Red Hat Advanced Cluster Management for Kubernetes 2.0

Troubleshooting

Troubleshooting

Last Updated: 2021-01-14

Red Hat Advanced Cluster Management for Kubernetes 2.0 Troubleshooting

Troubleshooting

法律通告

摘要

Troubleshooting in Red Hat Advanced Cluster Management for Kubernetes

目录

第 1 章 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.

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

1.1. MUST-GATHER

To get started, learn about the troubleshooting scenarios for users to run the **must-gather** command to debug the issues.

- **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.

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 Red Hat OpenShift Container Platform: Gathering data.

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-spoke.log** file that is located in the the **cluster-scoped-resources** directory:

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

   Check for managed clusters (spoke 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:v2.0.0 --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. 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 reinstallation failure
- Troubleshooting failed uninstallation because resources exist

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 all imported clusters offline after certificate change
- Troubleshooting a failed imported cluster secret after upgrade
- Troubleshooting OpenShift Container Platform version 3.11 cluster import failure
- Troubleshooting cluster in console with pending or failed status

Application management

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

- Troubleshooting application Kubernetes deployment version .

Governance and risk

To get to the original security guide, view Security.

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 the search collector pod

1.3. TROUBLESHOOTING REINSTALLATION FAILURE

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

1.3.1. Symptom: Reinstallation failure

If your pods do not start after you install Red Hat Advanced Cluster Management for Kubernetes, 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.

1.3.2. 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 CLI in the Red Hat OpenShift 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 v1.admission.cluster.open-cluster-management.io
v1beta1.webhook.certmanager.k8s.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
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
```

   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 .

1.4. TROUBLESHOOTING FAILED UNINSTALLATION BECAUSE RESOURCES EXIST

1.4.1. Symptom: Failed uninstallation because resources exist

When you uninstall Red Hat Advanced Cluster Management for Kubernetes, the installation fails with the following error message:

```
Cannot delete MultiClusterHub resource because ManagedCluster resource(s) exist
```

1.4.2. Resolving the problem: Failed uninstallation because resources exist

This error occurs when you try to uninstall the Red Hat Advanced Cluster Management hub cluster while it is still managing clusters. All clusters must be removed from management before uninstalling the hub cluster.

Detach all clusters that are still being managed by the hub cluster and try to uninstall again.

For more information about detaching clusters, see the *Removing a cluster from management* section by selecting the information for your provider in Creating a cluster with Red Hat Advanced Cluster Management for Kubernetes.

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 A FAILED IMPORTED CLUSTER SECRET AFTER UPGRADE

1.6.1. Symptom: Troubleshooting a failed imported cluster secret after upgrade

After an upgrade from Red Hat Advanced Cluster Management for Kubernetes version 2.0.0 to version 2.0.1, your cluster import in the Red Hat Advanced Cluster Management console might fail with the following message:

```
Failed to fetch import yaml secret
```

1.6.2. Identifying the problem: Troubleshooting a failed imported cluster secret after upgrade

To confirm that the issue can be resolved by the steps that follow, complete the following steps:

1. Run the following command to change to your Red Hat Advanced Cluster Management installation namespace:

```
oc project <namespace>
```

   Replace *<namespace>* with your Red Hat Advanced Cluster Management installation namespace. If you used the default value, it is **open-cluster-management**.

2. Run the following command to determine if the **managedcluster-import-controller** has the required permission:

```
oc get $(oc get clusterrole -o name | grep managedcluster-import-controller) -o yaml| grep apiservers
```

   If the command returns an empty response, then complete the step in the *Resolving the problem* section to fix the problem.

1.6.3. Resolving the problem: Troubleshooting a failed imported cluster secret after upgrade

To resolve the problem, enter the following command to restart the **multicluster-operators-standalone-subscription** service:

```
oc delete $(oc get pod -o name | grep multicluster-operators-standalone-subscription)
```

1.7. 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.

1.7.1. 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*.

1.7.2. 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>
```

   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**.

1.7.3. 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*.

1.8. TROUBLESHOOTING ALL IMPORTED CLUSTERS OFFLINE AFTER CERTIFICATE CHANGE

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

1.8.1. Symptom: All clusters offline after certificate change

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

1.8.2. Identifying the problem: All clusters offline after certificate change

After updating the information for a custom API server certificate, the 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
```

1.8.3. Resolving the problem: All 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
   ```

2. Delete the outdated secret on the managed cluster to make sure the **registration-agent** uses the latest bootstrap secret to recreate secrets:

   ```
   oc delete secret hub-kubeconfig-secret -n open-cluster-management-agent
   ```

3. Restart all pods in the **open-cluster-management-agent** namespace:

   ```
   oc delete po —all -n open-cluster-management-agent
   ```

4. Wait for 2-3 minutes for the cluster to connect, and for the **work-manager** to start.

5. Restart all pods in **open-cluster-management-agent-addon** namespace:

   ```
   oc delete po —all -n open-cluster-management-agent-addon
   ```

The pods stop and use the new certificate information as they restart.

1.9. 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.

1.9.1. 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.

1.9.2. 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.

1.9.3. 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 with Red Hat Advanced Cluster Management for Kubernetes for more information about creating a cluster.

1.10. TROUBLESHOOTING OPENSHIFT CONTAINER PLATFORM VERSION 3.11 CLUSTER IMPORT FAILURE

1.10.1. 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|...
```

1.10.2. 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*.

1.10.3. 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
     ```

1.11. 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.

1.11.1. 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 similiar 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"
```

1.11.2. 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**.

1.12. TROUBLESHOOTING THE SEARCH COLLECTOR POD

The **search-collector** crashes and the status displayed is a **CrashLoopback** error.

1.12.1. Symptom: Reinstallation failure

Afer you install IBM Cloud Pak for Multicloud Manager, the **search-collector** status displays a **CrashLoopback** error.

1.12.2. Resolving the problem: Reinstallation failure

You must increase the memory limit in the **search-collector** deployment. Complete the following steps to increase the memory limit:

1. Log in to your Red Hat OpenShift Container Platform hub cluster.

2. Access the **search-collector** deployment by running the following command:

```
oc edit deployment $(oc get deployment -l component=search-collector -o
jsonpath='{.items[0].metadata.name}')
```

3. Edit the container's resource limit for memory using the **.spec.template.spec.containers.resources.limits** field:

```
spec:
  template:
    spec:
      containers:
        resources:
          limits:
            memory: 863Mi
          requests:
            cpu: 25m
            memory: 64Mi
```

4. Apply and save changes to your deployment.

The memory limits for the **search-collector** deployment are increased.