



# Red Hat Advanced Cluster Management for Kubernetes 2.10

故障排除

故障排除



故障排除

## 法律通告

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## 摘要

查看集群的故障排除主题列表。您还可以使用 `must-gather` 命令来收集日志。

# 目录

<b>第 1 章 故障排除</b> .....	<b>3</b>
1.1. 记录的故障排除	3
1.2. 运行 MUST-GATHER 命令进行故障排除	5
1.3. 安装状态故障排除处于安装或待处理状态	7
1.4. 重新安装失败的故障排除	8
1.5. 在 RED HAT ADVANCED CLUSTER MANAGEMENT 升级后对 OCM-CONTROLLER 进行故障排除	9
1.6. 离线集群的故障排除	11
1.7. 受管集群导入失败的故障排除	11
1.8. 对带有待处理导入状态的集群进行故障排除	12
1.9. 对带有已存在错误的集群的故障排除	13
1.10. VMWARE VSPHERE 上创建集群的故障排除	13
1.11. 受管集群在 RED HAT OPENSTACK PLATFORM 上失败，并带有 UNKNOWN AUTHORITY 错误	16
1.12. OPENSIFT CONTAINER PLATFORM 版本 3.11 集群导入失败的故障排除	17
1.13. 证书更改后导入的集群离线故障排除	18
1.14. 删除集群后命名空间会保留	20
1.15. 导入集群时出现自动 AUTO-IMPORT-SECRET-EXISTS 错误	21
1.16. 对 VOLSYNC 的 CINDER CONTAINER STORAGE INTERFACE (CSI)驱动程序进行故障排除	21
1.17. 运行 MUST-GATHER 命令进行故障排除	22
1.18. 访问置备的 POSTGRESQL 数据库以进行故障排除	23
1.19. 使用数据库转储和恢复进行故障排除	25
1.20. 恢复合规性数据	26
1.21. 集群状态从离线变为可用的故障排除	27
1.22. 集群在控制台带有待处理或失败状态的故障排除	28
1.23. GRAFANA 故障排除	31
1.24. 未使用放置规则选择本地集群的故障排除	32
1.25. 对应用程序 KUBERNETES 部署版本进行故障排除	33
1.26. 带有降级条件的 KLUSTRLET 故障排除	34
1.27. OBJECT STORAGE 频道 SECRET 故障排除	36
1.28. 对可观察性功能进行故障排除	37
1.29. OPENSIFT 监控服务故障排除	38
1.30. METRICS-COLLECTOR 故障排除	39
1.31. POSTGRESQL 共享内存错误故障排除	39
1.32. 安装后无法连接 SUBMARINER 故障排除	41
1.33. SUBMARINER ADD-ON 状态降级的故障排除	42
1.34. 恢复状态故障排除完成并显示错误	45
1.35. 恢复 HUB 集群备份时会删除通用资源	46
1.36. 对多行 YAML 解析进行故障排除	47



# 第 1 章 故障排除

在使用故障排除指南前，您可以运行 `oc adm must-gather` 命令来收集详情、日志和步骤来调试问题。如需了解更多详细信息，请参阅[运行 must-gather 命令进行故障排除](#)。

另外，查看您基于角色的访问。详情请参阅[基于角色的访问控制](#)。

## 1.1. 记录的故障排除

查看与 Red Hat Advanced Cluster Management for Kubernetes 故障排除相关的主题：

### 安装

要查看安装任务的主要文档，请参阅[安装和升级](#)。

- [安装状态故障排除处于安装或待处理状态](#)
- [重新安装失败的故障排除](#)
- [在 Red Hat Advanced Cluster Management 升级后对 ocm-controller 进行故障排除](#)

### 备份和恢复

要查看备份和恢复的主要文档，请参阅[备份和恢复](#)。

- [恢复状态故障排除完成并显示错误](#)
- [恢复 hub 集群备份时会删除通用资源](#)

### 集群管理

要查看有关管理集群的主要文档，请参阅 `multicluster engine operator` [集群生命周期概述](#)。

- [离线集群的故障排除](#)
- [受管集群导入失败的故障排除](#)
- [对带有待处理导入状态的集群进行故障排除](#)
- [证书更改后导入的集群离线故障排除](#)
- [集群状态从离线变为可用的故障排除](#)
- [VMware vSphere 上创建集群的故障排除](#)
- [集群在控制台中带有待处理或失败状态的故障排除](#)
- [OpenShift Container Platform 版本 3.11 集群导入失败的故障排除](#)
- [带有降级条件的 Klusterlet 故障排除](#)
- [Object storage 频道 secret 故障排除](#)
- [删除集群后命名空间会保留](#)
- [导入集群时出现自动 Auto-import-secret-exists 错误](#)

- [对 VolSync 的 cinder Container Storage Interface \(CSI\)驱动程序进行故障排除](#)
- [对集群代理附加组件进行故障排除，忽略附加组件容限](#)

### 多集群全局 hub

要查看多集群全局 hub 的主要文档，请参阅 [{global-hub}](#)。

- [运行 must-gather 命令进行故障排除](#)
- [访问置备的 postgres 数据库以进行故障排除](#)
- [使用数据库转储和恢复进行故障排除](#)
- [恢复合规性数据](#)

### 应用程序管理

要查看有关应用程序管理的主要文档，请参阅[管理应用程序](#)。

- [对应用程序 Kubernetes 部署版本进行故障排除](#)
- [未选择本地集群的故障排除](#)

### 监管

- [对多行 YAML 解析进行故障排除](#)

要查看安全指南，请参阅[风险和合规性](#)。

### 控制台可观察性

控制台观察功能包括搜索，以及标头和导航功能。要查看可观察性指南，请参阅[控制台的观察性](#)。

- [grafana 故障排除](#)
- [对可观察性功能进行故障排除](#)
- [OpenShift 监控服务故障排除](#)
- [metrics-collector 故障排除](#)
- [PostgreSQL 共享内存错误故障排除](#)

### Submariner 网络和服务发现

本节列出了在 Red Hat Advanced Cluster Management 或 multicluster engine operator 中使用 Submariner 故障排除步骤。有关常规 Submariner 故障排除信息，请参阅 Submariner 文档中的[故障排除部分](#)。

要查看 Submariner 网络服务和发现的主要文档，请参阅 [Submariner multicluster networking 和 service discovery](#)。

- [安装后无法连接 Submariner 的故障排除 - 常规信息](#)
- [Submariner add-on 状态降级的故障排除](#)

## 1.2. 运行 MUST-GATHER 命令进行故障排除

要进行故障排除，参阅可以使用 `must-gather` 命令进行调试的用户情景信息，然后使用这个命令进行故障排除。

需要的访问权限：集群管理员

### 1.2.1. must-gather 情境

- **场景 1**：如果您的问题已被记录，使用 [已记录的故障排除](#) 文档部分进行解决。这个指南按照产品的主要功能进行组织。  
在这种情况下，您可以参阅本指南来查看您的问题的解决方案是否在文档中。例如，在创建集群时出现问题，您可能在这个指南的 [管理集群](#) 部分中找到解决方案。
- **情况 2**：如果这个指南中没有与您的问题相关的内容，运行 `must-gather` 命令并使用输出来调试问题。
- **情况 3**：无法使用 `must-gather` 命令的输出结果无法帮助解决您的问题，请向红帽支持提供您的输出。

### 1.2.2. must-gather 过程

请参阅以下流程来使用 `must-gather` 命令：

1. 了解 `must-gather` 命令以及按照 Red Hat OpenShift Container Platform 文档中的 [收集集群数据](#) 所需的先决条件。
2. 登录到您的集群。添加用于收集数据和目录的 Red Hat Advanced Cluster Management for Kubernetes 镜像。运行以下命令，在其中提供您要插入的镜像和输出目录：

```
oc adm must-gather --image=registry.redhat.io/rhacm2/acm-must-gather-rhel9:v2.10 --dest-dir=<directory>
```

3. 对于通常的用例，在登录到您的 `hub` 集群时运行 `must-gather` 命令。  
**注**：如果要检查您的受管集群，找到位于 `cluster-scoped-resources` 目录中的 `gather-managed.log` 文件：

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

检查 JOINED 和 AVAILABLE 栏没有被设置为 **True** 的受管集群。您可以在这些没有以 **True** 状态连接的集群中运行 `must-gather` 命令。

4. 进入您指定的目录查看输出。输出以以下级别进行组织：
  - 两个对等级别：`cluster-scoped-resources` 和 `namespace` 资源。
  - 每个对等级别下的子类：用于 `cluster-scope` 和 `namespace-scoped` 资源的自定义资源定义的 API 组。
  - 每个子类的下一级：按 `kind` 进行排序的 YAML 文件。

### 1.2.3. 在断开连接的环境中的 must-gather

在断开连接的环境中，按照以下步骤运行 `must-gather` 命令：

1. 在断开连接的环境中，将 RedHat operator 目录镜像镜像（mirror）到其 mirror registry 中。如需更多信息，请参阅[在断开连接的网路中安装](#)。
2. 运行以下命令收集所有信息，将 `<2.x>` 替换为 `&lt; acm-must-gather &gt;` 支持的版本，如 `2.10`，以及 `< multicluster-engine/must-gather >`，如 `2.5`。

```
REGISTRY=<internal.repo.address:port>
IMAGE1=$REGISTRY/rhacm2/acm-must-gather-rhel9:v<2.x>
oc adm must-gather --image=$IMAGE1 --dest-dir=<directory>
```

如果您在当前支持的某个版本或产品文档时遇到问题，请访问 [红帽支持](#)，您可以在其中进行故障排除、查看知识库文章、与支持团队连接，或者创建一个问题单。您必须使用您的红帽凭证登录。

