Red Hat Advanced Cluster Management for Kubernetes 2.2

Release notes

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

Last Updated: 2022-05-18

Red Hat Advanced Cluster Management for Kubernetes 2.2 Release notes

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

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. 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. Earlier versions of the documentation are also not supported.

- 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
  - Deprecations and removals
  - Red Hat Advanced Cluster Management for Kubernetes considerations for GDPR readiness

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

Red Hat Advanced Cluster Management for Kubernetes provides visibility of your entire Kubernetes domain with built-in governance, cluster lifecycle management, and application lifecycle management, along with observability. With this release, you can move towards managing clusters in more environments, GitOps integration for applications, and more. 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.

- 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 install your hub cluster in the Red Hat OpenShift Dedicated environment. For more information, see Installing while connected online .

1.1.2. Web console

You can now define the storage settings for search persistence. Persistence is enabled by default with the **searchCustomization** custom resource. For more information, see Search in the console.

1.1.2.1. Observing environments

- Certificates for observability are now automatically renewed before expiration is met. For more information, see Observability certificates.

- The following additional metrics are available with Red Hat Advanced Cluster Management and are included with telemetry, but are not displayed on the Red Hat Advanced Cluster Management Observe environments overview dashboard:
  - **visual_web_terminal_sessions_total**
  - **acm_managed_cluster_info**
- You can now add custom metrics to the observability service, which are collected from managed clusters. For more information, see Adding custom metrics.
- Observability automatically restricts configuration changes for observability resources (**observability-xxx**) in managed clusters to confirm that clusters are in a desired state. This is also applied in the hub cluster. Undesirable updates are reverted. See Customizing observability to learn how to customize the observability service.
- You can now design Grafana dashboards for your cluster. For more details, see Designing your Grafana dashboard.
- OpenShift Container Storage is now a supported storage solution for the observability service. For more information, see Enable observability service .

To learn more about observability, see Observing environments introduction.

1.1.3. Cluster management

- You can import and manage clusters in the Red Hat OpenShift Dedicated environment. You can also manage IBM Z managed clusters. See Importing a target managed cluster to the hub cluster for more information.
- You can display information that is specific to a cluster by using **clusterclaims**. For more information, see ClusterClaims.
- **Technology preview:** Submariner is integrated to provide direct networking across clusters that are managed by Red Hat Advanced Cluster Management. See Submariner documentation for more information.
- You can now limit permissions to create, manage, and import managed clusters to specific groups by creating and assigning a **clusterrole**. See Configuring a specific cluster management role for more information.
- Your list of cluster image sets is automatically updated to keep your list of cluster image sets current. For more information, see Maintaining a custom list of release images when connected .
- Automate an AnsibleTower task to run on a cluster, or on multiple clusters by creating an AnsibleJob. See Creating an AnsibleJob for a managed cluster for more information.

1.1.4. Application management

You can connect to a private repository using self-signed certificates, which is an improvement in Git connection ability. See Configuring application channel and subscription for a secure Git connection for more information.

Argo CD is integrated now so that you can manually sync any type of supported managed cluster. Enable the Argo CD cluster collection so that you can deploy applications from Argo CD to your managed clusters. See Configuring managed clusters for Argo CD to learn how to enable Argo CD.

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

1.1.5. Security and compliance

- You can now install gatekeeper with the Red Hat Advanced Cluster Management gatekeeper operator policy. See Installing gatekeeper using a gatekeeper operator policy for more information.
- You can now install the Red Hat OpenShift Container Platform compliance operator with the compliance operator policy. See Compliance operator policy for more details.
- You can now create and apply an Essential 8 (E8) scan policy to scan master and worker nodes for compliance with the E8 profiles. For more details, see E8 scan policy.
- You can now rotate internal managed certificates. For more information, see Certificates.

