Red Hat Advanced Cluster Management for Kubernetes 2.3

Troubleshooting

View a list of troubleshooting topics for your cluster. You can also use the must-gather command to collect logs.

Last Updated: 2022-08-31

Red Hat Advanced Cluster Management for Kubernetes 2.3 Troubleshooting

View a list of troubleshooting topics for your cluster. You can also use the must-gather command to collect logs.

Legal Notice

Abstract

View a list of troubleshooting topics for your cluster. You can also use the must-gather command to collect logs.

Table of Contents

CHAPTER 1. TROUBLESHOOTING

Before using the Troubleshooting guide, you can run the **oc adm must-gather** command to gather details, logs, and take steps in debugging issues. For more details, see Running the must-gather command to troubleshoot.

Additionally, check your role-based access. See Role-based access control for details.

1.1. DOCUMENTED TROUBLESHOOTING

View the list of troubleshooting topics for Red Hat Advanced Cluster Management for Kubernetes:

Installation

To get to the original installing tasks, view Installing.

- Troubleshooting installation status stuck in installing or pending
- Troubleshooting reinstallation failure

Cluster management

To get to the original cluster management tasks, view Managing your clusters.

- Troubleshooting an offline cluster
- Troubleshooting cluster with pending import status
- Troubleshooting imported clusters offline after certificate change
- Troubleshooting cluster status changing from offline to available
- Troubleshooting cluster creation on VMware vSphere
- Troubleshooting cluster in console with pending or failed status
- Troubleshooting OpenShift Container Platform version 3.11 cluster import failure
- Troubleshooting Klusterlet with degraded conditions
- Troubleshooting Klusterlet application manager on managed clusters
- Troubleshooting Object storage channel secret
- Troubleshooting managedcluster resource
- Namespace remains after deleting a cluster
- Auto-import-secret-exists error when importing a cluster

Application management

To get to the original application management, view Managing applications.

- Troubleshooting application Kubernetes deployment version .
- Troubleshooting standalone subscription memory problem .

- Troubleshooting application Git server connection.
- Troubleshooting local cluster not selected

Governance

To get to the original security guide, view Risk and compliance.

Console observability

Console observability includes Search and the Visual Web Terminal, along with header and navigation function. To get to the original observability guide, view Observability in the console.

- Troubleshooting grafana
- Troubleshooting observability
- Troubleshooting OpenShift monitoring services
- Undesired label value in managedcluster resource
- Troubleshooting search aggregator pod status
- Troubleshooting metrics-collector

1.2. RUNNING THE MUST-GATHER COMMAND TO TROUBLESHOOT

To get started with troubleshooting, learn about the troubleshooting scenarios for users to run the **must-gather** command to debug the issues, then see the procedures to start using the command.

Required access: Cluster administrator

1.2.1. Must-gather scenarios

- **Scenario one:** Use the Documented troubleshooting section to see if a solution to your problem is documented. The guide is organized by the major functions of the product.
  With this scenario, you check the guide to see if your solution is in the documentation. For instance, for trouble with creating a cluster, you might find a solution in the *Manage cluster* section.
- **Scenario two:** If your problem is not documented with steps to resolve, run the **must-gather** command and use the output to debug the issue.
- **Scenario three:** If you cannot debug the issue using your output from the **must-gather** command, then share your output with Red Hat Support.

1.2.2. Must-gather procedure

See the following procedure to start using the **must-gather** command:

1. Learn about the **must-gather** command and install the prerequisites that you need at Gathering data about your cluster in the Red Hat OpenShift Container Platform documentation.
2. Log in to your cluster. For the usual use-case, you should run the **must-gather** while you are logged into your *hub* cluster.

Note: If you want to check your managed clusters, find the **gather-managed.log** file that is located in the the **cluster-scoped-resources** directory:

```
<your-directory>/cluster-scoped-resources/gather-managed.log>
```

Check for managed clusters that are not set **True** for the JOINED and AVAILABLE column. You can run the **must-gather** command on those clusters that are not connected with **True** status.

3. Add the Red Hat Advanced Cluster Management for Kubernetes image that is used for gathering data and the directory. Run the following command, where you insert the image and the directory for the output:

```
oc adm must-gather --image=registry.redhat.io/rhacm2/acm-must-gather-rhel8:v2.3.0 --dest-dir=<directory>
```

4. Go to your specified directory to see your output, which is organized in the following levels:
   - Two peer levels: **cluster-scoped-resources** and **namespace** resources.
   - Sub-level for each: API group for the custom resource definitions for both cluster-scope and namespace-scoped resources.
   - Next level for each: YAML file sorted by **kind**.

1.2.3. Must-gather in a disconnected environment

Complete the following steps to run the **must-gather** command in a disconnected environment:

1. In a disconnected environment, mirror the Red Hat operator catalog images into their mirror registry. For more information, see Install on disconnected networks.
2. Run the following command to extract logs, which reference the image from their mirror registry:

```
REGISTRY=registry.example.com:5000
IMAGE=$REGISTRY/rhacm2/acm-must-gather-rhel8@sha256:ff9f37eb400dc1f7d07a9b6f2da9064992934b69847d17f59e385783c071b9d8

oc adm must-gather --image=$IMAGE --dest-dir=./data
```

1.3. TROUBLESHOOTING INSTALLATION STATUS STUCK IN INSTALLING OR PENDING

When installing Red Hat Advanced Cluster Management, the **MultiClusterHub** remains in **Installing** phase, or multiple pods maintain a **Pending** status.

1.3.1. Symptom: Stuck in Pending status

More than ten minutes passed since you installed **MultiClusterHub** and one or more components from the **status.components** field of the **MultiClusterHub** resource report **ProgressDeadlineExceeded**. Resource constraints on the cluster might be the issue.

Check the pods in the namespace where **Multiclusterhub** was installed. You might see **Pending** with a status similar to the following:

```
reason: Unschedulable
message: '0/6 nodes are available: 3 Insufficient cpu, 3 node(s) had taint {node-
role.kubernetes.io/master:
        }, that the pod didn't tolerate.'
```

In this case, the worker nodes resources are not sufficient in the cluster to run the product.

1.3.2. Resolving the problem: Adjust worker node sizing

If you have this problem, then your cluster needs to be updated with either larger or more worker nodes. See Sizing your cluster for guidelines on sizing your cluster.

1.4. TROUBLESHOOTING REINSTALLATION FAILURE

When reinstalling Red Hat Advanced Cluster Management for Kubernetes, the pods do not start.

1.4.1. Symptom: Reinstallation failure

If your pods do not start after you install Red Hat Advanced Cluster Management, it is likely that Red Hat Advanced Cluster Management was previously installed, and not all of the pieces were removed before you attempted this installation.

In this case, the pods do not start after completing the installation process.

