Red Hat Advanced Cluster Management for Kubernetes 2.6

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: 2023-10-31

Red Hat Advanced Cluster Management for Kubernetes 2.6 Release notes

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

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. RELEASE NOTES

Learn about the current release.

Note: The 2.4 and earlier versions of Red Hat Advanced Cluster Management are *removed* from service, and are no longer supported. The documentation might remain available, but is deprecated without any Errata or other updates available.

- What's new in Red Hat Advanced Cluster Management for Kubernetes
- Errata updates
- Known issues and limitations
- Deprecations and removals
- Red Hat Advanced Cluster Management for Kubernetes considerations for GDPR readiness
- FIPS readiness

If you experience issues with one of the currently supported releases, or the product documentation, go to Red Hat Support where you can troubleshoot, view Knowledgebase articles, connect with the Support Team, or open a case. You must log in with your credentials. You can also learn more about the Customer Portal documentation at Red Hat Customer Portal FAQ .

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

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

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

Learn more about what is new this release:

- Get an overview of Red Hat Advanced Cluster Management for Kubernetes from Welcome to Red Hat Advanced Cluster Management for Kubernetes.
- The open source *Open Cluster Management* repository is ready for interaction, growth, and contributions from the open community. To get involved, see open-cluster-management.io. You can access the GitHub repository for more information, as well.
- See the Multicluster architecture topic to learn more about major components of the product.
- The Getting started guide references common tasks that get you started, as well as the *Troubleshooting guide*.
- Web console
- Clusters
- Applications
- Governance

- Add-ons

1.1.1. Web console

- You can now access Grafana through the OpenShift Container Platform route with a URL. See Designing your Grafana dashboard for more information about the example URL:

  ```
  https://grafana-open-cluster-management-observability.{OPENSHIFT_INGRESS_DOMAIN}
  ```

- You can specify *AnsibleJob* templates while importing clusters and after creating clusters. See Configuring an *AnsibleJob* template to run on a cluster by using the console for more details.

Learn more about the console in the Console overview.

1.1.2. Clusters

Cluster lifecycle documentation is now located in The multicluster engine operator cluster lifecycle overview as documentation for the multicluster engine for Kubernetes operator operator.

- The *multicluster engine operator* is generally available as a software operator that enhances cluster fleet management. The multicluster engine operator supports Red Hat OpenShift Container Platform and Kubernetes cluster lifecycle management across clouds and data centers. Red Hat OpenShift Container Platform is a prerequisite for the multicluster engine operator.

- The permission cluster set **bind** grants permission to bind the cluster set to a namespace by creating a **ManagedClusterSetBinding**. See Assigning users or groups Role-Based Access Control permissions to your ManagedClusterSet for more details.

- When you create a managed cluster, a **ManagedClusterSet** called **global** is automatically created to ease management. See Global ManagedClusterSet for more details.

- You can add multiple hosts to an infrastructure environment at the same time by selecting the **By uploading a YAML** option. See Scaling hosts to an infrastructure environment for more details.

- The navigation menu option to access the infrastructure environments changed from **Infrastructure environment** to **Host inventory**.

- You can add additional workers to single-node OpenShift clusters created with the Central Infrastructure Management service. See Creating your cluster with the console for more details.

1.1.3. Applications

- You can optionally configure the Helm channel type to watch namespace-scoped resources so that any manual changes to those resources are reverted. See Configuring Helm to watch namespace resources.

- You can create and view *OpenShift*, *Flux*, and *Argo CD* application types. An **ApplicationSet** represents Argo applications that are generated from the controller. See see Console overview.

For other Application topics, see Managing applications.

1.1.4. Governance

- You can use the **configurationPolicyAnnotations** parameter in the policy generator configuration to specify key-value pair annotations on generated configuration policies. See Policy generator configuration reference table for more details.
- Configure the concurrency of the configuration policy controller for each managed cluster to change how many configuration policies it can evaluate at the same time. See Configuring the configuration policy controller for more details.
- Clean up resources by using the **pruneObjectBehavior** parameter. See Cleaning up resources that are created by policies
- Use the **everyEvent** mode to set your governance Ansible automation to run for every policy violation event. See the *Create a policy violation automation from the console* section in the Configuring Ansible Tower for governance .
- Select namespaces by label for the policy controllers by using the **matchLabels** and **matchExpressions** parameters. See Configuration policy YAML table for more information.
- Define file path expressions in the **include** and **exclude** parameters to select namespaces by name. See Configuration policy YAML table for more details.

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

1.1.5. Add-ons

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

- The VolSync operator is now generally available for for copying persistent volume claims with Red Hat Advanced Cluster Management. See VolSync persistent volume replication service for more information.

1.2. KNOWN ISSUES

Review the known issues for Red Hat Advanced Cluster Management for Kubernetes. The following list contains known issues for this release, or known issues that continued from the previous release. For your Red Hat OpenShift Container Platform cluster, see OpenShift Container Platform known issues .

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

1.2.1. Documentation known issues

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

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

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

1. Copy the link that is not resolving to the correct section and paste it in your browser address bar. For example, it might be: **https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_management_for_kubernetes/2.6/html/clusters/index#volsync**.

2. In the link, replace **html** with **html-single**. The new URL should read: **https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_management_for_kubernetes/2.6/html-single/clusters/index#volsync**

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

1.2.2. Installation known issues

1.2.2.1. Deprecated resources remain after upgrade to Errata releases

After you upgrade from 2.4.x to 2.5.x, and then to 2.6.x, deprecated resources in the managed cluster namespace might remain. You need to manually delete these deprecated resources if version 2.6.x was upgraded from 2.4.x:

Note: You need to wait 30 minutes or more before you upgrade from version 2.5.x to version 2.6.x.