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:

- Red Hat OpenShift Container Platform 4.5 is not supported with 2.2.4 and later errata releases. You must run Red Hat OpenShift Container Platform version 4.6 to upgrade to Red Hat Advanced Cluster Management version 2.2.4 or later. If you cannot upgrade your Red Hat OpenShift Container Platform version to 4.6, you can continue to use Red Hat Advanced Cluster Management version 2.2.3.
- 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 2.2.13

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

1.2.2. Errata 2.2.12

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

1.2.3. Errata 2.2.11

- Removes unsupported versions of OpenShift Container Platform from the list of available cluster image sets. (Bugzilla 2030859)
- Fixes an issue that prevented managed clusters from upgrading when observability was disabled because its **observabilityAddon** status was in a **degraded** state. This fix removes the **observabilityAddon** from the managed clusters when observability is disabled, which allows the upgrade to complete. (GitHub 19636)

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

1.2.4. Errata 2.2.10

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

1.2.5. Errata 2.2.9

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

1.2.6. Errata 2.2.8

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.2.8 updates:

- Fixes an issue that caused an application to be deleted when it was reconciled. The deletion was caused by a misconfigured application manifest, such as an incorrect kustomization, in a subscribed Git repository. With this fix, an error message is displayed in the application topology user interface when there is a misconfigured manifest, but the deployed application is not deleted. (Bugzilla 1972947)

- Decreases the number of unnecessary secrets that are created, so the **cert manager** can start with less memory consumption. (GitHub 13127)

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

1.2.7. Errata 2.2.7

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.2.7 updates:

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

1.2.8. Errata 2.2.6

Important: See information in the previous notes.

Red Hat OpenShift Container Platform 4.8 is supported with 2.2.6 and later. *Important: Red Hat OpenShift Container Platform 4.8 for bare metal does not work for this release.

View a summarized list of fixes that the Red Hat Advanced Cluster Management for Kubernetes Errata provides:

- Fixes issues with imported clusters. Imported clusters from OpenShift Container Platform 3.11 in metrics visibility appear as other OpenShift Container Platform clusters appear. (GitHub 13162)

- Fixes deployment failure with an application that subscribes to a Helm chart from a Git repository server with a custom CA certificate. Configure the application channel with either **insecureSkipVerify: true** to avoid custom CA verification, or add the custom CA certificates to the channel configuration. (GitHub 14467)

- Updates RedisGraph StatefulSet for metadata and specification updates to fix upgrade issues. (GitHub 13661)

- Fixes the compare and sort functionality of the Configuration Policy Controller to stop a continual increase of the Kubernetes resource on a managed cluster that is configured by the policy controller. (Bugzilla 1973772)

- Updates the configuration policy namespace selection to identify the policy as **NonCompliant** when any selected namespace is **NonCompliant**. (Bugzilla 1969845)

- Removes the use of the **SelfLink** attribute in **cert-manager** for compatibility with OpenShift Container Platform 4.8. (GitHub 13121)

- Fixes an issue with **SecurityConstraintContext** policy, which prevented the search operator pod from starting. Updates the Docker image to run with a non-root user by default and updates the security context in the search operator deployment with **runAsNonRoot**, which starts the container with a non-root user. (Bugzilla 1967953)

- Adds various container updates.

- Fixes an issue with bare metal asset fields not populating correctly. (GitHub 9850)

1.2.9. Errata 2.2.5

Important: See information in the previous notes.

View a summarized list of fixes that the Red Hat Advanced Cluster Management for Kubernetes Errata provides:

- Fixes multiple certificate issues.

- Adds labels to the Search redisgraph StatefulSet so that the pod is restarted automatically when certificates are refreshed. (GitHub 12299)

1.2.10. Errata 2.2.4

Important: See information in the previous notes.

View a summarized list of fixes that the Red Hat Advanced Cluster Management for Kubernetes Errata provides:

- Fixes issue with **placementrule** and **placementbinding** not being deleted when a disabled policy in the console was deleted. (GitHub 12689)

- Removes channel role and role binding that was created in the product system namespace. This fix avoids exposing system secrets to managed cluster service accounts. (GitHub 12319)

- Resolves a problem in Observability with the **endpoint-observability-operator** pod. Communication from managed cluster to hub cluster caused a **certificate signed by unknown authority** error if the hub cluster was installed on IBM cloud. (GitHub 11125)

- Fixes an issue with Hive controller not showing the correct version. (GitHub 12013)

- Fixes an issue with Observability crash. (Bugzilla 1967890)

1.2.11. Errata 2.2.3

View a summarized list of fixes that the Red Hat Advanced Cluster Management for Kubernetes Errata provides:

- Adds a control category to the compliance operator policy. (GitHub 11234)

- Updates the YAML editor to display an error when an invalid **imageSetRef** is specified in the **ClusterDeployment**. (Bugzilla 1946244)

- Fixes a console issue for imported clusters. (GitHub 11119)

- Adds quotation marks around **name** and **namespace** in YAML files to ensure the values are entered as a string. (Bugzilla 1936883)