## 1.2.4. 托管集群的 must-gather

如果托管 control plane 集群出现问题，您可以运行 `must-gather` 命令来收集信息，以帮助您进行故障排除。

### 1.2.4.1. 关于托管集群的 must-gather 命令

该命令为受管集群和托管集群生成输出。

- 多集群引擎 operator hub 集群中的数据：
  - 集群范围的资源：这些资源是管理集群的节点定义。
  - `hypershift-dump` 压缩文件：如果您需要与其他人员共享内容，该文件非常有用。
  - 命名空间资源：这些资源包括来自相关命名空间中的所有对象，如配置映射、服务、事件和日志。
  - 网络日志：这些日志包括 OVN 北向和南向数据库和每个数据库的状态。
  - 托管的集群：此级别的输出涉及托管集群内的所有资源。
- 来自托管集群的数据：
  - 集群范围的资源：这些资源包含所有集群范围的对象，如节点和 CRD。
  - 命名空间资源：这些资源包括来自相关命名空间中的所有对象，如配置映射、服务、事件和日志。

虽然输出不包含集群中的任何 secret 对象，但它可以包含对 secret 的名称的引用。

### 1.2.4.2. 先决条件

要通过运行 `must-gather` 命令来收集信息，您必须满足以下条件：

- 您必须确保 `kubeconfig` 文件已被加载，并指向 multicluster engine operator hub 集群。
- 您必须具有对 multicluster engine operator hub 集群的 `cluster-admin` 访问权限。
- 您必须具有 `HostedCluster` 资源的 `name` 值，以及部署自定义资源的命名空间。

### 1.2.4.3. 为托管集群输入 must-gather 命令

1. 输入以下命令收集有关托管集群的信息。在命令中，**hosted-cluster-namespace=HOSTEDCLUSTERNAMESPACE** 参数是可选的；如果没有包括它，则命令会像在 default 命名空间（即 **集群**）中运行。

```
oc adm must-gather --image=quay.io/stolostron/backplane-must-gather:SNAPSHOTNAME
/usr/bin/gather hosted-cluster-namespace=HOSTEDCLUSTERNAMESPACE hosted-cluster-
name=HOSTEDCLUSTERNAME
```

2. 要将命令的结果保存到压缩文件中，请包含 **--dest-dir= NAME** 参数，将 NAME 替换为您要保存结果的目录的名称：

```
oc adm must-gather --image=quay.io/stolostron/backplane-must-gather:SNAPSHOTNAME
/usr/bin/gather hosted-cluster-namespace=HOSTEDCLUSTERNAMESPACE hosted-cluster-
name=HOSTEDCLUSTERNAME --dest-dir=NAME ; tar -cvzf NAME.tgz NAME
```

#### 1.2.4.4. 在断开连接的环境中输入 **must-gather** 命令

在断开连接的环境中，按照以下步骤运行 **must-gather** 命令：

1. 在断开连接的环境中，将 RedHat operator 目录镜像镜像（mirror）到其 mirror registry 中。如需更多信息，请参阅 [在断开连接的网络中安装](#)。
2. 运行以下命令以提取日志，从其 mirror registry 中引用镜像：

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

oc adm must-gather --image=$IMAGE /usr/bin/gather hosted-cluster-
namespace=HOSTEDCLUSTERNAMESPACE hosted-cluster-
name=HOSTEDCLUSTERNAME --dest-dir=./data
```

#### 1.2.4.5. 其他资源

- 有关托管 control plane 故障排除的更多信息，请参阅 OpenShift Container Platform 文档中的 [对托管的 control plane 进行故障排除](#)。

### 1.3. 安装状态故障排除处于安装或待处理状态

在安装 Red Hat Advanced Cluster Management 时，**MultiClusterHub** 会一直处于 **Installing** 阶段，或多个 pods 一直处于 **Pending** 状态。

#### 1.3.1. 症状：一直处于 **Pending** 状态

安装 **MultiClusterHub** 后已超过了十分钟，一个或多个来自 **MultiClusterHub** 资源的 **status.components** 字段的组件报告 **ProgressDeadlineExceeded**。这可能是集群中的资源限制的问题。

检查安装了 **Multiclusterhub** 的命名空间中的 pod。您可能会看到 **Pending** 状态，如下所示：

```
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.'
```

在这种情况下，集群中 worker 节点资源不足以运行该产品。

### 1.3.2. 解决问题：调整 worker 节点大小

如果您有这个问题，则需要使用更大或多个 worker 节点来更新集群。如需了解[调整集群大小](#)的信息，请参阅[调整集群大小](#)。

## 1.4. 重新安装失败的故障排除

在重新安装 Red Hat Advanced Cluster Management for Kubernetes 时，pod 没有启动。

### 1.4.1. 症状：重新安装失败

在安装 Advanced Cluster Management 后，如果 pod 没有启动，这很可能是因为以前已安装了 Red Hat Advanced Cluster Management，在您进行最新安装时以前安装的一些组件没有被删除。

在本例中，pod 在完成安装过程后没有启动。

### 1.4.2. 解决问题：重新安装失败

如果您有这个问题，请完成以下步骤：

1. 根据[卸载](#)中介绍的步骤，执行卸载过程来删除当前的组件。
2. 按照[安装 Helm](#) 中的内容，安装 Helm CLI 二进制版本 3.2.0 或更新版本。
3. 确保您的 Red Hat OpenShift Container Platform CLI 被配置为运行 **oc** 命令。如需有关如何配置 **oc** 命令的更多信息，请参阅 OpenShift Container Platform 文档中的 OpenShift [CLI 入门](#)。
4. 将以下脚本复制到一个文件中：

```
#!/bin/bash
ACM_NAMESPACE=<namespace>
oc delete mch --all -n $ACM_NAMESPACE
oc delete apiservice v1.admission.cluster.open-cluster-management.io
v1.admission.work.open-cluster-management.io
oc delete clusterimageset --all
oc delete clusterrole multiclusterengines.multicluster.openshift.io-v1-admin
multiclusterengines.multicluster.openshift.io-v1-crdview
multiclusterengines.multicluster.openshift.io-v1-edit
multiclusterengines.multicluster.openshift.io-v1-view open-cluster-
management:addons:application-manager open-cluster-management:admin-aggregate open-
cluster-management:cert-policy-controller-hub open-cluster-management:cluster-manager-
admin-aggregate open-cluster-management:config-policy-controller-hub open-cluster-
management:edit-aggregate open-cluster-management:iam-policy-controller-hub open-
cluster-management:policy-framework-hub open-cluster-management:view-aggregate
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
discoveredclusters.discovery.open-cluster-management.io discoveryconfigs.discovery.open-
cluster-management.io
oc delete mutatingwebhookconfiguration ocm-mutating-webhook
managedclustermutators.admission.cluster.open-cluster-management.io multicluster-
```

```

observability-operator
oc delete validatingwebhookconfiguration
channels.apps.open.cluster.management.webhook.validator application-webhook-validator
multiclusterhub-operator-validating-webhook ocm-validating-webhook multicluster-
observability-operator multiclusterengines.multicluster.openshift.io

```

将脚本中的 **<namespace>** 替换为安装 Red Hat Advanced Cluster Management 的命名空间的名称。确保指定正确的命名空间，因为命名空间会被清理和删除。

5. 运行该脚本以删除以前安装中的内容。
6. 运行安装。参阅[在线安装](#)

## 1.5. 在 RED HAT ADVANCED CLUSTER MANAGEMENT 升级后对 OCM-CONTROLLER 进行故障排除

从 2.7.x 升级到 2.8.x 后，**multicluster-engine** 命名空间的 **ocm-controller** 会崩溃。

### 1.5.1. 症状：在 Red Hat Advanced Cluster Management 升级后对 ocm-controller 进行故障排除

尝试列出 **ManagedClusterSet** 和 **ManagedClusterSetBinding** 自定义资源定义后，会出现以下出错信息：

```

Error from server: request to convert CR from an invalid group/version: cluster.open-cluster-
management.io/v1beta1

```

前面的消息表示 **ManagedClusterSets** 和 **ManagedClusterSetBindings** 自定义资源定义从 **v1beta1** 迁移到 **v1beta2** 失败。

### 1.5.2. 解决问题：在 Red Hat Advanced Cluster Management 升级后对 ocm-controller 进行故障排除

要解决这个错误，您必须手动启动 API 迁移。完成以下步骤：

1. 将 **cluster-manager** 恢复到以前的版本。
  - a. 使用以下命令暂停 **multiclusterengine**：

```
oc annotate mce multiclusterengine pause=true
```

- b. 运行以下命令，将 **cluster-manager** 部署的镜像替换为之前的版本：

```

oc patch deployment cluster-manager -n multicluster-engine -p \ '{"spec":{"template":
{"spec":{"containers":[{"name":"registration-
operator","image":"registry.redhat.io/multicluster-engine/registration-operator-
rhel8@sha256:35999c3a1022d908b6fe30aa9b85878e666392dbbd685e9f3edcb83e3336d
19f"}]}}}'
export ORIGIN_REGISTRATION_IMAGE=$(oc get clustermanager cluster-manager -o
jsonpath='{.spec.registrationImagePullSpec}')

```

- c. 将 **ClusterManager** 资源中的注册镜像引用替换为以前的版本。运行以下命令：

```
oc patch clustermanager cluster-manager --type=json' -p='[{"op": "replace", "path":
"/spec/registrationImagePullSpec", "value": "registry.redhat.io/multicluster-
engine/registration-
rhel8@sha256:a3c22aa4326859d75986bf24322068f0aff2103cccc06e1001faaf79b939051
5"}]'
```

- 运行以下命令，将 **ManagedClusterSets** 和 **ManagedClusterSetBindings** 自定义资源定义恢复到之前的版本：

```
oc annotate crds managedclustersets.cluster.open-cluster-management.io operator.open-
cluster-management.io/version-
oc annotate crds managedclustersetbindings.cluster.open-cluster-management.io
operator.open-cluster-management.io/version-
```

- 重启 **cluster-manager** 并等待重新创建自定义资源定义。运行以下命令：

