Red Hat Advanced Cluster Management for Kubernetes 2.10

Release notes

Release notes

Last Updated: 2024-04-15

Red Hat Advanced Cluster Management for Kubernetes 2.10 Release notes

Release notes

Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

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 2.6 and earlier versions of Red Hat Advanced Cluster Management are *removed* from service, and are no longer supported. Documentation for versions 2.6 and earlier is not updated. 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
- Observability support

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.

Access the Support matrix to learn about hub cluster and managed cluster requirements and support.

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

- Clusters
- multicluster global hub
- Applications
- Observability
- Governance
- Backup and restore
- Networking

1.1.1. Cluster

Cluster lifecycle components and features are within the multicluster engine operator, which is a

software operator that enhances cluster fleet management. The multicluster engine operator supports OpenShift Container Platform and Kubernetes cluster lifecycle management across clouds and data centers. OpenShift Container Platform is a prerequisite for this technology.

- The documentation for multicluster engine operator (cluster) is found within the Cluster Lifecycle section of the product documentation.
- View What's new for multicluster engine operator 2.5 from *Cluster Lifecycle*.
- View tasks and support information at Cluster lifecycle overview.

1.1.2. multicluster global hub

You can use multicluster global hub with Red Hat Advanced Cluster Management backup and restore features. These features give you access to recovery solutions and basic resources. For more information, see Backup for multicluster global hub .

For other multicluster global hub topics, see multicluster global hub .

1.1.3. Applications

With the new **.status.subscription** field, you can see the overall subscription status instead of package status of only an individual package.

For other Application topics, see Managing applications.

1.1.4. Observability

- The hub collector metrics are now always collected and sent to the Red Hat Advanced Cluster Management Thanos instance. When you enable Observability, the service starts an **endpoint-operator** and **metrics-collector** pod in the **open-cluster-management-observability** namespace on your hub cluster. The **MultiClusterObservability** operator launches and manages the **endpoint-operator** and **metrics-collector** pods. The Observability add-on does not control the pods anymore. See Observability architecture to learn more.
- You can use Grafana dashboards to view your hosted control planes cluster capacity estimate, and existing hosted control planes resource utilizations. Hosted control plane observability is part of the cluster lifecycle, or multicluster engine operator,and Red Hat Advanced Cluster Management integration that you can see from Red Hat Advanced Cluster Management integration.

See Observability service introduction .

1.1.5. Governance

- **Technology Preview** Enable the policy compliance history API to store and query compliance history events for your hub cluster. See Policy compliance history API (Technology Preview) . To enable the API see, Policy compliance history (Technology Preview) .
- Configure the operations of the Gatekeeper operator webhook to manage admission events. See Managing Gatekeeper operator policies for details.
- Enable the Policy Generator to process Helm charts and add descriptions to policies. See the **policyDefaults.policyLabels** and **policies.policyLabels** optional specifications and more at Policy Generator configuration reference table .

- You can enable *diff logging* for **ConfigurationPolicy** resources by using the **recordDiff** parameter in the **ConfigurationPolicy** resource. The difference between the **object-template** and the object on the managed cluster is logged inside the **config-policy-controller** pod on the managed cluster. See Configure debug log for details.

- Enable the Policy Generator to process Helm charts and add descriptions to policies. See the Policy Generator configuration reference table for more details.

- You can now configure the concurrency of the governance framework. See Policy controller advanced configuration for more details.

- The Gatekeeper operator exposes a setting in the custom resource definition within the **auditFromCache** audit, which is disabled by default. You can enable **auditFromCache**, then set **config.gatekeeper.sh** for the sync details. See Managing Gatekeeper operator policies for details.

- Enable the **auditEventsInvolvedNamespace** to manage which namespace audit event to create, and the **admissionEventsInvolvedNamespace** to to manage which namespace admission event to create. See Managing Gatekeeper operator policies.

- **Technology Preview:** You can monitor and install Operator Lifecycle Manager (OLM) operators across your clusters by using the operator policy controller. See Operator policy controller (Technology Preview) for more information.

- **Technology preview:** Create and apply an **OperatorPolicy** resource to install OLM operators. See Installing an operator by using the *OperatorPolicy* resource (Technology Preview).

- Use **Placement** resources to define where you want your policies to be placed. See, Policy overview for more details.

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

1.1.6. Backup and restore

- The **backup-restore-enabled** policy includes a new template named, **OADP-channel**. Use the **OADP-channel** template to prevent your backup and restore operator from running with the wrong custom resource definitions. For more details, see Validating your backup or restore configurations.

- When you enable the backup component on the **MultiClusterHub**, the cluster backup and restore operator Helm chart installs policies. The new **backup-restore-auto-import** informs you about issues with the automatic managed clusters import feature. For more details, see Validating your backup or restore configurations.

See Backup and restore to learn about disaster recovery solutions for your hub cluster.

1.1.7. Networking

- You can deploy Submariner on IBM Power Systems Virtual Server. See Deploying Submariner by using the console to learn more.

- **Technology Preview:** You can also now deploy Submariner on Red Hat OpenShift on IBM Cloud. See Deploying Submariner by using the console to learn more.

See Networking.

1.1.8. Learn more about this release

- Get an overview of Red Hat Advanced Cluster Management for Kubernetes from Welcome to Red Hat Advanced Cluster Management for Kubernetes.
- See more release notes, such as *Known Issues and Limitations* in the Red Hat Advanced Cluster Management Release notes.
- See the Multicluster architecture topic to learn more about major components of the product.
- See support information and more in the Red Hat Advanced Cluster Management Troubleshooting guide.
- Access the open source *Open Cluster Management* repository for interaction, growth, and contributions from the open community. To get involved, see open-cluster-management.io. Visit the GitHub repository for more information.

1.2. ERRATA UPDATES

By default, Errata updates are automatically applied when released. The details are published here when the release is available.

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

See Upgrading by using the operator for more information about upgrades.

1.2.1. Errata 2.10.1

- Fixes a problem that might occur for users who use the Red Hat Advanced Cluster Management for Kubernetes backup and recovery function and backed up the **managedcluster** namespace without using the **cluster.open-cluster-management.io/backup: cluster-activation** label. The problem caused the managed cluster namespace to remain in the **Terminating** state after it is restored. (ACM-9780)
- Fixes the issue where a policy might temporarily be set to **noncompliant** with a message of **context cancelled** when the policy was being updated, while the **governance-policy-framework** pod was shutting down on the managed cluster. (ACM-10402)
- Fixes a problem that sometimes caused the console to briefly show a newly created policy as not found before refreshing to the policy details. (ACM-10416)
- Delivers updates to one or more product container images.

1.3. KNOWN ISSUES

Review the known issues for application management. 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.

For more about deprecations and removals, see Deprecations and removals.