- Upgrades the Kustomize API module version to **0.8.5** to support a Kubernetes-sigs fix. (GitHub 11362)

- Adds support for the **GET** cluster role for non-administrator users to access managed clusters. The clusters list that is created and returned by the **PlacementRule** contains only the clusters that the user can access. ((Bugzilla 1946244)

- Adds an upgrade for the Gatekeeper and Gatekeeper Operator base images. (Github 12038)

1.2.12. Errata 2.2.2

View a summarized list of fixes that the Red Hat Advanced Cluster Management for Kubernetes Errata provides:

- This errata addresses multiple security issues and container image updates.

- Resolves an issue with an existing policy in the *create policy* form displaying a blank page. (Bugzilla 1940588)

- Adds the gatekeeper operator policy, which is now available in the **Create policy** specification drop-down menu. (GitHub 10447)

- Fixes an *Application topology* deployment status issue. The Helm resources chart now displays the resource deployment status when the custom alias does not match the package name. (GitHub 10401)

- Fixes an issue with **ObservabilityAddon** in **terminating** status. (GitHub 10012)

- Adds the ability to create Azure clusters for all possible regions. (GitHub 9700)

- Fixes an issue with custom certificate authority on the hub cluster. Submariner agent can now connect. (GitHub 9894)

- Fixes an issue with **packageOverrides** that were incorrectly specified in subscription CR. Errors occurred on a pod on the hub cluster, or the **klusterlet-addon-appmgr** pod on the managed cluster. The log now ignores the override. (GitHub 9700)

- Updates the Visual Web Terminal CLI to support OpenShift Container Platform version 4.7. (GitHub 9640)

- Updates import cluster commands with double quotes to handle unescaped characters. Be sure to use base64 with the **-d** option. (GitHub 10748)

- Fixes an issue with the cluster YAML editor. (Bugzilla 1941778)

- Adds support for NodeJS version 14 from version 12 to limit vulnerabilities in the base image. (GitHub 9540)

- Updates the **ServiceExport** API version in the base image. (Bugzilla 1936528)

- Fixes a bare metal asset issue where assets that were originally referenced in **clusterdeployment** could not be reused for another **clusterdeployment**. (GitHub 9272)

- Fixes a bare metal issue that caused updates too frequently. (GitHub 9463)

- Changes the Application management default reconcile rate to 15 minutes. The reconcile rate is also now configurable. (GitHub 10644)

- Fixes resource issues with the default alert manager by removing the **KubeAPIServerLatency** rule. (GitHub 10693)

- Updates role-based access control. Added authorization for the **Viewer** role to create and delete the **ManagedClusterView** resource, and added authorization for **cluster-manager-admin** users to **get** and view logs for pods. (GitHub 11243, 11242)

1.2.13. Errata 2.2.1

View a summarized list of Red Hat Advanced Cluster Management for Kubernetes Errata 2.2.1 updates:

This errata release delivers a new set of 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 .

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

1.3.1. Installation known issues

1.3.1.1. OpenShift Container Platform cluster upgrade failed status

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

1.3.1.2. Certificate manager must not exist during an installation

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

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

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

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

1.3.2. Web console known issues

1.3.2.1. Node discrepancy between Cluster page and search results

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

1.3.2.2. LDAP user names are case-sensitive

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

1.3.2.3. Console features might not display in Firefox earlier versions

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

1.3.2.4. Unable to search using values with empty spaces

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

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

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

1.3.2.6. Secret content is no longer displayed

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

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

1.3.2.7. Restrictions for storage size in searchcustomization

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

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

1.3.2.8. YAML file is not displayed from the *Search* page

When you use the Safari v14.0.3 web browser, the YAML file is not displayed from the *Search* page. See Supported browsers for other supported browsers or upgrade your Safari version.

1.3.2.9. Restart *redisgraph* StatefulSet and pod

The **redisgraph** StatefulSet and pod are not refreshed when you upgrade from 2.2.z to 2.2.5. You must delete the **redisgraph** StatefulSet manually, so that the changes are picked up. Complete the following steps to fix this issue:

1. Install Red Hat Advanced Cluster Management 2.2.4.

2. Upgrade to Red Hat Advanced Cluster Management 2.2.5.

3. Log in as an administrator and check the **redisgraph** StatefulSet by running the following command:

   ```
   oc get statefulset search-redisgraph -n open-cluster-management
   ```

4. Notice that the StatefulSet and pod did not restart.

5. Delete the **redisgraph** StatefulSet with the following command:

   ```
   oc delete statefulset search-redisgraph -n open-cluster-management
   ```

6. Verifiy if the **redisgraph** pod is running successfully with the following command:

   ```
   oc get pod search-redisgraph-0 -n open-cluster-management
   ```

The **redisgraph** StatefulSet and pod have restarted.

