Red Hat Advanced Cluster Management for Kubernetes 2.5

集群

了解如何跨云供应商创建、导入和管理集群。

Last Updated: 2023-07-14

Red Hat Advanced Cluster Management for Kubernetes 2.5 集群

了解如何跨云供应商创建、导入和管理集群。

法律通告

摘要

了解如何跨云供应商创建、导入和管理集群。

目录

第 1 章 管理集群

了解如何使用 Red Hat Advanced Cluster Management for Kubernetes 控制台在云供应商中创建、导入和管理集群。通过以下主题了解如何在跨供应商环境中管理集群：

- 支持的供应商
- 扩展受管集群
- 发行镜像
- 创建和修改裸机资产
- 创建基础架构环境
- 管理凭证概述
- 创建集群
- 将目标受管集群导入到 hub 集群
- 在代理环境中创建集群
- 启用集群代理附加组件
- 配置特定的集群管理角色
- 管理集群标签
- 创建和管理 ManagedClusterSets（技术预览）
- 在放置中使用 ManagedClusterSet
- 管理集群池（技术预览）
- 配置 Ansible Tower 任务以在受管集群中运行
- 从集群池中声明集群
- 使用托管的 control plane 集群（技术预览）
- 发现简介
- 升级集群
- 从管理中移除集群
- 集群备份和恢复 Operator

1.1. 集群生命周期架构

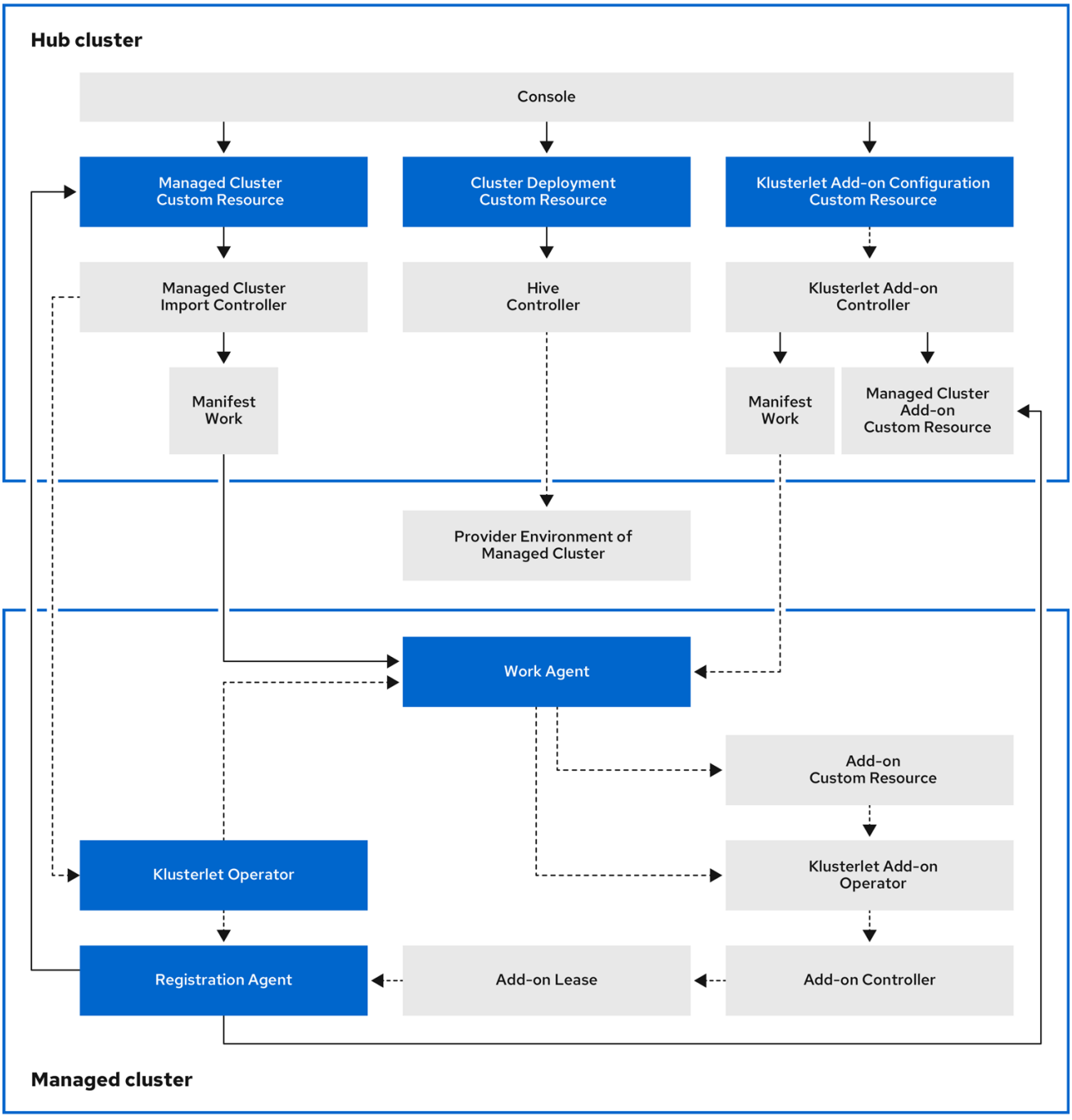
Red Hat Advanced Cluster Management for Kubernetes 有两个主要类型的集群： *hub 集群*和*受管集群*。

hub 集群是在其上安装的 Red Hat Advanced Cluster Management for Kubernetes 的主集群。您可以使用 hub 集群创建、管理和监控其他 Kubernetes 集群。

受管集群是由 hub 集群管理的 Kubernetes 集群。您可以使用 Red Hat Advanced Cluster Management hub 集群创建一些集群，同时也可以导入现有集群由 hub 集群管理。

当使用 Red Hat Advanced Cluster Management 创建受管集群时，集群会使用 Hive 资源的 Red Hat OpenShift Container Platform 集群安装程序创建。您可以通过阅读 OpenShift Container Platform 文档中的 OpenShift Container Platform 安装概述 来了解更多有关使用 OpenShift Container Platform 安装程序安装集群的过程的信息。

下图显示了安装 Red Hat Advanced Cluster Management for 集群管理的组件：



224_RHACM_1022

集群生命周期管理架构的组件包括以下项目：

hub 集群上的组件：

- 控制台：提供基于 Web 的界面，以管理 Red Hat Advanced Cluster Management 受管集群的集群生命周期。
- Hive Controller：置备使用 Red Hat Advanced Cluster Management 创建的集群。Hive Controller 还会分离并销毁由 Red Hat Advanced Cluster Management 创建的受管集群。
- Managed Cluster Import Controller：将 klusterlet operator 部署到受管集群。
- Klusterlet Add-on Controller：将 klusterlet 附加组件 Operator 部署到受管集群。

受管集群中的组件：

- Klusterlet Operator：在受管集群中部署注册和工作控制器。
- Registration Agent：将受管集群注册到 hub 集群。会自动创建以下权限来允许受管集群访问 hub 集群：
  - Clusterrole
    - 允许代理轮转其证书
    - 允许代理 **get/list/update/watch** 由 hub 集群管理的集群
    - 允许代理更新 hub 集群管理的集群状态
  - 在 hub 集群的 hub 集群命名空间中创建的角色
    - 允许受管集群注册代理 **get** 或 **update coordination.k8s.io** 租期
    - 允许代理 **get/list/watch** 受管集群附加组件
    - 允许代理更新受管集群附加组件的状态
- Work Agent：将清单应用到受管集群。会自动创建以下权限来允许受管集群访问 hub 集群：
  - 在 hub 集群的 hub 集群命名空间中创建的角色
    - 允许 Work Agent 将事件发送到 hub 集群
    - 允许代理 **get/list/watch/update manifestworks** 资源
    - 允许代理更新 **manifestworks** 资源的状态

1.2. 扩展受管集群（技术预览）

对于由 Red Hat Advanced Cluster Management 创建的集群，您可以自定义并调整受管集群规格，如虚拟机大小和节点数量。要扩展从其他供应商导入的受管集群，请参阅 供应商受管集群扩展。

技术预览：许多由 Red Hat Advanced Cluster Management for Kubernetes 管理的集群可以使用 Red Hat Advanced Cluster Management 控制台或命令行和 **MachinePool** 资源进行扩展。

- 使用 **MachinePool** 资源是 Red Hat Advanced Cluster Management 创建的裸机集群不支持的功能。
- **MachinePool** 资源是 Hub 集群上的 Kubernetes 资源，用于将 **MachineSet** 资源分组到受管集群上。
- **MachinePool** 资源统一配置一组计算机资源，包括区配置、实例类型和 root 存储。

- 使用 **MachinePool**，您可以手动配置所需的节点数量，或者配置受管集群中的节点的自动扩展。

1.2.1. 自动缩放

配置自动扩展可让集群根据需要进行扩展，从而降低资源成本，在流量较低时进行缩减，并通过向上扩展以确保在资源需求较高时有足够的资源。

1.2.1.1. 启用自动扩展

- 要使用 Red Hat Advanced Cluster Management 控制台在 **MachinePool** 资源上启用自动扩展，请完成以下步骤：

  1. 在 Red Hat Advanced Cluster Management 导航中，选择 **Infrastructure** > **Clusters**。
  2. 点目标集群的名称并选择 *Machine pool* 选项卡。
  3. 在 machine pool 页中，从目标机器池的 *Options* 菜单中选择 **Enable autoscale**。
  4. 选择机器设置副本的最小和最大数量。计算机集副本直接映射到集群中的节点。在点 **Scale** 后，更改可能需要几分钟时间来反映控制台。您可以通过单击 *Machine pools* 中的通知中的 **View machines** 来查看扩展操作的状态。

- 要使用命令行在 **MachinePool** 资源上启用自动扩展，请完成以下步骤：

  1. 输入以下命令查看您的机器池列表：

     ```
     oc get machinepools -n <managed-cluster-namespace>
     ```

     将 **managed-cluster-namespace** 替换为目标受管集群的命名空间。

  2. 输入以下命令为机器池编辑 YAML 文件：

     ```
     oc edit machinepool <name-of-MachinePool-resource> -n <namespace-of-managed-cluster>
     ```

     将 **name-of-MachinePool-resource** 替换为 **MachinePool** 资源的名称。

     将 **namespace-of-managed-cluster** 替换为受管集群的命名空间的名称。

  3. 从 YAML 文件删除 **spec.replicas** 字段。
  4. 在资源 YAML 中添加 **spec.autoscaling.minReplicas** 设置和 **spec.autoscaling.maxReplicas** 项。
  5. 将最小副本数添加到 **minReplicas** 设置。
  6. 将最大副本数添加到 **maxReplicas** 设置中。
  7. 保存文件以提交更改。

为机器池启用自动扩展。

1.2.1.2. 禁用自动扩展

您可以使用控制台或命令行禁用自动扩展。

- 要使用 Red Hat Advanced Cluster Management 控制台禁用自动扩展，请完成以下步骤：

  1. 在 Red Hat Advanced Cluster Management 导航中，选择 **Infrastructure** > **Clusters**。
  2. 点目标集群的名称并选择 *Machine pool* 选项卡。
  3. 在 machine pool 页面中，从目标机器池的 *Options* 菜单中选择 **Disable autoscale**。
  4. 选择您想要的机器集副本数量。机器集副本直接与集群中的节点映射。
     在点 **Scale** 后，在控制台中显示可能需要几分钟时间。您可以点 *Machine pools* 选项卡中的通知中的 **View machine** 来查看扩展的状态。

- 要使用命令行禁用自动扩展，请完成以下步骤：

  1. 输入以下命令查看您的机器池列表：

     ```
     oc get machinepools -n <managed-cluster-namespace>
     ```

     将 **managed-cluster-namespace** 替换为目标受管集群的命名空间。

  2. 输入以下命令为机器池编辑 YAML 文件：

     ```
     oc edit machinepool <name-of-MachinePool-resource> -n <namespace-of-managed-cluster>
     ```

     将 **name-of-MachinePool-resource** 替换为 **MachinePool** 资源的名称。

     将 **namespace-of-managed-cluster** 替换为受管集群的命名空间的名称。

  3. 从 YAML 文件中删除 **spec.autoscaling** 字段。
  4. 将 **spec.replicas** 字段添加到资源 YAML。
  5. 将副本数添加到 **replicas** 设置中。
  6. 保存文件以提交更改。

禁用自动扩展。

1.2.2. 手动扩展集群

如果您不想启用集群自动扩展，可以使用 Red Hat Advanced Cluster Management 控制台或命令行更改您希望集群维护的静态副本数。这有助于根据需要增加或缩小大小。

- 要使用 Red Hat Advanced Cluster Management 控制台手动扩展 **MachinePool** 资源，请完成以下步骤：

  1. 在 Red Hat Advanced Cluster Management 导航中，选择 **Infrastructure** > **Clusters**。
  2. 点目标集群的名称并选择 *Machine pool* 选项卡。
     注： 如果 *Autoscale* 字段中的值是 **Enabled**，您必须首先通过完成禁用自动扩展中的步骤来禁用自动扩展功能，然后再继续。
  3. 从机器池的 *Options* 菜单中，选择 **Scale 机器池**。
  4. 调整机器设置副本数量，以扩展计算机池。

- 要使用命令行扩展 **MachinePool** 资源，请完成以下步骤：

  1. 输入以下命令查看您的机器池列表：

     ```
     oc get machinepools -n <managed-cluster-namespace>
     ```

     将 **managed-cluster-namespace** 替换为目标受管集群的命名空间。

  2. 输入以下命令为机器池编辑 YAML 文件：

     ```
     oc edit machinepool <name-of-MachinePool-resource> -n <namespace-of-managed-cluster>
     ```

     将 **name-of-MachinePool-resource** 替换为 **MachinePool** 资源的名称。

     将 **namespace-of-managed-cluster** 替换为受管集群的命名空间的名称。

  3. 将 YAML 中的 **spec.replicas** 配置更新为副本数。

  4. 保存文件以提交更改。

注：导入的受管集群没有与 Red Hat Advanced Cluster Management 创建的集群相同的资源。因此，扩展集群的步骤有所不同。请参阅您的供应商的产品文档，其中包含有关如何扩展导入集群的信息。

例如，参阅适用于您使用版本的 OpenShift Container Platform 文档中的推荐的集群扩展实践和手动扩展 MachineSet。

1.3. 发行镜像

当使用 Red Hat Advanced Cluster Management for Kubernetes 在供应商处创建集群时，您必须指定用于新集群的发行镜像。发行镜像指定使用哪个版本的 Red Hat OpenShift Container Platform 来构建集群。

引用发行镜像的文件是在 **acm-hive-openshift-releases** GitHub 仓库中维护的 YAML 文件。Red Hat Advanced Cluster Management 使用这些文件在控制台中创建可用发行镜像的列表。这包括 OpenShift Container Platform 的最新快速频道镜像。控制台仅显示三个 OpenShift Container Platform 最新版本的最新发行镜像。例如，您可能在控制台选项中看到以下发行镜像：