Cluster management or *cluster lifecycle* is provided by the multicluster engine operator with or without Red Hat Advanced Cluster Management. See the following known issues and limitations for cluster

management that apply to Red Hat Advanced Cluster Management only. Most cluster management known issues are located in the cluster lifecyle documentation at cluster lifecyle known issues .

- Installation known issues
- Business continuity known issues
- Console known issues
- Application known issues
- Observability known issues
- Governance known issues
- Networking known issues

1.3.1. Installation known issues

Review the known issues for installation. 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.

For more about deprecations and removals, see Deprecations and removals.

1.3.1.1. Infrastructure operator error with ARM converged flow

When you install the **infrastructure-operator**, converged flow with ARM does not work. Set **ALLOW_CONVERGED_FLOW** to **false** to resolve this issue.

1. Run the following command to create a **ConfigMap** resource:

```
oc create -f
```

2. Apply your file by running **oc apply -f**. See the following file sample with **ALLOW_CONVERGED_FLOW** set to **false**:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-assisted-service-config
  namespace: assisted-installer
data:
  ALLOW_CONVERGED_FLOW: false
```

3. Annotate the **agentserviceconfig** with the following command:

```
oc annotate --overwrite AgentServiceConfig agent unsupported.agent-
install.openshift.io/assisted-service-configmap=my-assisted-service-config
```

The agent appears in the inventory when the issue is resolved.

1.3.1.2. 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
```

1.3.1.3. 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.

1.3.1.4. OpenShift Container Platform cluster upgrade failed status

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

1.3.1.5. 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.

1.3.2. Business continuity 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.

For more about deprecations and removals, see Deprecations and removals.

1.3.2.1. Backup and restore known issues

Backup and restore known issues and limitations are listed here, along with workarounds if they are available.

1.3.2.1.1. The *open-cluster-management-backup* namespace is stuck in the*Terminating* state

When the cluster-backup component is disabled on the **MultiClusterHub** resource, the **open-cluster-management-backup** namespace is stuck in the **Terminating** state if you have a Velero restore resource created by a Red Hat Advanced Cluster Management restore operation.

The **Terminating** state is a result of the Velero restore resources waiting on the **restores.velero.io/external-resources-finalizer** to complete. To workaround this issue, complete the following steps:

1. Delete all Red Hat Advanced Cluster Management restore resources and wait for the Velero restore to be cleaned up before you disable the cluster backup option on the **MultiClusterHub** resource.

2. If your **open-cluster-management-backup** namespace is already stuck in the **Terminating** state, edit all the Velero restore resources and remove the finalizers.

3. Allow the Velero resources to delete the namespaces and resources.

1.3.2.1.2. Bare metal managed clusters deployed with the Infrastructure Operator by using the ZTP flow perform reinstall nodes

If the resources for the bare metal cluster are backed up and restored to a secondary hub cluster by using the Red Hat Advanced Cluster Management back up and restore feature, the managed cluster reinstalls on the nodes, which destroys the existing managed cluster.

Note: This only affects bare metal clusters that were deployed by using zero touch provisioning, meaning that they have **BareMetalHost** resources that manage powering on and off bare metal nodes and attaching virtual media for booting.

If a **BareMetalHost** resource was not used in the deployment of the managed cluster, there is no negative impact.

To work around this issue, exclude managed **BareMetalHost** resources on the primary hub cluster from performing back up and restore to the secondary hub cluster.

Add the following label to the **BareMetalHost** resources on the primary hub cluster: **velero.io/exclude-from-backup: "true"**.

This label excludes any resource from the back up procedure.

When you exclude the **BareMetalHost** resource from restore, removing a cluster by using zero touch provisioning does not fully function because **BareMetalHost** manages power for the bare metal nodes.

1.3.2.1.3. *BackupSchedule* shows a *FailedValidation* status when using OADP 1.1.2, or later

After you enable the Red Hat Advanced Cluster Management backup and restore component and successfully create a **DataProtectionApplication** resource, a **BackupStorageLocation** resource is created with a status of **Available**. When you are using OADP version 1.1.2 or later, you might receive the following message after you create a **BackupSchedule** resource and the status is **FailedValidation**:

```
oc get backupschedule -n open-cluster-management-backup
NAME PHASE MESSAGE
rosa-backup-schedule FailedValidation Backup storage location is not available. Check
velero.io.BackupStorageLocation and validate storage credentials.
```

The error is caused by a missing value for **ownerReference** in the **BackupStorageLocation** resource. The value of the **DataProtectionApplication** resource should be used as the value of the **ownerReference**.

To work around the problem, manually add the **ownerReference** to the **BackupStorageLocation**:

1. Open the **oadp-operator.v1.1.2** file by running the following command:

   ```
   oc edit csv -n open-cluster-management-backup oadp-operator.v1.1.2
   ```

2. Edit the value of **spec.deployments.label.spec.replicas** by replacing the **1** with a **0** in the OADP operator CSV.

3. Patch the **ownerReference** annotations in the YAML script as shown in the following example:

   ```
   metadata:
   resourceVersion: '273482'
   name: dpa-sample-1
   uid: 4701599a-cdf5-48ac-9264-695a95b935a0
   namespace: open-cluster-management-backup
   ownerReferences: <<

   apiVersion: oadp.openshift.io/v1alpha1
   blockOwnerDeletion: true
   controller: true
   kind: DataProtectionApplication
   name: dpa-sample
   uid: 52acd151-52fd-440a-a846-95a0d7368ff7
   ```

4. Change the value of **spec.deployments.label.spec.replicas** back to **1** to start the data protection application process with the new settings.

1.3.2.1.4. Velero restore limitations

A new hub cluster can have a different configuration than the active hub cluster if the new hub cluster, where the data is restored, has user-created resources. For example, this can include an existing policy that was created on the new hub cluster before the backup data is restored on the new hub cluster.

Velero skips existing resources if they are not part of the restored backup, so the policy on the new hub cluster remains unchanged, resulting in a different configuration between the new hub cluster and active hub cluster.

To address this limitation, the cluster backup and restore operator runs a post restore operation to clean up the resources created by the user or a different restore operation when a **restore.cluster.open-cluster-management.io** resource is created.

For more information, see the Installing the backup and restore operator topic.

1.3.2.1.5. Passive configurations do not display managed clusters

Managed clusters are only displayed when the activation data is restored on the passive hub cluster.

1.3.2.1.6. 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 Installing the backup and restore operator topic.

1.3.2.1.7. 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.

1.3.2.1.8. Imported managed clusters show a*Pending Import* status

Managed clusters that are manually imported on the primary hub cluster show a **Pending Import** status when the activation data is restored on the passive hub cluster. For more information, see Connecting clusters by using a Managed Service Account.

1.3.2.1.9. The *appliedmanifestwork* is not removed from managed clusters after restoring the hub cluster

When the hub cluster data is restored on the new hub cluster, the **appliedmanifestwork** is not removed from managed clusters that have a placement rule for an application subscription that is not a fixed cluster set.

