Red Hat Advanced Cluster Management for Kubernetes 2.1

Release notes

Red Hat Advanced Cluster Management for Kubernetes Release notes

Last Updated: 2022-01-27

Red Hat Advanced Cluster Management for Kubernetes 2.1 Release notes

Red Hat Advanced Cluster Management for Kubernetes Release notes

Legal Notice

Abstract

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

Table of Contents

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

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.
- Upgrading to the most recent version of Red Hat Advanced Cluster Management is best practice.
  - What's new in Red Hat Advanced Cluster Management for Kubernetes
  - Errata updates
  - Known issues and limitations
  - Red Hat Advanced Cluster Management for Kubernetes considerations for GDPR readiness

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

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

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

1.1.1. Installation

- You can now manage your hub cluster. When you install Red Hat Advanced Cluster Management, the hub cluster is automatically imported and managed. See Installing while connected online for more information.

1.1.2. Web console

- Use the web console to access, view, and manage your cluster from a central view. You can access your Red Hat Advanced Cluster Management console from Red Hat OpenShift Container Platform, monitor cluster data and details, use the search component across your cluster and with Visual Web Terminal, and manage cluster labels. To learn more details about the console components, see Web console.
- You can now enable the multicluster observability service (**multicluster-observability-operator**) to view the health and optimization of your managed clusters. You can explore the collected metric data and logs from your managed clusters. For more information see, Observing environments.

1.1.3. Cluster management

- You can now create and manage clusters in a bare metal environment. See Creating a cluster on bare metal for more information.
- Additionally, you can create and manage Red Hat OpenShift Container Platform clusters on the VMware vSphere provider with Red Hat Advanced Cluster Management. See Creating a cluster on VMware vSphere for more information.
- You can now group clusters and give user access to the group by creating ManagedClusterSet resources. For more information, see ManagedClusterSets.
- You can integrate your Red Hat Advanced Cluster Management with the Red Hat OpenShift Update Service operator to upgrade your managed clusters in a disconnected environment. See Upgrading disconnected clusters for more information.

1.1.4. Application management

- Red Hat Advanced Cluster Management for Kubernetes application management improved usability with console configuration for managing resources. You can now create an application with supported channels in the console, create and edit an application, configure secret settings, and more. See Managing application resources.
- From **Advanced configuration**, you can view choose *Subscriptions*, *Placement rules*, or *Channels* to view your resources in a table. From the table, you can also edit these resources as YAML.
- Red Hat Advanced Cluster Management for Kubernetes Ansible Tower integration is available as technology preview so that you can deploy and manage Ansible jobs from the console. You can also view the job status in the *Resource topology*. See Application console.
- As part of application management, you can integrate Ansible Tower jobs into Git subscriptions. Automate tasks and integrate with external services, such as Slack and PagerDuty services. To learn more about working with Ansible, see Setting up Ansible Tower tasks (Technology preview).

For all the application management changes and improved documentation, see Managing applications.

1.1.5. Security and compliance

- Red Hat Advanced Cluster Management for Kubernetes supports several roles and uses Kubernetes authorization mechanisms. For more information, see Role-based access control.
- For the certificate policy controller, you can now use the **disallowedSANPattern** parameter to check DNS names against patterns. For more information, view the Certificate policy controller YAML table.
- You can now contribute to the open source community, **open-cluster-management/policy-collection**, by adding policies with the product governance framework. You can integrate third-party policies, such as Gatekeeper. For more information, see Integrate third-party policy controllers.
- You can now use the configuration policy controller to create an ETCD encryption policy. Use the ETCD encryption policy to enable encryption of sensitive data. For more information, see ETCD encryption policy .
- You can now create policies for the self-managed hub cluster (local hub cluster) by selecting **local-cluster** as the cluster binding. For more information, see Creating a security policy .

- You can now view the violation history of your policy from the *Status* tab. For more information, see Manage security policies .

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

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.

1.2.1. Errata 2.1.13

This errata release delivers updates to one or more of the product container images.

1.2.2. Errata 2.1.12

This errata release delivers updates to one or more of the product container images.

1.2.3. Errata 2.1.11

This errata release delivers updates to one or more of the product container images.

1.2.4. Errata 2.1.10

This errata release delivers updates to one or more of the product container images.

1.2.5. Errata 2.1.9

Updated selected containers in the images.