```
oc -n multicluster-engine delete pods -l app=cluster-manager
oc wait crds managedclustersets.cluster.open-cluster-management.io --for=jsonpath=
{.metadata.annotations[operator\open-cluster-management\io\version]}="2.3.3" --
timeout=120s
oc wait crds managedclustersetbindings.cluster.open-cluster-management.io --
for=jsonpath="{.metadata.annotations[operator\open-cluster-
management\io\version]}="2.3.3" --timeout=120s
```

- 使用以下命令启动存储版本迁移：

```
oc patch StorageVersionMigration managedclustersets.cluster.open-cluster-management.io -
-type=json' -p='[{"op": "replace", "path": "/spec/resource/version", "value": "v1beta1"}]
oc patch StorageVersionMigration managedclustersets.cluster.open-cluster-management.io -
-type=json' --subresource status -p='[{"op": "remove", "path": "/status/conditions"}]
oc patch StorageVersionMigration managedclustersetbindings.cluster.open-cluster-
management.io --type=json' -p='[{"op": "replace", "path": "/spec/resource/version",
"value": "v1beta1"}]
oc patch StorageVersionMigration managedclustersetbindings.cluster.open-cluster-
management.io --type=json' --subresource status -p='[{"op": "remove",
"path": "/status/conditions"}]'
```

- 运行以下命令等待迁移完成：

```
oc wait storageversionmigration managedclustersets.cluster.open-cluster-management.io --
for=condition=Succeeded --timeout=120s
oc wait storageversionmigration managedclustersetbindings.cluster.open-cluster-
management.io --for=condition=Succeeded --timeout=120s
```

- 将 **cluster-manager** 恢复回 Red Hat Advanced Cluster Management 2.10。它可能需要几分钟时间。运行以下命令：

```
oc annotate mce multiclusterengine pause-
oc patch clustermanager cluster-manager --type=json' -p='[{"op": "replace", "path":
"/spec/registrationImagePullSpec", "value": "$ORIGIN_REGISTRATION_IMAGE"}]'
```

### 1.5.2.1. 验证

要验证 Red Hat Advanced Cluster Management 是否已恢复，请运行以下命令：

```
oc get managedclusterset
oc get managedclustersetbinding -A
```

运行命令后，**ManagedClusterSets** 和 **ManagedClusterSetBindings** 资源会在没有错误消息的情况下列出。

## 1.6. 离线集群的故障排除

一些常见的原因会导致集群显示离线状态。

### 1.6.1. 症状：集群状态为离线

完成创建集群的步骤后，您无法从 Red Hat Advanced Cluster Management 控制台访问集群，集群的状态为**离线 (offline)**。

### 1.6.2. 解决问题：集群状态为离线

1. 确定受管集群是否可用。您可以在 Red Hat Advanced Cluster Management 控制台的 *Clusters* 区域中进行检查。  
如果不可用，请尝试重启受管集群。
2. 如果受管集群状态仍处于离线状态，完成以下步骤：
  - a. 在 hub 集群上运行 **oc get managedcluster <cluster\_name> -o yaml** 命令。将 **<cluster\_name>** 替换为集群的名称。
  - b. 找到 **status.conditionss** 部分。
  - c. 检查 **type: ManagedClusterConditionAvailable** 信息并解决相关的问题。

## 1.7. 受管集群导入失败的故障排除

如果集群导入失败，您可以执行一些步骤来确定集群导入失败的原因。

### 1.7.1. 症状：导入的集群不可用

完成导入集群的步骤后，您无法从 Red Hat Advanced Cluster Management for Kubernetes 控制台访问它。

### 1.7.2. 解决问题：导入的集群不可用

在尝试导入集群后，有几个可能的原因会导致导入集群的不可用。如果集群导入失败，请完成以下步骤，直到找到失败导入的原因：

1. 在 Red Hat Advanced Cluster Management hub 集群中，运行以下命令，以确保 Red Hat Advanced Cluster Management 导入控制器正在运行。

```
kubectl -n multicluster-engine get pods -l app=managedcluster-import-controller-v2
```

您应该会看到两个正在运行的 pod。如果任何一个 pod 没有运行，请运行以下命令来查看日志以确定原因：

```
kubectl -n multicluster-engine logs -l app=managedcluster-import-controller-v2 --tail=-1
```

2. 在 Red Hat Advanced Cluster Management hub 集群中，运行以下命令以确定受管集群导入 secret 是否由 Red Hat Advanced Cluster Management 导入控制器成功生成：

```
kubectl -n <managed_cluster_name> get secrets <managed_cluster_name>-import
```

如果导入 secret 不存在，请运行以下命令来查看导入控制器的日志条目，并确定它没有被创建的原因：

```
kubectl -n multicluster-engine logs -l app=managedcluster-import-controller-v2 --tail=-1 | grep importconfig-controller
```

3. 在 Red Hat Advanced Cluster Management hub 集群中，如果您的受管集群是 **local-cluster**，或者由 Hive 置备，或者具有 auto-import secret，请运行以下命令来检查受管集群的导入状态。

```
kubectl get managedcluster <managed_cluster_name> -o=jsonpath='{range .status.conditions[*]}.{type}"{"t"}{.status}"{"t"}{.message}"{"n"}{end}' | grep ManagedClusterImportSucceeded
```

如果 **ManagedClusterImportSucceeded** 条件为 **true**，则命令的结果表示失败的原因。

4. 检查受管集群的 Klusterlet 状态是否有降级条件。请参阅 [带有降级条件的 Klusterlet 故障排除](#)，以查找 Klusterlet 被降级的原因。

## 1.8. 对带有待处理导入状态的集群进行故障排除

如果在集群的控制台上持续接收到 *Pending import*（待处理到）信息时，请按照以下步骤排除此问题。

### 1.8.1. 症状：集群处于待处理导入状态

在使用 Red Hat Advanced Cluster Management 控制台导入一个集群后，出现在控制台中的集群带有 *Pending import* 状态。

### 1.8.2. 鉴别问题：集群处于待处理导入状态

1. 在受管集群中运行以下命令查看有问题的 Kubernetes pod 的名称：

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

2. 在受管集群中运行以下命令查找错误的日志条目：

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

把 *registration\_agent\_pod* 替换为在第 1 步中获得的 pod 名称。

3. 在返回的结果中搜索显示有网络连接问题的内容。示例包括：**no such host**。

### 1.8.3. 解决问题：集群处于待处理导入状态

1. 通过在 hub 集群中输入以下命令来检索有问题的端口号：

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

2. 确保来自受管集群的主机名可以被解析，并确保建立到主机和端口的出站连接。  
如果无法通过受管集群建立通信，集群导入就不完整。受管集群的集群状态将会是 *Pending import*。

## 1.9. 对带有已存在错误的集群的故障排除

如果您无法将 OpenShift Container Platform 集群导入到 Red Hat Advanced Cluster Management **MultiClusterHub** 并收到 **AlreadyExists** 错误，请按照以下步骤排除此问题。

### 1.9.1. 症状：导入 OpenShift Container Platform 集群时的 **Already exists** 错误日志

将 OpenShift Container Platform 集群导入到 Red Hat Advanced Cluster Management **MultiClusterHub** 时会显示错误日志：

```
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."
```

### 1.9.2. 鉴别问题：导入 OpenShift Container Platform 集群时存在 **Already**

运行以下命令，检查您要导入的集群中的任何与 Red Hat Advanced Cluster Management **MultiClusterHub** 相关的资源：

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

### 1.9.3. 解决问题：导入 OpenShift Container Platform 集群时存在 **Already**

运行以下命令以删除预先存在的资源：

```
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": []}]'
```

## 1.10. VMWARE VSPHERE 上创建集群的故障排除

如果您在 VMware vSphere 上创建 Red Hat OpenShift Container Platform 集群时遇到问题，请查看以下故障排除信息以查看它们是否解决了您的问题。

注：当集群创建过程在 VMware vSphere 上失败时，您将无法使用该链接来查看日志。如果发生这种情况，您可以通过查看 **thehive-controllers** pod 的日志来找出问题。**hive-controllers** 日志位于 **hive** 命名空间中。

### 1.10.1. 受管集群创建失败并显示证书 IP SAN 错误

#### 1.10.1.1. 症状：Managed 集群创建失败并显示证书 IP SAN 错误

在 VMware vSphere 上创建新的 Red Hat OpenShift Container Platform 集群后，集群会失败，并显示一个错误消息，显示证书 IP SAN 错误。

#### 1.10.1.2. 鉴别问题：管理的集群创建失败并显示证书 IP SAN 错误

受管集群的部署失败，并在部署日志中返回以下错误：

```
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
```

#### 1.10.1.3. 解决问题：管理的集群创建失败，并显示证书 IP SAN 错误

使用 VMware vCenter 服务器完全限定主机名，而不是凭证中的 IP 地址。您还可以更新 VMware vCenter CA 证书以包含 IP SAN。

### 1.10.2. 受管集群创建失败并显示未知证书颁发机构

#### 1.10.2.1. 症状：管理集群创建失败并显示未知证书颁发机构

在 VMware vSphere 上创建新的 Red Hat OpenShift Container Platform 集群后，集群会失败，因为证书由未知颁发机构签名。

#### 1.10.2.2. 鉴别问题：Managed 集群创建失败并显示未知证书颁发机构

受管集群的部署失败，并在部署日志中返回以下错误：

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

#### 1.10.2.3. 解决问题：管理的集群创建失败并显示未知证书颁发机构

确保您在创建凭证时从证书认证机构输入了正确的证书。

### 1.10.3. 受管集群创建带有过期证书失败

#### 1.10.3.1. 情况：集群创建失败并显示过期的证书

在 VMware vSphere 上创建新的 Red Hat OpenShift Container Platform 集群后，集群会失败，因为证书已过期或者无效。

#### 1.10.3.2. 鉴别问题：管理的集群创建失败并显示过期的证书

受管集群的部署失败，并在部署日志中返回以下错误：

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

### 1.10.3.3. 解决问题：管理的集群创建失败并显示过期的证书

确保同步了 ESXi 主机上的时间。

### 1.10.4. 受管集群创建失败且没有标记权限

#### 1.10.4.1. 症状：管理集群创建失败且没有足够特权进行标记

在 VMware vSphere 上创建新的 Red Hat OpenShift Container Platform 集群后，集群会失败，因为没有足够的权限进行标记。

#### 1.10.4.2. 鉴别问题：Managed 集群创建会失败，没有足够权限进行标记

受管集群的部署失败，并在部署日志中返回以下错误：