See the following example of a placement rule for an application subscription that is not a fixed cluster set:

```
spec:
  clusterReplicas: 1
  clusterSelector:
```

```
    matchLabels:
      environment: dev
```

As a result, the application is orphaned when the managed cluster is detached from the restored hub cluster.

To avoid the issue, specify a fixed cluster set in the placement rule. See the following example:

```
spec:
  clusterSelector:
    matchLabels:
      environment: dev
```

You can also delete the remaining **appliedmanifestwork** manually by running the folowing command:

```
oc delete appliedmanifestwork <the-left-appliedmanifestwork-name>
```

1.3.2.1.10. The *appliedmanifestwork* not removed and *agentID* is missing in the specification

When you are using Red Hat Advanced Cluster Management 2.6 as your primary hub cluster, but your restore hub cluster is on version 2.7 or later, the **agentID** is missing in the specification of **appliedmanifestworks** because the field is introduced in the 2.7 release. This results in the extra **appliedmanifestworks** for the primary hub on the managed cluster.

To avoid the issue, upgrade the primary hub cluster to Red Hat Advanced Cluster Management 2.7, then restore the backup on a new hub cluster.

Fix the managed clusters by setting the **spec.agentID** manually for each **appliedmanifestwork**.

1. Run the following command to get the **agentID**:

   ```
   oc get klusterlet klusterlet -o jsonpath='{.metadata.uid}'
   ```

2. Run the following command to set the **spec.agentID** for each **appliedmanifestwork**:

   ```
   oc patch appliedmanifestwork <appliedmanifestwork_name> --type=merge -p '{"spec":
   {"agentID": "'$AGENT_ID'"}}'
   ```

1.3.2.1.11. The *managed-serviceaccount* add-on status shows *Unknown*

The managed cluster **appliedmanifestwork addon-managed-serviceaccount-deploy** is removed from the imported managed cluster if you are using the Managed Service Account without enabling it on the multicluster engine for Kubernetes operator resource of the new hub cluster.

The managed cluster is still imported to the new hub cluster, but the **managed-serviceaccount** add-on status shows **Unknown**.

You can recover the **managed-serviceaccount** add-on after enabling the Managed Service Account in the multicluster engine operator resource. See Enabling automatic import to learn how to enable the Managed Service Account.

1.3.3. Console known issues

Review the known issues for the console. 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.

For more about deprecations and removals, see Deprecations and removals.

1.3.3.1. Cannot upgrade OpenShift Dedicated in console

From the console you can request an upgrade for OpenShift Dedicated clusters, but the upgrade fails with the **Cannot upgrade non openshift cluster** error message. Currently there is no workaround.

1.3.3.2. Search PostgreSQL pod is in CrashLoopBackoff state

The **search-postgres** pod is in **CrashLoopBackoff** state. If Red Hat Advanced Cluster Management is deployed in a cluster with nodes that have the **hugepages** parameter enabled and the **search-postgres** pod gets scheduled in these nodes, then the pod does not start.

Complete the following steps to increase the memory of the **search-postgres** pod:

1. Pause the **search-operator** pod with the following command:

```
oc annotate search search-v2-operator search-pause=true
```

2. Update the **search-postgres** deployment with a limit for the **hugepages** parameter. Run the following command to set the **hugepages** parameter to **512Mi**:

```
oc patch deployment search-postgres --type json -p '[{"op": "add", "path": "/spec/template/spec/containers/0/resources/limits/hugepages-2Mi", "value":"512Mi"}]'
```

3. Before you verify the memory usage for the pod, make sure your **search-postgres** pod is in the **Running** state. Run the following command:

```
oc get pod <your-postgres-pod-name>  -o jsonpath="Status: {.status.phase}"
```

4. Run the following command to verify the memory usage of the **search-postgres** pod:

```
oc get pod <your-postgres-pod-name> -o jsonpath='{.spec.containers[0].resources.limits.hugepages-2Mi}'
```

The following value appears, **512Mi**.

1.3.3.3. 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.

1.3.3.4. Horizontal scrolling does not work after provisioning hosted control plane cluster

After provisioning a hosted control plane cluster, you might not be able to scroll horizontally in the cluster overview of the Red Hat Advanced Cluster Management console if the **ClusterVersionUpgradeable** parameter is too long. You cannot view the hidden data as a result.

To work around the issue, zoom out by using your browser zoom controls, increase your Red Hat Advanced Cluster Management console window size, or copy and paste the text to a different location.

1.3.3.5. *EditApplicationSet* expand feature repeats

When you add multiple label expressions or attempt to enter your cluster selector for your **ApplicationSet**, you might receive the following message repeatedly, "Expand to enter expression". You can enter your cluster selection despite this issue.

1.3.4. Application known issues and limitations

Review the known issues for application management. 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.

For more about deprecations and removals, see Deprecations and removals.

See the following known issues for the Application lifecycle component.

1.3.4.1. Application Kubernetes Lease API missing for OpenShift Container Platform 3.11 managed clusters

The application add-on component uses the *Kubernetes Lease API* ,**leases.coordination.k8s.io**, which is missing for OpenShift Container Platform 3.11 users. The Kubernetes Lease API was introduced in Kubernetes 1.14, but OpenShift Container Platform 3.11 bundles Kubernetes version 1.11.

To resolve this issue, manually apply the following Kubernetes Lease API **CustomResourceDefinition** to the OpenShift Container Platform 3.11 managed cluster:

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: leases.coordination.k8s.io
spec:
  group: coordination.k8s.io
  names:
    kind: Lease
    listKind: LeaseList
    plural: leases
    singular: lease
    shortNames:
    - ls
  scope: Namespaced
```

```
versions:
- name: v1
  served: true    storage: true    schema:
    openAPIV3Schema:
      description: Lease defines a lease concept.
      type: object
      properties:
        apiVersion:
          type: string
        kind:
          type: string
        metadata:
          type: object
        spec:
          type: object
          properties:
            acquireTime:
              format: date-time
              type: string
            holderIdentity:
              type: string
            leaseDurationSeconds:
              format: int64
              type: integer
            leaseTransitions:
              format: int64
              type: integer
            renewTime:
              format: date-time
              type: string
          required:
          - holderIdentity
          - leaseDurationSeconds
          - renewTime
      required:
      - kind
      - metadata
      - spec
additionalPrinterColumns:
- JSONPath: .metadata.creationTimestamp
  name: Age
  type: date
subresources:
  status: {}
```

Note: Red Hat Advanced Cluster Management support of (OpenShift Container Platform 3.11 is deprecated.

1.3.4.2. Service account does not have automatic secrets

When you create a service account in Red Hat OpenShift Container Platform 4.15 provisioned by IBM VMware, the account does not automatically create a secret. Therefore, the Red Hat Advanced Cluster Management **gitopsCluster** controller fails to generate the managed cluster secret for the Argo CD push model. Red Hat OpenShift Container Platform 4.14 provisioned by IBM VMware and Red Hat OpenShift Container Platform 4.15 with any provider, do not observe this issue.

1.3.4.3. Editing subscription applications with *PlacementRule* does not display the subscription YAML in editor

After you create a subscription application that references a **PlacementRule** resource, the subscription YAML does not display in the YAML editor in the console. Use your terminal to edit your subscription YAML file.

1.3.4.4. Helm Chart with secret dependencies cannot be deployed by the Red Hat Advanced Cluster Management subscription

Using Helm Chart, you can define privacy data in a Kubernetes secret and refer to this secret within the **value.yaml** file of the Helm Chart.

The username and password are given by the referred Kubernetes secret resource **dbsecret**. For example, see the following sample **value.yaml** file:

```
credentials:
  secretName: dbsecret
  usernameSecretKey: username
  passwordSecretKey: password
```

The Helm Chart with secret dependencies is only supported in the Helm binary CLI. It is not supported in the operator SDK Helm library. The Red Hat Advanced Cluster Management subscription controller applies the operator SDK Helm library to install and upgrade the Helm Chart. Therefore, the Red Hat Advanced Cluster Management subscription cannot deploy the Helm Chart with secret dependencies.

1.3.4.5. Creating cluster secrets for Argo CD Push model is not supported

Customized cluster secrets cannot be created for the Argo CD Push model on your OpenShift Container Platform 3.11 managed clusters. This occurs because the managed service account add-on is not supported on OpenShift Container Platform 3.11 managed clusters.

1.3.4.6. Topology does not correctly display for Argo CD pull model **ApplicationSet** application

When you use the Argo CD pull model to deploy **ApplicationSet** applications and the application resource names are customized, the resource names might appear different for each cluster. When this happens, the topology does not display your application correctly.

1.3.4.7. Local cluster is excluded as a managed cluster for pull model

The hub cluster application set deploys to target managed clusters, but the local cluster, which is a managed hub cluster, is excluded as a target managed cluster.

As a result, if the Argo CD application is propagated to the local cluster by the Argo CD pull model, the local cluster Argo CD application is not cleaned up, even though the local cluster is removed from the placement decision of the Argo CD **ApplicationSet** resource.

To work around the issue and clean up the local cluster Argo CD application, remove the **skip-reconcile** annotation from the local cluster Argo CD application. See the following annotation:

```
annotations:
    argocd.argoproj.io/skip-reconcile: "true"
```

Additionally, if you manually refresh the pull model Argo CD application in the **Applications** section of the Argo CD console, the refresh is not processed and the **REFRESH** button in the Argo CD console is disabled.

To work around the issue, remove the **refresh** annotation from the Argo CD application. See the following annotation:

```
annotations:
   argocd.argoproj.io/refresh: normal
```

1.3.4.8. Argo CD controller and the propagation controller might reconcile simultaneously

Both the Argo CD controller and the propagation controller might reconcile on the same application resource and cause the duplicate instances of application deployment on the managed clusters, but from the different deployment models.

For deploying applications by using the pull model, the Argo CD controllers ignore these application resources when the Argo CD **argocd.argoproj.io/skip-reconcile** annotation is added to the template section of the **ApplicationSet**.

The **argocd.argoproj.io/skip-reconcile** annotation is only available in the GitOps operator version 1.9.0, or later. To prevent conflicts, wait until the hub cluster and all the managed clusters are upgraded to GitOps operator version 1.9.0 before implementing the pull model.

1.3.4.9. Resource fails to deploy

All the resources listed in the **MulticlusterApplicationSetReport** are actually deployed on the managed clusters. If a resource fails to deploy, the resource is not included in the resource list, but the cause is listed in the error message.

1.3.4.10. Resource allocation might take several minutes

For large environments with over 1000 managed clusters and Argo CD application sets that are deployed to hundreds of managed clusters, Argo CD application creation on the hub cluster might take several minutes. You can set the **requeueAfterSeconds** to **zero** in the **clusterDecisionResource** generator of the application set, as it is displayed in the following example file:

```
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: cm-allclusters-app-set
  namespace: openshift-gitops
spec:
  generators:
  - clusterDecisionResource:
      configMapRef: ocm-placement-generator
      labelSelector:
        matchLabels:
          cluster.open-cluster-management.io/placement: app-placement
      requeueAfterSeconds: 0
```

1.3.4.11. 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.

1.3.4.11.1. 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.

1.3.4.12. 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 overrided 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.

1.3.4.13. Policy resource not deployed unless by subscription administrator

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

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

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

1.3.4.14. 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 where **local-cluster: "true"** refers to your hub cluster:

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

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

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

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

1.3.4.15. 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.

1.3.4.16. 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 CR 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 steps:

1. Find the service account that is defined in the deployments. For example, see the following **nginx** deployments:

```
nginx-ingress-52edb
nginx-ingress-52edb-backend
```

2. Create an SCC CR in your namespace to assign the required permissions to the service account or accounts. See the following example, where **kind: SecurityContextConstraints** is added:

```
apiVersion: security.openshift.io/v1
 defaultAddCapabilities:
 kind: SecurityContextConstraints
 metadata:
   name: ingress-nginx
   namespace: ns-sub-1
 priority: null
 readOnlyRootFilesystem: false
 requiredDropCapabilities:
 fsGroup:
   type: RunAsAny
 runAsUser:
   type: RunAsAny
 seLinuxContext:
   type: RunAsAny
 users:
 - system:serviceaccount:my-operator:nginx-ingress-52edb
 - system:serviceaccount:my-operator:nginx-ingress-52edb-backend
```

1.3.4.17. Application channels require unique namespaces

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

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

1.3.4.18. 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.

1.3.4.19. Ansible Automation Platform operator access Ansible Automation Platform outside of a proxy

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

1.3.4.20. Application name requirements

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

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

1.3.4.21. 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.

1.3.4.22. No Application console topology filtering

The *Console* and *Topology* for *Application* changes for the 2.10. There is no filtering capability from the console Topology page.

1.3.4.23. 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.

1.3.5. Observability 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.

For more about deprecations and removals, see Deprecations and removals.

1.3.5.1. Observatorium API gateway pods in a restored hub cluster might have stale tenant data

The Observatorium API gateway pods in a restored hub cluster might contain stale tenant data after a backup and restore procedure because of a Kubernetes limitation. See Mounted ConfigMaps are updated automatically for more about the limitation.

As a result, the Observatorium API and Thanos gateway rejects metrics from collectors, and the Red Hat Advanced Cluster Management Grafana dashboards do not display data.

See the following errors from the Observatorium API gateway pod logs:

```
level=error name=observatorium caller=logchannel.go:129 msg="failed to forward metrics"
returncode="500 Internal Server Error" response="no matching hashring to handle tenant\n"
```

Thanos receives pods logs with the following errors:

```
caller=handler.go:551 level=error component=receive component=receive-handler tenant=xxxx
err="no matching hashring to handle tenant" msg="internal server error"
```

See the following procedure to resolve this issue:

1. Scale down the **observability-observatorium-api** deployment instances from **N** to **0**.
2. Scale up the **observability-observatorium-api** deployment instances from **0** to **N**.

Note: N = **2** by default, but might be greater than **2** in some custom configuration environments.

This restarts all Observatorium API gateway pods with the correct tenant information, and the data from collectors start displaying in Grafana in between 5-10 minutes.

1.3.5.2. Permission to add *PrometheusRules* and *ServiceMonitors* in *openshift-monitoring* namespace denied

Starting with Red Hat Advanced Cluster Management 2.9, you must use a label in your defined Red Hat Advanced Cluster Management hub cluster namespace. The label, **openshift.io/cluster-monitoring: "true"** causes the Cluster Monitoring Operator to scrape the namespace for metrics.

When Red Hat Advanced Cluster Management 2.9 is deployed or an installation is upgraded to 2.9, the Red Hat Advanced Cluster Management Observability **ServiceMonitors** and **PrometheusRule** resources are no longer present in the **openshift-monitoring** namespace.

1.3.5.3. Lack of support for proxy settings

The Prometheus **AdditionalAlertManagerConfig** resource of the observability add-on does not support proxy settings. You must disable the observability alert forwarding feature.

Complete the following steps to disable alert forwarding:

1. Go to the **MultiClusterObservability** resource.
2. Update the **mco-disabling-alerting** parameter value to **true**

The HTTPS proxy with a self-signed CA certificate is not supported.

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

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

1.3.5.5. Observability endpoint operator fails to pull image

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

For more information, see Enabling observability.

1.3.5.6. There is no data from ROKS clusters

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

1.3.5.7. There is no etcd data from ROKS clusters

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

1.3.5.8. Metrics are unavailable in the Grafana console

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

  "Annotation Query Failed"

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

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

  no project or cluster found

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

1.3.5.9. Prometheus data loss on managed clusters

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

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

1.3.5.10. Error ingesting out-of-order samples

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

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

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

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

1.3.5.11. 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
```

1.3.5.12. *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
```

1.3.5.13. Enabling *disableHubSelfManagement* causes empty list in Grafana dashboard

The Grafana dashboard shows an empty label list if the **disableHubSelfManagement** parameter is set to **true** in the **mulitclusterengine** custom resource. You must set the parameter to **false** or remove the parameter to see the label list. See disableHubSelfManagement for more details.

1.3.5.13.1. Endpoint URL cannot have fully qualified domain names (FQDN)

When you use the FQDN or protocol for the **endpoint** parameter, your observability pods are not enabled. The following error message is displayed:

```
Endpoint url cannot have fully qualified paths
```

Enter the URL without the protocol. Your **endpoint** value must resemble the following URL for your secrets:

```
endpoint: example.com:443
```

1.3.5.13.2. Grafana downsampled data mismatch

When you attempt to query historical data and there is a discrepancy between the calculated step value and downsampled data, the result is empty. For example, if the calculated step value is **5m** and the downsampled data is in a one-hour interval, data does not appear from Grafana.

This discrepancy occurs because a URL query parameter must be passed through the Thanos Query front-end data source. Afterwards, the URL query can perform additional queries for other downsampling levels when data is missing.

You must manually update the Thanos Query front-end data source configuration. Complete the following steps:

1. Go to the Query front-end data source.
2. To update your query parameters, click the *Misc* section.
3. From the *Custom query parameters* field, select **max_source_resolution=auto**.
4. To verify that the data is displayed, refresh your Grafana page.

Your query data appears from the Grafana dashboard.

1.3.5.14. Metrics collector does not detect proxy configuration

A proxy configuration in a managed cluster that you configure by using the **addonDeploymentConfig** is not detected by the metrics collector. As a workaround, you can enable the proxy by removing the managed cluster **ManifestWork**. Removing the **ManifestWork** forces the changes in the **addonDeploymentConfig** to be applied.

1.3.5.15. HTTPS proxy with a custom CA bundle is not supported

A proxy configuration in a managed cluster does not work when a custom CA bundle is required.

1.3.6. Governance known issues

Review the known issues for Governance. 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.

For more about deprecations and removals, see Deprecations and removals.

1.3.6.1. Container security operator is not available in OpenShift Container Platform 3.11

The container security operator is not available in OpenShift Container Platform 3.11. Therefore, you cannot take the policy template of **policy-imagemanifestvuln-sub** in the **ImageManifestVuln** policy and apply it for OpenShift Container Platform 3.11 clusters.

If you try to apply the **ImageManifestVuln** policy, you receive the following violation message:

> violation - couldn't find mapping resource with kind Subscription, please check if you have CRD deployed.

1.3.6.2. Governance resources not cleaned up properly when the component is disabled

Governance resources are not cleaned up properly. When the component is set to **false** or is disabled in the **MultiClusterHub** operator, the governance component is removed before it can clean up the add-ons that it manages.

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

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

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

1.3.6.4. 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 when 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.

1.3.6.5. 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

1.3.6.6. ConfigurationPolicy custom resource definition 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** custom resource definition might get stuck in a terminating state. If the **ConfigurationPolicy** custom resource

definition 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 custom resource definition is stuck:

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

If a deletion timestamp is on the resource, the custom resource definition 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 custom resource definition exits its terminating state. If the add-on has already been reinstalled, the custom resource definition is recreated automatically without a deletion timestamp.

1.3.6.7. *pruneObjectBehavior* does not work when modifying existing configuration policy

When you modify an existing configuration policy, **pruneObjectBehavior** does not work. View the following reasons why **pruneObjectBehavior** might not work:

- If you set **pruneObjectBehavior** to **DeleteAll** or **DeleteIfCreated** in a configuration policy, old resources that were created before modifying are not cleaned correctly. Only new resources from policy creations and policy updates are tracked and deleted when you delete the configuration policy.
- If you set **pruneObjectBehavior** to **None** or do not set the parameter value, old objects might be unintentionally deleted on the managed cluster. Specifically, this occurs when a user changes the **name**, **namespace**, **kind**, or **apiversion** in the template. The parameter fields can dynamically change when the **object-templates-raw** or **namespaceSelector** parameters change.

1.3.6.8. Policy status shows repeated updates when enforced

If a policy is set to **remediationAction: enforce** and is repeatedly updated, the Red Hat Advanced Cluster Management console shows repeated violations with successful updates. See the following two possible causes and solutions for the error:

- Another controller or process is also updating the object with different values.
  To resolve the issue, disable the policy and compare the differences between **objectDefinition** in the policy and the object on the managed cluster. If the values are different, another controller or process might be updating them. Check the **metadata** of the object to help identify why the values are different.
- The **objectDefinition** in the **ConfigurationPolicy** does not match because of Kubernetes processing the object when the policy is applied.
  To resolve the issue, disable the policy and compare the differences between **objectDefinition** in the policy and the object on the managed cluster. If the keys are different or missing,

Kubernetes might have processed the keys before applying them to the object, such as removing keys containing default or empty values.

1.3.6.9. Pod security policies not supported on OpenShift Container Platform 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
```

1.3.6.10. Duplicate policy template names create inconstistent results

When you create a policy with identical policy template names, you receive inconsistent results that are not detected, but you might not know the cause. For example, defining a policy with multiple configuration policies named **create-pod** causes inconsistent results. **Best practice:** Avoid using duplicate names for policy templates.