1.2.6. Errata 2.1.8

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.1.8 updates:

Important: You must run Red Hat OpenShift Container Platform version 4.6, or later, to upgrade to Errata 2.1.7 and later. If you cannot upgrade your Red Hat OpenShift Container Platform version 4.5 to a later version, you can continue to use Red Hat Advanced Cluster Management version 2.1.6.

- Resolves the issue with Observability **thanos-store-shard** pods in **crashloopback** state after upgrade to Red Hat OpenShift Container Platform 4.6.30. (GitHub 13081)
- Fixes an issue with **placementrule** and **placementbinding** not deleted when user deletes a disabled policy in the console. (GitHub 12689)
- Updated Search code to use data from other fields as a result of Kubernetes selfLink removal, which impacted Search logic that depended on those fields. (GitHub 12701)

1.2.7. Errata 2.1.7

Important: You must run Red Hat OpenShift Container Platform version 4.6, or later, to upgrade to Red

Hat Advanced Cluster Management version 2.1.7. If you cannot upgrade your Red Hat OpenShift Container Platform version 4.5 to a later version, you can continue to use Red Hat Advanced Cluster Management version 2.1.6.

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.1.7 updates:

- Fixed an issue that caused the Hive controller log to show the incorrect version information. (GitHub 12014)
- Added authorization for users with **view** permissions to create and delete the **ManagedClusterView** resource, which also enables users with **view** permissions to view the YAML file of the managed cluster resources. (GitHub 11243)
- Enabled users with a **cluster-manager-admin** role binding to run create, update, and delete operations on the **clusterimagesets** resource. This change allows a user with **cluster-manager-admin** privileges to provision clusters with Red Hat Advanced Cluster Management. (GitHub 11596)

1.2.8. Errata 2.1.6

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.1.6 updates:

- Updated the list of available Red Hat OpenShift Container Platform release ClusterImageSets when creating a new cluster. (GitHub 10760)
- Added quotation marks to the generated import command to avoid possible errors when the command is run. (Bugzilla 1934184)(GitHub 9983)

1.2.9. Errata 2.1.5

Note: OpenShift Container Platform version 4.7 is not supported on bare metal. You cannot create bare metal managed clusters with the Red Hat Advanced Cluster Management hub cluster when the hub cluster is hosted on OpenShift Container Platform version 4.7.

- Fixed log errors that occurred when **packageOverrides** was incorrectly specified in a subscription CR. Errors are now logged correctly and the incorrect **packageOverrides** specifications are ignored. (GitHub 10008)
- Updated the list of Azure regions that are available for adding a cluster. Bugzilla 1932430
- Fixed an issue that caused the *Application topology* page to display an unexpected error. (GitHub 9377)
- Fixed an issue with hub cluster subscription crashing when a Helm subscription is used to subscribe resources from a private Helm channel with only **spec.SecretRef** defined. Now the hub cluster subscription does not crash for this type of Helm subscription. The private Helm repository channel secret must be defined in the same channel namespace. Bugzilla 1932430
- Fixed duplicate Ansible prehook and posthook jobs that were created. Now only one Ansible prehook and posthook job is created and executed by application subscriptions. (Bugzilla 1920654)
- Updates to the *Overview* page to include resources from the hub cluster (local-cluster). (Bugzilla 1903446)

1.2.10. Errata 2.1.4

Updated selected containers in the images.

1.2.11. Errata 2.1.3

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.1.3 updates:

- Fixed panic error in **multicluster-operators-hub** pod so that **appsub** deploys successfully (Bugzilla 1921531).
- Fixed an issue with managed clusters that were created on VMware and did not use the values that were provided for worker pool CPU, memory, or disk size. (GitHub 8930)
- Fixed certificate policy controller not detecting created or deleted namespaces that match the selector in the policy. (GitHub 7639)
- Fixed Grafana ClusterRoleBinding object failure. (GitHub 7621)
- Fixed configuration policy controller crash when handling policies. (GitHub 7569)
- Fixed missing namespace when editing an existing provider connection. (GitHub 7501)
- Fixed routing issues in policy pages to show **No resource** instead of a loading animation when users navigate to a URL for a policy that does not exist. (GitHub 7445)
- Fixed issue with policy editor crashing when content was copied, pasted, and fixed an error in **.spec.policyTemplate** when the form did not update to display that there was a custom specification. (GitHub 7380)
- Added channel connection failure message, which can be found in the subscription status. (GitHub 7177)
- Excluded channels from removable application resources, which were listed in *Delete application* modal in the console. Now channels cannot be deleted in this modal. Only subscriptions and placement rules are removable in this modal with this fix. (GitHub 7153)
- Fixed display of NIST categories, standards, and controls for consistency across all policy elements and adjusted to NIST content. (GitHub 6954)
- Increased the search pod memory requests and limits in default install to handle most workloads without intervention: Memory limit for Search **redisgraph** pod to 4GB, memory request for Search API and Redisgraph pods to 128MB. (GitHub 6890)
- Fixed failure with Git channel connection to a private Git repository with missing **secretRef** leading to **multicluster-operators-hub-subscription** pod crash. (GitHub 8764)
- Fixed **cert-manager-webhook** failure to start because of permission problems with OpenShift Container Platform 4.6.10 installation. (GitHub 8517)
- Fixed a high availability configuration running too many competing compactors so that with the fix, there is only one compactor running. (GitHub 7676)
- Fixed a potential performance issue where some Grafana dashboards auto-refresh with an interval that was smaller than the metrics scrape interval. (GitHub 7665)
- Added support to import Red Hat OpenShift on IBM Cloud cluster to manage. (Bugzilla 1894778)

- Fixed Git webhook notification function to deploy the selected Git repository resources to target clusters through a subscription. (GitHub 6785)

- Fixed an issue with Application topology resources that deploy successfully but are not accessible offline. Now the cluster node displays a failed status if any remote clusters are offline. (GitHub 6298)

1.2.12. Errata 2.1.2

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.1.2 updates:

- Fixed an issue that caused the hub cluster to deny renewal requests for a registration agent certificate, which resulted in some registration agents going offline after one month. (GitHub 5628)

- Fixed an issue that caused a conflict between some cluster image sets when Red Hat Advanced Cluster Management was upgraded. (GitHub 7527)

- Fixed an issue that caused some certificates to be removed during upgrading. (GitHub 7533)

1.2.13. Errata 2.1.1

View a summarized list of Red Hat Advanced Cluster Management Errata 2.1.1 updates:

- Updated the **certificate** and **iam** policy controllers to fix an issue that prevented them from correctly maintaining the history of policy violations. (GitHub 6014)

- Increased VMware managed cluster default worker node values (4 CPU, 2 core, 16384 MB memory) to align with other providers. (GitHub 6206)

- Fixed an issue that caused a temporary error on the create resources page after you detach a managed cluster. (GitHub 6299)

- Fixed an issue in which the **Merge updates** option changed to **unset** after closing, modifying, and reopening an application. (GitHub 6349)

- Fixed an issue that prevented the complete clean up of a Microsoft Azure managed cluster after the addition of the cluster failed. (GitHub 6353)

- Fixed an issue that prevented the application topology from displaying the correct resource nodes after it deployed a **helm** type application to **local-cluster**. The application topology now displays all types of applications. (GitHub 6400)

- Application subscriptions: Enabled the **packageOverrides** YAML content for the Git **kustomization.yaml** file to use the path that is identified in the annotation of the subscription by default. (GitHub 6476)

- Fixed an issue that prevented subscription overrides from working when multiple subscriptions shared the same Git channel with the same branch. (GitHub 6476)

- Fixed an issue where policies using the **musthave** compliance type on a list of objects behaved similarly to the **mustonlyhave** compliance type. You can now specify as few as one field in a list of objects, and a **musthave** policy marks it as compliant as long as one object in the list has a field that matches the one specified in the policy. (GitHub 6492)