```
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\"
{"
```

#### 1.10.4.3. 解决问题：管理的集群创建没有足够权限进行标记

确保 VMware vCenter 所需的帐户权限正确。如需更多信息，请参阅[删除的镜像 registry](#)。

### 1.10.5. 受管集群创建失败并显示无效的 dnsVIP

#### 1.10.5.1. 症状：受管集群创建失败并显示无效的 dnsVIP

在 VMware vSphere 上创建新的 Red Hat OpenShift Container Platform 集群后，集群会失败，因为存在无效的 dnsVIP。

#### 1.10.5.2. 鉴别问题：Managed 集群创建失败并显示无效的 dnsVIP

如果您在尝试使用 VMware vSphere 部署新受管集群时看到以下消息，这是因为您有一个较老的 OpenShift Container Platform 发行版本镜像，它不支持 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
```

#### 1.10.5.3. 解决问题：受管集群创建失败并显示无效的 dnsVIP

从支持 VMware Installer Provisioned Infrastructure 的 OpenShift Container Platform 版本中选择一个发行镜像。

### 1.10.6. 受管集群创建带有不正确的网络类型失败

#### 1.10.6.1. 症状：集群创建失败并显示不正确的网络类型

在 VMware vSphere 上创建新的 Red Hat OpenShift Container Platform 集群后，集群会失败，因为指定的网络类型不正确。

#### 1.10.6.2. 鉴别问题：管理的集群创建失败并显示不正确的网络类型

如果您在尝试使用 VMware vSphere 部署新受管集群时看到以下消息，这是因为您有一个旧的 OpenShift Container Platform 镜像，它不支持 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"
```

#### 1.10.6.3. 解决问题：受管集群创建失败并显示不正确的网络类型

为指定的 VMware 集群选择一个有效的 VMware vSphere 网络类型。

### 1.10.7. 受管集群创建失败并显示磁盘更改错误

#### 1.10.7.1. 症状：因为错误处理磁盘更改导致添加 VMware vSphere 受管集群失败

在 VMware vSphere 上创建新的 Red Hat OpenShift Container Platform 集群后，集群会失败，因为在处理磁盘更改时会出现错误。

#### 1.10.7.2. 鉴别问题：添加 VMware vSphere 受管集群会因为处理磁盘更改出错而失败

日志中会显示类似以下内容的消息：

```
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)
```

#### 1.10.7.3. 解决问题：因为错误处理磁盘更改导致 VMware vSphere 受管集群失败

使用 VMware vSphere 客户端为用户授予 *Profile-driven Storage Privileges* 的所有权限。

### 1.11. 受管集群在 RED HAT OPENSTACK PLATFORM 上失败，并带有 UNKNOWN AUTHORITY 错误

如果您在 Red Hat OpenStack Platform 上创建 Red Hat OpenShift Container Platform 集群时遇到问题，请参阅以下故障排除信息来查看它们是否解决了您的问题。

### 1.11.1. 症状：受管集群创建失败，并带有 **unknown authority** 错误

在使用自签名证书在 Red Hat OpenStack Platform 上创建新的 Red Hat OpenShift Container Platform 集群后，集群会失败，并显示 **unknown authority** 错误。

### 1.11.2. 鉴别问题：管理的集群创建失败带有 **unknown authority** 错误

受管集群的部署失败，并返回以下出错信息：

**x509: certificate signed by unknown authority**

### 1.11.3. 解决问题：受管集群创建失败并显示 **unknown authority** 错误

验证是否已正确配置以下文件：

1. **clouds.yaml** 文件必须在 **cacert** 参数中指定 **ca.crt** 文件的路径。**cacert** 参数在生成 ignition shim 时传递给了 OpenShift 安装程序。请参见以下示例：

```
clouds:
  openstack:
    cacert: "/etc/pki/ca-trust/source/anchors/ca.crt"
```

2. **certificatesSecretRef** 参数必须引用一个其文件名与 **ca.crt** 文件匹配的 secret。请参见以下示例：

```
spec:
  baseDomain: dev09.red-chesterfield.com
  clusterName: txue-osspoke
  platform:
    openstack:
      cloud: openstack
      credentialsSecretRef:
        name: txue-osspoke-openstack-creds
      certificatesSecretRef:
        name: txue-osspoke-openstack-certificatebundle
```

要使用匹配文件名创建 secret，请运行以下命令：

```
oc create secret generic txue-osspoke-openstack-certificatebundle --from-file=ca.crt=ca.crt.pem -n $CLUSTERNAME
```

3. **ca.cert** 文件的大小必须小于 63 KB。

## 1.12. OPENSIFT CONTAINER PLATFORM 版本 3.11 集群导入失败的故障排除

### 1.12.1. 症状：OpenShift Container Platform 版本 3.11 集群导入失败

试图导入 Red Hat OpenShift Container Platform 版本 3.11 集群后，导入会失败，并显示类似以下内容的日志消息：

```
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|...
```

### 1.12.2. 鉴别问题：OpenShift Container Platform 版本 3.11 集群导入失败

这通常是因为安装的 **kubectrl** 命令行工具的版本为 1.11 或更早版本。运行以下命令，以查看您正在运行的 **kubectrl** 命令行工具的版本：

```
kubectrl version
```

如果返回的数据列出了 1.11 或更早版本，按照 [解决问题：OpenShift Container Platform 版本 3.11 集群导入失败](#) 中的内容进行解决。

### 1.12.3. 解决问题：OpenShift Container Platform 版本 3.11 集群导入失败

您可以通过完成以下步骤之一解决这个问题：

- 安装 **kubectrl** 命令行工具的最新版本。
  1. 下载 **kubectrl** 工具的最新版本：参阅 Kubernetes 文档中的 [安装和设置 kubectrl](#)。
  2. 升级 **kubectrl** 工具后再次导入集群。
- 运行包含导入命令的文件。
  1. 根据 [使用 CLI 导入受管集群](#) 进行操作。
  2. 在创建用于导入集群的命令时，将该命令复制到名为 **import.yaml** 的 YAML 文件中。
  3. 运行以下命令从文件中再次导入集群：

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

## 1.13. 证书更改后导入的集群离线故障排除

支持安装自定义 **apiserver** 证书，但在更改证书信息前导入的一个或多个集群处于离线状态。

### 1.13.1. 症状：证书更改后集群处于离线状态

完成更新证书 secret 的步骤后，在线的一个或多个集群现在在控制台中显示 **离线状态**。

### 1.13.2. 鉴别问题：证书更改后集群处于离线状态

更新自定义 API 服务器证书信息后，在新证书前导入并运行的集群会处于 **offline** 状态。

表示证书有问题的错误会出现在离线受管集群的 **open-cluster-management-agent** 命名空间中的 pod 日志中。以下示例与日志中显示的错误类似：

请参阅以下 **work-agent** 日志：

```
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
```

请参阅以下 **registration-agent** 日志：

```
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
```

### 1.13.3. 解决问题：证书更改后集群处于离线状态

如果您的受管集群是 **local-cluster**，或者受管集群是使用 Red Hat Advanced Cluster Management for Kubernetes 创建的，则必须等待 10 分钟或更长时间重新导入受管集群。

要立即重新导入受管集群，您可以删除 hub 集群上的受管集群导入 secret，并使用 Red Hat Advanced Cluster Management 重新导入它。运行以下命令：

```
oc delete secret -n <cluster_name> <cluster_name>-import
```

将 **<cluster\_name>** 替换为您要导入的受管集群的名称。

如果要重新导入使用 Red Hat Advanced Cluster Management 导入的受管集群，请完成以下步骤以再次导入受管集群：

1. 在 hub 集群中，运行以下命令来重新创建受管集群导入 secret：