You can delete from the console, or you can run a command similar to the following example for the resources you want to delete:

```
oc delete -n <managed cluster namespace> managedclusteraddons.addon.open-cluster-management.io <resource-name>
```

See the list of deprecated resources that might remain:

```
managedclusteraddons.addon.open-cluster-management.io:
policy-controller
manifestworks.work.open-cluster-management.io:
-klusterlet-addon-appmgr
-klusterlet-addon-certpolicyctrl
-klusterlet-addon-crds
-klusterlet-addon-iampolicyctrl
-klusterlet-addon-operator
-klusterlet-addon-policyctrl
-klusterlet-addon-workmgr
```

1.2.2.2. Pods might not come back up after upgrading Red Hat Advanced Cluster Management

After upgrading Red Hat Advanced Cluster Management to a new version, a few pods that belong to a **StatefulSet** might remain in a **failed** state. This infrequent event is caused by a known Kubernetes issue.

As a workaround for this problem, delete the failed pod. Kubernetes automatically relaunches it with the correct settings.

1.2.2.3. OpenShift Container Platform cluster upgrade failed status

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

1.2.2.4. Create MultiClusterEngine button not working

After installing Red Hat Advanced Cluster Management for Kubernetes in the Red Hat OpenShift Container Platform console, a pop-up window with the following message appears:

MultiClusterEngine required

Create a MultiClusterEngine instance to use this Operator.

The **Create MultiClusterEngine** button in the pop-up window message might not work. To work around the issue, select **Create instance** in the MultiClusterEngine tile in the Provided APIs section.

1.2.3. Web console known issues

1.2.3.1. 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.

1.2.3.2. Console features might not display in Firefox earlier version

There are known issues with dark theme styling for older versions of Firefox. Upgrade to the latest version for the best console compatibility.

For more information, see Supported browsers.

1.2.3.3. Restrictions for storage size in search customization

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
```

1.2.3.4. Search query parsing error

If an environment becomes large and requires more tests for scaling, the search queries can timeout which results in a parsing error message being displayed. This error is displayed after 30 seconds of waiting for a search query.

Extend the timeout time with the following command:

```
kubectl annotate route multicloud-console haproxy.router.openshift.io/timeout=Xs
```

1.2.3.5. Cannot edit namespace bindings for cluster set

When you edit namespace bindings for a cluster set with the **admin** role or **bind** role, you might encounter an error that resembles the following message:

ResourceError: managedclustersetbindings.cluster.open-cluster-management.io "<cluster-set>" is forbidden: User "<user>" cannot create/delete resource "managedclustersetbindings" in API group "cluster.open-cluster-management.io" in the namespace "<namespace>".

To resolve the issue, make sure you also have permission to create or delete a **ManagedClusterSetBinding** resource in the namespace you want to bind. The role bindings only allow you to bind the cluster set to the namespace.

1.2.4. Observability known issues

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

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

1.2.4.2. Observability endpoint operator fails to pull image

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

For more information, see Enabling observability.

1.2.4.3. There is no data from ROKS and HyperShift clusters

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

1.2.4.4. There is no etcd data from ROKS and HyperShift clusters

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

1.2.4.5. High CPU usage by the search-collector pod

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

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

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

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

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

1.2.4.7. Metrics are unavailable in the Grafana console

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

  "Annotation Query Failed"

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

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

  no project or cluster found

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

1.2.4.8. Prometheus data loss on managed clusters

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

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

1.2.4.9. Error ingesting out-of-order samples

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

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

The error message means that the time series data sent by a managed cluster, during a metrics collection interval is older than the time series data it sent in the previous collection interval. When this problem happens, data is discarded by the Thanos receivers and this might create a gap in the data

shown in Grafana dashboards. If the error is seen frequently, it is recommended to increase the metrics collection interval to a higher value. For example, you can increase the interval to 60 seconds.

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

1.2.4.10. Grafana deployment fails on managed clusters

The Grafana instance does not deploy to the managed cluster if the size of the manifest exceeds 50 thousand bytes. Only the **local-cluster** appears in Grafana after you deploy observability.

1.2.4.11. Grafana deployment fails after upgrade

If you have a **grafana-dev** instance deployed in earlier versions before 2.6, and you upgrade the environment to 2.6, the **grafana-dev** does not work. You must delete the existing **grafana-dev** instance by running the following command:

```
./setup-grafana-dev.sh --clean
```

Recreate the instance with the following command:

```
./setup-grafana-dev.sh --deploy
```

1.2.4.12. *klusterlet-addon-search* pod fails

The **klusterlet-addon-search** pod fails because the memory limit is reached. You must update the memory request and limit by customizing the **klusterlet-addon-search** deployment on your managed cluster. Edit the **ManagedclusterAddon** custom resource named **search-collector**, on your hub cluster. Add the following annotations to the **search-collector** and update the memory, **addon.open-cluster-management.io/search_memory_request=512Mi** and **addon.open-cluster-management.io/search_memory_limit=1024Mi**.

For example, if you have a managed cluster named **foobar**, run the following command to change the memory request to **512Mi** and the memory limit to **1024Mi**:

```
oc annotate managedclusteraddon search-collector -n foobar \
addon.open-cluster-management.io/search_memory_request=512Mi \
addon.open-cluster-management.io/search_memory_limit=1024Mi
```

1.2.5. Cluster management known issues

See the following known issues and limitations for cluster management:

1.2.5.1. Disconnected installation settings for cluster creation cannot be entered or are ignored if entered

When you create a cluster by using the bare metal provider and a disconnected installation, you must store all your settings in the credential in the *Configuration for disconnected installation* section. You cannot enter them in the cluster create console editor.

When creating a cluster by using the VMware vSphere or Red Hat OpenStack Platform providers and disconnected installation, if a certificate is required to access the mirror registry, you must enter it in the

Additional trust bundle field of your credential in the *Configuration for disconnected installation section* . If you enter that certificate in the cluster create console editor, it is ignored.

1.2.5.2. Credential with disconnected installer does not distinguish between the certificates

When creating a credential for the bare metal, VMware vSphere, or Red Hat OpenStack Platform provider, note that the *Additional trust bundle* field in the *Proxy and Configuration for disconnected installation* contains the same value since the installer does not distinguish between the certificates. You can still use these features independently, and you can enter multiple certificates in the field if different certificates are required for proxy and disconnected installation.

1.2.5.3. Manual removal of the VolSync CSV required on managed cluster when removing the add-on

When you remove the VolSync **ManagedClusterAddOn** from the hub cluster, it removes the VolSync operator subscription on the managed cluster but does not remove the cluster service version (CSV). To remove the CSV from the managed clusters, run the following command on each managed cluster from which you are removing VolSync:

```
oc delete csv -n openshift-operators volsync-product.v0.4.0
```

If you have a different version of VolSync installed, replace **v0.4.0** with your installed version.

1.2.5.4. Deleting a managed cluster set does not automatically remove its label

After you delete a **ManagedClusterSet**, the label that is added to each managed cluster that associates the cluster to the cluster set is not automatically removed. Manually remove the label from each of the managed clusters that were included in the deleted managed cluster set. The label resembles the following example: **cluster.open-cluster-management.io/clusterset:<ManagedClusterSet Name>**.

1.2.5.5. ClusterClaim error

If you create a Hive **ClusterClaim** against a **ClusterPool** and manually set the **ClusterClaimspec** lifetime field to an invalid golang time value, Red Hat Advanced Cluster Management stops fulfilling and reconciling all **ClusterClaims**, not just the malformed claim.

If this error occurs. you see the following content in the **clusterclaim-controller** pod logs, which is a specific example with the pool name and invalid lifetime included:

```
E0203 07:10:38.266841       1 reflector.go:138] sigs.k8s.io/controller-
runtime/pkg/cache/internal/informers_map.go:224: Failed to watch *v1.ClusterClaim: failed to list
*v1.ClusterClaim: v1.ClusterClaimList.Items: []v1.ClusterClaim:
v1.ClusterClaim.v1.ClusterClaim.Spec: v1.ClusterClaimSpec.Lifetime: unmarshalerDecoder: time:
unknown unit "w" in duration "1w", error found in #10 byte of ...|time":"1w"}},{"apiVe|..., bigger context
...|clusterPoolName":"policy-aas-hubs","lifetime":"1w"}},
{"apiVersion":"hive.openshift.io/v1","kind":"Cl|...
```

You can delete the invalid claim.

If the malformed claim is deleted, claims begin successfully reconciling again without any further interaction.

1.2.5.6. The product channel out of sync with provisioned cluster

The **clusterimageset** is in **fast** channel, but the provisioned cluster is in **stable** channel. Currently the product does not sync the **channel** to the provisioned OpenShift Container Platform cluster.

Change to the right channel in the OpenShift Container Platform console. Click **Administration** > **Cluster Settings** > **Details Channel**.

1.2.5.7. Restoring the connection of a managed cluster with custom CA certificates to its restored hub cluster might fail

After you restore the backup of a hub cluster that managed a cluster with custom CA certificates, the connection between the managed cluster and the hub cluster might fail. This is because the CA certificate was not backed up on the restored hub cluster. To restore the connection, copy the custom CA certificate information that is in the namespace of your managed cluster to the **<managed_cluster>-admin-kubeconfig** secret on the restored hub cluster.

Tip: If you copy this CA certificate to the hub cluster before creating the backup copy, the backup copy includes the secret information. When the backup copy is used to restore in the future, the connection between the hub and managed clusters will automatically complete.

1.2.5.8. The local-cluster might not be automatically recreated

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

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

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

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

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

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

1.2.5.10. Cluster provisioning with Infrastructure Operator fails

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

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

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

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

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

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

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

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

- 4.8.17 and earlier
- 4.9.6 and earlier

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

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

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

To recover from this case, complete the following steps:

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

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

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

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

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

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

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

1.2.5.12. Cluster provision with Ansible automation fails in proxy environment

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

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

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

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

1.2.5.14. Cannot delete managed cluster namespace manually

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

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

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

1.2.5.16. Hub cluster and managed clusters clock not synced

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

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

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

1.2.5.18. Automatic secret updates for provisioned clusters is not supported

When you change your cloud provider access key on the cloud provider side, you also need to update the corresponding credential for this cloud provider on the console of multicluster engine for Kubernetes operator. This is required when your credentials expire on the cloud provider where the managed cluster is hosted and you try to delete the managed cluster.

1.2.5.19. 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.

1.2.5.20. 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.

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

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

1.2.5.22. 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.

1.2.5.23. Cannot use Ansible Tower integration with an IBM Power or IBM Z system hub cluster

You cannot use the Ansible Tower integration when the Red Hat Advanced Cluster Management for Kubernetes hub cluster is running on IBM Power or IBM Z systems because the Ansible Automation Platform Resource Operator does not provide **ppc64le** or **s390x** images.

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

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

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

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

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

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

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

1.2.5.25. Cluster-proxy-addon does not start after upgrade

After you upgrade from version 2.4.x to 2.5.0, **cluster-proxy-addon** does not start and **cluster-proxy-addon-manager** raises a nil pointer exception.

To work around this issue, complete the following steps:

1. Disable **cluster-proxy-addon**. See Advanced configuration to learn more.
2. Delete the **cluster-proxy-signer** secret from the **open-cluster-management** namespace.
3. Enable **cluster-proxy-addon**.

1.2.5.26. OpenShift Container Platform 4.10.z does not support hosted control plane clusters with proxy configuration

When you create a hosting service cluster with a cluster-wide proxy configuration on OpenShift Container Platform 4.10.z, the **nodeip-configuration.service** service does not start on the worker nodes.

1.2.5.27. Cannot provision OpenShift Container Platform 4.11 cluster on Azure

Provisioning an OpenShift Container Platform 4.11 cluster on Azure fails due to an authentication operator timeout error. To work around the issue, use a different worker node type in the **install-config.yaml** file or set the **vmNetworkingType** parameter to **Basic**. See the following **install-config.yaml** example:

```
compute:
- hyperthreading: Enabled
  name: 'worker'
  replicas: 3
  platform:
    azure:
      type:  Standard_D2s_v3
      osDisk:
        diskSizeGB: 128
      vmNetworkingType: 'Basic'
```

1.2.5.28. Client cannot reach iPXE script

iPXE is an open source network boot firmware. See iPXE for more details.

When booting a node, the URL length limitation in some DHCP servers cuts off the **ipxeScript** URL in the **InfraEnv** custom resource definition, resulting in the following error message in the console:

no bootable devices

To work around the issue, complete the following steps:

1. Apply the **InfraEnv** custom resource definition when using an assisted installation to expose the **bootArtifacts**, which might resemble the following file:

```
status:
  agentLabelSelector:
    matchLabels:
      infraenvs.agent-install.openshift.io: qe2
  bootArtifacts:
```

```
    initrd: https://assisted-image-service-multicluster-engine.redhat.com/images/0000/pxe-initrd?api_key=0000000&arch=x86_64&version=4.11
    ipxeScript: https://assisted-service-multicluster-engine.redhat.com/api/assisted-install/v2/infra-envs/00000/downloads/files?api_key=000000000&file_name=ipxe-script
    kernel: https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.11/latest/rhcos-live-kernel-x86_64
    rootfs: https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.11/latest/rhcos-live-rootfs.x86_64.img
```

2. Create a proxy server to expose the **bootArtifacts** with short URLs.

3. Copy the **bootArtifacts** and add them them to the proxy by running the following commands:

```
for artifact in oc get infraenv qe2 -ojsonpath="{.status.bootArtifacts}" | jq ". | keys[]" | sed "s/\"//g"
do curl -k oc get infraenv qe2 -ojsonpath="{.status.bootArtifacts.${artifact}}"` -o $artifact
```

4. Add the **ipxeScript** artifact proxy URL to the **bootp** parameter in **libvirt.xml**.

1.2.5.29. Custom ingress domain is not applied correctly

You can specify a custom ingress domain by using the **ClusterDeployment** resource while installing a managed cluster, but the change is only applied after the installation by using the **SyncSet** resource. As a result, the **spec** field in the **clusterdeployment.yaml** file displays the custom ingress domain you specified, but the **status** still displays the default domain.

1.2.6. Application management known issues

See the following known issues for the application lifecycle component.

1.2.6.1. Application ObjectBucket channel type cannot use allow and deny lists

You cannot specify allow and deny lists with ObjectBucket channel type in the **subscription-admin** role. In other channel types, the allow and deny lists in the subscription indicates which Kubernetes resources can be deployed, and which Kubernetes resources should not be deployed.

1.2.6.2. Argo Application cannot be deployed on 3.x OpenShift Container Platform managed clusters

Argo **ApplicationSet** from the console cannot be deployed on 3.x OpenShift Container Platform managed clusters because the **Infrastructure.config.openshift.io** API is not available on on 3.x.

1.2.6.3. Changes to the multicluster_operators_subscription image do not take effect automatically

The **application-manager** add-on that is running on the managed clusters is now handled by the subscription operator, when it was previously handled by the klusterlet operator. The subscription operator is not managed the **multicluster-hub**, so changes to the **multicluster_operators_subscription** image in the **multicluster-hub** image manifest ConfigMap do not take effect automatically.

If the image that is used by the subscription operator is overridden by changing the **multicluster_operators_subscription** image in the **multicluster-hub** image manifest ConfigMap, the **application-manager** add-on on the managed clusters does not use the new image until the

subscription operator pod is restarted. You need to restart the pod.

1.2.6.4. Policy resource not deployed unless by subscription administrator

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

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

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

1.2.6.5. 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.

1.2.6.6. Edit role for application error

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

1. Run **oc edit clusterrole applications.app.k8s.io-v1beta2-edit -o yaml** to open the application edit cluster role.
2. Remove **create** and **delete** from the verbs list.
3. Save the change.

1.2.6.7. Edit role for placement rule error

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

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

1.2.6.8. Application not deployed after an updated placement rule

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

You can run **oc get pods -n open-cluster-management-agent-addon |grep application-manager** to verify.

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

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

1.2.6.9. 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
```

1.2.6.10. 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.

1.2.6.11. Ansible Automation Platform job fail

Ansible jobs fail to run when you select an incompatible option. Ansible Automation Platform only works when the **-cluster-scoped** channel options are chosen. This affects all components that need to perform Ansible jobs.

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

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

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

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

1.2.6.14. Application name requirements

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

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

1.2.6.15. Application console table limitations

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

- From the *Applications* table on the *Overview* page and the *Subscriptions* table on the *Advanced configuration* page, the *Clusters* column displays a count of clusters where application resources are deployed. Since applications are defined by resources on the local cluster, the local cluster is included in the search results, whether actual application resources are deployed on the local cluster or not.
- From the *Advanced configuration* table for *Subscriptions*, the *Applications* column displays the total number of applications that use that subscription, but if the subscription deploys child applications, those are included in the search result, as well.
- From the *Advanced configuration* table for *Channels*, the *Subscriptions* column displays the total number of subscriptions on the local cluster that use that channel, but this does not include subscriptions that are deployed by other subscriptions, which are included in the search result.

1.2.6.16. No Application console topology filtering

The *Console* and *Topology* for *Application* changes for the 2.6. There is no filtering capability from the console Topology page.

1.2.6.17. ApplicationSet resources do not show status in topology

When you create **ApplicationSet** applications that deploy resources to a different namespace than the namespace defined in the **ApplicationSet** YAML, the resource status does not appear in the topology.

