct section and paste it in your browser address bar. For example, it might be: **https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_management_for_kubernetes/2.4/html/clusters/index#volsync**.

2. In the link, replace **html** with **html-single**. The new URL should read: **https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_management_for_kubernetes/2.4/html-single/clusters/index#volsync**

3. Link to the new URL to find the specified section in the documentation.

1.3.2. Installation known issues

1.3.2.1. Upgrade from 2.2.x to 2.3.4 might cause klusterlet deletion

After you upgrade from 2.2.x to 2.3.4, the Klusterlet might be deleted. See the following procedure to work around this issue:

1. Upgrade the work-agent to 2.3.3.

   a. Create a JSON file **work-image-override.json** with the following JSON content:

   ```
   [
     {
       "image-name": "work-rhel8",
       "image-remote": "registry.redhat.io/rhacm2",
       "image-digest":
   "sha256:b6606f6bb6504acfb48f13cd5296473c17088caf380097ff7ce316f781c4f196",
       "image-key": "work"
     }
   ]
   ```

   b. Create a ConfigMap for image override on the hub cluster:

   ```
   kubectl -n open-cluster-management create configmap work-image-override --from-file=./work-image-override.json
   ```

   c. Enable the image override by annotating **mch** on the hub cluster.

   ```
   kubectl -n open-cluster-management annotate mch multiclusterhub --overwrite mch-imageOverridesCM=work-image-override
   ```

   d. Restart **multiclusterhub-operator** to enforce the change.

   ```
   kubectl -n open-cluster-management delete pod multiclusterhub-operator-xxxxx-xxxxx
   ```

- Wait for about 30 minutes to make sure that work-agent running on all managed clusters have been restarted with the overridden image.

2. Upgrade from 2.2.x to 2.3.4.
3. Disable the image override for work-agent.
4. After the upgrade is done, it's safe to remove the image override for work-agent.

```
kubectl -n open-cluster-management annotate mch multiclusterhub mch-imageOverridesCM-
--overwrite
kubectl -n open-cluster-management delete configmap work-image-override
```

1.3.2.2. 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.

1.3.2.3. 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.

1.3.3. Web console known issues

1.3.3.1. 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.

1.3.3.2. 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.

1.3.3.3. 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.

1.3.3.4. 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
```

1.3.4. Observability known issues

1.3.4.1. Duplicate local-clusters on Service-level Overview dashboard

When various hub clusters deploy Red Hat Advanced Cluster Management observability using the same S3 storage, *duplicate* **local-clusters** can be detected and displayed within the *Kubernetes/Service-Level Overview/API Server* dashboard. The duplicate clusters affect the results within the following panels: *Top Clusters*, *Number of clusters that has exceeded the SLO* , and *Number of clusters that are meeting the SLO*. The **local-clusters** are unique clusters associated with the shared S3 storage. To prevent multiple **local-clusters** from displaying within the dashboard, it is recommended for each unique hub cluster to deploy observability with a S3 bucket specifically for the hub cluster.

1.3.4.2. 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.

1.3.4.3. 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)**

1.3.4.4. 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.

1.3.4.5. High CPU usage by the search-collector pod

When search is disabled on a hub cluster that manages 1000 clusters, the **search-collector** pod crashes due to the out-of-memory error (OOM). Complete the following steps:

1. If search is disabled on the hub cluster, which means the **search-redisgraph-pod** is not deployed, reduce memory usage by scaling down the **search-collector** deployment to **0** replicas.

2. If search is enabled on the hub cluster, which means the **search-redisgraph-pod** is deployed, increase the allocated memory by editing the **search-collector** deployment.

1.3.4.6. 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.

1.3.4.7. 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.

1.3.4.8. Prometheus data loss on managed clusters

By default, Prometheus on OpenShift uses ephemeral storage. Prometheus loses all metrics data whenever it is restarted.

When observability is enabled or disabled on OpenShift Container Platform managed clusters that are managed by Red Hat Advanced Cluster Management, the observability endpoint operator updates the **cluster-monitoring-config ConfigMap** by adding additional alertmanager configuration that restarts the local Prometheus automatically.

1.3.5. Cluster management known issues

1.3.5.1. The local-cluster might not be automatically recreated

If the local-cluster is deleted while **disableHubSelfManagement** is set to **false**, the local-cluster is recreated by the **MulticlusterHub** operator. After you detach a local-cluster, the local-cluster might not be automatically recreated.

- To resolve this issue, modify a resource that is watched by the **MulticlusterHub** operator. See the following example:

  ```
  oc delete deployment multiclusterhub-repo -n <namespace>
  ```

- To properly detach the local-cluster, set the **disableHubSelfManagement** to true in the **MultiClusterHub**.

1.3.5.2. Selecting a subnet is required when creating an on-premises cluster

When you create an on-premises cluster using the Red Hat Advanced Cluster Management console, you must select an available subnet for your cluster. It is not marked as a required field.

1.3.5.3. 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.

1.3.5.4. 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
- 4.9.6 and earlier

To avoid this error, upgrade your OpenShift Container Platform to version 4.8.18 or later, or 4.9.7 or later.

1.3.5.5. Cannot hibernate an Azure Government cluster

When you try to hibernate an Azure Government cluster, the hibernation fails with the following error that is added to the provision pod log:

```
Confidential Client is not supported in Cross Cloud request
```

1.3.5.6. 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.

1.3.5.7. Cluster provision with Ansible automation fails in proxy environment

An AnsibleJob template that is configured to automatically provision a managed cluster might fail when both of the following conditions are met:

- The hub cluster has cluster-wide proxy enabled.
- The Ansible Tower can only be reached through the proxy.

1.3.5.8. Version of the klusterlet operator must be the same as the hub cluster

If you import a managed cluster by installing the klusterlet operator, the version of the klusterlet operator must be the same as the version of the hub cluster or the klusterlet operator will not work.

1.3.5.9. 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.

1.3.5.10. 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.

1.3.5.11. 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.

1.3.5.12. 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.

1.3.5.13. 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.

1.3.5.14. 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
```

1.3.5.15. 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}"} }]'
```

1.3.5.16. 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.

1.3.5.17. 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.

1.3.5.18. 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.

1.3.5.19. 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.

1.3.5.20. 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** or **s390x** 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 steps similar to the following example for your architecture type:

1. From the OpenShift Container Platform release registry, create a manifest list that includes **x86_64**, **s390x** and **ppc64le** release images.

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

   ```
   $ podman pull quay.io/openshift-release-dev/ocp-release:4.9.1-x86_64
   $ podman pull quay.io/openshift-release-dev/ocp-release:4.9.1-ppc64le
   $ podman pull quay.io/openshift-release-dev/ocp-release:4.9.1-s390x
   ```

   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 that apply to your environment:

   ```
   $ podman push quay.io/openshift-release-dev/ocp-release:4.9.1-x86_64 <private-repo>/ocp-release:4.9.1-x86_64
   $ podman push quay.io/openshift-release-dev/ocp-release:4.9.1-ppc64le <private-repo>/ocp-release:4.9.1-ppc64le
   $ podman push quay.io/openshift-release-dev/ocp-release:4.9.1-s390x <private-repo>/ocp-release:4.9.1-s390x
   ```

   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:4.9.1-x86_64
   $ podman manifest add mymanifest <private-repo>/ocp-release:4.9.1-ppc64le
   $ podman manifest add mymanifest <private-repo>/ocp-release:4.9.1-s390x
   ```

   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:4.9.1
   ```

   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: img4.9.1-appsub
spec:
  releaseImage: <private-repo>/ocp-release:4.9.1
```

   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.

1.3.5.21. Argo CD is not supported with IBM Power or IBM Z system 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 or IBM Z systems because there are no available **ppc64le** or **s390x** images.

1.3.5.22. Cannot use Ansible Tower integration with an IBM Power or IBM Z system 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 or IBM Z systems because the Ansible Automation Platform Resource Operator does not provide **ppc64le** or **s390x** images.

1.3.5.23. 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.

1.3.6. Application management known issues

1.3.6.1. Policy resource not deployed unless by subscription administrator

The **policy.open-cluster-management.io/v1** resources are no longer deployed by an application subscription by default for Red Hat Advanced Cluster Management version 2.4.

A subscription administrator needs to deploy the application subscription to change this default behavior.

See Creating an allow and deny list as subscription administrator for information. **policy.open-cluster-management.io/v1** resources that were deployed by existing application subscriptions in previous Red Hat Advanced Cluster Management versions remain, but are no longer reconciled with the source repository unless the application subscriptions are deployed by a subscription administrator.

1.3.6.2. 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.

1.3.6.3. 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.

1.3.6.4. 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.

1.3.6.5. 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.

1.3.6.6. 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.