- Resolved an issue that configures all Thanos receivers so that every time-series stores 3 copies. It also ensures that every time-series is successfully written to at least 2 Thanos receivers in the target hashring. (GitHub 6547)
- Fixed an issue that caused the **merge update** setting to clear after selecting the setting when creating the application with the **Create** wizard , then opening it in an editor. (GitHub 6554)
- Fixed an issue that caused policies to display a **noncompliant** status. (GitHub 6630)
- Fixed an issue that occurred when the Git webhook was enabled on channel and subscription, but the subscribed resources were not applied to the target clusters. (GitHub 6785)
- Resolved an issue that can cause the **create resource** command to fail with a **Forbidden** error on the first load. (GitHub 6798)
- Exposed the following additional metrics with the Red Hat Advanced Cluster Management observability components for persistent volumes:
  - **kubelet_volume_stats_available_bytes**
  - **kubelet_volume_stats_capacity_bytes**
  - **kube_persistentvolume_status_phase**
    These metrics are not explicitly exposed in any dashboards or alert rules, but you can query them and set custom alert rules for them. (GitHub 6891)
- Fixed selection and deleselection inconsistencies when creating a new Policy. (GitHub 6897)
- Fixed an issue that caused bare metal clusters to fail to upgrade to 2.1.0 due to memory errors. (GitHub 6898) (Bugzilla 1895799)
- Fixed an issue that required a pull secret in the **open-cluster-management-observability** namespace to successfully install the observability components. With this change, you are not required to create a pull secret to install the observability components. (GitHub 6911)
- Fixed an issue that caused the Governance and risk dashboard to take a long time to load. (GitHub 6925)
- Corrected a PATH error when starting a new Visual Web Terminal session. (GitHub 6928)
- Fixed a possible timing issue of the observability components in managed clusters changing to use incorrect images when the observability operator is restarted at runtime. (GitHub 6942)
- Added instructions for applying a fix to work around a failed application creation from a private Git repository. (GitHub 6952) (Bugzilla 1896341)
- Fixed an issue that prevented the **klusterlet-addon-controller** from being recognized when it is in a namespace other than the **open-cluster-management** namespace. (GitHub 6986)
- Fixed an issue that caused the configuration policy controller to crash when an object template checked a field for a list, but found something set to that field that is not the expected list. (GitHub 7135)
- Fixed an issue in which the template editor YAML filters out the placementRule **status: 'True'** setting when making changes to an application deployed on all online clusters.
  If you manually enter **status: 'True'** in the YAML editor for the placementRule before saving the updated application, the setting is retained. (GitHub 7152)

- Completed other general changes and bug fixes to code and documentation that are not listed.

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 .

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

1.3.1. Upgrade known issues

1.3.1.1. Upgrade from version 2.1.x to 2.3.2 degraded due to the Observability add-on

After you upgrade from 2.1.x to 2.3.2, some clusters might become degraded because the Observability add-on is not ready, or the image manifest ConfigMap is not read correctly during upgrade, which leads to an incorrect image.

To fix this issue, restart the **multicluster-observability-operator** pod by running the following command:

oc delete pod multicluster-observability-operator -n open-cluster-management

1.3.1.2. Upgrade to 2.1.x results in loss of certificates

After upgrading Red Hat Advanced Cluster Management from version 2.0 to version 2.1, the setting that specifies where an application is deployed is not pre-selected when you open the application template editor to make changes. If you make changes to the application settings in the application template editor, you must select application deployment setting before saving and closing the editor.

1.3.1.3. Upgrade to 2.1.1 results in loss of certificates

When your cluster upgrades to Red Hat Advanced Cluster Management version 2.1.1, you lose some or all of the certificates on your cluster. You can confirm this situation by entering one of the following commands:

```
oc get certificates -n open-cluster-management
```

or

```
oc get pods -n open-cluster-management | grep -vE "Completed|Running"
```

If there are fewer certificates returned than expected when you run the first command, or more than one pod is returned after running the second command, run the **generate-update-issue-cert-manifest.sh** script to update the certificates.

1.3.1.4. Upgrade to version 2.1.1 does not complete with ClusterImageSet error

In some cases, your upgrade of Red Hat Advanced Cluster Management for Kubernetes version 2.1.0 to Red Hat Advanced Cluster Management version 2.1.1 does not complete and displays an error that is similar to the following error:

```
failed to get candidate release: rendered manifests contain a resource
that already exists. Unable to continue with update: ClusterImageSet "img4.6.1-x86-64"
in namespace "" exists and cannot be imported into the current release: invalid
ownership metadata; label validation error: missing key "app.kubernetes.io/managed-by":
must be set to "Helm"; annotation validation error: missing key "meta.helm.sh/release-name":
must be set to "console-chart-c4cb5"; annotation validation error: missing key
"meta.helm.sh/release-namespace": must be set to "open-cluster-management"
```

This occurs when one or more ClusterImageSets on the existing version have the same names as the versions that are added with the upgrade, which causes a collision. To work around this issue, complete the following steps:

1. Stop the running upgrade.
2. Delete the ClusterImageSet or ClusterImageSets from your local environment that are identified in the error message.
3. Restart the upgrade.