1.4.2. Resolving the problem: Reinstallation failure

If you have this problem, complete the following steps:

1. Run the uninstallation process to remove the current components by following the steps in Uninstalling.
2. Install the Helm CLI binary version 3.2.0, or later, by following the instructions at Installing Helm.
3. Ensure that your Red Hat OpenShift Container Platform CLI is configured to run **oc** commands. See Getting started with the OpenShift CLI in the OpenShift Container Platform documentation for more information about how to configure the **oc** commands.
4. Copy the following script into a file:

```
#!/bin/bash
ACM_NAMESPACE=<namespace>
oc delete mch --all -n $ACM_NAMESPACE
helm ls --namespace $ACM_NAMESPACE | cut -f 1 | tail -n +2 | xargs -n 1 helm delete --
namespace $ACM_NAMESPACE
oc delete apiservice v1beta1.webhook.certmanager.k8s.io v1.admission.cluster.open-cluster-
management.io v1.admission.work.open-cluster-management.io
oc delete clusterimageset --all
oc delete configmap -n $ACM_NAMESPACE cert-manager-controller cert-manager-
cainjector-leader-election cert-manager-cainjector-leader-election-core
oc delete consolelink acm-console-link
oc delete crd klusterletaddonconfigs.agent.open-cluster-management.io
placementbindings.policy.open-cluster-management.io policies.policy.open-cluster-
management.io userpreferences.console.open-cluster-management.io
searchservices.search.acm.com
```

```
oc delete mutatingwebhookconfiguration cert-manager-webhook cert-manager-webhook-v1alpha1 ocm-mutating-webhook managedclustermutators.admission.cluster.open-cluster-management.io
oc delete oauthclient multicloudingress
oc delete rolebinding -n kube-system cert-manager-webhook-webhook-authentication-reader
oc delete scc kui-proxy-scc
oc delete validatingwebhookconfiguration cert-manager-webhook cert-manager-webhook-v1alpha1 channels.apps.open.cluster.management.webhook.validator application-webhook-validator multiclusterhub-operator-validating-webhook ocm-validating-webhook
```

Replace **<namespace>** in the script with the name of the namespace where Red Hat Advanced Cluster Management was installed. Ensure that you specify the correct namespace, as the namespace is cleaned out and deleted.

5. Run the script to remove the artifacts from the previous installation.

6. Run the installation. See Installing while connected online .

1.5. TROUBLESHOOTING AN OFFLINE CLUSTER

There are a few common causes for a cluster showing an offline status.

1.5.1. Symptom: Cluster status is offline

After you complete the procedure for creating a cluster, you cannot access it from the Red Hat Advanced Cluster Management console, and it shows a status of **offline**.

1.5.2. Resolving the problem: Cluster status is offline

1. Determine if the managed cluster is available. You can check this in the *Clusters* area of the Red Hat Advanced Cluster Management console.
   If it is not available, try restarting the managed cluster.

2. If the managed cluster status is still offline, complete the following steps:

   a. Run the **oc get managedcluster <cluster_name> -o yaml** command on the hub cluster. Replace **<cluster_name>** with the name of your cluster.

   b. Find the **status.conditions** section.

   c. Check the messages for **type: ManagedClusterConditionAvailable** and resolve any problems.

1.6. TROUBLESHOOTING CLUSTER WITH PENDING IMPORT STATUS

If you receive *Pending import* continually on the console of your cluster, follow the procedure to troubleshoot the problem.

1.6.1. Symptom: Cluster with pending import status

After importing a cluster by using the Red Hat Advanced Cluster Management console, the cluster appears in the console with a status of *Pending import*.

1.6.2. Identifying the problem: Cluster with pending import status

1. Run the following command on the managed cluster to view the Kubernetes pod names that are having the issue:

```
kubectl get pod -n open-cluster-management-agent | grep klusterlet-registration-agent
```

2. Run the following command on the managed cluster to find the log entry for the error:

```
kubectl logs <registration_agent_pod> -n open-cluster-management-agent
```

   Replace *registration_agent_pod* with the pod name that you identified in step 1.

3. Search the returned results for text that indicates there was a networking connectivity problem. Example includes: **no such host**.

1.6.3. Resolving the problem: Cluster with pending import status

1. Retrieve the port number that is having the problem by entering the following command on the hub cluster:

```
oc get infrastructure cluster -o yaml | grep apiServerURL
```

2. Ensure that the hostname from the managed cluster can be resolved, and that outbound connectivity to the host and port is occurring.
   If the communication cannot be established by the managed cluster, the cluster import is not complete. The cluster status for the managed cluster is *Pending import*.

1.7. TROUBLESHOOTING CLUSTER WITH ALREADY EXISTS ERROR

If you are unable to import an OpenShift Container Platform cluster into Red Hat Advanced Cluster Management **MultiClusterHub** and receive an **AlreadyExists** error, follow the procedure to troubleshoot the problem.

1.7.1. Symptom: Already exists error log when importing OpenShift Container Platform cluster

An error log shows up when importing an OpenShift Container Platform cluster into Red Hat Advanced Cluster Management **MultiClusterHub**:

```
error log:
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+,
unavailable in v1.22+; use apiextensions.k8s.io/v1 CustomResourceDefinition
Error from server (AlreadyExists): error when creating "STDIN":
customresourcedefinitions.apiextensions.k8s.io "klusterlets.operator.open-cluster-management.io"
already exists
The cluster cannot be imported because its Klusterlet CRD already exists.
Either the cluster was already imported, or it was not detached completely during a previous detach
process.
Detach the existing cluster before trying the import again."
```

1.7.2. Identifying the problem: Already exists when importing OpenShift Container Platform cluster

Check if there are any Red Hat Advanced Cluster Management-related resources on the cluster that you want to import to new the Red Hat Advanced Cluster Management **MultiClusterHub** by running the following commands:

```
oc get all -n open-cluster-management-agent
oc get all -n open-cluster-management-agent-addon
```

1.7.3. Resolving the problem: Already exists when importing OpenShift Container Platform cluster

Run the following commands to remove pre-existing resources:

```
oc delete namespaces open-cluster-management-agent open-cluster-management-agent-addon --wait=false
oc get crds | grep open-cluster-management.io | awk '{print $1}' | xargs oc delete crds --wait=false
oc get crds | grep open-cluster-management.io | awk '{print $1}' | xargs oc patch crds --type=merge -p '{"metadata":{"finalizers": []}}'
```

1.8. TROUBLESHOOTING CLUSTER CREATION ON VMWARE VSPHERE

If you experience a problem when creating a Red Hat OpenShift Container Platform cluster on VMware vSphere, see the following troubleshooting information to see if one of them addresses your problem.

Note: Sometimes when the cluster creation process fails on VMware vSphere, the link is not enabled for you to view the logs. If this happens, you can identify the problem by viewing the log of the **hive-controllers** pod. The **hive-controllers** log is in the **hive** namespace.

1.8.1. Managed cluster creation fails with certificate IP SAN error

1.8.1.1. Symptom: Managed cluster creation fails with certificate IP SAN error

After creating a new Red Hat OpenShift Container Platform cluster on VMware vSphere, the cluster fails with an error message that indicates a certificate IP SAN error.

1.8.1.2. Identifying the problem: Managed cluster creation fails with certificate IP SAN error

The deployment of the managed cluster fails and returns the following errors in the deployment log:

```
time="2020-08-07T15:27:55Z" level=error msg="Error: error setting up new vSphere SOAP client: Post https://147.1.1.1/sdk: x509: cannot validate certificate for xx.xx.xx.xx because it doesn't contain any IP SANs"
time="2020-08-07T15:27:55Z" level=error
```

1.8.1.3. Resolving the problem: Managed cluster creation fails with certificate IP SAN error

Use the VMware vCenter server fully-qualified host name instead of the IP address in the credential. You can also update the VMware vCenter CA certificate to contain the IP SAN.