- quay.io/openshift-release-dev/ocp-release:4.6.23-x86_64
- quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64

注：只有带有标签为： **visible: 'true'** 的发行镜像才可以在控制台中创建集群时选择。**ClusterImageSet** 资源中的此标签示例在以下内容中提供：

```
apiVersion: config.openshift.io/v1
kind: ClusterImageSet
metadata:
  labels:
    channel: fast
    visible: 'true'
  name: img4.10.1-x86-64-appsub
spec:
  releaseImage: quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64
```

存储了额外的发行镜像，但无法在控制台中看到。要查看所有可用发行镜像，请在 CLI 中运行 **kubectl get clusterimageset**。控制台中只有最新版本可促进创建带有最新发行镜像的集群。在某些情况下，您可能需要创建特定版本的集群，因此还会继续提供老版本。Red Hat Advanced Cluster Management 使用这些文件在控制台中创建可用发行镜像的列表。这包括 OpenShift Container Platform 的最新快速频道镜像。

仓库中包含 **clusterImageSets** 目录和 **subscription** 目录，它们是您使用发行镜像时使用的目录。

clusterImageSets 目录包含以下目录：

- Fast : 包含引用每个受支持 OpenShift Container Platform 版本的最新发行镜像版本的文件。此目录中的发行镜像经过测试、验证和支持。
- Releases : 包含引用每个 OpenShift Container Platform 版本（table、fast 和 candidate 频道）的所有发行镜像的文件 **请注意：**这些版本并没有经过测试并确定是稳定的。
- Stable : 包含引用每个受支持 OpenShift Container Platform 版本的最新两个稳定发行镜像版本的文件。

注： 默认情况下，当前发行镜像列表被更新一次。升级产品后，列表最多可能需要一小时，以反映该产品的新版本的建议发行镜像版本。

您可以通过三种方式对自己的 **ClusterImageSet** 进行策展：

这三种方法中的第一个步骤都是禁用包含自动更新最新快速频道镜像的订阅。使用 **multiclusterhub** 资源中的安装程序参数可以禁用对最新的 fast **ClusterImageSet** 的自动策展。通过切换 **spec.disableUpdateClusterImageSets** 参数为 **true** 和 **false**，Red Hat Advanced Cluster Management 安装的订阅会相应地被启用和禁用。如果要策展自己的镜像，请将 **spec.disableUpdateClusterImageSets** 设置为 **true**，以禁用订阅。

选项 1： 指定要在控制台创建集群时使用的特定 **ClusterImageSet** 的镜像引用。您指定的每个新条目都会保留，并可用于将来的所有集群置备。一个示例为：**quay.io/openshift-release-dev/ocp-release:4.6.8-x86_64**.

选项 2： 手动创建并应用来自 **acm-hive-openshift-releases** GitHub 仓库的 **ClusterImageSets** YAML 文件。

选项 3： 遵循 **acm-hive-openshift-releases** GitHub 仓库中的 **README.md**，启用对来自 fork 的 GitHub 仓库的 **ClusterImageSets** 的自动更新。

subscription 目录包含指定从哪里拉取发行镜像列表的文件。

Red Hat Advanced Cluster Management 的默认发行镜像位于 Quay.io 目录中。

被发行版本 2.5 的 acm-hive-openshift-releases GitHub 仓库 中引用的文件。

1.3.1. 创建发行镜像以在不同构架中部署集群

您可以通过手动创建包含这两个架构文件的发行镜像，在与 hub 集群架构不同的架构中创建集群。

例如，您可以从 **ppc64le**、**aarch64** 或 **s390x** 架构上运行的 hub 集群创建一个 **x86_64** 集群。如果使用两组文件创建发行镜像，集群创建成功，因为新发行镜像启用 OpenShift Container Platform 发行 registry 来提供多架构镜像清单。

要创建发行镜像，请按照以下构架类型完成类似如下的步骤：

1. 在 OpenShift Container Platform release registry 中，创建一个 清单列表，其中包含 **x86_64**、**s390x**、**aarch64** 和 **ppc64le** 发行镜像。

   a. 使用以下命令，从 Quay 仓库拉取环境中这两个架构的清单列表：

   ```
   podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64
   podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-ppc64le
   podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-s390x
   podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-aarch64
   ```

   b. 登录到维护镜像的私有存储库：

   ```
   podman login <private-repo>
   ```

   使用存储库的路径替换 **private-repo**。

   c. 运行以下命令，将发行镜像清单添加到私有存储库中：

   ```
   podman push quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64 <private-repo>/ocp-release:4.10.1-x86_64
   podman push quay.io/openshift-release-dev/ocp-release:4.10.1-ppc64le <private-repo>/ocp-release:4.10.1-ppc64le
   podman push quay.io/openshift-release-dev/ocp-release:4.10.1-s390x <private-repo>/ocp-release:4.10.1-s390x
   podman push quay.io/openshift-release-dev/ocp-release:4.10.1-aarch64 <private-repo>/ocp-release:4.10.1-aarch64
   ```

   使用存储库的路径替换 **private-repo**。

   d. 为新信息创建清单：

   ```
   podman manifest create mymanifest
   ```

   e. 将两个发行镜像的引用添加到清单列表中：

   ```
   podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-x86_64
   podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-ppc64le
   podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-s390x
   podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-aarch64
   ```

   使用存储库的路径替换 **private-repo**。

   f. 将清单列表中的列表与现有清单合并：

   ```
   podman manifest push mymanifest docker://<private-repo>/ocp-release:4.10.1
   ```

   使用存储库的路径替换 **private-repo**。

2. 在 hub 集群中，创建一个发行版本镜像来引用存储库中的清单。

   a. 创建一个 YAML 文件，其中包含类似以下示例的信息：

   ```
   apiVersion: hive.openshift.io/v1
   kind: ClusterImageSet
   metadata:
   ```

```
    labels:
      channel: fast
      visible: "true"
    name: img4.10.1-appsub
  spec:
    releaseImage: <private-repo>/ocp-release:4.10.1
```

使用存储库的路径替换 **private-repo**。

b. 在 hub 集群中运行以下命令以应用更改：

```
oc apply -f <file-name>.yaml
```

将 **file-name** 替换为您刚才创建的 YAML 文件的名称。

3. 在创建 OpenShift Container Platform 集群时选择新的发行镜像。

4. 如果使用 Red Hat Advanced Cluster Management 控制台部署受管集群，在集群创建过程中在 *Architecture* 字段中指定受管集群的架构。

创建流程使用合并的发行镜像来创建集群。

1.3.2. 同步可用发行镜像

版本镜像会频繁更新，因此可能需要同步发行镜像列表，以确保可以选择最新的可用版本。发行镜像位于用于发行版本 2.5 的 acm-hive-openshift-releases GitHub 仓库中。

发行镜像有三个级别的稳定性：

表 1.1. 发行镜像的稳定性级别

类别	描述
stable	已确认用于正确安装和构建集群的完整测试镜像。
fast	部分进行了测试，但稳定性可能低于稳定版本。
candidate	最新镜像，但未经测试。可能会有一些程序错误。

完成以下步骤以刷新列表：

1. 如果启用了安装程序管理的 **acm-hive-openshift-releases** 订阅，在 **multiclusterhub** 资源中将 **disableUpdateClusterImageSets** 的值设置为 **true** 来禁用订阅。

2. 克隆用于发行版本 2.5 的 acm-hive-openshift-releases GitHub 仓库。

3. 输入以下命令删除订阅：

```
oc delete -f subscribe/subscription-fast
```

4. 输入以下命令连接到稳定版本镜像并同步您的 Red Hat Advanced Cluster Management for Kubernetes hub 集群：

```
make subscribe-stable
```

注意：您只能在使用 Linux 或者 MacOS 操作系统时运行这个 **make** 命令。

大约一分钟后，最新的 **stable** 镜像列表将可用。

- 要同步并显示快速发行镜像，请输入以下命令：

  ```
  make subscribe-fast
  ```

  注意：您只能在使用 Linux 或者 MacOS 操作系统时运行这个 **make** 命令。

  运行此命令后，当前可用的镜像会在约一分钟内在可用的 **stable** 和 **fast** 发行镜像列表中出现。

- 要同步并显示 **candidate** 发行镜像，请输入以下命令：

  ```
  make subscribe-candidate
  ```

  注意：您只能在使用 Linux 或者 MacOS 操作系统时运行这个 **make** 命令。

  运行该命令后，当前可用的镜像会更新可用的 **stable**、**fast** 和 **candidate** 发行镜像列表。

5. 在创建集群时，查看 Red Hat Advanced Cluster Management 控制台中当前可用发行镜像的列表。

6. 使用以下格式输入命令来从这些频道中取消订阅以停止查看更新：

   ```
   oc delete -f subscribe/subscription-fast
   ```

1.3.3. 连接时维护自定义的发行镜像列表

您可能希望确保所有集群都使用同一发行镜像。为简化操作，您可以创建自己的自定义列表，在其中包含创建集群时可用的发行镜像。完成以下步骤以管理可用发行镜像：

1. 如果启用了安装程序管理的 **acm-hive-openshift-releases** 订阅，在 **multiclusterhub** 资源中将 **disableUpdateClusterImageSets** 的值设置为 **true** 来禁用它。

2. 对 acm-hive-openshift-releases GitHub 仓库 2.5 分支 进行分叉（fork）。

3. 通过将 **spec: pathname** 更改为访问已分叉的仓库的 GitHub 名称，而不是 **stolostron**，更新 **./subscribe/channel.yaml** 文件。此步骤指定 hub 集群在哪里检索发行镜像。您更新的内容应类似以下示例：

   ```
   spec:
     type: Git
     pathname: https://github.com/<forked_content>/acm-hive-openshift-releases.git
   ```

   将 **forked_content** 替换为已分叉仓库的路径。

4. 使用 Red Hat Advanced Cluster Management for Kubernetes 控制台创建集群时可用的镜像，将 YAML 文件添加到 *./clusterImageSets/stable/** 或 *./clusterImageSets/fast/** 目录中。
   提示： 您可以通过将更改合并到已分叉的存储库，从主存储库检索可用的 YAML 文件。

5. 将更改提交并合并到您的已分叉仓库。

6. 在克隆了 **acm-hive-openshift-releases** 仓库后，使用以下命令来更新 fast 镜像后同步 fast 发行镜像列表：

```
make subscribe-fast
```

注意：您只能在使用 Linux 或者 MacOS 操作系统时运行这个 **make** 命令。

运行此命令后，可用快速镜像列表会在约 1 分钟内更新为当前可用镜像。

7. 默认情况下，只列出 fast 镜像。要同步并显示稳定的发行镜像，请输入以下命令：

```
make subscribe-stable
```

注意：您只能在使用 Linux 或者 MacOS 操作系统时运行这个 **make** 命令。

运行此命令后，可用稳定镜像列表会在约 1 分钟内更新为当前可用镜像。

8. 默认情况下，Red Hat Advanced Cluster Management 会预加载几个 ClusterImageSets。您可以使用以下命令列出可用内容并删除默认值。

```
oc get clusterImageSets
oc delete clusterImageSet <clusterImageSet_NAME>
```

注： 如果您还没有通过将 **multiclusterhub** 资源中的 **disableUpdateClusterImageSets** 的值设置为 **true** 来禁用安装程序管理的 **ClusterImageSet** 的自动更新，则您删除的任何镜像都会被自动重新创建。

9. 在创建集群时，查看 Red Hat Advanced Cluster Management 控制台中当前可用发行镜像的列表。

1.3.4. 断开连接时维护自定义的发行镜像列表

在某些情况下，当节点集群没有互联网连接时，您需要维护一个自定义的发行镜像列表。您可以创建自己的自定义列表，在其中包含创建集群时可用的发行镜像。完成以下步骤以在断开连接的情况下管理可用发行镜像：

1. 在连接的系统中时，导航到 acm-hive-openshift-releases GitHub 仓库，以访问可用于版本 2.5 的集群镜像集。

2. 将 **clusterImageSets** 目录复制到可以访问断开连接的 Red Hat Advanced Cluster Management for Kubernetes hub 集群的系统中。

3. 通过完成以下步骤，添加受管集群和带有集群镜像的断开连接的存储库之间的映射：

   - 对于 OpenShift Container Platform 受管集群，请参阅配置镜像 registry 存储库镜像 以了解有关使用 **ImageContentSourcePolicy** 对象完成映射的信息。

   - 对于不是 OpenShift Container Platform 集群的受管集群，使用 **ManageClusterImageRegistry** CRD 覆盖镜像集的位置。如需有关如何为映射覆盖集群的信息，请参阅使用自定义 ManagedClusterImageRegistry CRD 导入集群。

4. 使用 Red Hat Advanced Cluster Management 控制台手动添加 **clusterImageSet** YAML 内容，为在创建集群时可用的镜像添加 YAML 文件。

5. 修改 OpenShift Container Platform 发行镜像的 **clusterImageSet** YAML 文件，以引用存储镜像的正确离线存储库。您的更新应类似以下示例：

```
apiVersion: hive.openshift.io/v1
kind: ClusterImageSet
metadata:
  name: img4.4.0-rc.6-x86-64
spec:
  releaseImage: IMAGE_REGISTRY_IPADDRESS_or_DNSNAME/REPO_PATH/ocp-release:4.4.0-rc.6-x86_64
```

确保在 YAML 文件中引用的离线镜像 registry 中载入镜像。

6. 通过为每个 **YAML** 文件输入以下命令来创建每个 clusterImageSets：

```
oc create -f <clusterImageSet_FILE>
```

将 **clusterImageSet_FILE** 替换为集群镜像集文件的名称。例如：

```
oc create -f img4.9.9-x86_64.yaml
```

在为您要添加的每个资源运行此命令后，可用发行镜像列表将变为可用。

7. 另外，您还可以将镜像 URL 直接粘贴到 Red Hat Advanced Cluster Management 的创建集群控制台中。如果镜像 URL 不存在，添加镜像 URL 会创建新的 clusterImageSet。

8. 在创建集群时，查看 Red Hat Advanced Cluster Management 控制台中当前可用发行镜像的列表。

1.4. 创建和修改裸机资产

弃用通知： 使用裸机资产创建裸机集群的步骤已弃用。请参阅在内部环境中为创建集群。

裸机资产是您配置为运行 OpenShift Container Platform 集群的虚拟或物理服务器。Red Hat Advanced Cluster Management for Kubernetes 会连接到管理员创建的裸机资产。然后您可以在受管集群中部署裸机资产。

hub 集群清单控制器定义一个名为 **BareMetalAsset** 的自定义资源定义 (CRD)，其中包含裸机资产清单记录。在置备受管集群时，清单控制器会将 **BareMetalAsset** 清单记录与受管集群中对应的 **BareMetalHost** 资源协调。

Red Hat Advanced Cluster Management 使用 **BareMetalAsset** CR 根据配置管理数据库 (CMDB) 或类似的系统中输入的记录置备集群硬件。外部工具或自动化轮询 CMDB，并使用 Red Hat Advanced Cluster Management API 在 hub 集群中创建对应的 **BareMetalAsset** 和对应的 **Secret** 资源，以便后续在受管集群中部署。