1.3.1.5. Upgrade to 2.1.1 disables klusterletaddonconfig CRDs

When your Red Hat Advanced Cluster Management upgrades from version 2.1.0 to version 2.1.1, the **klusterletaddonconfig** custom resource definitions (CRDs) might be reinstalled during the upgrade. If this happens, all of the add ons show a **Disabled** status on the *Cluster settings* pages. Complete the following steps to diagnose the problem and restore the klusterletaddonconfig CRDs:

1. Log on to the hub cluster by using the **oc login** command.
2. Run the following command to confirm that the deleted **klusterletaddonconfig** CRDs occurred because of the reinstalled CRDs:

   ```
   % oc get klusterletaddonconfig --all-namespaces
   ```

   If the returned content is **No resources found**, the reinstallation is likely the issue. Continue with step 3.

3. Save the following script to a file. For this example, the filename is **restore-addons.sh**:

   ```
   KUBECTL=oc
   ACM_NAMESPACE=open-cluster-management

   ACM_VERSION=$(${KUBECTL} get -n ${ACM_NAMESPACE} `${KUBECTL} get mch -
   oname -n ${ACM_NAMESPACE} | head -n1` -ojsonpath='{.status.desiredVersion}')
   if [ "${ACM_VERSION}" = ""  ]; then
   ACM_VERSION=2.1.1
   ```

```
fi

echo "ACM version: ${ACM_VERSION}"

for clusterName in `${KUBECTL} get managedcluster --ignore-not-found | grep -v "NAME" |
awk '{ print $1 }'`; do
    echo "Checking klusterletaddonconfig in ${clusterName} namespace."
    ${KUBECTL} get klusterletaddonconfig ${clusterName} -n ${clusterName} >/dev/null 2>&1
    if [ "$?" != "0"  ]; then
        echo "  klusterletaddonconfig in ${clusterName} is missing."
        echo "  Creating..."
        printf "  "
        cat <<EOF | ${KUBECTL}  apply -f -
apiVersion: agent.open-cluster-management.io/v1
kind: KlusterletAddonConfig
metadata:
  name: ${clusterName}
  namespace: ${clusterName}
spec:
  clusterLabels:
    cloud: auto-detect
    vendor: auto-detect
  clusterName: ${clusterName}
  clusterNamespace: ${clusterName}
  applicationManager:
    enabled: true
  certPolicyController:
    enabled: true
  iamPolicyController:
    enabled: true
  policyController:
    enabled: true
  searchCollector:
    enabled: true
  version: ${ACM_VERSION}
EOF

    fi
    echo "  Done."
done
```

Replace the value for **ACM_NAMESPACE** with the name of your namespace, if you did not install Red Hat Advanced Cluster Management in the **open-cluster-management** namespace.

4. Run the script from CLI. Your command should resemble the following command:

```
chmod +x restore-addons.sh && ./restore-addons.sh
```

Running the script recreates the missing **klusterletaddonconfig** CRDs in each managed cluster namespace.

1.3.1.6. OpenShift Container Platform cluster upgrade failed status

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

1.3.1.7. Upgrade from version 2.0.4 to version 2.1 leaves the ClusterServiceVersion in a pending state

After you upgrade from Red Hat Advanced Cluster Management version 2.0.4 to version 2.1 and run the **oc get csv** command. In the output, the **PHASE** of your Red Hat Advanced Cluster Management ClusterServiceVersion (CSV) is **Pending**, but the **NAME** is updated to **advanced-cluster-management.v2.1.0**.

To work around this issue, complete the following steps to find and create the missing **clusterRole** custom resource:

1. Find all **clusterrolebinding** resources that were deployed by the Red Hat Advanced Cluster Management 2.1 CSV by entering the following command:

   ```
   oc get clusterrolebinding |grep advanced-cluster-management
   ```

   Your output should resemble the following content:

   ```
   advanced-cluster-management.v2.1.0-86dfdf7c5d          ClusterRole/advanced-cluster-
   management.v2.1.0-86dfdf7c5d      9h
   advanced-cluster-management.v2.1.0-cd8d57f64           ClusterRole/advanced-cluster-
   management.v2.1.0-cd8d57f64       9h
   ```

2. Open each **clusterrolebinding** to find the **clusterRole** name that is associated to the **open-cluster-management** service account by entering a command that resembles the following command:

   ```
   oc get clusterrolebinding advanced-cluster-management.v2.1.0-cd8d57f64 -o yaml
   ```