1.3.2.10. 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.

1.3.2.11. Observability add-on stuck in terminating

When the managed cluster is detached forcefully, the **ObservabilityAddon** resource (***observability-addon***) in the cluster namespace is stuck in the *Terminating* status and cannot be removed. Also, the cluster namespace cannot be deleted.

To fix this problem, you can update the **ObervabilityAddon** resource in the cluster namespace. Update the resource by deleting the ***finalizers*** parameter in the metadata. Run the following command:

```
kubectl edit observabilityaddon observability-addon -n <CLUSTER_NAMESPACE>
```

CLUSTER_NAMESPACE is the cluster namespace for the detached cluster.

After the ***finalizers*** parameter is removed, the **ObervabilityAddon** resource is removed.

1.3.2.12. 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)**

1.3.2.13. Metrics data no longer collected after upgrade

After you upgrade Red Hat Advanced Cluster Management from 2.2.3 to 2.2.4, you might notice that the **metrics-collector** in the **open-cluster-management-addon-observability** namespace stops collecting data from your managed clusters. This is because the image manifests ConfigMap was upgraded after the multicluster observability operator was upgraded.

You might see the following Quay.io images being used in the **endpoint-observabililty-operator** YAML file: **quay.io/open-cluster-management/endpoint-monitoring-operator:2.2.0-6a5ea47fc39d51fb4fade6157843f2977442996e** and **quay.io/open-cluster-management/metrics-collector:2.2.0-ff79e6ec8783756b942a77f08b3ab763dfd2dc15**.

To fix this issue, delete **multicluster-observability-operator** pods that are in the **open-cluster-management** namespace in the hub cluster. After the new pod is created, a new **endpoint-observability-operator** deployment is created in your managed cluster with the correct images.

1.3.2.14. *MultiClusterObservability* CR displays incorrect status

After you upgrade Red Hat Advanced Cluster Management from 2.1 to 2.2.x, the **MultiClusterObservability** custom resource (CR) continues to display **Installing** from the OpenShift Container Platform console. You can confirm that **MultiClusterObservability** is deployed successfully by ensuring that the metrics data is displayed from the Grafana console.

1.3.2.15. Observability service metric gap

Some metric data is not displayed on the Grafana dashboard for Red Hat OpenShift Container Platform verson 4.8 clusters. You must upgrade to version 2.3. For more information, see Upgrading by using the operator.

1.3.3. Cluster management known issues

1.3.3.1. 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.

1.3.3.2. Cluster version does not immediately update in the console when the cluster is upgraded

When you upgrade a managed cluster or a local cluster on the **Cluster details** page, it might take up to 10 minutes after the upgrade process completes for the updated version number to display on the **Cluster details** page.

1.3.3.3. 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.

1.3.3.4. 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.

1.3.3.5. 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.

1.3.3.6. Console might report managed cluster policy inconsistency

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

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

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

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

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

1.3.3.7. Importing clusters might require two attempts

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

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

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

1.3.3.9. 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

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
```

1.3.3.10. 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}"} }]'
  ```

1.3.3.11. Cannot run management ingress as non-root user

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

1.3.3.12. 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.

1.3.3.13. 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.

1.3.3.14. Cannot upgrade OpenShift Container Platform managed clusters on Red Hat OpenShift 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 Red Hat OpenShift Dedicated environment.

1.3.3.15. Metrics are unavailable in the Grafana console

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

  "Annotation Query Failed"

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

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

  no project or cluster found

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

1.3.3.16. Related bare metal assets not destroyed after bare metal cluster is destroyed

Your bare metal asset might remain as an orphaned asset after you destroy the cluster that was associated with it. This happens when you have the required permissions to destroy a cluster, but not to destroy the bare metal asset. To ensure that you do not experience this issue, add a finalizer to the **ClusterDeployment** resource when you create a bare metal asset with Red Hat Advanced Cluster Management that references a cluster deployment:

```
kubectl patch clusterdeployments <name> -n <namespace> -p '{"metadata":{"finalizers":
["baremetalasset.inventory.open-cluster-management.io"]}}'
```

Replace ***name*** with the name of your cluster deployment.