1.2.6.18. Allow and deny list does not work in Object storage applications

The **allow** and **deny** list feature does not work in Object storage application subscriptions.

1.2.6.19. ApplicationSet topology status icon spins continuously

The **ApplicationSet** topology status icon spins continuously if an **ApplicationSet** application is deployed, but has no associated Argo applications.

1.2.6.20. Unsupported OpenShift Container Platform versions listed after hub cluster upgrade

After you upgrade your hub cluster from versions before 2.5 to 2.6, some unsupported OpenShift Container Platform versions are listed on the *Cluster* page in the console.

The stale **clusterImageSet** resources that are deployed by earlier versions before 2.5 subscription controller are not deleted after the upgrade. To resolve this, manually delete the **clusterImageSet** resources that have unsupported OpenShift Container Platform versions. For example, run the following command to delete the **img4.7.0-x86-64-appsub clusterImageSet**:

```
oc delete clusterimageset img4.7.0-x86-64-appsub
```

1.2.6.21. Cannot remove application subscription after restoring hub cluster to new hub cluster

When you restore the hub cluster data to a new hub cluster, the existing application subscription on the managed cluster is not deleted, even after the managed cluster is removed from the placement cluster decision list.

You can work around the issue by completing the following steps:

1. Navigate to the managed cluster.
2. Run the following command to get the orphan **AppliedManifestWork**:

   ```
   oc get appsub -n <appsub NS> <appsub Name> -o yaml
   ```

   The output might resemble the following:

   ```
     ownerReferences:
     - apiVersion: work.open-cluster-management.io/v1
       kind: AppliedManifestWork
       name: 6e01d06846c6ca2ac4ed6c9b0841e720af2de12a171108768f42285d7f873585-test-
   appsub-1-ns-git-app-1
       uid: f69fe90b-7f5f-483a-86b2-dcd5e041321a
   ```

3. Run the following command to delete the orphan **AppliedManifestWork**, which also deletes the application subscription:

   ```
   oc delete AppliedManifestWork
   6e01d06846c6ca2ac4ed6c9b0841e720af2de12a171108768f42285d7f873585-test-appsub-
   1-ns-git-app-1
   ```

1.2.6.22. ApplicationSet wizard does not fetch path automatically

After creating a new **ApplicationSet** with the same URL and branch as a previously created **ApplicationSet**, the ApplicationSet wizard does not fetch the path automatically.

To work around the issue, enter the path manually in the **Path** field.

1.2.7. Governance known issues

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

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

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

1.2.7.2. Gatekeeper operator installation fails

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

1.2.7.3. 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.

1.2.7.4. Operators deployed with policies do not support ARM

While installation into an ARM environment is supported, operators that are deployed with policies might not support ARM environments. The following policies that install operators do not support ARM environments:

- Red Hat Advanced Cluster Management policy for the Quay Container Security Operator
- Red Hat Advanced Cluster Management policy for the Compliance Operator

1.2.7.5. ConfigurationPolicy CRD is stuck in terminating

When you remove the **config-policy-controller** add-on from a managed cluster by disabling the policy controller in the **KlusterletAddonConfig** or by detaching the cluster, the **ConfigurationPolicy** CRD might get stuck in a terminating state. If the **ConfigurationPolicy** CRD is stuck in a terminating state, new policies might not be added to the cluster if the add-on is reinstalled later. You can also receive the following error:

> template-error; Failed to create policy template: create not allowed while custom resource definition is terminating

Use the following command to check if the CRD is stuck:

```
oc get crd configurationpolicies.policy.open-cluster-management.io -
o=jsonpath='{.metadata.deletionTimestamp}'
```

If a deletion timestamp is on the resource, the CRD is stuck. To resolve the issue, remove all finalizers from configuration policies that remain on the cluster. Use the following command on the managed cluster and replace **<cluster-namespace>** with the managed cluster namespace:

```
oc get configurationpolicy -n <cluster-namespace> -o name | xargs oc patch -n <cluster-namespace>
--type=merge -p '{"metadata":{"finalizers": []}}'
```

The configuration policy resources are automatically removed from the cluster and the CRD exits its terminating state. If the add-on has already been reinstalled, the CRD is recreated automatically without a deletion timestamp.

1.2.7.6. PruneObjectBehavior does not work when modifying existing configuration policy

When modifying an existing configuration policy, **DeleteAll** or **DeleteIfCreated** in the **pruneObjectBehavior** feature does not clean up old resources that were created before modifying. Only new resources from policy creations and policy updates are tracked and deleted when you delete the configuration policy.

1.2.7.7. Policy template issues

You might encounter the following issues when you edit policy templates for configuration policies:

- When you rename your configuration policy to a new name, a copy of the configuration policy with the older name remains.
- If you remove a configuration policy from a policy on your hub cluster, the configuration policy remains on your managed cluster but its status is not provided. To resolve this, disable your policy and reenable it. You can also delete the entire policy.

1.2.7.8. Pod security policies not supported on OpenShift 4.12 and later

The support of pod security policies is removed from OpenShift Container Platform 4.12 and later, and from Kubernetes v1.25 and later. If you apply a **PodSecurityPolicy** resource, you might receive the following non-compliant message:

```
violation - couldn't find mapping resource with kind PodSecurityPolicy, please check if you have CRD
deployed
```

1.2.8. Backup and restore known issues

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

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

1.2.8.2. Avoid backup collision

As hub clusters change from passive to primary clusters and back, different clusters can backup data at the same storage location. This can result in backup collisions, which means that the latest backups are generated by a passive hub cluster.