1.3.6.11. Governance deployments do not shut down without errors when disabled

When you disable governance deployments in the **MultiClusterHub** object, the deployments are not cleaned without errors. Complete the following steps to disable governance so that the deployments also get cleaned up:

1. Disable the **policyController** in the **KlusterletAddonConfig** for the managed cluster. If you do this for all managed clusters, run the following command:

```
for CLUSTER in $(oc get managedclusters -o jsonpath='{.items[].metadata.name}'); do
  oc patch -n ${CLUSTER} klusterletaddonconfig ${CLUSTER} --type=merge --
patch='{"spec":{"policyController":{"enabled":false}}}'
done
```

2. For local clusters only: Delete the **ManifestWork** for the local cluster and remove the finalizer on the **ManagedClusterAddon** if the **governance-policy-framework-uninstall** pod of a local cluster is in **CrashLoopBackOff**. Run the following commands:

```
oc delete manifestwork -n local-cluster -l open-cluster-management.io/addon-
name=governance-policy-framework
oc patch managedclusteraddon -n local-cluster governance-policy-framework --type=merge -
-patch='{"metadata":{"finalizers":[]}}'
```

3. Disable governance globally, if required, by setting the **grc** element in the **spec.overrides** section to **false** in the **MultiClusterHub** object. Run the following command:

```
oc edit multiclusterhub <name> -n <namespace>
```

4. For local clusters only: If there are any local cluster policies, you can delete the policies by running the following command:

```
oc delete policies -n local-cluster --all
```

5. To re-enable governance in the **KlusterletAddonConfig**, re-enable the **grc** element of the **spec.overrides** section in the **MultiClusterHub**. Run the following command:

```
for CLUSTER in $(oc get managedclusters -o jsonpath='{.items[].metadata.name}'); do
  oc patch -n ${CLUSTER} klusterletaddonconfig ${CLUSTER} --type=merge --
patch='{"spec":{"policyController":{"enabled":true}}}'
done
```

6. If the deployments are unsuccessful, the **governance-policy-addon-controller** might have a stale lease. Delete the lease by using the following command:

```
oc delete lease governance-policy-addon-controller-lock -n <namespace>
```

1.3.6.12. Database and policy compliance history API outage

There is built-in resilience for database and policy compliance history API outages, however, any compliance events that cannot be recorded by a managed cluster are queued in memory until they are successfully recorded. This means that if there is an outage and the **governance-policy-framework** pod on the managed cluster restarts, all queued compliance events are lost.

If you create or update a new policy during a database outage, any compliance events sent for this new policy cannot be recorded since the mapping of policies to database IDs cannot be updated. When the database is back online, the mapping is automatically updated and future compliance events from those policies are recorded.

1.3.6.13. PostgreSQL data loss

If there is data loss to the PostgreSQL server such as restoring to a backup without the latest data, the governance policy propagator on the {product-title-hsort} hub cluster must be restarted so that it can update the mapping of policies to database IDs. Until you restart the governance policy propagator, new compliance events associated with policies that once existed in the database are no longer recorded.

To restart the governance policy propagator, run the following command on the Red Hat Advanced Cluster Management hub cluster:

```
oc -n open-cluster-management rollout restart deployment/grc-policy-propagator
```

1.3.7. Known issues for networking

Review the known issues for Submariner. 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.

For more about deprecations and removals, see Deprecations and removals.

1.3.7.1. Submariner known issues

See the following known issues and limitations that might occur while using networking features.

1.3.7.1.1. Without *ClusterManagementAddon* submariner add-on fails

For versions 2.8 and earlier, when you install Red Hat Advanced Cluster Management, you also deploy

the **submariner-addon** component with the Operator Lifecycle Manager. If you did not create a **MultiClusterHub** custom resource, the **submariner-addon** pod sends an error and prevents the operator from installing.

The following notification occurs because the **ClusterManagementAddon** custom resource definition is missing:

> graceful termination failed, controllers failed with error: the server could not find the requested resource (post clustermanagementaddons.addon.open-cluster-management.io)

The **ClusterManagementAddon** resource is created by the **cluster-manager** deployment, however, this deployment becomes available when the **MultiClusterEngine** components are installed on the cluster.

If there is not a **MultiClusterEngine** resource that is already available on the cluster when the **MultiClusterHub** custom resource is created, the **MultiClusterHub** operator deploys the **MultiClusterEngine** instance and the operator that is required, which resolves the previous error.

1.3.7.1.2. Submariner add-on resources not cleaned up properly when managed clusters are imported

If the **submariner-addon** component is set to **false** within **MultiClusterHub** (MCH) operator, then the **submariner-addon** finalizers are not cleaned up properly for the managed cluster resources. Since the finalizers are not cleaned up properly, this prevents the **submariner-addon** component from being disabled within the hub cluster.

1.3.7.1.3. 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.

1.3.7.1.4. Submariner install plan limitation

The Submariner install plan does not follow the overall install plan settings. Therefore, the operator management screen cannot control the Submariner install plan. By default, Submariner install plans are applied automatically, and the Submariner addon is always updated to the latest available version corresponding to the installed Red Hat Advanced Cluster Management version. To change this behavior, you must use a customized Submariner subscription.

1.3.7.1.5. Limited headless services support

Service discovery is not supported for headless services without selectors when using Globalnet.

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

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

1.3.7.1.7. OVN Kubernetes requires OCP 4.11 and later

If you are using the OVN Kubernetes CNI network, you need Red Hat OpenShift 4.11 or later.