```
oc delete secret -n <cluster_name> <cluster_name>-import
```

将 **<cluster\_name>** 替换为您要导入的受管集群的名称。

2. 在 hub 集群中，运行以下命令来将受管集群导入 secret 公开给 YAML 文件：

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

将 **<cluster\_name>** 替换为您要导入的受管集群的名称。

3. 在受管集群中，运行以下命令应用 **import.yaml** 文件：

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

注：前面的步骤不会从 hub 集群中分离受管集群。步骤使用受管集群中的当前设置更新所需的清单，包括新证书信息。

## 1.14. 删除集群后命名空间会保留

当您删除受管集群时，该命名空间通常会作为移除集群过程的一部分被删除。在个别情况下，命名空间会和其中的一些工件一起被保留。在这种情况下，您必须手动删除命名空间。

### 1.14.1. 症状：删除集群后命名空间被保留

删除受管集群后，命名空间没有被删除。

### 1.14.2. 解决问题：删除集群后命名空间被保留

完成以下步骤以手动删除命名空间：

1. 运行以下命令以生成保留在 **<cluster\_name>** 命名空间中的资源列表：

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

使用您要删除的集群的命名空间名称替换 **cluster\_name**。

2. 输入以下命令来编辑列表，删除列表中状态不是 **Delete** 的资源：

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

将 **resource\_kind** 替换为资源类型。将 **resource\_name** 替换为资源的名称。使用资源的命名空间的名称替换 **namespace**。

3. 在元数据中找到 **finalizer** 属性。
4. 使用 vi 编辑器的 **dd** 命令删除非 Kubernetes finalizer。
5. 输入 **:wq** 命令保存列表并退出 vi 编辑器。
6. 输入以下命令来删除命名空间：

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

使用您要删除的命名空间的名称替换 **cluster-name**。

## 1.15. 导入集群时出现自动 **AUTO-IMPORT-SECRET-EXISTS** 错误

集群导入失败，并显示出错信息：auto import secret exists。

### 1.15.1. 症状：导入集群时出现 **Auto import secret exists** 错误

当导入 hive 集群以进行管理时，会显示 **auto-import-secret already exists** 错误。

### 1.15.2. 解决问题：导入集群时出现 **Auto import secret exists** 错误

当您试图导入之前由 Red Hat Advanced Cluster Management 管理的集群时，会出现这种情况。如果出现这种情况，当您尝试重新导入集群时，secret 会发生冲突。

要临时解决这个问题，请完成以下步骤：

1. 在 hub 集群中运行以下命令来手工删除存在的 **auto-import-secret**：

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

将 **cluster-namespace** 替换为集群的命名空间。

2. 使用[集群导入简介](#)中的流程再次导入集群。

## 1.16. 对 **VOLSYNC** 的 **CINDER CONTAINER STORAGE INTERFACE (CSI)** 驱动程序进行故障排除

如果您在 cinder Container Storage Interface (CSI) 驱动程序中使用 VolSync 或使用默认设置，您可能会遇到正在使用的 PVC 的错误。

### 1.16.1. 症状：**Volumesnapshot** 错误状态

您可以配置 VolSync **ReplicationSource** 或 **ReplicationDestination** 以使用快照。另外，您可以在 **ReplicationSource** 和 **ReplicationDestination** 中配置 **storageclass** 和 **volumesnapshotclass**。cinder **volumesnapshotclass** 上有一个名为 **force-create** 的参数，默认值为

**false**。 **volumesnapshotclass** 上的这个 **force-create** 参数意味着 cinder 不允许使用 **volumesnapshot**。因此， **volume snapshot** 处于错误状态。

### 1.16.2. 解决问题：将参数设置为 true

1. 为 cinder CSI 驱动程序创建新 **volumesnapshotclass**。
2. 将参数 **force-create** 更改为 **true**。请参见以下 YAML 示例：

```
apiVersion: snapshot.storage.k8s.io/v1
deletionPolicy: Delete
driver: cinder.csi.openstack.org
kind: VolumeSnapshotClass
metadata:
  annotations:
    snapshot.storage.kubernetes.io/is-default-class: 'true'
  name: standard-csi
parameters:
  force-create: 'true'
```

## 1.17. 运行 MUST-GATHER 命令进行故障排除

运行 **must-gather** 命令来收集详情、日志并采取调试问题的步骤。当您请求支持时，这个调试信息会很 有用。**oc adm must-gather** CLI 命令收集从集群收集调试问题通常需要的信息，包括：

- 资源定义
- 服务日志

### 1.17.1. 先决条件

您必须满足以下先决条件才能运行 **must-gather** 命令：

- 以具有 **cluster-admin** 角色的用户访问全局 hub 和受管 hub 集群。
- 安装了 OpenShift Container Platform CLI (oc)。

### 1.17.2. 运行 must-gather 命令

完成以下步骤，使用 **must-gather** 命令收集信息：

1. 了解 **must-gather** 命令以及使用 OpenShift Container Platform 文档中的 [收集集群数据](#) 所需的先决条件。
2. 登录到您的全局 hub 集群。对于典型的用例，在登录到全局 hub 集群时运行以下命令：

```
oc adm must-gather --image=quay.io/stolostron/must-gather:SNAPSHOTNAME
```

如果要检查您的受管 hub 集群，请在这些集群中运行 **must-gather** 命令。

3. 可选：如果要将结果保存到 **SOMENAME** 目录中，您可以运行以下命令而不是上一步中的结果：

```
oc adm must-gather --image=quay.io/stolostron/must-gather:SNAPSHOTNAME --dest-dir=
<SOMENAME> ; tar -cvzf <SOMENAME>.tgz <SOMENAME>
```

您可以为目录指定不同的名称。

**注意：**命令包括创建 gzipped tarball 文件所需的添加。

从 **must-gather** 命令收集以下信息：

- 两个对等级别：**cluster-scoped-resources** 和 **namespaces** 资源。
- 每个对等级别下的子类：用于 cluster-scope 和 namespace-scoped 资源的自定义资源定义的 API 组。
- 每个的下一级别：按 kind 排序的 YAML 文件。
- 对于全局 hub 集群，您可以在命名空间资源中检查 **PostgresCluster** 和 **Kafka**。
- 对于全局 hub 集群，您可以在命名空间资源的 pod 中检查多集群全局 hub 相关 pod 和日志。
- 对于受管 hub 集群，您可以检查多集群全局 hub 代理 pod，并在命名空间资源的 pod 中检查日志。

## 1.18. 访问置备的 POSTGRESQL 数据库以进行故障排除

您可以访问置备的 PostgreSQL 数据库，以查看对 multicluster global hub 问题进行故障排除的信息。根据服务类型，可以通过三种方式访问置备的 PostgreSQL 数据库。

- **使用 ClusterIP 服务**

1. 运行以下命令来确定您的 postgres 连接 URI：

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-
template='{{index (.data) "uri" | base64decode}}'
```

2. 运行以下命令来访问数据库：

```
oc exec -it $(kubectl get pods -n multicluster-global-hub-postgres -l postgres-
operator.crunchydata.com/role=master -o jsonpath='{.items..metadata.name}') -c
database -n multicluster-global-hub-postgres -- psql -U postgres -d hoh -c "SELECT 1"
```

## •

**使用 NodePort 服务**

1.

运行以下命令，将服务修改为 **NodePort**，将主机设置为节点 IP，并将端口设置为 **32432**：

```
oc patch postgrescluster hoh -n multicluster-global-hub-postgres -p '{"spec":{"service":{"type":"NodePort", "nodePort": 32432}}}' --type merge
```

2.

运行以下命令来添加您的用户名：

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "user" | base64decode}}'
```

3.

运行以下命令来添加您的密码：

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "password" | base64decode}}'
```

4.

运行以下命令来添加数据库名称：

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "dbname" | base64decode}}'
```

## •

**LoadBalancer**

1.

运行以下命令，将服务类型设置为 **LoadBalancer**：

```
oc patch postgrescluster hoh -n multicluster-global-hub-postgres -p '{"spec":{"service":{"type":"LoadBalancer"}}}' --type merge
```

**默认端口为 5432**

2.

运行以下命令来设置主机名：

```
kubectl get svc -n multicluster-global-hub-postgres hoh-ha -ojsonpath='{.status.loadBalancer.ingress[0].hostname}'
```

3. 运行以下命令来添加您的用户名：

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "user" | base64decode}}'
```

4. 运行以下命令来添加您的密码：

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "password" | base64decode}}'
```

5. 运行以下命令来添加数据库名称：

```
oc get secrets -n multicluster-global-hub-postgres hoh-pguser-postgres -o go-template='{{index (.data) "dbname" | base64decode}}'
```

## 1.19. 使用数据库转储和恢复进行故障排除

在生产环境中，定期备份您的 PostgreSQL 数据库作为数据库管理任务。备份也可用于调试多集群全局 hub。

### 1.19.1. 转储用于 debugging 的数据库输出

有时，您需要转储 multicluster global hub 数据库中的输出来调试问题。PostgreSQL 数据库提供 `pg_dump` 命令行工具来转储数据库的内容。要从 localhost 数据库服务器转储数据，请运行以下命令：

```
pg_dump hoh > hoh.sql
```

要转储位于带有压缩格式的远程服务器中的多集群全局 hub 数据库，请使用命令行选项来控制连接详情，如下例所示：

```
pg_dump -h my.host.com -p 5432 -U postgres -F t hoh -f hoh-$(date +%d-%m-%y_%H-%M).tar
```

### 1.19.2. 从转储中恢复数据库

要恢复 PostgreSQL 数据库，您可以使用 `psql` 或 `pg_restore` 命令行工具。`psql` 工具用于恢复 `pg_dump` 创建的纯文本文件：

```
psql -h another.host.com -p 5432 -U postgres -d hoh < hoh.sql
```

`pg_restore` 工具用于通过一个非纯文本格式(custom、tar 或 目录)从 `pg_dump` 创建的存档中恢复 PostgreSQL 数据库：

```
pg_restore -h another.host.com -p 5432 -U postgres -d hoh hoh-$(date +%d-%m-%y_%H-%M).tar
```

## 1.20. 恢复合规性数据

**Grafana Datasource** 主要来自名为 `history.local_compliance` 的表中。其记录由一个总结例程生成，该例程在每天的 00:00:00 开始。您通常不需要手动运行摘要过程。在某些情况下，运行合规作业时可能会出现意外错误，因此需要手动登录到数据库以运行整个摘要过程来恢复未生成的数据。您可以按照[手动运行总结流程的步骤](#)来恢复数据。

### 1.20.1. 可选：手动将现有表升级到分区表

如果您在 GA 之前安装了多集群全局 hub 的早期版本，则需要升级表使其与当前多集群全局 hub operator 兼容。升级的主要目的是将 `event.local_policies`、`event.local_root_policies` 和 `history.local_compliance` 表转换为分区表。

以下示例显示了使用 2023-08 设置的日期的 `event.local_policies` 表转换。其他两个表的升级步骤类似。

1. 确保目标已分区。

```
SELECT relname, relkind FROM pg_class WHERE relname = 'local_policies';
```

表输出类似以下示例：

relname	relkind	local_policies	r
---------	---------	----------------	---

如果 `relkind` 是 `p`，则当前表将被分区。如果是，您可以跳过剩余的步骤并升级其他表。

2. 将常规表转换为分区表。