1.3.6.7. 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
```

1.3.6.8. 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.

1.3.6.9. Ansible Automation Platform (early access) job fail

When the Ansible Automation Platform (early access) is installed, **AnsibleJobs** fails to run. To submit prehook and posthook **AnsibleJobs** through Red Hat Advanced Cluster Management, use the **early-access-cluster-scoped** option. The option is available in Ansible Automation Platform (early access) version **2.0.1+0.1635279521** and later.

1.3.6.10. Ansible Automation Platform operator access Ansible Tower outside of a proxy

The Ansible Automation Platform (AAP) operator cannot access Ansible Tower outside of a proxy-enabled OpenShift Container Platform cluster. To resolve, you can install the Ansible tower within the proxy. See install steps that are provided by Ansible Tower.

1.3.6.11. Template information does not show when editing a Helm Argo application in version 2.4

When a Helm Argo application is created and then edited, the template information appears empty while the YAML file is correct. Upgrade to Errata 2.4.1 to fix the error.

1.3.6.12. 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'
```

1.3.6.13. 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.

1.3.7. Governance known issues

1.3.7.1. 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.

1.3.7.2. Unable to log out from Red Hat Advanced Cluster Management

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.

1.3.7.3. Placement resource limitations

As a reminder, a policy must use either a **PlacementRule** or a **Placement** resource to control the deployment of policies to specific managed clusters. If you create policies that use the **Placement** resource, you might encounter the following limitations:

- The placement information is not shown when you view the details of the policy from the console.
- The placement information is not removed when the policy is deleted from the console.
- When you edit the policy from the console, the placement details are not updated.

Use the command line interface (CLI) to make updates to the policies when you use the **Placement** resource.

1.3.7.4. 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.

1.3.8. Backup and restore known issues

1.3.8.1. Backup and restore feature does not work on IBM Power and IBM Z

The backup and restore feature for the hub cluster requires the OpenShift API for Data Protection (OADP) operator. The OADP operator is not available on the IBM Power or IBM Z architectures.

1.3.8.2. Application and policy show no resource status on managed cluster after a restore operation

When a restore operation is run on a new hub cluster, using data backed up from another hub cluster, the application and policy shows no status for resources on managed clusters. This happens because the search and policy add-ons are not reset to point to the new hub cluster.

From the new hub cluster, you must restart the **addon-certpolicyctrl** and **addon-search** for all managed clusters. Run the following commands to restart the pods:

```
oc get pods -n open-cluster-management-agent-addon | grep search | awk '{print $1}' | xargs kubectl delete pod
```

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.

1.4.1.1. API deprecations

Product or category	Affected item	Version	Recommended action	More details and links
Applications	The **v1alpha1** API is removed completely. GitOps clusters API is upgraded to **V1beta1**.	2.4	Use **V1beta1**.	None

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	Managing secrets	2.4	Use policy hub templates for secrets instead.	See Manage security policies.
Governance and risk console	**pod-security-policy**	2.4	None	None

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 and risk	Custom policy controller	2.3	None	None
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

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
Red Hat Advanced Cluster Management console	Visual Web Terminal (Technology Preview)	2.4	Use the terminal instead	None

Product or category	Affected item	Version	Recommended action	More details and links
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
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
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

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

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.

1.6. FIPS READINESS

FIPS readiness has been completed for Red Hat Advanced Cluster Management for Kubernetes. Red Hat Advanced Cluster Management uses the same tools to make sure cryptography calls are passed to the Red Hat Enterprise Linux (RHEL) certified cryptographic modules that are used by Red Hat OpenShift Container Platform. For more details on OpenShift FIPS support see, Support for FIPS cryptography.

1.6.1. Limitations

Read the following limitations with Red Hat Advanced Cluster Management and FIPS.

- Red Hat OpenShift Container Platform does not support FIPS on the IBM Power (**ppc64le**) and IBM Z (**s390x**) architectures.
- The following Technology preview components are not FIPS ready:
  - Hub cluster backup and restore
  - Infrastructure Operator for Red Hat OpenShift
  - Submariner
  - VolSync
- Persistent Volume Claim (PVC) and S3 storage that is used by the search and observability components must be encrypted when you configure the provided storage. Red Hat Advanced Cluster Management does not provide storage encryption, see the OpenShift Container Platform documentation, Support for FIPS cryptography .
- When you provision managed clusters from Red Hat Advanced Cluster Management, you must set **fips: true** in the **install-config.yaml** file before you deploy the new managed cluster.