Replace ***namespace*** with the namespace of your cluster resource.

If you delete the cluster deployment, you must remove the finalizer manually by entering the following command:

```
kubectl patch clusterdeployments <name> -n <namespace> -p '{"metadata":{"finalizers":[]}}'
```

Replace ***name*** with the name of your cluster deployment.

Replace ***namespace*** with the namespace of your cluster resource.

1.3.4. Application management known issues

1.3.4.1. 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.

1.3.4.2. 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
```

1.3.4.3. Work manager add-on search details

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

1.3.4.4. Deployable resource with empty specification does not work

Applying a Deployable resource with no specification crashes the pod **multicluster-operators-application** container **multicluster-operators-deployable**. A deployable needs to contain specifications.

If you accidentally create the resource without a specification, delete the unnecessary deployable and restart the **multicluster-operators-application** pod.

See the following example of a Deployable that is empty and crashes the pod:

```
apiVersion: apps.open-cluster-management.io/v1
kind: Deployable
metadata:
  labels:
    app: simple-app-tester
  name: simple-app-tester-deployable
  namespace: grp-proof-of-concept-acm
```

1.3.4.5. Topology ReplicationController or ReplicaSet resources missing

When you deploy an application that directly creates a **ReplicationController** or **ReplicaSet** resource, the Pod resources are not displayed in the *Application topology*. You can use the **Deployment** or **DeploymentConfig** resources instead for creating Pod resources.

1.3.4.6. Application topology displays incorrect Ansible job status

Ansible tests run as hooks for the subscription and not as regular tasks. You need to store Ansible tasks in a prehook and posthook folder. You can choose to deploy the Ansible tasks as regular tasks and not as hooks, but the Application topology Ansible job status is not reported correctly in this case.

1.3.4.7. Application Ansible hook stand-alone mode

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

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

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

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

    ```
    apiVersion: apps.open-cluster-management.io/v1
    kind: PlacementRule
    metadata:
      name: <towhichcluster>
      namespace: hello-openshift
    spec:
    ```

```
  clusterSelector:
    matchLabels:
      local-cluster: "true" #this points to your hub cluster
```

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

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

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

1.3.4.8. Application Deploy on local cluster limitation

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

To resolve the issue, see the following procedure:

1. Deselect the **Deploy on local cluster** option in the console.

2. Select the **Deploy application resources only on clusters matching specified labels** option.

3. Create the following label: **local-cluster : 'true'**.

1.3.4.9. Namespace channel subscription remains in failed state

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

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

1. Log in to your hub cluster.

2. Manually add a label to the subscription using the following command:

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

1.3.4.10. 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.

1.3.4.11. 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.

1.3.4.12. 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.

1.3.4.13. 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
```

1.3.4.14. Application channels require unique namespaces

Creating more than one channel in the same namespace can cause errors with the hub cluster.

For instance, namespace **charts-v1** is used by the installer as a Helm type channel, so do not create any additional channels in **charts-v1**. Ensure that you create your channel in a unique namespace. All channels need an individual namespace, except GitHub channels, which can share a namespace with another GitHub channel.

1.3.5. Application management limitations

1.3.5.1. Application console tables

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

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

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

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

1.3.6. Security known issues

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

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

1.4. DEPRECATIONS AND REMOVALS

Learn when parts of the product are deprecated or removed from Red Hat Advanced Cluster Management for Kubernetes.

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.

Note: A *stabilized* item is not deprecated or removed from a release of a product, but is no longer updated or developed. For instance, if an API is replaced with a new version in a release, but no longer updated, it can be listed in this topic as *stabilized*. The API is still available for one to three more releases before deprecated or removed.

Consider the alternative actions in the *Recommended action* and details that are provided in a table only if stabilized functions are listed for the current release.

1.4.1. Deprecations

A *deprecated* component, feature, or service is supported, but no longer recommended for use and might become obsolete. 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
Application management	**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

1.4.1.1. Guidance for API deprecations

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

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

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

1.4.2. Removals

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

Product or category	Affected item	Version	Recommended action	More details and links
Observability Topology	Topology access from *Observe environments* removed completely	2.2	None	Application topology is located in *Application management* and no longer in the *Observability console*.
Application management	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.