使用以下步骤为由 Red Hat Advanced Cluster Management 管理的集群创建和管理裸机资产。

- 先决条件
- 使用控制台创建裸机资产
- 使用 CLI 创建裸机资产
- 使用控制台批量导入裸机资产
- 修改裸机资产

- 删除裸机资产
- 使用 REST API 创建裸机资产

1.4.1. 先决条件

创建裸机资产前需要满足以下先决条件：

- 在 OpenShift Container Platform 版本 4.6 或更高版本上部署了 Red Hat Advanced Cluster Management hub 集群。
- 访问 Red Hat Advanced Cluster Management hub 集群以连接到裸机资产。
- 配置了裸机资产，以及登录凭证（包含登录和管理该资产所需的权限）。
  注：裸机资产凭证包括您的管理员提供的资产的以下项：**用户名**密码 **Baseboard Management Controller (BMC) Address** 引导 NIC MAC 地址

1.4.2. 使用控制台创建裸机资产

要使用 Red Hat Advanced Cluster Management for Kubernetes 控制台创建裸机资产，请进入 **Infrastructure** > **Bare metal assets**。选择 **Create bare metal asset**，并在控制台中完成这个流程。

裸机资产的名称在创建集群时标识它。

裸机资产、受管裸机集群及其相关 secret 必须位于同一命名空间中。

+ 有权访问此命名空间的用户可以在创建集群时将此资产与集群相关联。

基板管理控制器地址是启用与主机通信的控制器。支持以下协议：

- IPMI，如需更多信息，请参阅 IPMI 2.0 规格。
- iDRAC，如需更多信息，请参阅Dell Remote Access Controller 9（iDRAC9）的支持 。
- iRMC，如需更多信息，请参阅数据手册：FUJITSU Software ServerView Suite 集成远程管理控制器 - iRMC S5。
- Redfish，如需更多信息，请参阅 Redfish 规格。

引导 NIC MAC 地址是主机网络连接 NIC 的 MAC 地址，用于在裸机资产上置备主机。

您可以继续在裸机上创建集群。

1.4.3. 使用 CLI 创建裸机资产

使用 **BareMetalAsset** CR 为集群中的特定命名空间创建裸机资产。每个 **BareMetalAsset** 都有一个对应的 **Secret**，其中包含 Baseboard Management Controller (BMC) 凭证以及同一命名空间中的 secret 名称。

1.4.3.1. 先决条件

- 在 hub 集群中安装 Red Hat Advanced Cluster Management for Kubernetes。
- 安装 Red Hat OpenShift CLI (oc)。
- 以具有 **cluster-admin** 权限的用户身份登录。

1.4.3.2. 创建裸机资产

1. 在您的环境中安装并置备裸机资产。
2. 打开 BMC，并记录硬件的 IPMI 或 Redfish BMC 地址和 MAC 地址。
3. 创建以下 **BareMetalAsset** 和 **Secret CR，并将文件保存为baremetalasset-cr.yaml** ：

```
apiVersion: inventory.open-cluster-management.io/v1alpha1
kind: BareMetalAsset
metadata:
  name: <baremetalasset-machine>
  namespace: <baremetalasset-namespace>
spec:
  bmc:
    address: ipmi://<out_of_band_ip>:<port>
    credentialsName: baremetalasset-machine-secret
  bootMACAddress: "00:1B:44:11:3A:B7"
  hardwareProfile: "hardwareProfile"
  role: "<role>"
  clusterName: "<cluster name>"
---
apiVersion: v1
kind: Secret
metadata:
  name: baremetalasset-machine-secret
type: Opaque
data:
  username: <username>
  password: <password>
```

- 使用裸机资产所在机器的名称替换 **baremetalasset-machine**。创建后，受管集群中的 **BareMetalHost** 的名称与 hub 集群上对应的 **BareMetalAsset** 的名称相同。**BareMetalHost** 名称应始终与对应的 **BareMetalAsset** 名称匹配。
- 使用创建裸机资产的集群命名空间替换 **baremetalasset-namespace**。
- 将 **out_of_band_ip** 和 **port** 替换为裸机资产的地址和端口。对于 Redfish 寻址，请使用以下地址格式： **redfish://<out-band-ip>/redfish/v1/Systems/1**。
- 根据计算机角色类型，将 **role** 替换为 **worker**、**master** 或保留为空。**role** 设置用于将裸机资产与集群中的特定机器角色类型匹配。**BareMetalAsset** 资源不应用于指定机器角色类型来填充另一个角色。**role** 角色值被用作键为 **inventory.open-cluster-management.io/role** 的标签的值。这可让集群管理应用程序或用户查询用于特定角色的清单。
- 使用集群的名称替换 **cluster_name**，集群管理应用程序或用户使用该名称查询与特定集群关联的清单。保留这个值为空以创建裸机资产，而不将其添加到集群部署中。
- 使用您的 secret 的用户名替换 **username**。
- 将 **password** 替换为您的 secret 的密码。

4. 运行以下命令来创建 **BareMetalAsset** CR：

```
oc create -f baremetalasset-cr.yaml
```

5. 检查 **BareMetalAsset** 是否已成功创建：

```
oc get baremetalassets -A
```

输出示例：

```
NAMESPACE        NAME                              AGE
ocp-example-bm    baremetalasset-machine           2m
ocp-example-bm    csv-f24-h27-000-r630-master-1-1  4d21h
```

1.4.4. 使用控制台批量导入裸机资产

您可以使用 CSV 格式列表使用 Red Hat Advanced Cluster Management for Kubernetes 控制台批量导入裸机资产。

1.4.4.1. 先决条件

- 在一个管理一个或多个 spoke 集群的 hub 集群上安装 Red Hat Advanced Cluster Management。
- 安装 OpenShift Container Platform CLI（oc）。
- 以具有 **cluster-admin** 权限的用户身份登录。

1.4.4.2. 导入资产

要导入一组裸机资产，请完成以下步骤：

1. 在 Red Hat Advanced Cluster Management 控制台中，在导航菜单中选择 **Cluster management** > **Bare metal asset**。
2. 选择 **Import asset**，并导入包含裸机资产数据的 CSV 文件。CSV 文件必须具有以下标题列：

```
hostName, hostNamespace, bmcAddress, macAddress, role (optional), username, password
```

1.4.5. 修改裸机资产

如果您需要修改裸机资产的设置，请完成以下步骤：

1. 在 Red Hat Advanced Cluster Management for Kubernetes 控制台导航中，选择：**Infrastructure** > **Bare metal assets**。
2. 选择表中您要修改的资产的选项菜单。
3. 选择 **Edit asset**。

1.4.6. 删除裸机资产

当裸机资产不再用于任何集群时，您可以将其从可用的裸机资产列表中删除。删除未使用的资产既可以简化您的可用资产列表，又可以防止意外选择该资产。

要在控制台中删除裸机资产，请完成以下步骤：

1. 在 Red Hat Advanced Cluster Management for Kubernetes 控制台导航中，选择：**Infrastructure** > **Bare metal assets**。

2. 选择表中要删除的资产的选项菜单。

3. 选择 **Delete asset**。

1.4.7. 使用 REST API 创建裸机资产

您可以使用 OpenShift Container Platform REST API 管理要在 Red Hat Advanced Cluster Management 集群中使用的裸机资产。当您有单独的 CMDB 应用程序或数据库来管理环境中的裸机资产时，这很有用。

1.4.7.1. 先决条件

- 在 hub 集群中安装 Red Hat Advanced Cluster Management for Kubernetes。
- 安装 OpenShift Container Platform CLI（oc）。
- 以具有 **cluster-admin** 权限的用户身份登录。

1.4.7.2. 创建裸机资产

要使用 REST API 创建裸机资产，请执行以下操作：

1. 获取 hub 集群的登录令牌，并通过命令行登录集群。例如：

```
oc login --token=<login_token> --server=https://<hub_cluster_api_url>:6443
```

2. 使用您要添加到集群的裸机资产详情修改以下 curl 命令，并运行命令。

```
$ curl --location --request POST '<hub_cluster_api_url>:6443/apis/inventory.open-cluster-management.io/v1alpha1/namespaces/<bare_metal_asset_namespace>/baremetalassets?fieldManager=kubectl-create' \
--header 'Authorization: Bearer <login_token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "apiVersion": "inventory.open-cluster-management.io/v1alpha1",
    "kind": "BareMetalAsset",
    "metadata": {
        "name": "<baremetalasset_name>",
        "namespace": "<bare_metal_asset_namespace>"
    },
    "spec": {
        "bmc": {
            "address": "ipmi://<ipmi_address>",
            "credentialsName": "<credentials-secret>"
        },
        "bootMACAddress": "<boot_mac_address>",
        "clusterName": "<cluster_name>",
        "hardwareProfile": "hardwareProfile",
        "role": "worker"
    }
}'
```

- **Replacebaremetalasset-name** 使用裸机资产的名称。创建后，受管集群中的 **BareMetalHost** 的名称与 hub 集群上对应的 **BareMetalAsset** 的名称相同。**BareMetalHost** 名称应始终与对应的 **BareMetalAsset** 名称匹配。

- 使用创建裸机资产的集群命名空间替换 **baremetalasset-namespace**。
- 将 **out_of_band_ip** 和 **port** 替换为裸机资产的地址和端口。对于 Redfish 寻址，请使用以下地址格式： **redfish://<out-band-ip>/redfish/v1/Systems/1**。
- 根据计算机角色类型，将 **role** 替换为 **worker**、**master** 或保留为空。**role** 设置用于将裸机资产与集群中的特定机器角色类型匹配。**BareMetalAsset** 资源不应用于指定机器角色类型来填充另一个角色。**role** 角色值被用作键为 **inventory.open-cluster-management.io/role** 的标签的值。这可让集群管理应用程序或用户查询用于特定角色的清单。
- 使用集群的名称替换 **cluster_name**，集群管理应用程序或用户使用该名称查询与特定集群关联的清单。保留这个值为空以创建裸机资产，而不将其添加到集群部署中。
  注：对于上一个 curl 命令，假设 API 服务器通过 HTTPS 提供，并可以安全地访问。在开发或测试环境中，您可以传递 **--insecure** 参数。

提示： 您可以将 **--v=9** 附加到 **oc** 命令来查看生成的操作的原始输出。这对于确定 **oc** 命令的 REST API 路由非常有用。

1.5. 创建基础架构环境

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台创建一个基础架构环境，以管理主机并在这些主机上创建集群。

- 前提条件
- 启用中央基础架构管理服务
  - 手动创建 Provisioning 自定义资源 (CR)
  - 在 Amazon Web Services 上启用中央基础架构管理
- 使用控制台创建基础架构环境

基础架构环境支持以下功能：

- 集群的零接触置备：使用脚本部署集群。如需更多信息，请参阅 Red Hat OpenShift Container Platform 文档中的在断开连接的环境中部署分布式单元。
- 最后绑定：启用主机由基础架构管理员引导，并且集群的创建者可以稍后将集群绑定到该主机。在使用较晚绑定时，集群创建者不需要具有基础架构的管理员特权。
- 双堆栈：部署具有 IPv4 和 IPv6 地址的集群。双堆栈使用 **OVN-Kubernetes** 网络实现来支持多个子网。
- 添加远程 worker 节点：在集群创建并运行后向集群添加远程 worker 节点，从而提供在其他位置添加节点以进行备份的灵活性。
- 使用 NMState 的静态 IP：使用 NMState API 为您的环境定义静态 IP 地址。

1.5.1. 先决条件

在创建基础架构环境前，请查看以下先决条件：

- 您必须在 hub 集群中部署了 OpenShift Container Platform。

- 对于联网环境，通过互联网访问 Red Hat Advanced Cluster Management hub 集群；对于非联网环境，连接到一个可访问互联网的内部或镜像 registry，以获取创建环境所需的镜像。
- hub 集群中中央基础架构管理 (CIM) 功能配置的实例。请参阅启用中央基础架构管理服务。
- 您需要 OpenShift Container Platform pull secret。如需更多信息，请参阅使用镜像 pull secret。
- 默认情况下，您的 SSH 密钥位于 ~/**.ssh/id_rsa.pub** 文件中。
- 您需要一个配置的存储类。
- **断开连接的环境：** 完成 OpenShift Container Platform 文档中的 准备断开连接的环境的步骤。

1.5.2. 启用中央基础架构管理服务

Central Infrastructure Management 服务随 {mce-short} 提供，并部署 OpenShift Container Platform 集群。当您在 hub 集群上启用 **MultiClusterHub** Operator 时，CIM 被部署，但必须启用。

要启用 CIM 服务，请完成以下步骤：

重要： 只有在以下其中一个平台上安装了 hub 集群：裸机、Red Hat OpenStack Platform、VMware vSphere 或使用用户置备的基础架构(UPI)方法安装，且平台为 **None**，请完成以下步骤。如果您的 hub 集群位于任何其他平台上，请跳过这一步。

1. 运行以下命令，修改 **Provisioning** 资源以允许 Bare Metal Operator 监视所有命名空间：

```
oc patch provisioning provisioning-configuration --type merge -p '{"spec":
{"watchAllNamespaces": true }}'
```

2. **对于断开连接的环境：** 在与基础架构 Operator *相同的命名空间中创建一个* **ConfigMap**，为您的镜像 registry 指定 **ca-bundle.crt** 和 **registry.conf** 的值。您的文件 **ConfigMap** 应该类似以下示例：

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: <mirror-config>
  namespace: "<infrastructure-operator-namespace>"
  labels:
    app: assisted-service
data:
  ca-bundle.crt: |
    -----BEGIN CERTIFICATE-----
    certificate contents
    -----END CERTIFICATE-----

  registries.conf: |
    unqualified-search-registries = ["registry.access.redhat.com", "docker.io"]

    [[registry]]
      prefix = ""
      location = "quay.io/edge-infrastructure"
      mirror-by-digest-only = false

      [[registry.mirror]]
      location = "mirror1.registry.corp.com:5000/edge-infrastructure"
```

1.5.2.1. 创建 *AgentServiceConfig* 自定义资源

通过完成以下步骤来创建 **AgentServiceConfig** 自定义资源：

1. **仅限断开连接的环境：** 在 **agent_service_config.yaml** 文件中保存以下 **YAML** 内容，并根据需要替换值：

```
apiVersion: agent-install.openshift.io/v1beta1
kind: AgentServiceConfig
metadata:
 name: agent
spec:
  databaseStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <db_volume_size>
  filesystemStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <fs_volume_size>
  mirrorRegistryRef:
    name: <mirror_config>
  unauthenticatedRegistries:
    - <unauthenticated_registry>
  imageStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <img_volume_size>
  osImages:
    - openshiftVersion: "<ocp_version>"
      version: "<ocp_release_version>"
      url: "<iso_url>"
      rootFSUrl: "<root_fs_url>"
      cpuArchitecture: "x86_64"
```

将 **mirror_config** 替换为包含您的镜像 registry 配置详情的 **ConfigMap** 名称。

如果您使用不需要身份验证的镜像 registry，请包含可选的 **unauthenticated_registry** 参数。此列表上的条目不会被验证，或者需要在 pull secret 中有一个条目。

2. **仅限连接的环境：** 在 **agent_service_config.yaml** 文件中保存以下 **YAML** 内容：