   Your output should resemble the following content:

   ```
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRoleBinding
   metadata:
     name: advanced-cluster-management.v2.1.0-cd8d57f64
   roleRef:
     apiGroup: rbac.authorization.k8s.io
     kind: ClusterRole
     name: advanced-cluster-management.v2.1.0-cd8d57f64
   subjects:
   - kind: ServiceAccount
     name: multicluster-operators
     namespace: open-cluster-management
   ```

3. Manually create any missing **clusterRole** entries by adding content that resembles the following content to your **.yaml** file:

   ```
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRole
   metadata:
     name: advanced-cluster-management.v2.1.0-cd8d57f64
   rules:
   - apiGroups:
     - '*'
   ```

```
resources:
- '*'
verbs:
- '*'
```

1.3.2. Installation known issues

1.3.2.1. Certificate manager must not exist during an installation

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

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

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

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

1.3.3. 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 versions

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

1.3.3.4. 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.

1.3.3.5. 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.

1.3.3.6. 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

1.3.3.7. Observability not working due to *MultiClusterObservability* CR name

When you deploy **MultiClusterObservability** custom resource (CR) with a unique name, the metrics data is not collected. The metrics are not collected because the **metrics-collector** is not created. When you deploy observability, Red Hat Advanced Cluster Management supports using only the default name, **observability**, for the **MultiClusterObservability** CR.

1.3.4. Cluster management known issues

1.3.4.1. New bare metal asset options might not be displayed

After you create and save a bare metal asset, you can select the bare metal asset in the table and apply selected actions. With this issue, you might not see the available actions after selecting a new bare metal asset. Refresh your browser window to restore the actions at the beginning of the table.

1.3.4.2. Cannot create bare metal managed clusters on OpenShift Container Platform version 4.7

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 4.7.

1.3.4.3. Create resource dropdown 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.

1.3.4.4. 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.

1.3.4.5. Console might report managed cluster policy inconsistency

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

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

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

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

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

1.3.4.6. Importing clusters might require two attempts

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

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

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

1.3.4.8. 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
```

1.3.4.9. 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}"} }]'
```

1.3.4.10. Cannot run management ingress as non-root user

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

1.3.4.11. 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.

1.3.4.12. 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.

1.3.4.13. 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.

1.3.5. Application management known issues

1.3.5.1. Application deployment window error

When you create an application with a deployment window that is set to **Active within specified interval**, the deployment window might not be calculated correctly, resulting in the application being deployed in undefined times.

1.3.5.2. Resource topology status not deployed

If your Helm subscription does not have **packageAlias** defined, the resource Topology displays remote cluster resources as **Not deployed**.

See Configuring package overrides to define the appropriate **packageName** and the **packageAlias**.

1.3.5.3. 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. Uncheck 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'**

1.3.5.4. Merge updates option in the console is unselected when you edit your app

In the application console, when you edit your applcation, the **Merge updates** is unselected. You need to select the option again if it was previously selected and you still want to merge your updates.

To verify that merging updates was successful, ensure that the **reconcile-option: merge** is in the YAML subscription annotations. Complete the following steps in the console:

1. Click the **Subscription** node in the resource Topology diagram in the console.
2. Click the **View Resource YAML** button in the subscription details pop-up window.

3. Verify that the **apps.open-cluster-management.io/reconcile-option: merge** annotation is created on the subscription **.yaml** file.

1.3.5.5. Git branch and URL path fields not populated if a private Git URL exists

If you create an application with a *private* Git repo, and then click **Create application** to create another Git type, the former URL is not populated in the fields in the console.

The application Editor does not display the channel credential details in this case. If you change the repository authentication information for an existing channel repository, the product cannot manage existing applications that subscribe to that repository.

To resolve this issue, you can update the credential information on the channel resource, or you can delete and recreate the channel.

Use a YAML editor to update the channel resource with the newest credentials. See the sample section of link:../manage_applications#managing-apps-with-git-repositories[Managing apps with Git repositories].

1.3.5.6. Console pipeline cards might display different data

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

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

1.3.5.7. Namespace channel

Namespace channel might be functional in code but is currently not a documented option.