The passive hub cluster produces backups because the **BackupSchedule.cluster.open-cluster-management.io** resource is enabled on the hub cluster, but it should no longer write backup data since the hub cluster is no longer a primary hub cluster. Run the following command to check if there is a backup collision:

```
oc get backupschedule -A
```

You might receive the following status:

```
NAMESPACE       NAME              PHASE           MESSAGE
openshift-adp   schedule-hub-1    BackupCollision   Backup acm-resources-schedule-20220301234625, from cluster with id [be97a9eb-60b8-4511-805c-298e7c0898b3] is using the same storage location. This is a backup collision with current cluster [1f30bfe5-0588-441c-889e-eaf0ae55f941] backup. Review and resolve the collision then create a new BackupSchedule resource to  resume backups from this cluster.
```

Avoid backup collisions by setting the **BackupSchedule.cluster.open-cluster-management.io** resource **status** to **BackupCollision**. The **Schedule.velero.io** resources that are created by the **BackupSchedule** resource are automatically deleted.

The backup collision is reported by the **hub-backup-pod** policy. The administrator must verify which hub cluster writes data to the storage location. Then remove the **BackupSchedule.cluster.open-cluster-management.io** resource from the passive hub cluster, and recreate a new **BackupSchedule.cluster.open-cluster-management.io** resource on the primary hub cluster to resume the backup.

See Enabling the backup and restore operator for more information.

1.2.8.3. Velero restore limitations

View the following restore limitations:

- The new hub cluster is not identical to the initial hub cluster, where the data is restored, when there is an existing policy on the new hub cluster before the backup data is restored on the initial hub cluster. The policy should not be running on the new hub cluster since this is a policy that is unavailable with the backup resources.
- Since Velero skips existing resources, the policy on the new hub cluster is unchanged. Therefore, the policy is not the same as the one backed up on the initial hub cluster.
- The new hub cluster has a different configuration from the active hub cluster when a user reapplies the backup on the new hub cluster. Since there is an existing policy on the hub cluster from a previous restore, it is not restored again. Even when the backup contains the expected updates, the policy contents are not updated by Velero on the new hub cluster.

To address the previously mentioned limitations, when a **restore.cluster.open-cluster-management.io** resource is created, the cluster backup and restore operator runs a set of steps to prepare for restore by cleaning the hub cluster before Velero restore begins.

For more information, see *Clean the hub cluster before restore* in the Enabling the backup and restore operator topic.

1.2.8.4. Imported managed clusters are not displayed

Managed clusters that are manually imported on the primary hub cluster show only when the activation data is restored on the passive hub cluster.

1.2.8.5. Cluster backup and restore upgrade limitation

If you upgrade your cluster from 2.5 to 2.6 with the **enableClusterBackup** parameter set to **true**, the following message appears:

```
When upgrading from version 2.4 to 2.5, cluster backup must be disabled
```

Before you upgrade your cluster, disable cluster backup and restore by setting the **enableClusterBackup** parameter to **false**. The **components** section in your **MultiClusterHub** resource might resemble the following YAML file:

You can reenable the backup and restore component when the upgrade is complete. View the following sample:

```
overrides:
    components:
      - enabled: true
        name: multiclusterhub-repo
      - enabled: true
        name: search
      - enabled: true
        name: management-ingress
      - enabled: true
        name: console
      - enabled: true
        name: insights
      - enabled: true
        name: grc
      - enabled: true
        name: cluster-lifecycle
      - enabled: true
        name: volsync
      - enabled: true
        name: multicluster-engine
      - enabled: false
        name: cluster-proxy-addon
      - enabled: true
        name: cluster-backup
  separateCertificateManagement: false
```

If you have manually installed OADP, you must manually uninstall OADP before you upgrade. After the upgrade is successful and backup and restore is reenabled, OADP is installed automatically.

1.2.8.6. Managed cluster resource not restored

When you restore the settings for the **local-cluster** managed cluster resource and overwrite the **local-cluster** data on a new hub cluster, the settings are misconfigured. Content from the previous hub cluster **local-cluster** is not backed up because the resource contains **local-cluster** specific information, such as the cluster URL details.

You must manually apply any configuration changes that are related to the **local-cluster** resource on the restored cluster. See *Prepare the new hub cluster* in the Enabling the backup and restore operator topic.

1.2.8.7. Restored Hive managed clusters might not be able to connect with the new hub cluster

When you restore the backup of the changed or rotated certificate of authority (CA) for the Hive managed cluster, on a new hub cluster, the managed cluster fails to connect to the new hub cluster. The connection fails because the **admin kubeconfig** secret for this managed cluster, available with the backup, is no longer valid.

You must manually update the restored **admin kubeconfig** secret of the managed cluster on the new hub cluster.

1.2.8.8. Creating DataProtectionApplication resource causes error

When creating the **DataProtectionApplication** resource in OADP 1.0, the resource status might create an error message resembling the following:

```
Route.route.openshift.io "oadp-dpa-sample-1-aws-registry-route" is invalid: spec.host: Invalid value: "oadp-dpa-sample-1-aws-registry-route-open-cluster-management-backup.dns.name.here": must be no more than 63 characters
```

To fix the issue, set the **backupImages** parameter to **false**. See the following example:

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: oadp-dpa-sample-1
  namespace: open-cluster-management-backup
spec:
  backupImages: false
  backupLocations:
```

1.2.9. Submariner known issues

1.2.9.1. Only OpenShift SDN is supported as a CNI network provider when using Globalnet

You can use both OpenShift SDN and OVN Kubernetes CNI networks with Submariner, unless you are using Globalnet. Only OpenShift SDN is supported when you use Globalnet.

1.2.9.2. Some Red Hat Enterprise Linux nodes are not supported as worker nodes

When deploying Submariner on a cluster that includes Red Hat Enterprise Linux worker nodes with the kernel version between 4.18.0-359.el8.x86_64 and 4.18.0-372.11.1.el8_6.x86_64, application workloads fail to communicate with remote clusters.

1.2.9.3. Not all of the infrastructure providers that Red Hat Advanced Cluster Management can manage are supported

Submariner is not supported with all of the infrastructure providers that Red Hat Advanced Cluster Management can manage. Refer to the Red Hat Advanced Cluster Management support matrix for a list of supported providers.

1.2.9.4. Preparing the Red Hat OpenStack Platform infrastructure from the Red Hat Advanced Cluster Management console is not supported

Automatic cloud preparation for Red Hat OpenStack clusters is not supported for Submariner from the product-title-short} console. You can use the Red Hat Advanced Cluster Management APIs to prepare the clouds manually.

1.2.9.5. Headless services with Globalnet is not supported in some cases

You can use headless services with Globalnet, except when you access the exported headless service from a client that resides in the same cluster by using the **clusterset.local** domain name. When you use the **clusterset.local** domain name to access the headless service, the **globalIP** that is associated with the headless service is not routable in the cluster and is returned to the client.

You can use the **cluster.local** domain name to access the local headless services.

1.2.9.6. Air-gapped clusters are not supported

Submariner is not validated for clusters that are provisioned in an air-gapped environment.

1.2.9.7. Numerous gateways cannot be deployed

You cannot deploy multiple gateways.

1.2.9.8. Deployments that use VXLAN when NAT is enabled are not supported

Only non-NAT deployments support Submariner deployments with the VXLAN cable driver.

1.2.9.9. OVN Kubernetes support limitations

Using the OVN Kubernetes CNI network provider requires Red Hat OpenShift 4.11 or later. OVN Kubernetes does not support Globalnet.

1.2.9.10. Globalnet limitations

Globalnet is not supported with Red Hat OpenShift Data Foundation disaster recovery solutions. Make sure to use a non-overlapping range of private IP addresses for the cluster and service networks in each cluster for regional disaster recovery scenarios.

1.2.9.11. Microsoft Azure cluster set name length limitation

The length of a Microsoft Azure managed cluster set name must be 20 characters or fewer.

1.2.9.12. Red Hat OpenShift Container Platform 4.12 not supported on Microsoft Azure

OpenShift Container Platform 4.12 on Microsoft Azure is not supported due to a gateway node issue.

1.3. ERRATA UPDATES

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

Important: For reference, Errata links and GitHub numbers might be added to the content and used internally. Links that require access might not be available for the user.

FIPS notice: If you do not specify your own ciphers in **spec.ingress.sslCiphers**, then the **multiclusterhub-operator** provides a default list of ciphers. For 2.4, this list includes two ciphers that are *not* FIPS approved. If you upgrade from a version 2.4.x or earlier and want FIPS compliance, remove the following two ciphers from the **multiclusterhub** resource: **ECDHE-ECDSA-CHACHA20-POLY1305** and **ECDHE-RSA-CHACHA20-POLY1305**.

1.3.1. Errata 2.6.8

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

1.3.2. Errata 2.6.7

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

1.3.3. Errata 2.6.6

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

1.3.4. Errata 2.6.5

- Reduces the default number of images displayed in the search drop-down to 2500 to increase performance. (ACM-2800)
- The **must-gather** command now collects the Red Hat OpenShift Container Platform version number. (ACM-2857)
- Fixes an issue that caused the **max_item_size** setting in the **MEMCACHED** index to not propogate changes to all **MEMCACHED** clients. (ACM-4684)

1.3.5. Errata 2.6.4

- Fixes an issue that causes **ClusterCurator** upgrades to time out by adding the **spec.upgrade.monitorTimeout** setting to the **ClusterCurator** API. (ACM-2024)
- Makes **ClusterCurator** usable with GitOps tools such as ArgoCD by avoiding **spec** changes in the **ClusterCurator** custom resource after an operation is completed. (ACM-2197)

1.3.6. Errata 2.6.3

- Fixes an issue that causes a service denial for all policies when adding a custom label with a specific key and value to a policy.
- Fixes an issue that caused applications to be unavailable after installation and partial configuration of ArgoCD.
- Fixes a bug that caused the continuous creation of **ClusterRoleBindings** for the **open-cluster-management-agent**. (Bugzilla #2134796)

1.3.7. Errata 2.6.2

- Fixes an issue that caused the **klusterlet-work-agent** to log a nil pointer panic on managed clusters. (Bugzilla #2041540)
- Fixes a bug that caused an **inform musthave ConfigurationPolicy** to incorrectly say certain **Role** or **ClusterRole** objects are non-compliant, and improves the **enforce** behavior. (Bugzilla #2041540)
- Updates the validation procedure of objects that are specified in a policy to avoid an infinite loop. (Bugzilla #2041540)

1.3.8. Errata 2.6.1

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

1.4. DEPRECATIONS AND REMOVALS

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

Important: The 2.4 and earlier versions of Red Hat Advanced Cluster Management are *removed* and no longer supported. The documentation might remain available, but is deprecated without any Errata or other updates available.

Best practice: Upgrade to the most recent version of Red Hat Advanced Cluster Management.

1.4.1. API deprecations and removals

Red Hat Advanced Cluster Management follows the Kubernetes deprecation guidelines for APIs. See the Kubernetes Deprecation Policy for more details about that policy. Red Hat Advanced Cluster Management APIs are only deprecated or removed outside of the following timelines:

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

1.4.1.1. API deprecations

Product or category	Affected item	Version	Recommended action	More details and links
BareMetalAsset	The BareMetalAsset **v1alpha1** API is deprecated.	2.6	Do not use this API.	None

Product or category	Affected item	Version	Recommended action	More details and links
Discovery	The DiscoveredCluster and DiscoveryConfig **v1alpha1** APIs are deprecated. Discovery API is upgraded to **V1**.	2.5	Use **V1**.	None
Placements	The **v1alpha1** API is upgraded to **v1beta1** because **v1alpha1** is deprecated.	2.5	Use **v1beta1**.	The field **spec.prioritizerPolicy.configurations.name** in **Placement** API **v1alpha1** is removed. Use **spec.prioritizerPolicy.configurations.scoreCoordinate.builtIn** in **v1beta1**.