```
-- start a transaction
BEGIN;
-- Rename the legacy TABLE_NAME
```

```

ALTER TABLE event.local_policies RENAME TO local_policies_old;
-- Partition tables: https://github.com/stolostron/multicluster-global-
hub/blob/main/operator/pkg/controllers/hubofhubs/database/2.tables.sql#L283-L318
CREATE TABLE IF NOT EXISTS event.local_policies (
  event_name character varying(63) NOT NULL,
  policy_id uuid NOT NULL,
  cluster_id uuid NOT NULL,
  leaf_hub_name character varying(63) NOT NULL,
  message text,
  reason text,
  count integer NOT NULL DEFAULT 0,
  source jsonb,
  created_at timestamp without time zone DEFAULT now() NOT NULL,
  compliance local_status.compliance_type NOT NULL,
  -- Rename the constraint to avoid conflicts
  CONSTRAINT local_policies_unique_partition_constraint UNIQUE (event_name,
count, created_at)
) PARTITION BY RANGE (created_at);
-- Create partitions, load the old data to the previous partition table
CREATE TABLE IF NOT EXISTS event.local_policies_2023_08 PARTITION OF
event.local_policies FOR VALUES FROM ('2023-08-01') TO ('2023-09-01');
CREATE TABLE IF NOT EXISTS event.local_policies_2023_07 PARTITION OF
event.local_policies FOR VALUES FROM ('2000-01-01') TO ('2023-08-01');

-- Move the records from regular table to partition table
INSERT INTO event.local_policies SELECT * FROM event.local_policies_old;
DROP TABLE IF EXISTS event.local_policies_old;
-- commit the transaction
COMMIT;

```

您可以根据表名称和当前日期替换以下值：

- `event.local_policies_2023_08` 是带有当前月份作为后缀的分区名称，例如这个示例中的 8 月
- `'2023-08-01'` 和 `'2023-09-01'` 是当前月份分区的最小和最大界限
- `event.local_policies_2023_07` 是之前月份的后缀(July)的分区名称
- `'2000-01-01'` 和 `'2023-08-01'` 以前月分区的最小和最大边界

### 1.21. 集群状态从离线变为可用的故障排除

在没有对环境或集群进行任何手工更改的情况下，受管集群的状态在 `offline`（离线）和 `available`（可用）间转换。

### 1.21.1. 症状：集群状态从离线变为可用

当将受管集群连接到 hub 集群的网络不稳定时，hub 集群所报告的受管集群的状态在离线和可用之间不断转换。

hub 集群和受管集群之间的连接通过 `leaseDurationSeconds interval` 值验证的租期进行维护。如果在 `leaseDurationSeconds` 的值的五次尝试中未验证租期，则集群被标记为 离线。

例如，在五分钟后集群标记为 离线，`leaseDurationSeconds` 间隔为 60 秒。由于连接问题或延迟等原因，此配置可能不足，从而导致不稳定。

### 1.21.2. 解决问题：集群状态从离线变为可用

五个验证尝试是默认的，且无法更改，但您可以更改 `leaseDurationSeconds` 间隔。

确定您希望集群标记为 离线 的时间（以分钟为单位），然后将该值乘以 60 以转换为秒。然后默认划分五个尝试。结果为您的 `leaseDurationSeconds` 值。

1. 输入以下命令编辑 hub 集群上的 `ManagedCluster` 规格，但将 `cluster-name` 替换为受管集群的名称：

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

2. 在 `ManagedCluster` 规格中增加 `leaseDurationSeconds` 的值，如下例所示：

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: <cluster-name>
spec:
  hubAcceptsClient: true
  leaseDurationSeconds: 60
```

3. 保存并应用该文件。

## 1.22. 集群在控制台中带有待处理或失败状态的故障排除

如果您在控制台中看到您创建的集群的状态为 *Pending* 或 *Failed*，请按照以下步骤排除问题。

### 1.22.1. 症状：集群在控制台中带有待处理或失败状态

在使用 Red Hat Advanced Cluster Management 控制台创建一个新集群后，在控制台中集群会一直显示 *Pending* 或 *Failed* 状态。

### 1.22.2. 鉴别问题：集群在控制台中显示待处理或失败状态

如果集群显示 *Failed* 状态，进入集群的详情页面并使用提供的日志的链接。如果没有找到日志或集群显示 *Pending* 状态，请按照以下步骤检查日志：

- 

#### 流程 1

- 1.

在 hub 集群中运行以下命令，查看在命名空间中为新集群创建的 Kubernetes pod 的名称：

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

使用您创建的集群名称替换 *new\_cluster\_name*。

- 2.

如果没有 pod 在列出的名称中包括 *provision* 字符串，则按照流程 2 继续进行。如果存在其标题中带有 *provision* 字符串的 pod，则在 hub 集群中运行以下命令以查看该 pod 的日志：

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

将 *new\_cluster\_name\_provision\_pod\_name* 替换为您创建的集群的名称，后接包含 *provision* 的 pod 名称。

- 3.

搜索日志中可能会解释造成问题的原因的错误信息。

- 

#### 流程 2

如果没有在其名称中带有 *provision* 的 pod，则代表问题在进程早期发生。完成以下步骤以

查看日志：

1. 在 **hub** 集群中运行以下命令：

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

使用您创建的集群名称替换 `new_cluster_name`。如需有关集群安装日志的更多信息，请参阅 [Red Hat OpenShift 文档中的 收集安装日志](#) 的内容。

2. 检查是否在资源的 `Status.Conditions.Message` 和 `Status.Conditions.Reason` 条目中存在有关此问题的额外信息。

### 1.22.3. 解决问题：集群在控制台中显示待处理或失败状态

在日志中找到错误后，确定如何在销毁集群并重新创建它之前解决相关的错误。

以下示例包括了一个选择不支持的区的日志错误，以及解决它所需的操作：

```
No subnets provided for zones
```

When you created your cluster, you selected one or more zones within a region that are not supported.在重新创建集群时完成以下操作之一以解决此问题：

- 在区域里（region）选择不同的区（zone）。
- 如果列出了其它区，则省略不支持的区。
- 为集群选择不同的区域。

在处理了日志中记录的问题后，销毁集群并重新创建它。

有关创建集群的更多信息，请参阅 [集群创建简介](#)。

## 1.23. GRAFANA 故障排除

当您在 Grafana explorer 中查询一些耗时的指标时，您可能会遇到一个网关超时错误。

### 1.23.1. 症状：Grafana explorer 网关超时

如果您在 Grafana explorer 中查询一些耗时的指标时遇到 Gateway Time-out 错误，则超时可能是由 open-cluster-management-observability 命名空间中的 Grafana 造成的。

### 1.23.2. 解决问题：配置 Grafana

如果您有这个问题，请完成以下步骤：

1. 验证 Grafana 的默认配置是否有预期的超时设置：

- a. 要验证 Grafana 的默认超时设置，请运行以下命令：

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

应显示以下超时设置：

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

- b. 要验证 Grafana 的默认数据源查询超时时间，请运行以下命令：

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

应显示以下超时设置：

```
queryTimeout: 300s
```

- 2.

如果您验证了 Grafana 的默认配置有预期的超时设置，您可以运行以下命令来在 `open-cluster-management-observability` 命名空间中配置 Grafana：

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

刷新 Grafana 页面，并尝试再次查询指标。网关超时错误不再显示。

## 1.24. 未使用放置规则选择本地集群的故障排除

受管集群使用放置规则选择，但不会选择 `local-cluster`（这也是管理的 `hub` 集群）。放置规则用户没有在 `local-cluster` 命名空间中获取 `managedcluster` 资源的权限。

### 1.24.1. 症状：不选择为受管集群对本地集群进行故障排除

所有受管集群都使用放置规则选择，但 `local-cluster` 不是。放置规则用户没有在 `local-cluster` 命名空间中获取 `managedcluster` 资源的权限。

### 1.24.2. 解决问题：未选择作为受管集群的故障排除本地集群

已弃用：`PlacementRule`

要解决这个问题，您需要在 `local-cluster` 命名空间中授予 `managedcluster` 管理权限。完成以下步骤：

1.

确认受管集群列表包含 `local-cluster`，并且放置规则的 `decisions` 列表不显示 `local-cluster`。运行以下命令并查看结果：

```
% oc get managedclusters
```

请参阅 `local-cluster` 被加入的示例输出，但它不在 `PlacementRule` 的 YAML 中：

```
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.

在 YAML 文件中创建一个 Role，以便在 local-cluster 命名空间中授予 managedcluster 管理权限。请参见以下示例：

```

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: managedcluster-admin-user-zisis
  namespace: local-cluster
rules:
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  verbs:
  - get

```

3.

创建 RoleBinding 资源，向放置规则用户授予 local-cluster 命名空间的访问权限。请参见以下示例：

```

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

```

### 1.25. 对应用程序 KUBERNETES 部署版本进行故障排除

可能不支持带有已弃用 Kubernetes apiVersion 的受管集群。有关已弃用 API 版本的详情，请参阅 [Kubernetes 问题](#)。

### 1.25.1. 症状：应用程序部署版本

如果 **Subscription YAML** 文件中的一个或多个应用程序资源使用弃用的 **API**，您可能会收到与以下类似的错误信息：

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

或者，带有新 **Kubernetes API** 版本的 **YAML** 文件名为 **old.yaml**，您可能会收到以下错误：

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

### 1.25.2. 解决：应用程序部署版本

1.

更新资源中的 **apiVersion**。例如，如果 **subscription YAML** 文件中显示 **Deployment kind** 的错误，您需要将 **apiVersion** 从 **extensions/v1beta1** 更新至 **apps/v1**。

请参见以下示例：

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

2.

在受管集群中运行以下命令来验证可用版本：