1.3.5.8. 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
```

1.3.5.9. Deployable resources in a namespace channel

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

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

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

-

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

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

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. Red Hat OpenShift Operator Lifecycle Manager default settings change the setting for the product. To workaround the issue, see the following procedure:

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

1.3.5.11. 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. Red Hat OpenShift Operator Lifecycle Manager default settings change the setting for the product. To workaround the issue, see the following procedure:

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

1.3.5.12. 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.

1.3.6. Security known issues

1.3.6.1. Internal error 500 during login to the console

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

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

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

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

```
oc edit multiclusterhub multiclusterhub
```

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

```
customCAConfigmap: custom-ca
```

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

1.3.6.2. Recovering *cert-manager* after removing the helm release

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

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

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

2. Verify that the helm release is recreated and the pods are running.

3. Make sure the certificate is generated by running the following command:

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

   You might receive the following response:

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

4. Update the other components with this certificate, by downloading and running **generate-update-issuer-cert-manifest.sh** script.

5. Verify that all of the secrets from **oc get certificates.certmanager.k8s.io** have the ready state **True**.

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

1.4.1. Notice

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

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

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

1.4.2. Table of Contents

1.4.3. GDPR

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

1.4.3.1. Why is GDPR important?

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

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors

- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

1.4.3.2. Read more about GDPR

- EU GDPR Information Portal
- Red Hat GDPR website

1.4.4. Product Configuration for GDPR

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

1.4.5. Data Life Cycle

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

As such, the Red Hat Advanced Cluster Management for Kubernetes platform deals primarily with technical data that is related to the configuration and management of the platform, some of which might be subject to GDPR. The Red Hat Advanced Cluster Management for Kubernetes platform also deals with information about users who manage the platform. This data will be described throughout this document for the awareness of clients responsible for meeting GDPR requirements.

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

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

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

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

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

1.4.5.2. Personal data used for online contact

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

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

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

1.4.6. Data Collection

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

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

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

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

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

1.4.7. Data storage

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

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

- **Platform Configuration Data:** The Red Hat Advanced Cluster Management for Kubernetes platform configuration can be customized by updating a configuration YAML file with properties for general settings, Kubernetes, logs, network, Docker, and other settings. This data is used as input to the Red Hat Advanced Cluster Management for Kubernetes platform installer for deploying one or more nodes. The properties also include an administrator user ID and password that are used for bootstrap.

- **Kubernetes Configuration Data:** Kubernetes cluster state data is stored in a distributed key-value store, **etcd**.

- **User Authentication Data, including User IDs and passwords:** User ID and password management are handled through a client enterprise LDAP directory. Users and groups that are defined in LDAP can be added to Red Hat Advanced Cluster Management for Kubernetes platform teams and assigned access roles. Red Hat Advanced Cluster Management for Kubernetes platform stores the email address and user ID from LDAP, but does not store the password. Red Hat Advanced Cluster Management for Kubernetes platform stores the group name and upon login, caches the available groups to which a user belongs. Group membership is not persisted in any long-term way. Securing user and group data at rest in the enterprise LDAP must be considered. Red Hat Advanced Cluster Management for Kubernetes platform also includes an authentication service, Open ID Connect (OIDC) that interacts with the enterprise directory and maintains access tokens. This service uses 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.

1.4.8. Data access

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

- Web user interface (the console)

- Kubernetes **kubectl** CLI

- Red Hat Advanced Cluster Management for Kubernetes CLI

- oc CLI

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

1.4.8.1. Authentication

The Red Hat Advanced Cluster Management for Kubernetes platform authentication manager accepts user credentials from the console and forwards the credentials to the backend OIDC provider, which validates the user credentials against the enterprise directory. The OIDC provider then returns an authentication cookie (**auth-cookie**) with the content of a JSON Web Token (**JWT**) to the

authentication manager. The JWT token persists information such as the user ID and email address, in addition to group membership at the time of the authentication request. This authentication cookie is then sent back to the console. The cookie is refreshed during the session. It is valid for 12 hours after you sign out of the console or close your web browser.

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

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

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

1.4.8.2. Role Mapping

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

1.4.8.3. Authorization

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

1.4.8.4. Pod Security

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

1.4.9. Data Processing

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

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

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

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

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

1.4.10. Data Deletion

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

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

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

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

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

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

1.4.11. Capability for Restricting Use of Personal Data

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

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

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

- Red Hat Advanced Cluster Management for Kubernetes platform administrators can use Red Hat Advanced Cluster Management for Kubernetes platform features to stop processing an individual's data.

1.4.12. Appendix

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

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