1.8.2. Managed cluster creation fails with unknown certificate authority

1.8.2.1. Symptom: Managed cluster creation fails with unknown certificate authority

After creating a new Red Hat OpenShift Container Platform cluster on VMware vSphere, the cluster fails because the certificate is signed by an unknown authority.

1.8.2.2. Identifying the problem: Managed cluster creation fails with unknown certificate authority

The deployment of the managed cluster fails and returns the following errors in the deployment log:

```
Error: error setting up new vSphere SOAP client: Post https://vspherehost.com/sdk: x509: certificate signed by unknown authority"
```

1.8.2.3. Resolving the problem: Managed cluster creation fails with unknown certificate authority

Ensure you entered the correct certificate from the certificate authority when creating the credential.

1.8.3. Managed cluster creation fails with expired certificate

1.8.3.1. Symptom: Managed cluster creation fails with expired certificate

After creating a new Red Hat OpenShift Container Platform cluster on VMware vSphere, the cluster fails because the certificate is expired or is not yet valid.

1.8.3.2. Identifying the problem: Managed cluster creation fails with expired certificate

The deployment of the managed cluster fails and returns the following errors in the deployment log:

```
x509: certificate has expired or is not yet valid
```

1.8.3.3. Resolving the problem: Managed cluster creation fails with expired certificate

Ensure that the time on your ESXi hosts is synchronized.

1.8.4. Managed cluster creation fails with insufficient privilege for tagging

1.8.4.1. Symptom: Managed cluster creation fails with insufficient privilege for tagging

After creating a new Red Hat OpenShift Container Platform cluster on VMware vSphere, the cluster fails because there is insufficient privilege to use tagging.

1.8.4.2. Identifying the problem: Managed cluster creation fails with insufficient privilege for tagging

The deployment of the managed cluster fails and returns the following errors in the deployment log:

```
time="2020-08-07T19:41:58Z" level=debug msg="vsphere_tag_category.category: Creating..."
time="2020-08-07T19:41:58Z" level=error
time="2020-08-07T19:41:58Z" level=error msg="Error: could not create category: POST
https://vspherehost.com/rest/com/vmware/cis/tagging/category: 403 Forbidden"
time="2020-08-07T19:41:58Z" level=error
time="2020-08-07T19:41:58Z" level=error msg="  on ../tmp/openshift-install-436877649/main.tf line
```

```
54, in resource \"vsphere_tag_category\" \"category\":"
time="2020-08-07T19:41:58Z" level=error msg="  54: resource \"vsphere_tag_category\" \"category\"
{"
```

1.8.4.3. Resolving the problem: Managed cluster creation fails with insufficient privilege for tagging

Ensure that your VMware vCenter required account privileges are correct. See Image registry removed during information for more information.

1.8.5. Managed cluster creation fails with invalid dnsVIP

1.8.5.1. Symptom: Managed cluster creation fails with invalid dnsVIP

After creating a new Red Hat OpenShift Container Platform cluster on VMware vSphere, the cluster fails because there is an invalid dnsVIP.

1.8.5.2. Identifying the problem: Managed cluster creation fails with invalid dnsVIP

If you see the following message when trying to deploy a new managed cluster with VMware vSphere, it is because you have an older OpenShift Container Platform release image that does not support VMware Installer Provisioned Infrastructure (IPI):

```
failed to fetch Master Machines: failed to load asset \\\"Install Config\\\": invalid \\\"install-
config.yaml\\\" file: platform.vsphere.dnsVIP: Invalid value: \\\"\\\": \\\"\\\" is not a valid IP
```

1.8.5.3. Resolving the problem: Managed cluster creation fails with invalid dnsVIP

Select a release image from a later version of OpenShift Container Platform that supports VMware Installer Provisioned Infrastructure.

1.8.6. Managed cluster creation fails with incorrect network type

1.8.6.1. Symptom: Managed cluster creation fails with incorrect network type

After creating a new Red Hat OpenShift Container Platform cluster on VMware vSphere, the cluster fails because there is an incorrect network type specified.

1.8.6.2. Identifying the problem: Managed cluster creation fails with incorrect network type

If you see the following message when trying to deploy a new managed cluster with VMware vSphere, it is because you have an older OpenShift Container Platform image that does not support VMware Installer Provisioned Infrastructure (IPI):

```
time="2020-08-11T14:31:38-04:00" level=debug msg="vsphereprivate_import_ova.import:
Creating..."
time="2020-08-11T14:31:39-04:00" level=error
time="2020-08-11T14:31:39-04:00" level=error msg="Error: rpc error: code = Unavailable desc =
transport is closing"
time="2020-08-11T14:31:39-04:00" level=error
```

```
time="2020-08-11T14:31:39-04:00" level=error
time="2020-08-11T14:31:39-04:00" level=fatal msg="failed to fetch Cluster: failed to generate asset
\"Cluster\": failed to create cluster: failed to apply Terraform: failed to complete the change"
```

1.8.6.3. Resolving the problem: Managed cluster creation fails with incorrect network type

Select a valid VMware vSphere network type for the specified VMware cluster.

1.8.7. Managed cluster creation fails with an error processing disk changes

1.8.7.1. Symptom: Adding the VMware vSphere managed cluster fails due to an error processing disk changes

After creating a new Red Hat OpenShift Container Platform cluster on VMware vSphere, the cluster fails because there is an error when processing disk changes.

1.8.7.2. Identifying the problem: Adding the VMware vSphere managed cluster fails due to an error processing disk changes

A message similar to the following is displayed in the logs:

```
ERROR
ERROR Error: error reconfiguring virtual machine: error processing disk changes post-clone: disk.0:
ServerFaultCode: NoPermission: RESOURCE (vm-71:2000), ACTION (queryAssociatedProfile):
RESOURCE (vm-71), ACTION (PolicyIDByVirtualDisk)
```

1.8.7.3. Resolving the problem: Adding the VMware vSphere managed cluster fails due to an error processing disk changes

Use the VMware vSphere client to give the user **All privileges** for *Profile-driven Storage Privileges*.

1.9. TROUBLESHOOTING OPENSHIFT CONTAINER PLATFORM VERSION 3.11 CLUSTER IMPORT FAILURE

1.9.1. Symptom: OpenShift Container Platform version 3.11 cluster import failure

After you attempt to import a Red Hat OpenShift Container Platform version 3.11 cluster, the import fails with a log message that resembles the following content:

```
customresourcedefinition.apiextensions.k8s.io/klusterlets.operator.open-cluster-management.io
configured
clusterrole.rbac.authorization.k8s.io/klusterlet configured
clusterrole.rbac.authorization.k8s.io/open-cluster-management:klusterlet-admin-aggregate-clusterrole
configured
clusterrolebinding.rbac.authorization.k8s.io/klusterlet configured
namespace/open-cluster-management-agent configured
secret/open-cluster-management-image-pull-credentials unchanged
serviceaccount/klusterlet configured
deployment.apps/klusterlet unchanged
klusterlet.operator.open-cluster-management.io/klusterlet configured
Error from server (BadRequest): error when creating "STDIN": Secret in version "v1" cannot be
```

```
handled as a Secret:
v1.Secret.ObjectMeta:
v1.ObjectMeta.TypeMeta: Kind: Data: decode base64: illegal base64 data at input byte 1313, error
found in #10 byte of ...|dhruy45="},"kind":"|..., bigger context
...|tye56u56u568yuo7i67i67i67o556574i"},"kind":"Secret","metadata":{"annotations":{"kube|...
```

1.9.2. Identifying the problem: OpenShift Container Platform version 3.11 cluster import failure

This often occurs because the installed version of the **kubectl** command-line tool is 1.11, or earlier. Run the following command to see which version of the **kubectl** command-line tool you are running:

```
kubectl version
```

If the returned data lists version 1.11, or earlier, complete one of the fixes in *Resolving the problem: OpenShift Container Platform version 3.11 cluster import failure*.