```
kubectl explain <resource>
```

3.

检查 **VERSION**。

## 1.26. 带有降级条件的 KLUSTERLET 故障排除

**Klusterlet** 降级条件可帮助诊断受管集群中的 **Klusterlet** 代理的状态。如果 **Klusterlet** 处于 **degraded** 条件，受管集群中的 **Klusterlet** 代理可能会出错，需要进行故障排除。对于设置为 **True** 的 **Klusterlet** 降级条件，请参见以下信息。

### 1.26.1. 症状：Klusterlet 处于降级状况

在受管集群中部署 Klusterlet 后，`KlusterletRegistrationDegraded` 或 `KlusterletWorkDegraded` 条件会显示 `True` 的状态。

### 1.26.2. 鉴别问题：Klusterlet 处于降级状况

1.

在受管集群中运行以下命令查看 Klusterlet 状态：

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

2.

检查 `KlusterletRegistrationDegraded` 或 `KlusterletWorkDegraded` 以查看该条件是否被设置为 `True`。请根据[解决这个问题](#)的内容处理降级问题。

### 1.26.3. 解决问题：Klusterlet 处于降级状况

请查看以下降级状态列表，以及如何尝试解决这些问题：

- 如果 `KlusterletRegistrationDegraded` 条件的状态为 `True` 且状况原因为：`BootstrapSecretMissing`，您需要在 `open-cluster-management-agent` 命名空间中创建一个 `bootstrap secret`。
- 如果 `KlusterletRegistrationDegraded` 条件显示为 `True`，且状况原因为 `BootstrapSecretError` 或 `BootstrapSecretUnauthorized`，则当前的 `bootstrap secret` 无效。删除当前的 `bootstrap secret`，并在 `open-cluster-management-agent` 命名空间中重新创建有效的 `bootstrap secret`。
- 如果 `KlusterletRegistrationDegraded` 和 `KlusterletWorkDegraded` 显示为 `True`，且状况原因为 `HubKubeConfigSecretMissing`，请删除 Klusterlet 并重新创建它。
- 如果 `KlusterletRegistrationDegraded` 和 `KlusterletWorkDegraded` 显示为 `True`，则状况原因为：`ClusterNameMissing`、`KubeConfigMissing`、`HubConfigSecretError` 或 `HubConfigSecretUnauthorized`，从 `open-cluster-management-agent` 命名空间中删除 hub 集群 `kubeconfig secret`。注册代理将再次引导以获取新的 hub 集群 `kubeconfig secret`。
- 如果 `KlusterletRegistrationDegraded` 显示 `True`，且状况原因为 `GetRegistrationDeploymentFailed` 或 `UnavailableRegistrationPod`，您可以检查条件信息以获取问题详情并尝试解决。

- 如果 `KlusterletWorkDegraded` 显示为 `True`，且状况原因为 `GetWorkDeploymentFailed`，或 `UnavailableWorkPod`，您可以检查条件消息以获取问题详情并尝试解决。

## 1.27. OBJECT STORAGE 频道 SECRET 故障排除

如果更改 `SecretAccessKey`，Object 存储频道的订阅将无法自动获取更新的 `secret`，您会收到一个错误。

### 1.27.1. 症状：对象存储频道 secret

Object 存储频道的订阅无法自动获取更新的 `secret`。这可能会阻止订阅 `operator` 协调并将资源从对象存储部署到受管集群的过程。

### 1.27.2. 解决问题：对象存储频道 secret

您需要手动输入凭证来创建 `secret`，然后引用频道中的 `secret`。

1.

注解订阅 CR，以便生成一个协调的单个订阅 `operator`。请参阅以下 `data` 规格：

```
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. 运行 `oc annotate` 进行测试：

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

运行命令后，您可以进入 **Application** 控制台以验证资源是否已部署到受管集群。或者您可以登录到受管集群，以查看应用程序资源是否在给定命名空间中创建。

## 1.28. 对可观察性功能进行故障排除

安装可观察组件后，该组件可能会卡住，并显示 **Installing** 状态。

### 1.28.1. 症状：MultiClusterObservability 资源处于没有就绪的状态

如果在安装并创建 **Observability** 自定义资源定义 (CRD) 后，可观察性 (**observability**) 状态会一直处于 **Installing** 状态，则可能是因为没有为 `spec:storageConfig:storageClass` 参数定义值。或者，可观察性组件自动找到默认的 `storageClass`，但如果没有存储值，则组件会停留在 **Installing** 状态下。

### 1.28.2. 解决这个问题：MultiClusterObservability 资源处于没有就绪的状态

如果您有这个问题，请完成以下步骤：

1. 验证是否安装了可观察性组件：
  - a. 要验证 `multicluster-observability-operator`，请运行以下命令：

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

- b. 要验证是否存在正确的 **CRD**，请运行以下命令：

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

在启用组件前，必须显示以下 **CRD**：

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

2.

如果您为裸机集群创建自己的 `storageClass`，[请参阅使用 NFS 的持久性存储](#)。

3.

为确保可观察性组件可以找到默认的 `storageClass`，请更新 `multicluster-observability-operator` 自定义资源定义中的 `storageClass` 参数。您的参数可能类似以下值：

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

安装完成后，可观察组件状态会被更新为 `Ready` 状态。如果安装无法完成，则会显示 `Fail` 状态。

## 1.29. OPENSIFT 监控服务故障排除

受管集群中的 `Observability` 服务需要从 `OpenShift Container Platform` 监控堆栈中提取指标数据。如果 `OpenShift Container Platform` 监控堆栈没有处于就绪状态，则不会安装 `metrics-collector`。

### 1.29.1. 症状：OpenShift 监控服务未就绪

`endpoint-observability-operator-x` pod 会检查 `prometheus-k8s` 服务是否在 `openshift-monitoring` 命名空间中可用。如果这个服务没有出现在 `openshift-monitoring` 命名空间中，则不会部署 `metrics-collector`。您可能会收到以下出错信息：`Failed to get prometheus resource`。

### 1.29.2. 解决问题：OpenShift 监控服务未就绪

如果您有这个问题，请完成以下步骤：

1.

登录您的 `OpenShift Container Platform` 集群。

2.

访问 `openshift-monitoring` 命名空间，验证 `prometheus-k8s` 服务是否可用。

3.

在受管集群的 `open-cluster-management-addon-observability` 命名空间中重启 `endpoint-observability-operator-x` pod。

## 1.30. METRICS-COLLECTOR 故障排除

当 `observability-client-ca-certificate` secret 没有在受管集群中被重新刷新时，您可能会收到一个内部服务器错误。

### 1.30.1. 症状：metrics-collector 无法验证 observability-client-ca-certificate

可能有一个受管集群，其中的指标不可用。如果出现这种情况，您可能会从 `metrics-collector` 部署中收到以下错误：

```
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")
```

### 1.30.2. 解决问题：metrics-collector 无法验证 observability-client-ca-certificate

如果您有这个问题，请完成以下步骤：

1. 登录到受管集群。
2. 删除名为 `observability-controller-open-cluster-management.io-observability-signer-client-cert` 的 secret，该 secret 位于 `open-cluster-management-addon-observability` 命名空间中。运行以下命令：

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

**注：**`observability-controller-open-cluster-management.io-observability-signer-client-cert` 会自动使用新证书重新创建。

重新创建 `metrics-collector` 部署并更新 `observability-controller-open-cluster-management.io-observability-signer-client-cert` secret。

## 1.31. POSTGRESQL 共享内存错误故障排除

如果您有一个大型环境，您可能会遇到 PostgreSQL 共享内存错误，这会影响到应用程序的搜索结果和拓扑视图。

### 1.31.1. 症状 : PostgreSQL 共享内存错误

错误消息重装以下内容会出现在 `search-api` 日志中 : `ERROR: could not resize shared memory segment "/PostgreSQL.1083654800" to 25031264 bytes: No space left on device (SQLSTATE 53100)`

### 1.31.2. 解决问题 : PostgreSQL 共享内存错误

要解决这个问题, 更新 `search-postgres ConfigMap` 中找到的 PostgreSQL 资源。完成以下步骤以更新资源 :

1. 运行以下命令来切换到 `open-cluster-management` 项目 :

```
oc project open-cluster-management
```

2. 增加 `search-postgres pod` 内存。以下命令将内存增加到 `16Gi` :

```
oc patch search -n open-cluster-management search-v2-operator --type json -p '{"op": "add", "path": "/spec/deployments/database/resources", "value": {"limits": {"memory": "16Gi"}, "requests": {"memory": "32Mi", "cpu": "25m"}}}'
```

3. 运行以下命令以防止搜索 `Operator` 覆盖您的更改 :

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

4. 运行以下命令以更新 `search-postgres YAML` 文件中的资源 :

```
oc edit cm search-postgres -n open-cluster-management
```

有关增加资源的信息, 请参见以下示例 :

```
postgresql.conf: |-
work_mem = '128MB' # Higher values allocate more memory
max_parallel_workers_per_gather = '0' # Disables parallel queries
shared_buffers = '1GB' # Higher values allocate more memory
```

在退出前, 请确保保存您的更改。

5. 运行以下命令以重启 **postgres** 和 **api pod**。

```
oc delete pod search-postgres-xyz search-api-xzy
```

6. 要验证您的更改，请打开 **search-postgres** YAML 文件，并确认您对 **postgresql.conf**: 所做的更改是否存在：

```
oc get cm search-postgres -n open-cluster-management -o yaml
```

有关添加环境变量的更多信息，请参阅[搜索自定义和配置](#)。

## 1.32. 安装后无法连接 SUBMARINER 故障排除

如果在配置 **Submariner** 后 **Submariner** 没有正确运行，请完成以下步骤以诊断问题。

### 1.32.1. 症状：在安装后无法连接 **Submariner**

在安装后，**Submariner** 网络不会被通信。

### 1.32.2. 鉴别问题：Submariner 在安装后无法连接

如果在部署 **Submariner** 后没有建立网络连接，请开始故障排除步骤。请注意，部署 **Submariner** 的过程可能需要几分钟时间。

### 1.32.3. 解决问题：Submariner 在安装后无法连接

当 **Submariner** 在部署后无法正常工作，请完成以下步骤：

1. 检查以下要求，以确定 **Submariner** 的组件是否正确部署：
  - **submariner-addon pod** 在 **hub** 集群的 **open-cluster-management** 命名空间中运行。