```
apiVersion: agent-install.openshift.io/v1beta1
kind: AgentServiceConfig
metadata:
 name: agent
spec:
  databaseStorage:
    accessModes:
    - ReadWriteOnce
```

```
    resources:
      requests:
        storage: <db_volume_size>
  filesystemStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <fs_volume_size>
  imageStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <img_volume_size>
```

使用 **databaseStorage** 字段的卷大小替换 **db_volume_size**，如 **10Gi**。这个值指定为存储集群分配的存储量，如数据库表和数据库视图。如果有多个集群，您可能需要使用较高的值。

将 **fs_volume_size** 替换为 **filesystemStorage** 字段的卷大小，例如，每个集群 **200M** 和每个支持的 OpenShift Container Platform 版本 **2-3G**。所需的最小值为 **100G**。这个值指定为存储集群的日志、清单和 **kubeconfig** 文件分配了多少存储。如果有多个集群，您可能需要使用较高的值。

将 **img_volume_size** 替换为 **imageStorage** 字段的卷大小，例如每个操作系统镜像的 **2G**。最小值为 **50G**。这个值指定为集群镜像分配多少存储。您需要为每个运行的 Red Hat Enterprise Linux CoreOS 实例提供 1 GB 的镜像存储。如果 Red Hat Enterprise Linux CoreOS 有多个集群和实例，您可能需要使用更高的值。

将 **ocp_version** 替换为要安装的 OpenShift Container Platform 版本，如 **4.9**。

将 **ocp_release_version** 替换为特定的安装版本，例如：**49.83.202103251640-0**。

使用 ISO url 替换 **iso_url**，例如 **https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.10/4.10.3/rhcos-4.10.3-x86_64-live.x86_64.iso**。您可以在以下位置找到其他值：https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.10/4.10.3/。

使用 root FS 镜像 URL 替换 **root_fs_url**，例如 **https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.10/4.10.3/rhcos-4.10.3-x86_64-live-rootfs.x86_64.img**。您可以在以下位置找到其他值：https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.10/4.10.3/。

3. 运行以下命令来创建 **AgentServiceConfig** 自定义资源：

```
oc create -f agent_service_config.yaml
```

输出可能类似以下示例：

```
agentserviceconfig.agent-install.openshift.io/agent created
```

您可以通过检查 **assisted-service** 和 **assisted-image-service** 部署，确定 pod 已就绪并在运行，来验证其状态是否正常。使用控制台继续创建基础架构环境。

1.5.2.2. 手动创建 Provisioning 自定义资源 (CR)

使用以下命令手动创建 **Provisioning** CR 来为自动置备启用服务：

```
oc create -f provisioning-configuration.yaml
```

您的 CR 可能类似以下示例：

```
apiVersion: metal3.io/v1alpha1
kind: Provisioning
metadata:
  name: provisioning-configuration
spec:
  provisioningNetwork: Disabled
  watchAllNamespaces: true
```

1.5.2.3. 在 Amazon Web Services 上启用中央基础架构管理

如果您在 Amazon Web Services 上运行 hub 集群并希望启用 CIM 服务，请在 启用 CIM 后完成以下步骤：

1. 确保已在 hub 中登录，并通过运行以下命令查找在 **assisted-image-service** 上配置的唯一域：

   ```
   oc get routes --all-namespaces | grep assisted-image-service
   ```

   您的域可能类似以下示例：**assisted-image-service-multicluster-engine.apps.<yourdomain>.com**

2. 确保已在 hub 中登录，并使用 **NLB type** 参数创建带有唯一域的新 **IngressController**。请参见以下示例：

   ```
   apiVersion: operator.openshift.io/v1
   kind: IngressController
   metadata:
     name: ingress-controller-with-nlb
     namespace: openshift-ingress-operator
   spec:
     domain: nlb-apps.<domain>.com
     routeSelector:
        matchLabels:
          router-type: nlb
     endpointPublishingStrategy:
       type: LoadBalancerService
       loadBalancer:
         scope: External
         providerParameters:
           type: AWS
           aws:
             type: NLB
   ```

3. 将 **<yourdomain>** 添加到 **IngressController** 中的 **domain** 参数，方法是使用 **<yourdomain>** 替换 **nlb-apps.<domain>.com** 的 **<domain>**。

4. 使用以下命令应用新的 **IngressController** ：

   ```
   oc apply -f ingresscontroller.yaml
   ```

5. 运行以下命令来编辑 **assisted-image-service** 路由以使用 **nlb-apps** 位置：

```
oc edit route assisted-image-service -n <namespace>
```

提示：默认命名空间是您安装 :mce: 的位置。

6. 在 **assisted-image-service** 路由中添加以下行：

```
metadata:
  labels:
    router-type: nlb
  name: assisted-image-service
```

7. 在 **assisted-image-service** 路由中，找到 **spec.host** 的 URL 值。URL 可能类似以下示例：**assisted-image-service-multicluster-engine.apps.<yourdomain>.com**

8. 将 URL 中的 **apps** 替换为 **nlb-apps**，以匹配新 **IngressController** 中配置的域。

要验证 CIM 服务是否在 Amazon Web Services 上启用，请完成以下步骤：

1. 运行以下命令验证 pod 是否健康：

```
oc get pods -n multicluster-engine | grep assist
```

2. 创建新的基础架构环境，并确保下载 URL 使用新的 **nlb-apps** URL。

1.5.3. 使用控制台创建基础架构环境

要从 Red Hat Advanced Cluster Management 控制台创建基础架构环境，请完成以下步骤：

1. 在导航菜单中导航到 **Infrastructure** > **Infrastructure environments**，再点 **Create infrastructure environment**。

2. 在您的基础架构环境设置中添加以下信息：

   - 名称：您的环境的唯一名称。
   - 网络类型：指定可以将哪些类型的主机添加到您的环境中。您只能在使用裸机主机时使用静态 IP 选项。
   - 位置：指定主机的地理位置。地理位置可用于在创建集群时轻松确定集群中的数据的存储位置。
   - 标签：您可以在基础架构环境中添加标签的可选字段，以便您可以更轻松地将环境与具有特征的其他环境进行分组。您为网络类型和位置所做的选择将自动添加到标签列表中。
   - pull secret：用于访问 OpenShift Container Platform 资源的 OpenShift Container Platform pull secret。
   - SSH 公钥：实现与主机安全通信的 SSH 密钥。默认情况下，这通常在您的 **~/.ssh/id_rsa.pub** 文件中。
   - 如果要在所有集群中启用代理设置，请选择设置来启用它。这要求您输入以下信息：
     - HTTP 代理 URL：在访问发现服务时使用的 URL。

- HTTPS 代理 URL：访问发现服务时应使用的安全代理 URL。请注意，格式必须是 **http**，因为尚不支持 **https**。
- 无代理域：应当绕过代理的以逗号分隔的域列表。使用一个句点 (.) 开始的域名，包含该域中的所有子域。添加星号 (*) 以绕过所有目的地的代理。

现在，您可以通过将主机添加到基础架构环境来继续。

要访问基础架构环境，请在控制台中选择 **Infrastructure** > **Host inventory**。从列表中选择您的基础架构环境，以查看该基础架构环境的详细信息和主机。

1.5.4. 将主机添加到基础架构环境中

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台将主机添加到基础架构环境中。通过添加主机，可以在创建集群时选择已配置的主机。

完成以下步骤以添加主机：

1. 在 Red Hat Advanced Cluster Management 导航中，选择 **Infrastructure** > **Infrastructure environments**。
2. 选择您要添加主机以查看其设置的基础架构环境。
3. 选择 *Hosts* 选项卡来查看已添加到该环境中的主机并添加主机。可用主机可能需要几分钟时间才会出现在表中。
4. 选择 **Discovery ISO** 或 **Baseboard Management Controller(BMC)** 来输入主机的信息。
5. 如果您选择了 **Discovery ISO** 选项，请完成以下步骤：
   a. 复制控制台中提供的命令，以下载 ISO 或选择 **Download Discovery ISO**。
   b. 在可引导设备上运行 命令，以启动每个主机。
   c. 若要提高安全性，您需要为每个发现的主机选择**批准主机**。这一额外步骤可在您的 ISO 文件由未经授权的人员更改并运行时提供一些保护。
   d. 将主机（即 **localhost**）重命名为唯一名称。
6. 如果您选择 **Baseboard Management Controller(BMC)** 选项，请完成以下步骤：
   注：只有 Red Hat Advanced Cluster Management hub 集群平台为裸机、Red Hat OpenStack Platform、VMware vSphere 或是使用用户置备的基础架构(UPI)方法安装的，可以使用添加主机的 BMC 选项，平台为 **None**。
   a. 添加主机的 BMC 连接详情。
   b. 选择 **Add host** 以开始引导过程。主机通过使用发现 ISO 镜像自动引导，并在主机启动时添加到主机列表中。
   当您使用 BMC 选项添加主机时，该主机会被自动批准。

现在，您可以在此基础架构环境中创建一个内部集群。如需了解更多与创建集群相关的信息，请参阅在内部环境中创建集群。

1.6. 创建集群

了解如何使用 Red Hat Advanced Cluster Management for Kubernetes 跨云供应商创建 Red Hat OpenShift Container Platform 集群。

multicluster-engine 使用 OpenShift Container Platform 提供的 Hive operator 为除内部集群和托管 control plane 之外的所有供应商置备集群。在置备内部集群时，multicluster-engine 使用 OpenShift Container Platform 提供的中央基础架构管理 (CIM) 和辅助安装程序功能。托管 control plane 托管的集群使用 HyperShift operator 置备。

- 在集群创建过程中配置额外清单
- 在 Amazon Web Services 上创建集群
- 在 Microsoft Azure 上创建集群
- 在 Google Cloud Platform 上创建集群
- 在 VMware vSphere 上创建集群
- 在 Red Hat OpenStack Platform 上创建集群
- 在 Red Hat Virtualization 上创建集群
- 在裸机上创建集群
- 在内部环境中创建集群

1.6.1. 在集群创建过程中配置额外清单

您可以在创建集群的过程中配置额外的 Kubernetes 资源清单。如果您需要为配置网络或设置负载均衡器等场景配置额外清单，这可以提供帮助。

在创建集群前，您需要添加对 **ClusterDeployment** 资源的引用，该资源指定包含其他资源清单的 **ConfigMap**。

注： **ClusterDeployment** 资源和 **ConfigMap** 必须位于同一命名空间中。以下示例演示了您的内容看起来如何。

- 带有资源清单的 ConfigMap
  包含具有另一个 **ConfigMap** 资源的清单的 **ConfigMap**。资源清单 **ConfigMap** 可以包含多个键，并在 **data.<resource_name>\.yaml** 模式中添加资源配置。

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: <my-baremetal-cluster-install-manifests>
  namespace: <mynamespace>
data:
  99_metal3-config.yaml: |
    kind: ConfigMap
    apiVersion: v1
    metadata:
      name: metal3-config
      namespace: openshift-machine-api
    data:
      http_port: "6180"
      provisioning_interface: "enp1s0"
```

```
    provisioning_ip: "172.00.0.3/24"
    dhcp_range: "172.00.0.10,172.00.0.100"
    deploy_kernel_url: "http://172.00.0.3:6180/images/ironic-python-agent.kernel"
    deploy_ramdisk_url: "http://172.00.0.3:6180/images/ironic-python-agent.initramfs"
    ironic_endpoint: "http://172.00.0.3:6385/v1/"
    ironic_inspector_endpoint: "http://172.00.0.3:5150/v1/"
    cache_url: "http://192.168.111.1/images"
    rhcos_image_url: "https://releases-art-
rhcos.svc.ci.openshift.org/art/storage/releases/rhcos-
4.3/43.81.201911192044.0/x86_64/rhcos-43.81.201911192044.0-
openstack.x86_64.qcow2.gz"
```

- 使用引用的资源清单 **ConfigMap** 的 ClusterDeployment
  资源清单 **ConfigMap** 在 **spec.provisioning.manifestsConfigMapRef** 下引用。

```
apiVersion: hive.openshift.io/v1
kind: ClusterDeployment
metadata:
  name: <my-baremetal-cluster>
  namespace: <mynamespace>
  annotations:
    hive.openshift.io/try-install-once: "true"
spec:
  baseDomain: test.example.com
  clusterName: <my-baremetal-cluster>
  controlPlaneConfig:
    servingCertificates: {}
  platform:
    baremetal:
      libvirtSSHPrivateKeySecretRef:
        name: provisioning-host-ssh-private-key
  provisioning:
    installConfigSecretRef:
      name: <my-baremetal-cluster-install-config>
    sshPrivateKeySecretRef:
      name: <my-baremetal-hosts-ssh-private-key>
    manifestsConfigMapRef:
      name: <my-baremetal-cluster-install-manifests>
    imageSetRef:
      name: <my-clusterimageset>
    sshKnownHosts:
    - "10.1.8.90 ecdsa-sha2-nistp256
AAAAE2VjZHNhLXvVVVKUYVkuyvkuygkuyTCYTytfkufTYAAAAIbmlzdHAyNTYAAABBBKWjJR
zeUVuZs4yxSy4eu45xiANFIIbwE3e1aPzGD58x/NX7Yf+S8eFKq4RrsfSaK2hVJyJjvVIhUsU9z2s
BJP8="
  pullSecretRef:
    name: <my-baremetal-cluster-pull-secret>
```

1.6.2. 在 Amazon Web Services 上创建集群

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台在 Amazon Web Services (AWS) 上创建一个 Red Hat OpenShift Container Platform 集群。

在创建集群时，创建过程使用 Hive 资源中的 OpenShift Container Platform 安装程序。如果在完成此步骤后集群创建有疑问，请参阅 OpenShift Container Platform 文档中的在 AWS 上安装 有关流程的更多信息。

- 先决条件
- 使用控制台创建集群
- 将集群添加到现有集群集合中

1.6.2.1. 先决条件

在 AWS 上创建集群前请查看以下先决条件：

- 您必须已部署 Red Hat Advanced Cluster Management for Kubernetes hub 集群。
- 您需要对 Red Hat Advanced Cluster Management for Kubernetes hub 集群的互联网访问，以便它在 Amazon Web Services 上创建 Kubernetes 集群。
- 您需要 AWS 凭证。如需更多信息，请参阅为 Amazon Web Services 创建凭证。
- 您在 AWS 中需要配置的域。有关如何配置域的说明，请参阅配置 AWS 帐户。
- 您需要具有 Amazon Web Services (AWS) 登录凭证，其中包括用户名、密码、访问密钥 ID 和 secret 访问密钥。请参阅了解和获取您的安全凭证。
- 您必须具有 OpenShift Container Platform 镜像 pull secret。请参阅使用镜像 pull secret。