1.9.3. Resolving the problem: OpenShift Container Platform version 3.11 cluster import failure

You can resolve this issue by completing one of the following procedures:

- Install the latest version of the **kubectl** command-line tool.
  1. Download the latest version of the **kubectl** tool from: Install and Set Up kubectl in the Kubernetes documentation.
  2. Import the cluster again after upgrading your **kubectl** tool.
- Run a file that contains the import command.
  1. Start the procedure in Importing a managed cluster with the CLI .
  2. When you create the command to import your cluster, copy that command into a YAML file named **import.yaml**.
  3. Run the following command to import the cluster again from the file:

     ```
     oc apply -f import.yaml
     ```

1.10. TROUBLESHOOTING IMPORTED CLUSTERS OFFLINE AFTER CERTIFICATE CHANGE

Installing a custom **apiserver** certificate is supported, but one or more clusters that were imported before you changed the certificate information can have an **offline** status.

1.10.1. Symptom: Clusters offline after certificate change

After you complete the procedure for updating a certificate secret, one or more of your clusters that were online are now displaying an **offline** status in the Red Hat Advanced Cluster Management for Kubernetes console.

1.10.2. Identifying the problem: Clusters offline after certificate change

After updating the information for a custom API server certificate, clusters that were imported and running before the new certificate are now in an **offline** state.

The errors that indicate that the certificate is the problem are found in the logs for the pods in the **open-cluster-management-agent** namespace of the offline managed cluster. The following examples are similar to the errors that are displayed in the logs:

Log of work-agent:

```
E0917 03:04:05.874759       1 manifestwork_controller.go:179] Reconcile work test-1-klusterlet-addon-workmgr fails with err: Failed to update work status with err Get "https://api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/namespaces/test-1/manifestworks/test-1-klusterlet-addon-workmgr": x509: certificate signed by unknown authority
E0917 03:04:05.874887       1 base_controller.go:231] "ManifestWorkAgent" controller failed to sync "test-1-klusterlet-addon-workmgr", err: Failed to update work status with err Get "api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/namespaces/test-1/manifestworks/test-1-klusterlet-addon-workmgr": x509: certificate signed by unknown authority
E0917 03:04:37.245859       1 reflector.go:127] k8s.io/client-go@v0.19.0/tools/cache/reflector.go:156: Failed to watch *v1.ManifestWork: failed to list *v1.ManifestWork: Get "api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/namespaces/test-1/manifestworks?resourceVersion=607424": x509: certificate signed by unknown authority
```

Log of registration-agent:

```
I0917 02:27:41.525026       1 event.go:282] Event(v1.ObjectReference{Kind:"Namespace", Namespace:"open-cluster-management-agent", Name:"open-cluster-management-agent", UID:"", APIVersion:"v1", ResourceVersion:"", FieldPath:""}): type: 'Normal' reason: 'ManagedClusterAvailableConditionUpdated' update managed cluster "test-1" available condition to "True", due to "Managed cluster is available"
E0917 02:58:26.315984       1 reflector.go:127] k8s.io/client-go@v0.19.0/tools/cache/reflector.go:156: Failed to watch *v1beta1.CertificateSigningRequest: Get "https://api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/managedclusters?allowWatchBookmarks=true&fieldSelector=metadata.name%3Dtest-1&resourceVersion=607408&timeout=9m33s&timeoutSeconds=573&watch=true"": x509: certificate signed by unknown authority
E0917 02:58:26.598343       1 reflector.go:127] k8s.io/client-go@v0.19.0/tools/cache/reflector.go:156: Failed to watch *v1.ManagedCluster: Get "https://api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/managedclusters?allowWatchBookmarks=true&fieldSelector=metadata.name%3Dtest-1&resourceVersion=607408&timeout=9m33s&timeoutSeconds=573&watch=true": x509: certificate signed by unknown authority
E0917 02:58:27.613963       1 reflector.go:127] k8s.io/client-go@v0.19.0/tools/cache/reflector.go:156: Failed to watch *v1.ManagedCluster: failed to list *v1.ManagedCluster: Get "https://api.aaa-ocp.dev02.location.com:6443/apis/cluster.management.io/v1/managedclusters?allowWatchBookmarks=true&fieldSelector=metadata.name%3Dtest-1&resourceVersion=607408&timeout=9m33s&timeoutSeconds=573&watch=true"": x509: certificate signed by unknown authority
```

1.10.3. Resolving the problem: Clusters offline after certificate change

To manually restore your clusters after updating your certificate information, complete the following steps for each managed cluster:

1. Manually import the cluster again. Red Hat OpenShift Container Platform clusters that were created from Red Hat Advanced Cluster Management will resynchronize every 2 hours, so you can skip this step for those clusters.

   a. On the hub cluster, display the import command by entering the following command:

   ```
   oc get secret -n ${CLUSTER_NAME} ${CLUSTER_NAME}-import -
   ojsonpath='{.data.import\.yaml}' | base64 --decode  > import.yaml
   ```

   Replace *CLUSTER_NAME* with the name of the managed cluster that you are importing.

   b. On the managed cluster, apply the **import.yaml** file:

   ```
   oc apply -f import.yaml
   ```

1.11. NAMESPACE REMAINS AFTER DELETING A CLUSTER

When you remove a managed cluster, the namespace is normally removed as part of the cluster removal process. In rare cases, the namespace remains with some artifacts in it. In that case, you must manually remove the namespace.

1.11.1. Symptom: Namespace remains after deleting a cluster

After removing a managed cluster, the namespace is not removed.

1.11.2. Resolving the problem: Namespace remains after deleting a cluster

Complete the following steps to remove the namespace manually:

1. Run the following command to produce a list of the resources that remain in the <cluster_name> namespace:

   ```
   oc api-resources --verbs=list --namespaced -o name | grep -E
   '^secrets|^serviceaccounts|^managedclusteraddons|^roles|^rolebindings|^manifestworks|^lease
   |^managedclusterinfo|^appliedmanifestworks' | xargs -n 1 oc get --show-kind --ignore-not-
   found -n <cluster_name>
   ```

   Replace *cluster_name* with the name of the namespace for the cluster that you attempted to remove.