- 以下 pod 在每个受管集群的 `submariner-operator` 命名空间中运行：
  - `submariner-addon`
  - `submariner-gateway`
  - `submariner-routeagent`
  - `submariner-operator`
  - `Submariner-globalnet` (仅在 `ClusterSet` 中启用了 `Globalnet` 时)
  - `submariner-lighthouse-agent`
  - `submariner-lighthouse-coredns`
  - `Submariner-networkplugin-syncer` (仅在指定的 CNI 值是 `OVNKubernetes` 时)
  - `submariner-metrics-proxy`
- 2. 运行 `subctl diagnose all` 命令来检查所需 pod 的状态，但 `submariner-addon` pod 除外。
- 3. 确保运行 `must-gather` 命令来收集有助于解决问题的日志。

### 1.33. SUBMARINER ADD-ON 状态降级的故障排除

将 `Submariner add-on` 添加到集群集中的集群后，`Connection status`、`Agent status`、和 `Gateway nodes` 会显示集群的意外状态。

### 1.33.1. 症状：Submariner 附加组件状态为 degraded

将 Submariner add-on 添加到集群中，在 *Gateway nodes*, *Agent status*, 和 *Connection status* 中显示以下状态：

- 带有标签的网关节点
  - **Progressing:** 标记网关节点的进程已启动。。
  - **Nodes not labeled:** 网关节点没有标记，可能因为标记它们的进程还没有完成。
  - **Nodes not labeled:** 网关节点尚未标记，可能会因为进程正在等待另一个进程完成。
  - **Nodes labeled:** 网关节点已标记。
- 代理状态
  - **progressing :** Submariner 代理的安装已启动。
  - **degraded :** Submariner 代理没有运行，可能会因为它仍在进行中。
- 连接状态
  - **progressing :** 在 Submariner add-on 启动时建立连接的过程。
  - **degraded :** 连接未就绪。如果您刚才安装了附加组件，则进程可能仍在进行中。如果在连接已经建立并运行后，有两个集群丢失了相互的连接。当有多个集群时，如果任何集群处于断开连接状态，则所有集群都会显示一个 **Degraded** 状态。

它还会显示哪些集群已连接，以及哪些集群处于断开连接状态。

### 1.33.2. 解决问题：Submariner 附加组件状态为 degraded

- 当流程完成时，降级状态通常会自行解决。您可以点击表中的状态来查看进程的当前步骤。您可以使用这些信息来确定该过程是否完成，并且需要执行其他故障排除步骤。

- 对于无法解决的问题，请完成以下步骤以排除这个问题：

1. 您可以使用带有 `subctl` 程序的 `diagnose` 命令，在存在以下条件时在 Submariner 连接上运行一些测试：

- a. **Agent 状态或 Connection 状态处于 Degraded 状态。** `diagnose` 命令提供有关此问题的详细分析。

- b. 控制台中都不是绿色的，但网络连接无法正常工作。`diagnose` 命令有助于确认控制台外没有其他连接或部署问题。最好在任意部署之后运行 `diagnose` 命令来识别问题

如需有关如何运行命令的更多信息，请参阅 Submariner 中的 [diagnose](#)。

2. 如果问题继续处于 **Connection** 状态，您可以先运行 `subctl` 工具的 `diagnose` 诊断命令来获取两个 Submariner 集群之间的连接的详细状态。该命令的格式是：

```
subctl diagnose all --kubeconfig <path-to-kubeconfig-file>
```

使用 `kubeconfig` 文件的路径替换 `path-to-kubeconfig-file`。有关命令的更多信息，请参阅 Submariner 文档中的 [diagnose](#) 部分。

3. 检查防火墙设置。有时候，连接的问题是由防止集群进行通信的防火墙权限问题造成的。这可能导致 **Connection** 状态显示为 **degraded**。运行以下命令检查防火墙问题：

```
subctl diagnose firewall inter-cluster <path-to-local-kubeconfig> <path-to-remote-cluster-kubeconfig>
```

将 `path-to-local-kubeconfig` 替换为一个集群的 `kubeconfig` 文件的路径。

将 `path-to-remote-kubeconfig` 替换为其他集群的 `kubeconfig` 文件的路径。您可以运行 `subctl` 工具程序的 `verify` 命令来测试两个 Submariner 集群间的连接。该命令的基本格式

为：

4. 如果问题继续，并处于 **Connection** 状态，您可以使用 **subctl** 工具运行 **verify** 命令以测试两个 **Submariner** 集群之间的连接。该命令的基本格式为：

```
subctl verify --kubeccontexts <cluster1>,<cluster2> [flags]
```

将 **cluster1** 和 **cluster2** 替换为您要测试的集群的名称。有关该命令的更多信息，请参阅 **Submariner** 文档中的 [验证](#)。

5. 在故障排除步骤解决此问题后，使用 **benchmark** 命令和 **subctl** 工具建立一个基础，以便在您运行其他诊断时进行比较。

有关该命令选项的更多信息，请参阅 **Submariner** 文档中的 [基准](#) 部分。

### 1.34. 恢复状态故障排除完成并显示错误

恢复备份后，资源会被正确恢复，但 **Red Hat Advanced Cluster Management** 恢复资源会显示 **FinishedWithErrors** 状态。

#### 1.34.1. 症状：恢复状态故障排除完成并显示错误

**Red Hat Advanced Cluster Management** 显示 **FinishedWithErrors** 状态，由 **Red Hat Advanced Cluster Management** 恢复所创建的一个或多个 **Velero** 恢复资源显示 **PartiallyFailed** 状态。

#### 1.34.2. 解决问题：恢复状态完成并出错

如果您从空备份中恢复，可以安全地忽略 **FinishedWithErrors** 状态。

**Red Hat Advanced Cluster Management for Kubernetes** 恢复显示所有 **Velero** 恢复资源的累积状态。如果一个状态是 **PartiallyFailed**，另一个状态为 **Completed**，则您会看到累积的状态 **PartiallyFailed**，代表至少有一个问题。

要解决这个问题，请检查所有带有 **PartiallyFailed** 状态的单独 **Velero** 恢复资源的状态，并查看日志以了解更多详情。您可以直接从对象存储获取日志，或使用 **DownloadRequest** 自定义资源从 **OADP Operator** 下载它。

要从控制台创建 `DownloadRequest`，请完成以下步骤：

1. 进入 `Operators > Installed Operators > Create DownloadRequest`。
2. 选择 `BackupLog` 作为 `Kind`，并按照控制台说明完成 `DownloadRequest` 创建。

### 1.35. 恢复 HUB 集群备份时会删除通用资源

当您恢复一个 `hub` 集群备份，并在 `Restore.cluster.open-cluster-management.io` 资源中使用 `cleanupBeforeRestore: CleanupRestored` 参数，由 `acm-resources-generic-schedule` 备份创建的资源可能会被删除。

#### 1.35.1. 症状：恢复 `hub` 集群备份时会删除通用资源

在 `acm-resources-generic-schedule` 备份中备份的资源不会出现在恢复的 `hub` 集群中。如果检查备份 `Operator` 日志，您会看到类似如下的消息：

```
_2023-06-08T13:42:48.066572033Z 2023-06-08T13:42:48.066Z INFO Deleting resource
DRPlacementControl [c1-helloworld-placement-1-drpc.c1-helloworld] {"controller": "restore",
"controllerGroup": "cluster.open-cluster-management.io", "controllerKind": "Restore", "restore":
{"name": "restore-acm", "namespace": "open-cluster-management-backup"}}
```

#### 1.35.2. 解决问题：在恢复 `hub` 集群备份时会删除通用资源

如果出现以下条件，则会删除这些资源：

- 您有由 `acm-resources-generic-schedule` 备份的资源备份，它们与带有 `cluster.open-cluster-management.io/backup` 标签的 `Secret` 或 `ConfigMap` 资源类型不匹配。
- 您可以运行使用 `Restore.cluster.open-cluster-management.io` 资源的恢复，并将 `cleanupBeforeRestore: CleanupRestored` 值设置为 `CleanupRestored`。
- 最新的 Red Hat Advanced Cluster Management 备份集不包含 `acm-resources-schedule` 备份，因此选择了旧版本的备份。因此，`acm-resources-schedule` 备份的时间戳与 `acm-resources-generic-schedule` 备份不同。在恢复后操作过程中，当处理 `CleanRestore` 选项时，会清理所有通用资源，因为它们没有与 `acm-resources-schedule` 备份相同的时间戳。

要解决这个问题，请再次运行恢复操作，并将 `cleanupBeforeRestore` 值设置为 `None`。

### 1.36. 对多行 YAML 解析进行故障排除

当您想使用 `fromSecret` 功能将 `Secret` 资源的内容添加到 `Route` 资源中时，内容会被错误地显示。

#### 1.36.1. 症状：对多行 YAML 解析进行故障排除

当受管集群和 `hub` 集群与证书数据进行隐藏的集群时，则内容不会被解析为模板 `JSON` 字符串。您可能会收到以下出错信息：

```
message: >-  
  [spec.tls.caCertificate: Invalid value: "redacted ca certificate  
  data": failed to parse CA certificate: data does not contain any  
  valid RSA or ECDSA certificates, spec.tls.certificate: Invalid  
  value: "redacted certificate data": data does not contain any valid  
  RSA or ECDSA certificates, spec.tls.key: Invalid value: "": no key specified]
```

#### 1.36.2. 解决问题：对多行 YAML 解析进行故障排除

配置证书策略，以获取 `hub` 集群和受管集群 `fromSecret` 值。使用 `autoindent` 功能更新您的证书策略，其内容如下：

```
  tls:  
    certificate: |  
      {{ print "{{hub fromSecret \"open-cluster-management\" \"minio-cert\" \"tls.crt\"  
hub}}" | base64dec | autoindent }}
```