注：如果更改了云供应商访问密钥，则必须手动更新置备的集群访问密钥。如需更多信息，请参阅已知问题，不支持置备的集群的自动 secret 更新。

1.6.2.2. 使用控制台创建集群

要从 Red Hat Advanced Cluster Management 控制台创建集群，请进入 **Infrastructure** > **Clusters**。在 *Clusters* 页面上，点 **Create cluster** 并完成控制台中的步骤。

注： 此过程用于创建集群。如果您有一个要导入的现有集群，请参阅将目标受管集群导入到 hub 集群 以了解这些步骤。

如果您需要创建凭证，请参阅为 Amazon Web Services 创建凭证。

集群的主机名用于集群的主机名。

重要： 当您创建集群时，Red Hat Advanced Cluster Management 控制器为集群及其资源创建一个命名空间。确保只在该命名空间中包含该集群实例的资源。销毁集群会删除命名空间和所有资源。

提示： 在控制台中输入信息时，选择 **YAML: On** 查看内容更新。

1.6.2.3. 将集群添加到现有集群集合中

如果要将集群添加到现有的集群集中，则需要在集群设置上具有正确的权限来添加它。如果在创建集群时没有 **cluster-admin** 权限，则必须选择一个具有 **clusterset-admin** 权限的集群集。如果您在指定的集群集中没有正确的权限，集群创建会失败。如果您没有任何集群集选项，请联络您的集群管理员，为集群集提供 **clusterset-admin** 权限。

每个受管集群都必须与受管集群集关联。如果您没有将受管集群分配给 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。

如果已有与您使用 AWS 帐户配置的所选凭证关联的基本 DNS 域，则该值会在字段中填充。您可以修改它的值来覆盖它。此名称用于集群的主机名。如需更多信息，请参阅配置 AWS 帐户。

发行镜像标识用于创建集群的 OpenShift Container Platform 镜像的版本。如果要使用的版本可用，您可以从镜像列表中选择镜像。如果您要使用的镜像不是标准镜像，您可以输入您要使用的镜像的 URL。有关发行镜像的更多信息，请参阅发行镜像。

节点池包括 control plane 池和 worker 池。control plane 节点共享集群活动的管理。该信息包括以下字段：

- 架构： 如果受管集群的架构类型与 hub 集群的架构不同，请为池中机器的说明集合架构输入一个值。有效值为 *amd64*, *ppc64le*, *s390x*, 和 *arm64*。
- Zones：指定您要运行 control plane 池的位置。您可以为更分布式的 control plane 节点选择区域中的多个区。距离更近的区域可能会提供更快的性能，但距离更远的区域可能更为分散。
- 实例类型：指定 control plane 节点的实例类型。您可以在实例创建后更改实例的类型和大小。
- 根存储：指定要为集群分配的 root 存储量。

您可以在 worker 池中创建零个或多个 worker 节点，以运行集群的容器工作负载。它们可以位于单个 worker 池中，也可以分布在多个 worker 池中。如果指定了零个 worker 节点，control plane 节点也充当 worker 节点。可选信息包括以下字段：

- Zones：指定您要运行 worker 池的位置。您可以为更加分散的节点组群选择区域中的多个区。距离更近的区域可能会提供更快的性能，但距离更远的区域可能更为分散。
- 实例类型：指定 worker 池的实例类型。您可以在实例创建后更改实例的类型和大小。
- 节点数：指定 worker 池的节点计数。定义 worker 池时需要此设置。
- 根存储：指定分配给 worker 池的根存储量。定义 worker 池时需要此设置。

集群需要网络详情，并使用 IPv6 需要多个网络。您可以通过点 **Add network** 来添加额外网络。

凭证中提供的代理信息会自动添加到代理字段中。您可以使用信息原样，覆盖这些信息，或者在要启用代理时添加信息。以下列表包含创建代理所需的信息：

- HTTP 代理 URL：指定应当用作 **HTTP** 流量的代理的 URL。
- HTTPS 代理 URL：指定用于 **HTTPS** 流量的安全代理 URL。如果没有提供值，则使用相同的值 **HTTP Proxy URL**，用于 **HTTP** 和 **HTTPS**。
- 无代理域：应当绕过代理的以逗号分隔的域列表。使用一个句点 (.) 开始的域名，包含该域中的所有子域。添加一个星号 * 以绕过所有目的地的代理。
- Additional trust bundle：指定访问镜像 registry 所需的证书文件内容。

当您在创建集群前查看信息并选择性地自定义它时，您可以选择 **YAML: On** 查看面板中的 **install-config.yaml** 文件内容。如果有更新，您可以使用自定义设置编辑 YAML 文件。

注： 您不必运行 **kubectl** 命令，它为导入集群提供集群详情。当您创建集群时，它由 Red Hat Advanced Cluster Management 管理自动配置。

如需了解更多与访问集群相关的信息，继续访问集群。

1.6.3. 在 Microsoft Azure 上创建集群

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台在 Microsoft Azure 或 Microsoft Azure Government 上部署 Red Hat OpenShift Container Platform 集群。

在创建集群时，创建过程使用 Hive 资源中的 OpenShift Container Platform 安装程序。如果在完成此步骤后集群创建有疑问，请参阅 OpenShift Container Platform 文档中的在 Azure 上安装 有关流程的更多信息。

- 先决条件
- 使用控制台创建集群
- 将集群添加到现有集群集合中

1.6.3.1. 先决条件

在 Azure 上创建集群前请查看以下先决条件：

- 您必须已部署 Red Hat Advanced Cluster Management for Kubernetes hub 集群。
- 可通过互联网访问 Red Hat Advanced Cluster Management for Kubernetes hub 集群，以便它在 Azure 或 Azure Government 上创建 Kubernetes 集群
- 您需要一个 Azure 凭证。如需更多信息，请参阅为 Microsoft Azure 创建凭证。
- 您需要在 Azure 或 Azure Government 中配置了域。有关如何配置域的说明，请参阅为 Azure 云服务配置自定义域名。
- 您需要 Azure 登录凭证，其中包括用户名和密码。请参阅 Microsoft Azure Portal。
- 您需要 Azure 服务主体，其中包括 **clientId**、**clientSecret** 和 **tenantId**。请参阅 azure.microsoft.com。
- 您需要 OpenShift Container Platform 镜像 pull secret。请参阅使用镜像 pull secret。

注：如果更改了云供应商访问密钥，则必须手动更新置备的集群访问密钥。如需更多信息，请参阅已知问题，不支持置备的集群的自动 secret 更新。

1.6.3.2. 使用控制台创建集群

要从 Red Hat Advanced Cluster Management for Kubernetes 控制台创建集群，请进入 **Infrastructure** > **Clusters**。在 *Clusters* 页面上，点 **Create cluster** 并完成控制台中的步骤。

注： 此过程用于创建集群。如果您有一个要导入的现有集群，请参阅将目标受管集群导入到 hub 集群以了解这些步骤。

如果您需要创建凭证，请参阅为 Microsoft Azure 创建凭证。

集群的主机名用于集群的主机名。

重要： 当您创建集群时，Red Hat Advanced Cluster Management 控制器为集群及其资源创建一个命名空间。确保只在该命名空间中包含该集群实例的资源。销毁集群会删除命名空间和所有资源。

提示： 在控制台中输入信息时，选择 **YAML: On** 查看内容更新。

1.6.3.3. 将集群添加到现有集群集合中

如果要将集群添加到现有的集群集中，则需要在集群设置上具有正确的权限来添加它。如果在创建集群时没有 **cluster-admin** 权限，则必须选择一个具有 **clusterset-admin** 权限的集群集。如果您在指定的集群集中没有正确的权限，集群创建会失败。如果您没有任何集群集选项，请联络您的集群管理员，为集群集提供 **clusterset-admin** 权限。

每个受管集群都必须与受管集群集关联。如果您没有将受管集群分配给 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。

如果已有与您为 Azure 帐户配置的所选凭证关联的基本 DNS 域，则该值会在那个字段中填充。您可以修改它的值来覆盖它。如需更多信息，请参阅为 Azure 云服务配置自定义域名。此名称用于集群的主机名。

发行镜像标识用于创建集群的 OpenShift Container Platform 镜像的版本。如果要使用的版本可用，您可以从镜像列表中选择镜像。如果您要使用的镜像不是标准镜像，您可以输入您要使用的镜像的 URL。有关发行镜像的更多信息，请参阅发行镜像。

Node 池包括 control plane 池和 worker 池。control plane 节点共享集群活动的管理。该信息包括以下可选字段：

- region：指定一个您要运行节点池的区域。您可以为更分布式的 control plane 节点选择区域中的多个区。距离更近的区域可能会提供更快的性能，但距离更远的区域可能更为分散。
- 架构： 如果受管集群的架构类型与 hub 集群的架构不同，请为池中机器的说明集合架构输入一个值。有效值为 *amd64*, *ppc64le*, *s390x*, 和 *arm64*。
- control plane 池的实例类型和 Root 存储分配（必需）。您可以在实例创建后更改实例的类型和大小。

您可以在 worker 池中创建一个或多个 worker 节点，以运行集群的容器工作负载。它们可以位于单个 worker 池中，也可以分布在多个 worker 池中。如果指定了零个 worker 节点，control plane 节点也充当 worker 节点。该信息包括以下字段：

- Zones：指定您要运行 worker 池的位置。您可以为更加分散的节点组群选择区域中的多个区。距离更近的区域可能会提供更快的性能，但距离更远的区域可能更为分散。
- 实例类型：您可以在实例创建后更改实例的类型和大小。

您可以通过点 **Add network** 来添加额外网络。如果您使用的是 IPv6 地址，您必须有多个网络。

凭证中提供的代理信息会自动添加到代理字段中。您可以使用信息原样，覆盖这些信息，或者在要启用代理时添加信息。以下列表包含创建代理所需的信息：

- HTTP 代理 URL：用作 **HTTP** 流量的代理的 URL。
- HTTPS 代理 URL：用于 **HTTPS** 流量的安全代理 URL。如果没有提供值，则使用相同的值 **HTTP Proxy URL**，用于 **HTTP** 和 **HTTPS**。
- 无代理域：应当绕过代理的以逗号分隔的域列表。使用一个句点 (.) 开始的域名，包含该域中的所有子域。添加一个星号 * 以绕过所有目的地的代理。
- Additional trust bundle：访问镜像 registry 所需的证书文件内容。

当您在创建集群前查看信息并选择性地自定义它时，您可以点 **YAML** 切换 **On** 查看面板中的 **install-config.yaml** 文件内容。如果有更新，您可以使用自定义设置编辑 YAML 文件。

注： 您不必运行 **kubectl** 命令，它为导入集群提供集群详情。当您创建集群时，它由 Red Hat Advanced Cluster Management 管理自动配置。

如需了解更多与访问集群相关的信息，继续访问集群。

1.6.4. 在 Google Cloud Platform 上创建集群

按照步骤在 Google Cloud Platform (GCP) 上创建 Red Hat OpenShift Container Platform 集群有关 GCP 的更多信息，请参阅 Google Cloud Platform。

在创建集群时，创建过程使用 Hive 资源中的 OpenShift Container Platform 安装程序。如果您在完成此步骤后集群创建有疑问，请参阅 OpenShift Container Platform 文档中的在 GCP 上安装以了解有关此过程的更多信息。

- 先决条件
- 使用控制台创建集群
- 将集群添加到现有集群集合中

1.6.4.1. 先决条件

在 GCP 上创建集群前请查看以下先决条件：

- 您必须已部署 Red Hat Advanced Cluster Management for Kubernetes hub 集群。
- 您需要对 Red Hat Advanced Cluster Management for Kubernetes hub 集群的互联网访问，以便它在 GCP 上创建 Kubernetes 集群。
- 您必须具有 GCP 凭证。如需更多信息，请参阅为 Google Cloud Platform 创建凭证。
- 您必须在 GCP 中配置了域。有关如何配置域的说明，请参阅设置自定义域。
- 您需要 GCP 登录凭证，其中包括用户名和密码。
- 您必须具有 OpenShift Container Platform 镜像 pull secret。请参阅使用镜像 pull secret。

注：如果更改了云供应商访问密钥，则必须手动更新置备的集群访问密钥。如需更多信息，请参阅已知问题，不支持置备的集群的自动 secret 更新。

1.6.4.2. 使用控制台创建集群

要从 Red Hat Advanced Cluster Management for Kubernetes 控制台创建集群，请进入 **Infrastructure** > **Clusters**。在 *Clusters* 页面上，点 **Create cluster** 并完成控制台中的步骤。

注： 此过程用于创建集群。如果您有一个要导入的现有集群，请参阅将目标受管集群导入到 hub 集群以了解这些步骤。

如果您需要创建凭证，请参阅为 Google Cloud Platform 创建凭证。如需更多信息，请参阅为 Google Cloud Platform 创建凭证。

集群名称用于集群的主机名。在命名 GCP 集群时有一些限制。这些限制包括，名称不要以 **goog** 开始；名称的任何部分都不要包含与 **google** 类似的内容。如需了解完整的限制列表，请参阅 Bucket 命名指南。

重要： 当您创建集群时，Red Hat Advanced Cluster Management 控制器为集群及其资源创建一个命名空间。确保只在该命名空间中包含该集群实例的资源。销毁集群会删除命名空间和所有资源。

提示： 在控制台中输入信息时，选择 **YAML: On** 查看内容更新。

1.6.4.3. 将集群添加到现有集群集合中

如果要将集群添加到现有的集群集中，则需要在集群设置上具有正确的权限来添加它。如果在创建集群时没有 **cluster-admin** 权限，则必须选择一个具有 **clusterset-admin** 权限的集群集。如果您在指定的集群集中没有正确的权限，集群创建会失败。如果您没有任何集群集选项，请联络您的集群管理员，为集群集提供 **clusterset-admin** 权限。

每个受管集群都必须与受管集群集关联。如果您没有将受管集群分配给 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。

如果已有与 GCP 帐户所选凭证关联的基本 DNS 域，则该值会在字段中填充。您可以修改它的值来覆盖它。如需更多信息，请参阅设置自定义域。此名称用于集群的主机名。

发行镜像标识用于创建集群的 OpenShift Container Platform 镜像的版本。如果要使用的版本可用，您可以从镜像列表中选择镜像。如果您要使用的镜像不是标准镜像，您可以输入您要使用的镜像的 URL。有关发行镜像的更多信息，请参阅发行镜像。

Node 池包括 control plane 池和 worker 池。control plane 节点共享集群活动的管理。该信息包括以下字段：

- Region：指定您要运行 control plane 池的区域。距离更近的区域可能会提供更快的性能，但距离更远的区域可能更为分散。
- 架构： 如果受管集群的架构类型与 hub 集群的架构不同，请为池中机器的说明集合架构输入一个值。有效值为 *amd64*, *ppc64le*, *s390x*, 和 *arm64*。
- 实例类型：您可以在实例创建后更改实例的类型和大小。