2. Delete each identified resource on the list that does not have a status of **Delete** by entering the following command to edit the list:

   ```
   oc edit <resource_kind> <resource_name> -n <namespace>
   ```

   Replace *resource_kind* with the kind of the resource. Replace *resource_name* with the name of the resource. Replace *namespace* with the name of the namespace of the resource.

3. Locate the **finalizer** attribute in the in the metadata.

4. Delete the non-Kubernetes finalizers by using the vi editor **dd** command.

5. Save the list and exit the **vi** editor by entering the **:wq** command.

6. Delete the namespace by entering the following command:

```
oc delete ns <cluster-name>
```

Replace *cluster-name* with the name of the namespace that you are trying to delete.

1.12. AUTO-IMPORT-SECRET-EXISTS ERROR WHEN IMPORTING A CLUSTER

Your cluster import fails with an error message that reads: auto import secret exists.

1.12.1. Symptom: Auto import secret exists error when importing a cluster

When importing a hive cluster for management, an **auto-import-secret already exists** error is displayed.

1.12.2. Resolving the problem: Auto-import-secret-exists error when importing a cluster

This problem occurs when you attempt to import a cluster that was previously managed by Red Hat Advanced Cluster Management. When this happens, the secrets conflict when you try to reimport the cluster.

To work around this problem, complete the following steps:

1. To manually delete the existing **auto-import-secret**, run the following command on the hub cluster:

```
oc delete secret auto-import-secret -n <cluster-namespace>
```

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

2. Import your cluster again using the procedure in Importing a target managed cluster to a hub cluster.

Your cluster is imported.

1.13. TROUBLESHOOTING CLUSTER STATUS CHANGING FROM OFFLINE TO AVAILABLE

The status of the managed cluster alternates between **offline** and **available** without any manual change to the environment or cluster.

1.13.1. Symptom: Cluster status changing from offline to available

When the network that connects the managed cluster to the hub cluster is unstable, the status of the managed cluster that is reported by the hub cluster cycles between **offline** and **available**.

1.13.2. Resolving the problem: Cluster status changing from offline to available

To attempt to resolve this issue, complete the following steps:

1. Edit your **ManagedCluster** specification on the hub cluster by entering the following command:

```
oc edit managedcluster <cluster-name>
```

   Replace *cluster-name* with the name of your managed cluster.

2. Increase the value of **leaseDurationSeconds** in your **ManagedCluster** specification. The default value is 5 minutes, but that might not be enough time to maintain the connection with the network issues. Specify a greater amount of time for the lease. For example, you can raise the setting to 20 minutes.

1.14. TROUBLESHOOTING CLUSTER IN CONSOLE WITH PENDING OR FAILED STATUS

If you observe *Pending* status or *Failed* status in the console for a cluster you created, follow the procedure to troubleshoot the problem.

1.14.1. Symptom: Cluster in console with pending or failed status

After creating a new cluster by using the Red Hat Advanced Cluster Management for Kubernetes console, the cluster does not progress beyond the status of *Pending* or displays *Failed* status.

1.14.2. Identifying the problem: Cluster in console with pending or failed status

If the cluster displays *Failed* status, navigate to the details page for the cluster and follow the link to the logs provided. If no logs are found or the cluster displays *Pending* status, continue with the following procedure to check for logs:

- Procedure 1

  1. Run the following command on the hub cluster to view the names of the Kubernetes pods that were created in the namespace for the new cluster:

```
oc get pod -n <new_cluster_name>
```

     Replace ***new_cluster_name*** with the name of the cluster that you created.

  2. If no pod that contains the string **provision** in the name is listed, continue with Procedure 2. If there is a pod with **provision** in the title, run the following command on the hub cluster to view the logs of that pod:

```
oc logs <new_cluster_name_provision_pod_name> -n <new_cluster_name> -c hive
```

     Replace **new_cluster_name_provision_pod_name** with the name of the cluster that you created, followed by the pod name that contains **provision**.

  3. Search for errors in the logs that might explain the cause of the problem.

- Procedure 2
  If there is not a pod with **provision** in its name, the problem occurred earlier in the process. Complete the following procedure to view the logs:

  1. Run the following command on the hub cluster:

```
oc describe clusterdeployments -n <new_cluster_name>
```

Replace **new_cluster_name** with the name of the cluster that you created. For more information about cluster installation logs, see Gathering installation logs in the Red Hat OpenShift documentation.

2. See if there is additional information about the problem in the *Status.Conditions.Message* and *Status.Conditions.Reason* entries of the resource.

1.14.3. Resolving the problem: Cluster in console with pending or failed status

After you identify the errors in the logs, determine how to resolve the errors before you destroy the cluster and create it again.

The following example provides a possible log error of selecting an unsupported zone, and the actions that are required to resolve it:

```
No subnets provided for zones
```

When you created your cluster, you selected one or more zones within a region that are not supported. Complete one of the following actions when you recreate your cluster to resolve the issue:

- Select a different zone within the region.
- Omit the zone that does not provide the support, if you have other zones listed.
- Select a different region for your cluster.

After determining the issues from the log, destroy the cluster and recreate it.

See Creating a cluster for more information about creating a cluster.

1.15. TROUBLESHOOTING APPLICATION GIT SERVER CONNECTION

Logs from the **open-cluster-management** namespace display failure to clone the Git repository.

1.15.1. Symptom: Git server connection

The logs from the subscription controller pod **multicluster-operators-hub-subscription-<random-characters>** in the **open-cluster-management** namespace indicates that it fails to clone the Git repository. You receive a **x509: certificate signed by unknown authority** error, or **BadGateway** error.

1.15.2. Resolving the problem: Git server connection

Important: Upgrade if you are on a previous version.

1. Save apps.open-cluster-management.io_channels_crd.yaml as the same file name.
2. On the Red Hat Advanced Cluster Management cluster, run the following command to apply the file:

```
oc apply -f apps.open-cluster-management.io_channels_crd.yaml
```

3. In the **open-cluster-management** namespace, edit the **advanced-cluster-management.v2.2.0** CSV, run the following command and edit:

```
oc edit csv advanced-cluster-management.v2.2.0 -n open-cluster-management
```

   Find the following containers:

   - **multicluster-operators-standalone-subscription**
   - **multicluster-operators-hub-subscription**
     Replace the container images with the following:

```
quay.io/open-cluster-management/multicluster-operators-subscription:2.2-PR337-91af6cb37d427d22160b2c055589a4418dada4eb
```

   The update recreates the following pods in the **open-cluster-management** namespace:

   - **multicluster-operators-standalone-subscription-<random-characters>**
   - **multicluster-operators-hub-subscription-<random-characters>**

4. Check that the new pods are running with the new docker image. Run the following command, then find the new docker image:

```
oc get pod multicluster-operators-standalone-subscription-<random-characters> -n open-cluster-management -o yaml
oc get pod multicluster-operators-hub-subscription-<random-characters> -n open-cluster-management -o yaml
```

1. Update the images on managed clusters.
   On the hub cluster, run the following command by replacing **CLUSTER_NAME** with the actual managed cluster name:

```
oc annotate klusterletaddonconfig -n CLUSTER_NAME CLUSTER_NAME klusterletaddonconfig-pause=true --overwrite=true
```

2. Run the following command, replacing **CLUSTER_NAME** with the actual managed cluster name:

```
oc edit manifestwork -n CLUSTER_NAME  CLUSTER_NAME-klusterlet-addon-appmgr
```

3. Find **spec.global.imageOverrides.multicluster_operators_subscription** and set the value to:

```
quay.io/open-cluster-management/multicluster-operators-subscription:2.2-PR337-91af6cb37d427d22160b2c055589a4418dada4eb
```

   This recreates the **klusterlet-addon-appmgr-<random-characters>** pod in **open-cluster-management-agent-addon** namespace on the managed cluster.