1.3.7.1.8. Self-signed certificates might prevent connection to broker

Self-signed certificates on the broker might prevent joined clusters from connecting to the broker. The connection fails with certificate validation errors. You can disable broker certificate validation by setting **InsecureBrokerConnection** to **true** in the relevant **SubmarinerConfig** object. See the following example:

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  insecureBrokerConnection: true
```

1.3.7.1.9. Submariner only supports OpenShift SDN or OVN Kubernetes

Submariner only supports Red Hat OpenShift Container Platform clusters that use the OpenShift SDN or the OVN-Kubernetes Container Network Interface (CNI) network provider.

1.3.7.1.10. Command limitation on Microsoft Azure clusters

The **subctl diagnose firewall inter-cluster** command does not work on Microsoft Azure clusters.

1.3.7.1.11. Automatic upgrade not working with custom *CatalogSource* or *Subscription*

Submariner is automatically upgraded when Red Hat Advanced Cluster Management for Kubernetes is upgraded. The automatic upgrade might fail if you are using a custom **CatalogSource** or **Subscription**.

To make sure automatic upgrades work when installing Submariner on managed clusters, you must set the **spec.subscriptionConfig.channel** field to **stable-0.15** in the **SubmarinerConfig** custom resource for each managed cluster.

1.3.7.1.12. Uninstall Submariner before removing *ManagedCluster* from a *ManageClusterSet*

If you remove a cluster from a **ClusterSet**, or move a cluster to a different **ClusterSet**, the Submariner installation is no longer valid.

You must uninstall Submariner before moving or removing a **ManagedCluster** from a **ManageClusterSet**. If you don't uninstall Submariner, you cannot uninstall or reinstall Submariner anymore and Submariner stops working on your **ManagedCluster**.

1.3.8. Multicluster global hub Operator known issues

Review the known issues for the multicluster global hub Operator. The following list contains known issues for this release, or known issues that continued from the previous release. For your OpenShift Container Platform cluster, see OpenShift Container Platform known issues .

1.3.8.1. Kafka operator keeps restarting

In the Federal Information Processing Standard (FIPS) environment, the Kafka operator keeps restarting because of the out-of-memory (OOM) state. To fix this issue, set the resource limit to at least **512M**. For detailed steps on how to set this limit, see amq stream doc.

1.3.8.2. Backup and restore known issues

If your original multicluster global hub cluster crashes, the multicluster global hub loses its generated events and **cron** jobs. Even if you restore the new multicluster global hub cluster, the events and **cron** jobs are not restored. To workaround this issue, you can manually run the **cron** job, see Running the summarization process manually.

1.3.8.3. Managed cluster displays but is not counted

A managed cluster that is not created successfully, meaning **clusterclaim id.k8s.io** does not exist in the managed cluster, is not counted in the policy compliance dashboards, but shows in the policy console.

1.3.8.4. The multicluster global hub is installed on OpenShift Container Platform 4.13 hyperlinks might redirect home

If the multicluster global hub Operator is installed on OpenShift Container Platform 4.13, all hyperlinks that link to the managed clusters list and detail pages in dashboards might redirect to the Red Hat Advanced Cluster Management home page.

You need to manually go to your target page.

1.3.8.5. The standard group filter cannot pass to the new page

In the **Global Hub Policy Group Compliancy Overview**hub dashboards, you can check one data point by clicking **View Offending Policies for standard group**, but after you click this link to go to the offending page, the standard group filter cannot pass to the new page.

This is also an issue for the **Cluster Group Compliancy Overview**.

1.3.8.6. Cannot redirect to OpenShift Container Platform 3.11 cluster *Observability* page

If a managed hub cluster imports an OpenShift Container Platform 3.11 cluster (deprecated) as managed cluster, it cannot redirect to the *Observability* page in the **Global Hub** > **Overview** dashboard.

You need to manually navigate to your target page.

1.3.8.7. Compliance *cron* job error

Use the compliance **cron** job to view a summary of the daily job status of the policy by compliance and event tables. When you reimport the managed hub cluster to the multicluster global hub cluster with a different name, this reimport might cause two events from the same policy but with different managed hub cluster names. These different events cause the daily compliance **cron** job status to run the error, **pq: ON CONFLICT DO UPDATE command cannot affect row a second time**.

To work around this error, complete the following steps:

1. Restart the managed hub cluster.
2. Restore the compliance history data for yesterday by running the following SQL:

```
INSERT INTO history.local_compliance (policy_id, cluster_id, leaf_hub_name,
compliance_date, compliance, compliance_changed_frequency)
WITH compliance_aggregate AS (
   SELECT
      cluster_id,
      policy_id,
      leaf_hub_name,
```