您可以在 worker 池中创建一个或多个 worker 节点，以运行集群的容器工作负载。它们可以位于单个 worker 池中，也可以分布在多个 worker 池中。如果指定了零个 worker 节点，control plane 节点也充当 worker 节点。该信息包括以下字段：

- 实例类型：您可以在实例创建后更改实例的类型和大小。
- 节点数：定义 worker 池时需要此设置。

网络详细信息是必需的，需要使用 IPv6 地址的多个网络。您可以通过点 **Add network** 来添加额外网络。

凭证中提供的代理信息会自动添加到代理字段中。您可以使用信息原样，覆盖这些信息，或者在要启用代理时添加信息。以下列表包含创建代理所需的信息：

- HTTP 代理 URL：用作 **HTTP** 流量的代理的 URL。
- HTTPS 代理 URL：用于 **HTTPS** 流量的安全代理 URL。如果没有提供值，则使用相同的值 **HTTP Proxy URL**，用于 **HTTP** 和 **HTTPS**。
- 无代理域：应当绕过代理的以逗号分隔的域列表。使用一个句点 (.) 开始的域名，包含该域中的所有子域。添加一个星号 * 以绕过所有目的地的代理。
- Additional trust bundle：访问镜像 registry 所需的证书文件内容。

当您在创建集群前查看信息并选择性地自定义它时，您可以选择 **YAML: On** 查看面板中的 **install-config.yaml** 文件内容。如果有更新，您可以使用自定义设置编辑 YAML 文件。

注： 您不必运行 **kubectl** 命令，它为导入集群提供集群详情。当您创建集群时，它由 Red Hat Advanced Cluster Management 管理自动配置。

如需了解更多与访问集群相关的信息，继续访问集群。

1.6.5. 在 VMware vSphere 上创建集群

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台在 VMware vSphere 上部署 Red Hat OpenShift Container Platform 集群。

在创建集群时，创建过程使用 Hive 资源中的 OpenShift Container Platform 安装程序。如果在完成此步骤后集群创建有疑问，请参阅 OpenShift Container Platform 文档中的在 vSphere 上安装 有关流程的更多信息。

- 先决条件
- 使用控制台创建集群
- 将集群添加到现有集群集合中

1.6.5.1. 先决条件

在 vSphere 上创建集群前请查看以下先决条件：

- 您必须在 OpenShift Container Platform 版本 4.6 或更高版本上部署了 Red Hat Advanced Cluster Management hub 集群。
- 您需要对 Red Hat Advanced Cluster Management hub 集群的互联网访问，以便它在 vSphere 上创建 Kubernetes 集群。
- 您需要 vSphere 凭证。如需更多信息，请参阅为 VMware vSphere 创建凭证。
- 您需要 OpenShift Container Platform 镜像 pull secret。请参阅使用镜像 pull secret。
- 您必须具有要部署的 VMware 实例的以下信息：
  - API 和 Ingress 实例所需的静态 IP 地址
  - 以下的 DNS 记录：
    - **api.<cluster_name>.<base_domain>**，它必须指向静态 API VIP
    - ***.apps.<cluster_name>.<base_domain>**，它必须指向 Ingress VIP 的静态 IP 地址

备注：当使用 VMware vSphere 或 Red Hat OpenStack Platform 供应商和断开连接的安装创建集群时，如果需要一个证书才能访问镜像 registry，您必须在 *为断开连接的安装配置* 部分的 *附加信任捆绑包* 字段中输入它。您不能在集群创建控制台编辑器中输入它们。

1.6.5.2. 使用控制台创建集群

要从 Red Hat Advanced Cluster Management for Kubernetes 控制台创建集群，请进入 **Infrastructure** > **Clusters**。在 *Clusters* 页面上，点 **Create cluster** 并完成控制台中的步骤。

注：此过程用于创建集群。如果您有一个要导入的现有集群，请参阅将目标受管集群导入到 hub 集群 以了解这些步骤。

如果您需要创建凭证，请参阅为 VMware vSphere 创建凭证，以了解有关创建凭证的更多信息。

集群名称用于集群的主机名。

重要：当您创建集群时，Red Hat Advanced Cluster Management 控制器为集群及其资源创建一个命名空间。确保只在该命名空间中包含该集群实例的资源。销毁集群会删除命名空间和所有资源。

提示： 在控制台中输入信息时，选择 **YAML: On** 查看内容更新。

1.6.5.3. 将集群添加到现有集群集合中

如果要将集群添加到现有的集群集中，则需要在集群设置上具有正确的权限来添加它。如果在创建集群时没有 **cluster-admin** 权限，则必须选择一个具有 **clusterset-admin** 权限的集群集。如果您在指定的集群集中没有正确的权限，集群创建会失败。如果您没有任何集群集选项，请联络您的集群管理员，为集群集提供 **clusterset-admin** 权限。

每个受管集群都必须与受管集群集关联。如果您没有将受管集群分配给 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。

如果已有与您为 vSphere 帐户配置的所选凭证关联的基域，则该值会在字段中填充。您可以修改它的值来覆盖它。如需更多信息，请参阅使用自定义在 vSphere 上安装集群。这个值必须与创建 prerequisites 部分中列出的 DNS 记录的名称匹配。此名称用于集群的主机名。

发行镜像标识用于创建集群的 OpenShift Container Platform 镜像的版本。如果要使用的版本可用，您可以从镜像列表中选择镜像。如果您要使用的镜像不是标准镜像，您可以输入您要使用的镜像的 URL。有关发行镜像的更多信息，请参阅发行镜像

注： 仅支持 OpenShift Container Platform 版本 4.5.x 或更高版本的发行镜像。

节点池包括 control plane 池和 worker 池。control plane 节点共享集群活动的管理。该信息包括 *Architecture* 字段。查看以下字段描述：

- 架构：如果受管集群的架构类型与 hub 集群的架构不同，请为池中机器的说明集合架构输入一个值。有效值为 *amd64*, *ppc64le*, *s390x*, 和 *arm64*。

您可以在 worker 池中创建一个或多个 worker 节点，以运行集群的容器工作负载。它们可以位于单个 worker 池中，也可以分布在多个 worker 池中。如果指定了零个 worker 节点，control plane 节点也充当 worker 节点。信息包括 *每个插槽的内核数*, *CPU*, *Memory_min MB*, *_Disk size*（GiB 为单位）和 *节点数*。

需要网络信息。使用 IPv6 需要多个网络。一些所需网络信息包括以下字段：

- vSphere 网络名：指定 VMware vSphere 网络名称。
- API VIP：指定用于内部 API 通信的 IP 地址。
  注： 这个值必须与您用来创建 prerequisites 部分中列出的 DNS 记录的名称匹配。如果没有提供，DNS 必须预先配置，以便 **api.** 可以正确解析。
- Ingress VIP：指定用于入口流量的 IP 地址。
  注： 这个值必须与您用来创建 prerequisites 部分中列出的 DNS 记录的名称匹配。如果没有提供，则必须预先配置 DNS，以便 **test.apps** 可以被正确解析。

您可以通过点 **Add network** 来添加额外网络。如果您使用的是 IPv6 地址，您必须有多个网络。

凭证中提供的代理信息会自动添加到代理字段中。您可以使用信息原样，覆盖这些信息，或者在要启用代理时添加信息。以下列表包含创建代理所需的信息：

- HTTP 代理 URL：指定应当用作 **HTTP** 流量的代理的 URL。
- HTTPS 代理 URL：指定用于 **HTTPS** 流量的安全代理 URL。如果没有提供值，则使用相同的值 **HTTP Proxy URL**，用于 **HTTP** 和 **HTTPS**。
- 没有代理域：提供以逗号分隔的域列表，这些域应绕过代理。使用一个句点 (.) 开始的域名，包含该域中的所有子域。添加一个星号 * 以绕过所有目的地的代理。

- Additional trust bundle：指定访问镜像 registry 所需的证书文件内容。

您可以点 **Disconnected** 安装来定义断开连接的安装镜像。有关限制的详情，请参阅 Disconnected installation settings for cluster creation cannot be entered or are ignored if entered。

您可以点击 **Add Automation template** 来创建模板。

当您在创建集群前查看信息并选择性地自定义它时，您可以点 **YAML** 切换 **On** 查看面板中的 **install-config.yaml** 文件内容。如果有更新，您可以使用自定义设置编辑 YAML 文件。

注： 您不必运行 **kubectl** 命令，它为导入集群提供集群详情。当您创建集群时，它由 Red Hat Advanced Cluster Management 管理自动配置。

如需了解更多与访问集群相关的信息，继续访问集群。

1.6.6. 在 Red Hat OpenStack Platform 上创建集群

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台在 Red Hat OpenStack Platform 上部署 Red Hat OpenShift Container Platform 集群。

在创建集群时，创建过程使用 Hive 资源中的 OpenShift Container Platform 安装程序。如果在完成此步骤后集群创建问题，请参阅 OpenShift Container Platform 文档中的在 OpenStack 上安装 OpenStack 以了解有关流程的更多信息。

- 先决条件
- 使用控制台创建集群
- 将集群添加到现有集群集合中

1.6.6.1. 先决条件

在 Red Hat OpenStack Platform 上创建集群前请查看以下先决条件：

- 您必须在 OpenShift Container Platform 版本 4.6 或更高版本上部署了 Red Hat Advanced Cluster Management hub 集群。
- 您需要可通过互联网访问 Red Hat Advanced Cluster Management hub 集群，以便它在 Red Hat OpenStack Platform 上创建 Kubernetes 集群。
- 您必须具有 Red Hat OpenStack Platform 凭证。如需更多信息，请参阅为 Red Hat OpenStack Platform 创建凭证。
- 您需要 OpenShift Container Platform 镜像 pull secret。请参阅使用镜像 pull secret。
- 您要部署的 Red Hat OpenStack Platform 实例需要以下信息：
  - control plane 和 worker 实例的 flavor 名称，如 **m1.xlarge**
  - 外部网络的网络名称，以提供浮动 IP 地址
  - API 和入口实例所需的浮动 IP 地址
  - 以下的 DNS 记录：
    - **api.<cluster_name>.<base_domain>**，它必须指向 API 的浮动 IP 地址

- ***.apps.<cluster_name>.<base_domain>**，它必须指向 ingress 的浮动 IP 地址

1.6.6.2. 使用控制台创建集群

要从 Red Hat Advanced Cluster Management for Kubernetes 控制台创建集群，请进入 **Infrastructure** > **Clusters**。在 *Clusters* 页面上，点 **Create cluster** 并完成控制台中的步骤。

注： 此过程用于创建集群。如果您有一个要导入的现有集群，请参阅将目标受管集群导入到 hub 集群以了解这些步骤。

如果需要创建凭证，请参阅为 Red Hat OpenStack Platform 创建凭证。

集群的主机名用于集群的主机名。名称必须包含少于 15 个字符。这个值必须与创建凭证先决条件中列出的 DNS 记录的名称匹配。

重要： 当您创建集群时，Red Hat Advanced Cluster Management 控制器为集群及其资源创建一个命名空间。确保只在该命名空间中包含该集群实例的资源。销毁集群会删除命名空间和所有资源。

提示： 在控制台中输入信息时，选择 **YAML: On** 查看内容更新。

1.6.6.3. 将集群添加到现有集群集合中

如果要将集群添加到现有的集群集中，则需要在集群设置上具有正确的权限来添加它。如果在创建集群时没有 **cluster-admin** 权限，则必须选择一个具有 **clusterset-admin** 权限的集群集。如果您在指定的集群集中没有正确的权限，集群创建会失败。如果您没有任何集群集选项，请联络您的集群管理员，为集群集提供 **clusterset-admin** 权限。

每个受管集群都必须与受管集群集关联。如果您没有将受管集群分配给 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。

如果已有与您为 Red Hat OpenStack Platform 帐户配置的所选凭证关联的基本 DNS 域，则该值会在字段中填充。您可以修改它的值来覆盖它。如需更多信息，请参阅 Red Hat OpenStack Platform 文档中的管理域。此名称用于集群的主机名。

发行镜像标识用于创建集群的 OpenShift Container Platform 镜像的版本。如果要使用的版本可用，您可以从镜像列表中选择镜像。如果您要使用的镜像不是标准镜像，您可以输入您要使用的镜像的 URL。有关发行镜像的更多信息，请参阅发行镜像。仅支持 OpenShift Container Platform 版本 4.6.x 或更高版本的发行镜像。

节点池包括 control plane 池和 worker 池。control plane 节点共享集群活动的管理。该信息包括以下字段：

- **可选架构**：如果受管集群的架构类型与 hub 集群的架构不同，请为池中机器的说明集合架构输入一个值。有效值为 *amd64*, *ppc64le*, *s390x*, 和 *arm64*。
- control plane 池的实例类型：您可以在实例创建后更改实例的类型和大小。

您可以在 worker 池中创建一个或多个 worker 节点，以运行集群的容器工作负载。它们可以位于单个 worker 池中，也可以分布在多个 worker 池中。如果指定了零个 worker 节点，control plane 节点也充当 worker 节点。该信息包括以下字段：

- 实例类型：您可以在实例创建后更改实例的类型和大小。
- 节点数：指定 worker 池的节点数。定义 worker 池时需要此设置。

集群需要网络详情。您必须为 IPv4 网络提供一个或多个网络的值。对于 IPv6 网络，您必须定义多个网络。

您可以通过点 **Add network** 来添加额外网络。如果您使用的是 IPv6 地址，您必须有多个网络。

凭证中提供的代理信息会自动添加到代理字段中。您可以使用信息原样，覆盖这些信息，或者在要启用代理时添加信息。以下列表包含创建代理所需的信息：

- HTTP 代理 URL：指定应当用作 **HTTP** 流量的代理的 URL。
- HTTPS 代理 URL：用于 **HTTPS** 流量的安全代理 URL。如果没有提供值，则使用相同的值 **HTTP Proxy URL**，用于 **HTTP** 和 **HTTPS**。
- 无代理域：定义应绕过代理的域列表。使用一个句点 (.) 开始的域名，包含该域中的所有子域。添加一个星号 * 以绕过所有目的地的代理。
- Additional trust bundle：指定访问镜像 registry 所需的证书文件内容。

您可以点 **Disconnected** 安装来定义断开连接的安装镜像。有关限制的详情，请参阅 Disconnected installation settings for cluster creation cannot be entered or are ignored if entered。

当您在创建集群前查看信息并选择性地自定义它时，您可以点 **YAML** 切换 **On** 查看面板中的 **install-config.yaml** 文件内容。如果有更新，您可以使用自定义设置编辑 YAML 文件。

注： 您不必运行 **kubectl** 命令，它为导入集群提供集群详情。当您创建集群时，它由 Red Hat Advanced Cluster Management 管理自动配置。

如需了解更多与访问集群相关的信息，继续访问集群。

1.6.7. 在 Red Hat Virtualization 上创建集群

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台在 Red Hat Virtualization 上创建一个 Red Hat OpenShift Container Platform 集群。

在创建集群时，创建过程使用 Hive 资源中的 OpenShift Container Platform 安装程序。如果在完成此步骤后集群创建有疑问，请参阅 OpenShift Container Platform 文档中的在 RHV 上安装 有关流程的更多信息。