4. Check that the new pod is running with the new docker image.

5. When you create an application through the console or the CLI, add `insecureSkipVerify: true' in the channel spec manually. See the following example:

```
apiVersion: apps.open-cluster-management.io/v1
kind: Channel
metadata:
labels:
  name: sample-channel
  namespace: sample
spec:
  type: GitHub
  pathname: <Git URL>
  insecureSkipVerify: true
```

1.16. TROUBLESHOOTING GRAFANA

When you query some time-consuming metrics in the Grafana explorer, you might encounter a **Gateway Time-out** error.

1.16.1. Symptom: Grafana explorer gateway timeout

If you hit the **Gateway Time-out** error when you query some time-consuming metrics in the Grafana explorer, it is possible that the timeout is caused by the **multicloud-console** route in the **open-cluster-management** namespace.

1.16.2. Resolving the problem: Configure the *multicloud-console* route

If you have this problem, complete the following steps:

1. Verify that the default configuration of Grafana has expected timeout settings:
   a. To verify that the default timeout setting of Grafana, run the following command:

   ```
   oc get secret grafana-config -n open-cluster-management-observability -o jsonpath="
   {.data.grafana\.ini}" | base64 -d | grep dataproxy -A 4
   ```

   The following timeout settings should be displayed:

   ```
   [dataproxy]
   timeout = 300
   dial_timeout = 30
   keep_alive_seconds = 300
   ```

   b. To verify the default data source query timeout for Grafana, run the following command:

   ```
   oc get secret/grafana-datasources -n open-cluster-management-observability -o
   jsonpath="{.data.datasources\.yaml}" | base64 -d | grep queryTimeout
   ```

   The following timeout settings should be displayed:

   ```
   queryTimeout: 300s
   ```

2. If you verified the default configuration of Grafana has expected timeout settings, then you can configure the **multicloud-console** route in the **open-cluster-management** namespace by running the following command:

```
oc annotate route multicloud-console -n open-cluster-management --overwrite
haproxy.router.openshift.io/timeout=300s
```

Refresh the Grafana page and try to query the metrics again. The **Gateway Time-out** error is no longer displayed.

1.17. TROUBLESHOOTING LOCAL CLUSTER NOT SELECTED WITH PLACEMENT RULE

The managed clusters are selected with a placement rule, but the **local-cluster** (hub cluster that is also managed) is not selected. The placement rule user is not granted to permission to create deployable resources in the **local-cluster** namespace.

1.17.1. Symptom: Troubleshooting local cluster not selected

All managed clusters are selected with a placement rule, but the **local-cluster** is not. The placement rule user is not granted permission to create the deployable resources in the **local-cluster** namespace.

1.17.2. Resolving the problem: Troubleshooting local cluster not selected

To resolve this issue, you need to grant the deployable administrative permission in the **local-cluster** namespace. Complete the following steps:

1. Confirm that the list of managed clusters does include **local-cluster**, and that the placement rule **decisions** list does not display the local cluster. Run the following command and view the results:

```
% oc get managedclusters
```

```
NAME            HUB ACCEPTED   MANAGED CLUSTER URLS   JOINED   AVAILABLE
AGE
local-cluster   true                                  True     True       56d
cluster1        true                                  True     True       16h
```

```
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: all-ready-clusters
  namespace: default
spec:
  clusterSelector: {}
status:
  decisions:
  - clusterName: cluster1
    clusterNamespace: cluster1
```

2. Create a **Role** in your **.yaml** file to grant the deployable administrative permission in the **local-cluster** namespace. See the following example:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
```

```
  name: deployables-admin-user-zisis
  namespace: local-cluster
rules:
- apiGroups:
  - apps.open-cluster-management.io
  resources:
  - deployables
  verbs:
  - '*'
```

3. Create a **RoleBinding** resource to grant the placement rule user access to the **local-cluster** namespace. See the following example:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: deployables-admin-user-zisis
  namespace: local-cluster
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: deployables-admin-user-zisis
  namespace: local-cluster
subjects:
- kind: User
  name: zisis
  apiGroup: rbac.authorization.k8s.io
```

1.18. TROUBLESHOOTING APPLICATION KUBERNETES DEPLOYMENT VERSION

A managed cluster with a deprecated Kubernetes **apiVersion** might not be supported. See the Kubernetes issue for more details about the deprecated API version.

1.18.1. Symptom: Application deployment version

If one or more of your application resources in the Subscription YAML file uses the deprecated API, you might receive an error similar to the following error:

```
failed to install release: unable to build kubernetes objects from release manifest: unable to recognize
"": no matches for
kind "Deployment" in version "extensions/v1beta1"
```

Or with new Kubernetes API version in your YAML file named **old.yaml** for instance, you might receive the following error:

```
error: unable to recognize "old.yaml": no matches for kind "Deployment" in version
"deployment/v1beta1"
```

1.18.2. Resolving the problem: Application deployment version

1. Update the **apiVersion** in the resource. For example, if the error displays for *Deployment* kind in the subscription YAML file, you need to update the **apiVersion** from **extensions/v1beta1** to **apps/v1**.
   See the following example:

   ```
   apiVersion: apps/v1
   kind: Deployment
   ```

2. Verify the available versions by running the following command on the managed cluster:

   ```
   kubectl explain <resource>
   ```

3. Check for **VERSION**.

1.19. TROUBLESHOOTING STANDALONE SUBSCRIPTION MEMORY

The **multicluster-operators-standalone-subscription** pod restarts regularly because of a memory issue.

1.19.1. Symptom: Standalone subscription memory

When Operator Lifecycle Manager (OLM) deploys all operators, not only the multicluster-subscription-operator, the **multicluster-operators-standalone-subscription** pod restarts because not enough memory is allocated to the standalone subscription container.

The memory limit of the **multicluster-operators-standalone-subscription** pod was increased to 2GB in the multicluster subscription community operator CSV, but this resource limit setting is ignored by OLM.

1.19.2. Resolving the problem: Standalone subscription memory

1. After installation, find the operator subscription CR that subscribes the multicluster subscription community operator. Run the following command:

   ```
   % oc get sub -n open-cluster-management acm-operator-subscription
   ```

2. Edit the operator subscription custom resource by appending the **spec.config.resources .yaml** file to define resource limits.
   Note: Do not create a new operator subscription custom resource that subscribes the same multicluster subscription community operator. Because two operator subscriptions are linked to one operator, the operator pods are **"killed"** and restarted by the two operator subscription custom resources.