```
        CASE
            WHEN bool_or(compliance = 'non_compliant') THEN 'non_compliant'
            WHEN bool_or(compliance = 'unknown') THEN 'unknown'
            ELSE 'compliant'
        END::local_status.compliance_type AS aggregated_compliance
    FROM
        event.local_policies lp
    WHERE
        created_at BETWEEN CURRENT_DATE - INTERVAL '1 days' AND CURRENT_DATE -
INTERVAL '0 day'
        AND EXISTS (
            SELECT 1
            FROM status.leaf_hubs lh
            WHERE lh.leaf_hub_name = lp.leaf_hub_name AND deleted_at IS NULL
        )
    GROUP BY
        cluster_id, policy_id, leaf_hub_name
)
SELECT
    policy_id,
    cluster_id,
    leaf_hub_name,
    (CURRENT_DATE - INTERVAL '1 day') AS compliance_date,
    aggregated_compliance,
    (
        SELECT COUNT(1) FROM (
            SELECT
                created_at,
                compliance,
                LAG(compliance) OVER (PARTITION BY cluster_id, policy_id ORDER BY
created_at ASC) AS prev_compliance
            FROM
                event.local_policies lp
            WHERE
                (lp.created_at BETWEEN CURRENT_DATE - INTERVAL '1 day' AND
CURRENT_DATE - INTERVAL '0 day')
                AND lp.cluster_id = ca.cluster_id AND lp.policy_id = ca.policy_id
            ORDER BY
                created_at ASC
        ) AS subquery
        WHERE compliance <> prev_compliance
    ) AS compliance_changed_frequency
FROM
    compliance_aggregate ca
ORDER BY
    cluster_id, policy_id;
ON CONFLICT (leaf_hub_name, policy_id, cluster_id, compliance_date)
DO UPDATE SET
    compliance = EXCLUDED.compliance,
    compliance_changed_frequency = EXCLUDED.compliance_changed_frequency;
```

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 2.6 and earlier versions of Red Hat Advanced Cluster Management are *removed* and no longer supported. Documentation for versions 2.6 and earlier are not updated. 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 removals

Product or category	Affected item	Version	Recommended action	More details and links
ManagedClusterSets	The **v1beta1** API is removed.	2.9	Use **v1beta2** instead.	ManagedClusterSets.cluster.open-cluster-management.io
ManagedClusterSetBindings	The **v1beta1** API is removed.	2.9	Use **v1beta2** instead.	ManagedClusterSetBindings.cluster.open-cluster-management.io
HypershiftDeployment	The **HypershiftDeployment** API is removed.	2.7	Do not use this API.	
BareMetalAssets	The **v1alpha1** API is removed.	2.7	Do not use this API.	Baremetalassets.inventory.open-cluster-management.io
Placements	The **v1alpha1** API is removed.	2.7	Use **v1beta1** instead.	Placements.cluster.open-cluster-management.io

Product or category	Affected item	Version	Recommended action	More details and links
PlacementDecisions	The **v1alpha1** API is removed.	2.7	Use **v1beta1** instead.	PlacementDecisions.cluster.open-cluster-management.io
ClusterManagementAddOn	The field **addOnConfiguration** is deprecated in the **ClusterManagementAddOn** spec.	2.7	Use the **supportedConfigs** field.	None
ManagedClusterAddOn	The field **addOnConfiguration** is deprecated in the **ManagedClusterAddOn** spec.	2.7	Use the **supportedConfigs** field.	None

1.4.2. Red Hat Advanced Cluster Management deprecations

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

Product or category	Affected item	Version	Recommended action	More details and links
Features supported on OpenShift Container Platform 3.11	Various components	2.9	None	Life Cycle Policy
Governance	IAM policy controller	2.9	None	
Governance	Container security operator	OpenShift Container Platform 3.11	None	See Container security operator is not available in OpenShift Container Platform 3.11

Product or category	Affected item	Version	Recommended action	More details and links
Installer	**ingress.sslCiphers** field in **operator.open-cluster-management.io_multiclusterhubs_crd.yaml**	2.9	None	See Advanced Configuration for configuring install. If you uppgrade your Red Hat Advanced Cluster Management for Kubernetes version and originally had a **MultiClusterHub** custom resource with the **spec.ingress.sslCiphers** field defined, the field is still recognized, but is deprecated and has no effect.
Applications and Governance	**PlacementRule**	2.8	Use **Placement** anywhere that you might use **PlacementRule**.	While **PlacementRule** is still available, it is not supported and the console displays **Placement** by default.
Installer	**customCAConfigmap** field in **operator.open-cluster-management.io_multiclusterhubs_crd.yaml**	2.7	None	See Advanced Configuration for configuring install.

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

Product or category	Affected item	Version	Recommended action	More details and links
Search	**SearchCustomizations.open-cluster-management.io** custom resource definition is removed.	2.7	Use **search.open-cluster-management.io/v1alpha1** to customize search.	None
Search	RedisGraph was replaced by PostgreSQL as the internal database.	2.7	No change required.	The search component is reimplemented by using PostgreSQL as the internal database.
Console	Standalone web console	2.7	Use the integrated web console.	See Accessing your console for more information.

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

1.5.1. Notice

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

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

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

1.5.2. Table of Contents

- Data Collection
- Data Storage
- Data Access
- Data Processing
- Data Deletion
- Capability for Restricting Use of Personal Data
- Appendix

1.5.3. GDPR

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

1.5.3.1. Why is GDPR important?

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

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

1.5.3.2. Read more about GDPR

- EU GDPR Information Portal
- Red Hat GDPR website

1.5.4. Product Configuration for GDPR

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

1.5.5. Data Life Cycle

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

As such, the Red Hat Advanced Cluster Management for Kubernetes platform deals primarily with technical data that is related to the configuration and management of the platform, some of which

might be subject to GDPR. The Red Hat Advanced Cluster Management for Kubernetes platform also deals with information about users who manage the platform. This data will be described throughout this document for the awareness of clients responsible for meeting GDPR requirements.

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

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

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

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

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

1.5.5.2. Personal data used for online contact

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

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

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

1.5.6. Data Collection

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

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

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

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

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

1.5.7. Data storage

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

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

- **Platform Configuration Data:** The Red Hat Advanced Cluster Management for Kubernetes platform configuration can be customized by updating a configuration YAML file with properties for general settings, Kubernetes, logs, network, Docker, and other settings. This data is used as input to the Red Hat Advanced Cluster Management for Kubernetes platform installer for deploying one or more nodes. The properties also include an administrator user ID and password that are used for bootstrap.
- **Kubernetes Configuration Data:** Kubernetes cluster state data is stored in a distributed key-value store, **etcd**.
- **User Authentication Data, including User IDs and passwords:** User ID and password management are handled through a client enterprise LDAP directory. Users and groups that are defined in LDAP can be added to Red Hat Advanced Cluster Management for Kubernetes platform teams and assigned access roles. Red Hat Advanced Cluster Management for Kubernetes platform stores the email address and user ID from LDAP, but does not store the password. Red Hat Advanced Cluster Management for Kubernetes platform stores the group name and upon login, caches the available groups to which a user belongs. Group membership is not persisted in any long-term way. Securing user and group data at rest in the enterprise LDAP must be considered. Red Hat Advanced Cluster Management for Kubernetes platform also includes an authentication service, Open ID Connect (OIDC) that interacts with the enterprise directory and maintains access tokens. This service uses ETCD as a backing store.

- **Service authentication data, including user IDs and passwords:** Credentials that are used by Red Hat Advanced Cluster Management for Kubernetes platform components for inter-component access are defined as Kubernetes Secrets. All Kubernetes resource definitions are persisted in the **etcd** key-value data store. Initial credentials values are defined in the platform configuration data as Kubernetes Secret configuration YAML files. For more information, see 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

Red Hat Advanced Cluster Management for Kubernetes is designed for FIPS. When running on Red Hat OpenShift Container Platform in FIPS mode, OpenShift Container Platform uses the Red Hat Enterprise Linux cryptographic libraries submitted to NIST for FIPS Validation on only the architectures that are supported by OpenShift Container Platform. For more information about the NIST validation program, see Cryptographic Module Validation Program. For the latest NIST status for the individual versions of the RHEL cryptographic libraries submitted for validation, see Compliance Activities and Government Standards.

If you plan to manage clusters with FIPS enabled, you must install Red Hat Advanced Cluster

Management on an OpenShift Container Platform cluster configured to operate in FIPS mode. The hub cluster must be in FIPS mode because cryptography that is created on the hub cluster is used on managed clusters.

To enable FIPS mode on your managed clusters, set **fips: true** when you provision your OpenShift Container Platform managed cluster. You cannot enable FIPS after you provision your cluster. For more information, see OpenShift Container Platform documentation, Do you need extra security for your cluster?

1.6.1. Limitations

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

- 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, Configuring persistent storage.
- When you provision managed clusters using the Red Hat Advanced Cluster Management console, select the following checkbox 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.

1.7. OBSERVABILITY SUPPORT

- Red Hat Advanced Cluster Management is tested with and fully supported by Red Hat OpenShift Data Foundation, formerly Red Hat OpenShift Container Platform.
- Red Hat Advanced Cluster Management supports the function of the multicluster observability operator on user-provided third-party object storage that is S3 API compatible. The observability service uses Thanos supported, stable object stores.
- Red Hat Advanced Cluster Management support efforts include reasonable efforts to identify root causes. If you open a support ticket and the root cause is the S3 compatible object storage that you provided, then you must open an issue using the customer support channels.