- 前提条件
- 使用控制台创建集群
- 将集群添加到现有集群集合中

1.6.7.1. 先决条件

在 Red Hat Virtualization 上创建集群前请查看以下先决条件：

- 您必须已部署 Red Hat Advanced Cluster Management for Kubernetes hub 集群。
- 您需要对 Red Hat Advanced Cluster Management for Kubernetes hub 集群的互联网访问，以便它在 Red Hat Virtualization 上创建 Kubernetes 集群。
- 您需要 Red Hat Virtualization 凭证。如需更多信息，请参阅为 Red Hat Virtualization 创建凭证。
- 您需要 oVirt Engine 虚拟机配置的域和虚拟机代理。有关如何配置域的说明，请参阅 Red Hat OpenShift Container Platform 文档中的在 RHV 上安装。
- 您必须具有 Red Hat Virtualization 登录凭证，其中包括您的红帽客户门户网站用户名和密码。

- 您需要 OpenShift Container Platform 镜像 pull secret。您可以从以下位置下载 pull secret： Pull secret。如需有关 pull secret 的更多信息，请参阅使用镜像 pull secret。

注：如果更改了云供应商访问密钥，则必须手动更新置备的集群访问密钥。如需更多信息，请参阅已知问题，不支持置备的集群的自动 secret 更新。

1.6.7.2. 使用控制台创建集群

要从 Red Hat Advanced Cluster Management for Kubernetes 控制台创建集群，请进入 **Infrastructure** > **Clusters**。在 *Clusters* 页面上，点 **Create cluster** 并完成控制台中的步骤。

注： 此过程用于创建集群。如果您有一个要导入的现有集群，请参阅将目标受管集群导入到 hub 集群以了解这些步骤。

如果您需要创建凭证，请参阅为 Red Hat Virtualization 创建凭证 。

集群名称用于集群的主机名。

重要： 当您创建集群时，Red Hat Advanced Cluster Management 控制器为集群及其资源创建一个命名空间。确保只在该命名空间中包含该集群实例的资源。销毁集群会删除命名空间和所有资源。

提示： 在控制台中输入信息时，选择 **YAML: On** 查看内容更新。

1.6.7.3. 将集群添加到现有集群集合中

如果要将集群添加到现有的集群集中，则需要在集群设置上具有正确的权限来添加它。如果在创建集群时没有 **cluster-admin** 权限，则必须选择一个具有 **clusterset-admin** 权限的集群集。如果您在指定的集群集中没有正确的权限，集群创建会失败。如果您没有任何集群集选项，请联络您的集群管理员，为集群集提供 **clusterset-admin** 权限。

每个受管集群都必须与受管集群集关联。如果您没有将受管集群分配给 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。

如果已有与您为 Red Hat Virtualization 帐户配置的所选凭证关联的基本 DNS 域，则该值会在那个字段中填充。您可以覆盖该值来更改它。

发行镜像标识用于创建集群的 OpenShift Container Platform 镜像的版本。如果要使用的版本可用，您可以从镜像列表中选择镜像。如果您要使用的镜像不是标准镜像，您可以输入您要使用的镜像的 URL。有关发行镜像的更多信息，请参阅发行镜像。

节点池的信息包括 control plane 池的内核、插槽、内存和磁盘大小。这三个 control plane 节点共享集群活动的管理。该信息包括 *Architecture* 字段。查看以下字段描述：

- 架构： 如果受管集群的架构类型与 hub 集群的架构不同，请为池中机器的说明集合架构输入一个值。有效值为 *amd64*, *ppc64le*, *s390x*, 和 *arm64*。

worker 池信息需要池名称、内核数量、内存分配、磁盘大小分配和 worker 池的节点数。worker 池中的 worker 节点可以在单个 worker 池中，也可以分布到多个 worker 池中。

您预先配置的 oVirt 环境中需要以下网络详情。

- ovirt 网络名称
- API VIP：指定用于内部 API 通信的 IP 地址。
  注： 这个值必须与您用来创建 prerequisites 部分中列出的 DNS 记录的名称匹配。如果没有提供，DNS 必须预先配置，以便 **api.** 可以正确解析。

- Ingress VIP：指定用于入口流量的 IP 地址。
  注： 这个值必须与您用来创建 prerequisites 部分中列出的 DNS 记录的名称匹配。如果没有提供，则必须预先配置 DNS，以便 **test.apps** 可以被正确解析。

- Network type：默认值为 **OpenShiftSDN**。OVNKubernetes 是使用 IPv6 的必要设置。

- Cluster network CIDR：此 IP 地址可用于 pod IP 地址的数量和列表。这个块不能与另一个网络块重叠。默认值为 **10.128.0.0/14**。

- 网络主机前缀：为每个节点设置子网前缀长度。默认值为 **23**。

- Service network CIDR：为服务提供 IP 地址块。这个块不能与另一个网络块重叠。默认值为 **172.30.0.0/16**。

- Machine CIDR：提供 OpenShift Container Platform 主机使用的 IP 地址块。这个块不能与另一个网络块重叠。默认值为 **10.0.0.0/16**。
  您可以通过点 **Add network** 来添加额外网络。如果您使用的是 IPv6 地址，您必须有多个网络。

凭证中提供的代理信息会自动添加到代理字段中。您可以使用信息原样，覆盖这些信息，或者在要启用代理时添加信息。以下列表包含创建代理所需的信息：

- HTTP 代理 URL：指定应当用作 **HTTP** 流量的代理的 URL。

- HTTPS 代理 URL：指定用于 **HTTPS** 流量的安全代理 URL。如果没有提供值，则使用相同的值 **HTTP Proxy URL**，用于 **HTTP** 和 **HTTPS**。

- 没有代理域：提供以逗号分隔的域列表，这些域应绕过代理。使用一个句点 (.) 开始的域名，包含该域中的所有子域。添加一个星号 * 以绕过所有目的地的代理。

- Additional trust bundle：指定访问镜像 registry 所需的证书文件内容。

当您在创建集群前查看信息并选择性地自定义它时，您可以点 **YAML** 切换 **On** 查看面板中的 **install-config.yaml** 文件内容。如果有更新，您可以使用自定义设置编辑 YAML 文件。

注： 您不必运行 **kubectl** 命令，它为导入集群提供集群详情。当您创建集群时，它由 Red Hat Advanced Cluster Management 管理自动配置。

如需了解更多与访问集群相关的信息，继续访问集群。

1.6.8. 在裸机上创建集群

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台在裸机环境中创建一个 Red Hat OpenShift Container Platform 集群。

弃用通知： 使用裸机资产创建裸机集群的步骤已弃用。裸机资产将在以后的版本中删除。

在创建集群时，请注意，创建过程会将 OpenShift Container Platform 安装程序与 Hive 资源一起使用。如果您在完成此步骤后集群创建有疑问，请参阅在 OpenShift Container Platform 文档中的在裸机上安装来获得更详细的信息。

- 先决条件

- 创建裸机集群

1.6.8.1. 先决条件

在裸机环境中创建集群前请查看以下先决条件：

- 您必须在 OpenShift Container Platform 版本 4.6 或更高版本上部署了 Red Hat Advanced Cluster Management for Kubernetes hub 集群。
- 您需要对 Red Hat Advanced Cluster Management for Kubernetes hub 集群（连接）或连接到连接到互联网（断开连接）的内部或镜像 registry 的连接，以检索创建集群所需的镜像。
- 您需要一个运行 bootstrap 虚拟机的临时外部 KVM 主机，用于创建 Hive 集群。如需更多信息，请参阅准备置备主机。
- 部署的 Red Hat Advanced Cluster Management for Kubernetes hub 集群必须能够路由到 provisioning 网络。
- 您需要裸机服务器登录凭证，其中包括上一项目中 bootstrap 虚拟机的 libvirt URI、SSH 私钥以及 SSH 已知主机列表。如需更多信息，请参阅为 OpenShift 安装设置环境。
- 您需要配置一个配置的裸机凭证。如需更多信息，请参阅为裸机创建凭证。
- 您必须有裸机环境的登录凭证，其中包括用户名、密码和基板管理控制器地址。
- 如果您要启用证书验证，则需要配置一个裸机资产。如需更多信息，请参阅创建和修改裸机资产。
- 您需要 OpenShift Container Platform 镜像 pull secret。如需更多信息，请参阅使用镜像 pull secret。
  备注：
  - 裸机资产、受管裸机集群及其相关 secret 必须位于同一命名空间中。
  - 如果更改了云供应商访问密钥，则必须手动更新置备的集群访问密钥。如需更多信息，请参阅已知问题，不支持置备的集群的自动 secret 更新。
  - 当使用裸机供应商并以断开连接安装的方式创建集群时，您必须将所有设置保存在 *Configuration for disconnected installation* 部分的凭证中。您不能在集群创建控制台编辑器中输入它们。

1.6.8.2. 创建裸机集群

要从 Red Hat Advanced Cluster Management for Kubernetes 控制台创建集群，请进入 **Infrastructure** > **Clusters**。在 *Clusters* 页面上，点 **Create cluster** 并完成控制台中的步骤。

注： 此过程用于创建集群。如果您有一个要导入的现有集群，请参阅将目标受管集群导入到 hub 集群以了解这些步骤。

如果您需要创建凭证，请参阅为裸机创建凭证，以了解有关创建凭证的更多信息。

对于裸机集群，集群名称不能是一个任意名称。它与集群 URL 相关联。确保使用的集群名称与您的 DNS 和网络设置一致。

重要： 当您创建集群时，Red Hat Advanced Cluster Management 控制器为集群及其资源创建一个命名空间。确保只在该命名空间中包含该集群实例的资源。销毁集群会删除命名空间和所有资源。

提示： 在控制台中输入信息时，选择 **YAML: On** 查看内容更新。

如果要将集群添加到现有的集群集中，则需要在集群设置上具有正确的权限来添加它。如果在创建集群时没有 **cluster-admin** 权限，则必须选择一个具有 **clusterset-admin** 权限的集群集。如果您在指定的集群集中没有正确的权限，集群创建会失败。如果您没有任何集群集选项，请联络您的集群管理员，为集群集提供 **clusterset-admin** 权限。

每个受管集群都必须与受管集群集关联。如果您没有将受管集群分配给 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。

此基础域用于创建到 OpenShift Container Platform 集群组件的路由。它在集群供应商的 DNS 中被配置为授权起始(SOA)记录。此名称用于集群的主机名。

如果已有与您为裸机供应商帐户配置的所选凭证关联的基本 DNS 域，则该值会在那个字段中填充。您可以通过覆盖它来更改值，但无法在创建集群后更改名称。如需更多信息，请参阅 OpenShift Container Platform 文档中的在裸机上安装。

发行镜像标识用于创建集群的 OpenShift Container Platform 镜像的版本。如果您要使用的镜像不是标准镜像，您可以输入您要使用的镜像的 URL。有关发行镜像的更多信息，请参阅发行镜像。

主机列表从现有的裸机资产生成，并与您的凭证相关联。确保您在裸机主机上运行最新的固件，或者置备可能会失败。您必须最少选择三个与虚拟机监控程序在同一网桥网络上的裸机资产。如果您没有创建任何裸机资产，您可以在继续创建或导入它们前创建或导入它们，方法是选择 **Import asset**。有关创建裸机资产的更多信息，请参阅创建和修改裸机资产。或者，您可以选择 **Disable certificate verify** 来忽略要求。

下表显示了网络选项及其描述：

参数	描述	必需/可选
置备网络 CIDR	用于置备的网络的 CIDR。示例格式为： 172.30.0.0/16。	必需
置备网络接口	连接到置备网络的 control plane 节点上的网络接口名称。	必需
置备网络桥接	附加到置备网络中的桥接名称。	必需
外部网络桥接	附加到外部网络的管理程序桥接名称。	必需
API VIP	用于内部 API 通信的虚拟 IP。DNS 必须预先配置为有 A/AAAA 或 CNAME 记录，以便 **api.<cluster_name>.<Base DNS domain>** 路径可以正确解析。	必需
Ingress VIP	用于入口流量的虚拟 IP。DNS 必须预先配置为有 A/AAAA 或 CNAME 记录，以便 ***.apps.<cluster_name>.<Base DNS domain>** 路径可以正确解析。	选填

参数	描述	必需/可选
Network type	要部署的 Pod 网络供应商插件。OpenShiftSDN 插件是 OpenShift Container Platform 4.3 中唯一支持的插件。OVNKubernetes 插件在 OpenShift Container Platform 版本 4.3、4.4 和 4.5 中仅以技术预览提供。它通常包括在 OpenShift Container Platform 版本 4.6 及更高版本中。OVNKubernetes 必须和 IPv6 一起使用。默认值为 **OpenShiftSDN**。	必需
Cluster network CIDR	从其中分配 Pod IP 地址的 IP 地址块。OpenShiftSDN 网络插件支持多个集群网络。多个集群网络的地址块不得互相重叠。请选择足够大的地址池，以适配预期的工作负载。默认值为 10.128.0.0/14。	必需
Network host prefix	分配给每个单独节点的子网前缀长度。例如，如果 hostPrefix 设为 23，则每个节点从给定的 CIDR 中分配一个 /23 子网，允许 510（2^(32-23)-2）个 pod IP 地址。默认值为 23。	必需
Service network CIDR	服务的 IP 地址块。OpenShiftSDN 只允许一个 serviceNetwork 块。该地址不能与任何其他网络块相重叠。默认值为 172.30.0.0/16。	必需
Machine CIDR	OpenShift Container Platform 主机使用的 IP 地址块。该地址块不得与任何其他网络块重叠。默认值为 10.0.0.0/16。	必需

如果您使用的是 IPv6 地址，您必须有多个网络。

凭证中提供的代理信息会自动添加到代理字段中。您可以使用信息原样，覆盖这些信息，或者在要启用代理时添加信息。以下列表包含创建代理所需的信息：

- HTTP 代理 URL：用作 **HTTP** 流量的代理的 URL。
- HTTPS 代理 URL：用于 **HTTPS** 流量的安全代理 URL。如果没有提供值，则使用相同的值 **HTTP Proxy URL**，用于 **HTTP** 和 **HTTPS**。
- 无代理域：应当绕过代理的以逗号分隔的域列表。使用一个句点 (.) 开始的域名，包含该域中的所有子域。添加一个星号 * 以绕过所有目的地的代理。
- Additional trust bundle：访问镜像 registry 所需的证书文件内容。

当您在创建集群前查看信息并选择性地自定义它时，您可以选择 **YAML: On** 查看面板中的 **install-config.yaml** 文件内容。如果有更新，您可以使用自定义设置编辑 YAML 文件。

注： 您不必运行 **kubectl** 命令，它为导入集群提供集群详情。创建集群时，它由 Red Hat Advanced Cluster Management 管理自动配置。

如需了解更多与访问集群相关的信息，继续访问集群。

1.6.9. 在内部环境中创建集群

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台创建内部 Red Hat OpenShift Container Platform 集群。建议使用这个进程，而不是弃用的裸机进程。