   See the following updated **.yaml** file example:

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: multicluster-operators-subscription-alpha-community-operators-openshift-
   marketplace
     namespace: open-cluster-management
   spec:
     channel: release-2.2
   ```

```
  config:
    resources:
      limits:
        cpu: 750m
        memory: 2Gi
      requests:
        cpu: 150m
        memory: 128Mi
  installPlanApproval: Automatic
  name: multicluster-operators-subscription
  source: community-operators
  sourceNamespace: openshift-marketplace
```

3. After the resource is saved, ensure that the standalone subscription pod is restarted with 2GB memory limit. Run the following command:

```
% oc get pods -n open-cluster-management multicluster-operators-standalone-subscription-7c8cbf885f-c94kz -o yaml
```

```
apiVersion: v1
kind: Pod
...
spec:
  containers:
  - image: quay.io/open-cluster-management/multicluster-operators-subscription:community-2.2
...
    resources:
      limits:
        cpu: 750m
        memory: 2Gi
      requests:
        cpu: 150m
        memory: 128Mi
...
status:
  qosClass: Burstable
```

1.20. TROUBLESHOOTING KLUSTERLET WITH DEGRADED CONDITIONS

The Klusterlet degraded conditions can help to diagnose the status of Klusterlet agents on managed cluster. If a Klusterlet is in the degraded condition, the Klusterlet agents on managed cluster might have errors that need to be troubleshooted. See the following information for Klusterlet degraded conditions that are set to **True**.

1.20.1. Symptom: Klusterlet is in the degraded condition

After deploying a Klusterlet on managed cluster, the **KlusterletRegistrationDegraded** or **KlusterletWorkDegraded** condition displays a status of *True*.

1.20.2. Identifying the problem: Klusterlet is in the degraded condition

1. Run the following command on the managed cluster to view the Klusterlet status:

```
kubectl get klusterlets klusterlet -oyaml
```

2. Check **KlusterletRegistrationDegraded** or **KlusterletWorkDegraded** to see if the condition is set to **True**. Proceed to *Resolving the problem* for any degraded conditions that are listed.

1.20.3. Resolving the problem: Klusterlet is in the degraded condition

See the following list of degraded statuses and how you can attempt to resolve those issues:

- If the **KlusterletRegistrationDegraded** condition with a status of *True* and the condition reason is: *BootStrapSecretMissing*, you need create a bootstrap secret on **open-cluster-management-agent** namespace.
- If the **KlusterletRegistrationDegraded** condition displays *True* and the condition reason is a *BootstrapSecretError*, or *BootstrapSecretUnauthorized*, then the current bootstrap secret is invalid. Delete the current bootstrap secret and recreate a valid bootstrap secret on **open-cluster-management-agent** namespace.
- If the **KlusterletRegistrationDegraded** and **KlusterletWorkDegraded** displays *True* and the condition reason is *HubKubeConfigSecretMissing*, delete the Klusterlet and recreate it.
- If the **KlusterletRegistrationDegraded** and **KlusterletWorkDegraded** displays *True* and the condition reason is: *ClusterNameMissing*, *KubeConfigMissing*, *HubConfigSecretError*, or *HubConfigSecretUnauthorized*, delete the hub cluster kubeconfig secret from **open-cluster-management-agent** namespace. The registration agent will bootstrap again to get a new hub cluster kubecofnig secret.
- If the **KlusterletRegistrationDegraded** displays *True* and the condition reason is *GetRegistrationDeploymentFailed*, or *UnavailableRegistrationPod*, you can check the condition message to get the problem details and attempt to resolve.
- If the **KlusterletWorkDegraded** displays *True* and the condition reason is *GetWorkDeploymentFailed* ,or *UnavailableWorkPod*, you can check the condition message to get the problem details and attempt to resolve.

1.21. TROUBLESHOOTING KLUSTERLET APPLICATION MANAGER ON MANAGED CLUSTERS

When you upgrade from Red Hat Advanced Cluster Management for Kubernetes, the **klusterlet-addon-appmgr** pod on Red Hat OpenShift Container Platform managed clusters version 4.5 and 4.6 are **OOMKilled**.

1.21.1. Symptom: Klusterlet application manager on managed cluster

You receive an error for the **klusterlet-addon-appmgr** pod on Red Hat OpenShift Container Platform managed clusters version 4.5 and 4.6: **OOMKilled**.

1.21.2. Resolving the problem: Klusterlet application manager on managed cluster

For Red Hat Advanced Cluster Management for Kubernetes 2.1.x and 2.2, you need to manually increase the memory limit of the pod to **8Gb**. See the following steps.

1. On your hub cluster, annotate the **klusterletaddonconfig** to pause replication. See the following command:

```
oc annotate klusterletaddonconfig -n ${CLUSTER_NAME} ${CLUSTER_NAME}
klusterletaddonconfig-pause=true --  overwrite=true
```

2. On your hub cluster, scale down the **klusterlet-addon-operator**. See the following command:

```
oc edit manifestwork ${CLUSTER_NAME}-klusterlet-addon-operator -n ${CLUSTER_NAME}
```

3. Find the **klusterlet-addon-operator** Deployment and add **replicas: 0** to the spec to scale down.

```
- apiVersion: apps/v1
  kind: Deployment
  metadata:
    labels:
      app: cluster1
    name: klusterlet-addon-operator
    namespace: open-cluster-management-agent-addon
    spec:
      replicas: 0
```

   On the managed cluster, the **open-cluster-management-agent-addon/klusterlet-addon-operator** pod will be terminated.

4. Log in to the managed cluster to manually increase the memory limit in the **appmgr** pod. Run the following command:

```
% oc edit deployments -n open-cluster-management-agent-addon klusterlet-addon-appmgr
```

   For example, if the limit is 5G, increase the limit to 8G.

```
resources:
  limits:
    memory: 2Gi  -> 8Gi
  requests:
    memory: 128Mi -> 256Mi
```

1.22. TROUBLESHOOTING OBJECT STORAGE CHANNEL SECRET

If you change the **SecretAccessKey**, the subscription of an Object storage channel cannot pick up the updated secret automatically and you receive an error.

1.22.1. Symptom: Object storage channel secret

The subscription of an Object storage channel cannot pick up the updated secret automatically. This prevents the subscription operator from reconciliation and deploys resources from Object storage to the managed cluster.

1.22.2. Resolving the problem: Object storage channel secret

You need to manually input the credentials to create a secret, then refer to the secret within a channel.

1. Annotate the subscription CR in order to generate a reconcile single to subscription operator. See the following **data** specification:

```
apiVersion: apps.open-cluster-management.io/v1
kind: Channel
metadata:
  name: deva
  namespace: ch-obj
  labels:
    name: obj-sub
spec:
  type: ObjectBucket
  pathname: http://ec2-100-26-232-156.compute-1.amazonaws.com:9000/deva
  sourceNamespaces:
    - default
  secretRef:
    name: dev
---
apiVersion: v1
kind: Secret
metadata:
  name: dev
  namespace: ch-obj
  labels:
    name: obj-sub
data:
  AccessKeyID: YWRtaW4=
  SecretAccessKey: cGFzc3dvcmRhZG1pbg==
```

2. Run **oc annotate** to test:

```
oc annotate appsub -n <subscription-namespace> <subscription-name> test=true
```

After you run the command, you can go to the Application console to verify that the resource is deployed to the managed cluster. Or you can log in to the managed cluster to see if the application resource is created at the given namespace.

1.23. TROUBLESHOOTING OBSERVABILITY

After you install the observability component, the component might be stuck and an **Installing** status is displayed.

1.23.1. Symptom: MultiClusterObservability resource status stuck

If the observability status is stuck in an **Installing** status after you install and create the Observability custom resource definition (CRD), it is possible that there is no value defined for the **spec:storageConfig:storageClass** parameter. Alternatively, the observability component automatically finds the default **storageClass**, but if there is no value for the storage, the component remains stuck with the **Installing** status.

1.23.2. Resolving the problem: MultiClusterObservability resource status stuck

If you have this problem, complete the following steps:

1. Verify that the observability components are installed:

   a. To verify that the **multicluster-observability-operator**, run the following command:

   ```
   kubectl get pods -n open-cluster-management|grep observability
   ```

   b. To verify that the appropriate CRDs are present, run the following command:

   ```
   kubectl get crd|grep observ
   ```

   The following CRDs must be displayed before you enable the component:

   ```
   multiclusterobservabilities.observability.open-cluster-management.io
   observabilityaddons.observability.open-cluster-management.io
   observatoria.core.observatorium.io
   ```

2. If you create your own storageClass for a Bare Metal cluster, see How to create an NFS provisioner in the cluster or out of the cluster.

3. To ensure that the observability component can find the default storageClass, update the **storageClass** parameter in the **multicluster-observability-operator** CRD. Your parameter might resemble the following value:

```
storageclass.kubernetes.io/is-default-class: "true"
```

The observability component status is updated to a *Ready* status when the installation is complete. If the installation fails to complete, the *Fail* status is displayed.

1.24. TROUBLESHOOTING OPENSHIFT MONITORING SERVICE

Observability service in a managed cluster needs to scrape metrics from the OpenShift Container Platform monitoring stack. The **metrics-collector** is not installed if the OpenShift Container Platform monitoring stack is not ready.

1.24.1. Symptom: OpenShift monitoring service is not ready

The **endpoint-observability-operator-x** pod checks if the **prometheus-k8s** service is available in the **openshift-monitoring** namespace. If the service is not present in the **openshift-monitoring** namespace, then the **metrics-collector** is not deployed. You might receive the following error message: **Failed to get prometheus resource**.

1.24.2. Resolving the problem: OpenShift monitoring service is not ready

If you have this problem, complete the following steps:

1. Log in to your OpenShift Container Platform cluster.

2. Access the **openshift-monitoring** namespace to verify that the **prometheus-k8s** service is available.

3. Restart **endpoint-observability-operator-x** pod in the **open-cluster-management-addon-observability** namespace of the managed cluster.

1.25. UNDESIRED LABEL VALUE IN MANAGEDCLUSTER RESOURCE

When you import a managed cluster, the observability components are installed by default. Your placement rule might resemble the following information:

```
status:
  decisions:
  - clusterName: sample-managed-cluster
    clusterNamespace: sample-managed-cluster
```

If the managed cluster is not included in the placement rule, the observability components are not installed.

1.25.1. Symptom: Undesired label value in managedcluster resource

If you find that the imported cluster is not included, the observability service for your managed cluster resource might be disabled.

Remember: When you enable the service, the **vendor:OpenShift** label is added to represent the target managed cluster. Observability service is only supported on OpenShift Container Platform managed cluster.

1.25.2. Resolving the problem: Undesired label value in managedcluster resource

If you have this problem, enable the observability service for the target managed cluster and update labels in the **managedcluster** resource.

Complete the following steps:

1. Log in to your Red Hat Advanced Cluster Management cluster.
2. Change the **observability** parameter value to **enabled** by updating the placement rule. Run the following command:

   ```
   oc edit placementrule -n open-cluster-management-observability
   ```

3. Verify that OpenShift is listed as vendor for the target managed cluster by running the following command:

   ```
   oc get managedcluster <CLUSTER NAME> -o yaml
   ```

   Update the **metadata.labels.vendor** parameter value to **OpenShift**.

1.26. TROUBLESHOOTING SEARCH AGGREGATOR POD STATUS

The **search-aggregator** fail to run.

1.26.1. Symptom 1: Search aggregator pod in Not Ready state

Search aggregator pods are in a **Not Ready** state if the **redisgraph-user-secret** is updated. You might receive the following error:

```
E0113 15:04:42.427931       1 pool.go:93] Error authenticating Redis client. Original error: ERR invalid
```

```
password
W0113 15:04:42.428100       1 healthProbes.go:36] Unable to reach Redis.
E0113 15:04:44.265777       1 pool.go:93] Error authenticating Redis client. Original error: ERR invalid
password
W0113 15:04:44.266003       1 healthProbes.go:36] Unable to reach Redis.
E0113 15:04:46.316869       1 pool.go:93] Error authenticating Redis client. Original error: ERR invalid
password
W0113 15:04:46.317029       1 healthProbes.go:36] Unable to reach Redis.
```

1.26.2. Resolving the problem: Search aggregator pod in Not Ready state

If you have this problem, delete the **search-aggregator** and **search-api** pods to restart the pods. Run the following commands to delete the previously mentioned pods.

```
oc delete pod -n open-cluster-management <search-aggregator>

oc delete pod -n open-cluster-management <search-api>
```

1.26.3. Symptom 2: Search redisgraph pod in Pending state

The **search-redisgraph** pod fail to run when it is in **Pending** state.

1.26.4. Resolving the problem: Search redisgraph pod in Pending state

If you have this problem complete the following steps:

1. Check the pod events on the hub cluster namespace with the following command:

   ```
   oc describe pod search-redisgraph-0
   ```

2. If you have created a **searchcustomization** CR, check if the storage class and storage size is valid, and check if a PVC can be created. List the PVC by running the following command:

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

3. Make sure the PVC can be bound to the **search-redisgraph-0** pod. If the problem is still not resolved , delete the StatefulSet **search-redisgraph**. The search operator recreates the StatefulSet. Run the following command:

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

1.27. TROUBLESHOOTING METRICS-COLLECTOR

When the **observability-client-ca-certificate** secret is not refreshed in the managed cluster, you might receive an internal server error.

1.27.1. Symptom: metrics-collector cannot verify observability-client-ca-certificate

There might be a managed cluster, where the metrics are unavailable. If this is the case, you might receive the following error from the **metrics-collector** deployment:

```
error: response status code is 500 Internal Server Error, response body is x509: certificate signed by unknown authority (possibly because of "crypto/rsa: verification error" while trying to verify candidate authority certificate "observability-client-ca-certificate")
```

1.27.2. Resolving the problem: metrics-collector cannot verify observability-client-ca-certificate

If you have this problem, complete the following steps:

1. Log in to your managed cluster.
2. Delete the secret named, **observability-controller-open-cluster-management.io-observability-signer-client-cert** that is in the **open-cluster-management-addon-observability** namespace. Run the following command:

   ```
   oc delete observability-controller-open-cluster-management.io-observability-signer-client-cert -n open-cluster-management-addon-observability
   ```

   Note: The **observability-controller-open-cluster-management.io-observability-signer-client-cert** is automatically recreated with new certificates.

The **metrics-collector** deployment is recreated and the **observability-controller-open-cluster-management.io-observability-signer-client-cert** secret is updated.