PlacementDecisions	The **v1alpha1** API is upgraded to **v1beta1** because **v1alpha1** is deprecated.	2.5	Use **v1beta1**.	None
Applications	The **v1alpha1** API is removed completely. GitOps clusters API is upgraded to **V1beta1**.	2.5	Use **V1beta1**.	None
Applications	**deployables.apps.open-cluster-management.io**	2.5	None	The deployable API remains just for upgrade path. Any deployable CR create, update, or delete will not get reconciled.

Product or category	Affected item	Version	Recommended action	More details and links
ManagedClusterSets	The **v1alpha1** API is upgraded to **v1beta1** because **v1alpha1** is deprecated.	2.4	Use **v1beta1**.	None
ManagedClusterSetBindings	The **v1alpha1** API is upgraded to **v1beta1** because **v1alpha1** is deprecated.	2.4	Use **v1beta1**.	None

1.4.1.2. API removals

Product or category	Affected item	Version	Recommended action	More details and links
CertPolicyController	The **v1** API is deprecated.	2.6	Do not use this API.	CertPolicyController.agent.open-cluster-management.io
ApplicationManager	The **v1** API is deprecated.	2.6	Do not use this API.	ApplicationManager.agent.open-cluster-management.io
IAMPolicyController	The **v1** API is deprecated.	2.6	Do not use this API.	IAMPolicyController.agent.open-cluster-management.io
PolicyController	The **v1** API is deprecated.	2.6	Do not use this API.	PolicyController.agent.open-cluster-management.io
SearchCollector	The **v1** API is deprecated.	2.6	Do not use this API.	SearchCollector.agent.open-cluster-management.io
WorkManager	The **v1** API is deprecated.	2.6	Do not use this API.	WorkManager.agent.open-cluster-management.io

1.4.2. Red Hat Advanced Cluster Management deprecations

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

Product or category	Affected item	Version	Recommended action	More details and links
Console	Standalone web console	2.6	Use the integrated web console.	Launch from the perspective switcher. See Accessing your console for more information.
Observability	**data.custom_rules.yaml.groups.rules** is deprecated	2.5	Use **data.custom_rules.yaml.groups.recording_rules**.	See Customizing observability.
Installer	**enableClusterProxyAddon** and **enableClusterBackup** fields in **operator.open-cluster-management.io_multiclusterhubs_crd.yaml**	2.5	None	See Advanced Configuration for configuring install.
Applications	Managing secrets	2.4	Use policy hub templates for secrets instead.	See Manage security policies.
Governance console	**pod-security-policy**	2.4	None	None
Governance	Gatekeeper operator	2.6	Install with a subscription instead.	See Managing gatekeeper operator policies.

1.4.3. Removals

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

Product or category	Affected item	Version	Recommended action	More details and links
Clusters	ManifestWork	2.6	None	None

Product or category	Affected item	Version	Recommended action	More details and links
Clusters	Configuring a Red Hat Ansible job using labels	2.6	Configure the Red Hat Ansible job by using the console.	See Configuring an AnsibleJob template to run on a cluster by using the console for more information.
Clusters	Cluster creation using bare metal assets	2.6	Create an infrastructure environment with the console	See Creating a cluster in an on-premises environment for the proceding process.
Add-on operator	Installation of built-in managed cluster add-ons	2.6	None	None
Applications	Deployable controller	2.5	None	The Deployable controller removed.
Red Hat Advanced Cluster Management console	Visual Web Terminal (Technology Preview)	2.4	Use the terminal instead	None
Governance	Custom policy controller	2.6	No action is required	None
Governance	The unused **LabelSelector** parameter is removed from the configuration policy.	2.6	None	See the Kubernetes configuration policy controller documentation.

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 Secrets in the Kubernetes documentation.

1.5.8. Data access

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

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

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

1.5.8.1. Authentication

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

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

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

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

1.5.8.2. Role Mapping

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

1.5.8.3. Authorization

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

1.5.8.4. Pod Security

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

1.5.9. Data Processing

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

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

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

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

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

1.5.10. Data Deletion

Red Hat Advanced Cluster Management for Kubernetes platform provides commands, application programming interfaces (APIs), and user interface actions to delete data that is created or collected by the product. These functions enable users to delete technical data, such as service user IDs and

passwords, IP addresses, Kubernetes node names, or any other platform configuration data, as well as information about users who manage the platform.

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

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

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

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

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

1.5.11. Capability for Restricting Use of Personal Data

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

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

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

1.5.12. Appendix

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

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

1.6. FIPS READINESS

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

If you plan to manage clusters with FIPS enabled, you must have a FIPS-ready hub cluster because cryptography that is created on the hub cluster is stored on the managed cluster. Enable FIPS with the **fips: true** setting when you provision your OpenShift Container Platform managed cluster. You cannot enable FIPS after you provision your cluster.

1.6.1. Limitations

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

- Red Hat OpenShift Container Platform only supports FIPS on the **x86_64** architecture.
- Integrity Shield is a Technology Preview component that is not FIPS ready.
- Persistent Volume Claim (PVC) and S3 storage that is used by the search and observability components must be encrypted when you configure the provided storage. Red Hat Advanced Cluster Management does not provide storage encryption, see the OpenShift Container Platform documentation, Support for FIPS cryptography .
- When you provision managed clusters using the Red Hat Advanced Cluster Management console, select the following check box in the *Cluster details* section of the managed cluster creation to enable the FIPS standards:

  > FIPS with information text: Use the Federal Information Processing Standards (FIPS) modules provided with Red Hat Enterprise Linux CoreOS instead of the default Kubernetes cryptography suite file before you deploy the new managed cluster.