最佳实践：使用此流程创建单节点 OpenShift(SNO)集群。您可以在 VMware vSphere、Red Hat OpenStack、Red Hat Virtualization Platform 和裸机环境中创建单节点 OpenShift 集群。没有与安装集群的平台集成，因为平台值设置为 **platform=none**。单节点 OpenShift 集群仅包含一个节点，用于托管 control plane 服务和用户工作负载。当您要最小化集群资源占用空间时，此配置很有用。

您还可以使用零接触置备功能测试在边缘资源中置备多个单节点 OpenShift 集群的步骤，它在 Red Hat OpenShift Container Platform 中是一个技术预览功能。有关该流程的更多信息，请参阅 OpenShift Container Platform 文档中的在断开连接的环境中部署分布式单元。

- 前提条件
- 使用控制台创建集群

1.6.9.1. 先决条件

在内部环境中创建集群前需要满足以下先决条件：

- 您必须在 OpenShift Container Platform 版本 4.9 或更高版本上部署了 Red Hat Advanced Cluster Management hub 集群。
- 您需要配置了主机的基础架构环境。如需更多信息，请参阅 创建基础架构环境。
- 您需要对 Red Hat Advanced Cluster Management for Kubernetes hub 集群（连接）或连接到连接到互联网（断开连接）的内部或镜像 registry 的连接，以检索创建集群所需的镜像。
- 您需要配置了内部凭证。如需更多信息，请参阅为内部环境创建凭证。
- 您需要 OpenShift Container Platform 镜像 pull secret。如需更多信息，请参阅使用镜像 pull secret。

1.6.9.2. 使用控制台创建集群

要从 Red Hat Advanced Cluster Management for Kubernetes 控制台创建集群，请进入 **Infrastructure** > **Clusters**。在 *Clusters* 页面上，点 **Create cluster** 并完成控制台中的步骤。

以下选项可用于您的支持安装：

- **使用现有的发现的主机** ：从现有基础架构环境中的主机列表中选择您的主机。
- **发现新主机** ：发现不在现有基础架构环境中的主机。发现您自己的主机，而不是使用已在基础架构环境中的主机。

如果您需要创建凭证，请参阅为内部环境创建凭证。

集群的名称用于集群的主机名。

重要： 当您创建集群时，Red Hat Advanced Cluster Management 控制器为集群及其资源创建一个命名空间。确保只在该命名空间中包含该集群实例的资源。销毁集群会删除命名空间和所有资源。

提示： 在控制台中输入信息时，选择 **YAML: On** 查看内容更新。

如果要将集群添加到现有的集群集中，则需要在集群设置上具有正确的权限来添加它。如果在创建集群时没有 **cluster-admin** 权限，则必须选择一个具有 **clusterset-admin** 权限的集群集。如果您在指定的集群集中没有正确的权限，集群创建会失败。如果您没有任何集群集选项，请联络您的集群管理员，为集群集提供 **clusterset-admin** 权限。

每个受管集群都必须与受管集群集关联。如果您没有将受管集群分配给 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。

如果已有与您为供应商帐户配置的所选凭证关联的基本 DNS 域，则该值会在那个字段中填充。您可以通过覆盖它来更改值，但创建集群后无法更改此设置。此基础域用于创建到 OpenShift Container Platform 集群组件的路由。它在集群供应商的 DNS 中被配置为授权起始(SOA)记录。

OpenShift 版本 标识用于创建集群的 OpenShift Container Platform 镜像的版本。如果要使用的版本可用，您可以从镜像列表中选择镜像。如果您要使用的镜像不是标准镜像，您可以输入您要使用的镜像的 URL。有关发行镜像的更多信息，请参阅发行镜像。

当您选择 4.9 或更高版本的 OpenShift 版本时，会显示 **Install single node OpenShift(SNO)** 的选项。单节点 OpenShift 集群仅包含一个节点，用于托管 control plane 服务和用户工作负载。您不能在创建后向单节点 OpenShift 集群添加附加节点。

如果您希望集群是一个单节点 OpenShift 集群，请选择单节点 OpenShift 选项。

注： 单节点 OpenShift control plane 需要 8 个 CPU 内核，而多节点 control plane 集群的 control plane 节点只需要 4 个 CPU 内核。

查看并保存集群后，您的集群将保存为集群草稿。您可以通过在 *Clusters* 页面中选择集群名称来关闭创建过程，并在稍后完成该过程。

如果您使用的是现有主机，请选择是否要自行选择主机，还是自动选择它们。主机数量取决于您选择的节点数量。例如，SNO 集群只需要一个主机，而标准的三节点集群需要三个主机。

主机位置列表中显示了满足此集群要求的可用*主机位置*。对于主机的分发和更加高可用性的配置，请选择多个位置。

如果您发现新主机没有现存基础架构环境，请完成将主机添加到基础架构环境，从第 4 步开始以定义您的主机。

绑定主机以及验证通过后，通过添加以下 IP 地址完成集群的网络信息：

- API VIP：指定用于内部 API 通信的 IP 地址。
  注： 这个值必须与您用来创建 prerequisites 部分中列出的 DNS 记录的名称匹配。如果没有提供，DNS 必须预先配置，以便 **api.** 可以正确解析。

- Ingress VIP：指定用于入口流量的 IP 地址。
  注： 这个值必须与您用来创建 prerequisites 部分中列出的 DNS 记录的名称匹配。如果没有提供，则必须预先配置 DNS，以便 **test.apps** 可以被正确解析。

您可以在 *Clusters* 导航页面中查看安装状态。

如需了解更多与访问集群相关的信息，继续访问集群。

1.6.10. 休眠创建的集群（技术预览）

您可以休眠使用 Red Hat Advanced Cluster Management for Kubernetes 创建的集群来节省资源。休眠集群需要的资源比正在运行的资源要少得多，因此您可以通过将集群移入和停止休眠状态来降低供应商成本。此功能只适用于在以下环境中由 Red Hat Advanced Cluster Management 创建的集群：

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform

1.6.10.1. 使用控制台休眠集群

要使用 Red Hat Advanced Cluster Management 控制台休眠由 Red Hat Advanced Cluster Management 创建的集群，请完成以下步骤：

1. 在 Red Hat Advanced Cluster Management 导航菜单中选择 **Infrastructure** > **Clusters**。确保已选中 *Manage* cluster 选项卡。
2. 从集群的 *Options* 菜单中选择 **Hibernate cluster**。**注：**如果 *Hibernate cluster* 选项不可用，您就无法休眠该群集。当集群被导入且不是由 Red Hat Advanced Cluster Management 创建时，会出现这种情况。

当进程完成后，*Clusters* 页面中的集群状态为 **Hibernating**。

提示：您可以通过在 *Clusters* 页面上选择要休眠的集群并选择 **Actions** > **Hibernate clusters**来休眠多个集群。

您选择的集群正在休眠。

1.6.10.2. 使用 CLI Hibernate 集群

要使用 CLI 休眠由 Red Hat Advanced Cluster Management 创建的集群，请完成以下步骤：

1. 输入以下命令编辑您要休眠的集群设置：

   ```
   oc edit clusterdeployment <name-of-cluster> -n <namespace-of-cluster>
   ```

   将 **name-of-cluster** 替换为您要休眠的集群的名称。

   将 **namespace-of-cluster** 替换为您要休眠的集群的命名空间。

2. 将 **spec.powerState** 的值改为 **Hibernating**。
3. 输入以下命令查看集群的状态：

   ```
   oc get clusterdeployment <name-of-cluster> -n <namespace-of-cluster> -o yaml
   ```

   将 **name-of-cluster** 替换为您要休眠的集群的名称。

   将 **namespace-of-cluster** 替换为您要休眠的集群的命名空间。

   当集群休眠过程完成时，集群的 type 值为 **type=Hibernating**。

您选择的集群正在休眠。

1.6.10.3. 使用控制台恢复休眠集群的一般操作

要使用 Red Hat Advanced Cluster Management 控制台恢复一个休眠集群的正常操作，请完成以下步骤：

1. 在 Red Hat Advanced Cluster Management 导航菜单中选择 **Infrastructure** > **Clusters**。确保已选中 *Manage* cluster 选项卡。
2. 从您要恢复的集群的 *Options* 菜单中选择 **Resume cluster**。

当进程完成后，*Clusters* 页面中的集群状态为 **Ready**。

提示：您可以通过在 *Clusters* 页面上选择要恢复的集群并选择 **Actions** > **Resume clusters**来休眠多个集群。

所选集群恢复正常操作。

1.6.10.4. 使用 CLI 恢复休眠集群的一般操作

要使用 CLI 恢复休眠集群的一般操作，请完成以下步骤：

1. 输入以下命令来编辑集群的设置：

   ```
   oc edit clusterdeployment <name-of-cluster> -n <namespace-of-cluster>
   ```

   将 **name-of-cluster** 替换为您要休眠的集群的名称。

   将 **namespace-of-cluster** 替换为您要休眠的集群的命名空间。

2. 将 **spec.powerState** 的值改为 **Running**。
3. 输入以下命令查看集群的状态：

   ```
   oc get clusterdeployment <name-of-cluster> -n <namespace-of-cluster> -o yaml
   ```

   将 **name-of-cluster** 替换为您要休眠的集群的名称。

   将 **namespace-of-cluster** 替换为您要休眠的集群的命名空间。

   完成恢复集群的过程后，集群的 type 值为 **type=Running**。

所选集群恢复正常操作。

1.7. 将目标受管集群导入到 HUB 集群

您可以从不同的 Kubernetes 云供应商导入集群。导入后，目标集群就成为 Red Hat Advanced Cluster Management for Kubernetes hub 集群的受管集群。除非另有指定，否则在可以访问 hub 集群和目标受管集群的任意位置完成导入任务。

hub 集群无法管理*任何*其他 hub 集群,但可以管理自己。hub 集群被配置为自动导入和自助管理。您不需要手动导入 hub 集群。

但是，如果您删除 hub 集群并尝试再次导入它，则需要添加 **local-cluster:true** 标签。

从以下说明中进行选择以通过控制台或 CLI 设置受管集群：

所需的用户类型或访问权限级别：集群管理员

- 使用控制台导入现有集群
- 使用 CLI 导入受管集群
- 修改集群的 klusterlet 附加组件设置

1.7.1. 使用控制台导入现有集群

安装 Red Hat Advanced Cluster Management for Kubernetes 后，就可以导入集群来进行管理。您可以从控制台和 CLI 导入。

按照以下步骤从控制台导入。在此过程中，您需要通过终端来进行身份验证。

- 先决条件
- 导入集群
- 删除集群

1.7.1.1. 先决条件

- 您需要一个已部署的 Red Hat Advanced Cluster Management for Kubernetes hub 集群。如果要导入裸机集群，则必须在 Red Hat OpenShift Container Platform 版本 4.8 或更高版本上安装了 hub 集群。
- 您需要一个要管理的集群，以及互联网连接。
- 安装 **kubectl**。要安装 **kubectl**，请参阅 Kubernetes 文档中的*安装和设置 kubectl*。
- 您需要 **base64** 命令行工具。
- **注：** 如果要导入不是由 OpenShift Container Platform 创建的集群，则需要定义 **multiclusterhub.spec.imagePullSecret**。安装 Red Hat Advanced Cluster Management 时可能会创建此 secret。
  如果您需要创建新服务，请完成以下步骤：

  1. 从 cloud.redhat.com 下载 Kubernetes pull secret。
  2. 将 pull secret 添加到 hub 集群的命名空间。
  3. 运行以下命令，在 hub 集群的命名空间中创建新 secret：

     ```
     oc create secret generic pull-secret -n <open-cluster-management> --from-file=.dockerconfigjson=<path-to-pull-secret> --type=kubernetes.io/dockerconfigjson
     ```

     将 **open-cluster-management** 替换为 hub 集群的命名空间的名称。hub 集群的默认命名空间是 **open-cluster-management**。

     将 **path-to-pull-secret** 替换为您下载的 pull secret 的路径。

     在导入时，secret 会自动复制到受管集群。

     如需了解更多与 pull secret 相关的信息，请参阅使用镜像 pull secret 或了解并创建服务帐户。

如需有关如何定义此 secret 的更多信息，请参阅自定义镜像 pull secret。

- 确保您要导入的集群中已删除代理。必须删除 **open-cluster-management-agent** 和 **open-cluster-management-agent-addon** 命名空间以避免错误。
- 有关在 Red Hat OpenShift Dedicated 环境中导入，请参阅以下备注：
  - 您必须在 Red Hat OpenShift Dedicated 环境中部署了 hub 集群。
  - Red Hat OpenShift Dedicated 的默认权限是 dedicated-admin，但不包含创建命名空间的所有权限。您必须具有 **cluster-admin** 权限才能使用 Red Hat Advanced Cluster Management for Kubernetes 导入和管理集群。

所需的用户类型或访问权限级别：集群管理员

1.7.1.2. 导入集群

您可以从 Red Hat Advanced Cluster Management for Kubernetes 控制台中为每个可用的云供应商导入现有集群。

注： hub 集群无法管理不同的 hub 集群。hub 集群被设置为自动导入和管理自身，因此您不必手动导入 hub 集群来管理自己。

1. 在导航菜单中选择 **Infrastructure** > **Clusters**。
2. 在 *Managed cluster* 选项卡中，点 **Import cluster**。
3. 为集群提供名称。默认情况下，命名空间用于集群名称和命名空间。

重要： 当您创建集群时，Red Hat Advanced Cluster Management 控制器为集群及其资源创建一个命名空间。确保只在该命名空间中包含该集群实例的资源。销毁集群会删除命名空间和所有资源。

1. 如果要将其添加到现有集群集 中，请指定 *Cluster set*，则指定具有 **cluster-admin** 权限的现有集群集。如果在创建集群时没有 **cluster-admin** 权限，则必须选择一个具有 **clusterset-admin** 权限的集群集。如果您在指定的集群集中没有正确的权限，集群创建会失败。如果没有要选择的集群设置选项，请联系集群管理员，为集群集提供 **clusterset-admin** 权限。
   每个受管集群都必须与受管集群集关联。如果您没有将受管集群分配给 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。
2. **可选：** 添加任何 *附加标签*。
   注： 如果您导入 Red Hat OpenShift Dedicated 集群，且没有添加 **vendor=OpenShiftDedicated** 标签指定一个厂商，或者添加了 **vendor=auto-detect** 标签，则 **managed-by=platform** 标签会自动添加到集群中。您可以使用此添加标签将集群标识为一个 Red Hat OpenShift Dedicated 集群，并作为一个组来获取 Red Hat OpenShift Dedicated 集群。
3. 选择您要用来标识从以下选项导入的集群的 *导入模式* ：
   - **手动运行导入命令**：根据您提供的信息生成可复制和运行的导入命令。点 **Save import and generate code** 生成用于部署 **open-cluster-management-agent-addon** 的命令。此时会显示确认信息。
     a. 在 *Import an existing cluster* 窗口中，选择 **Copy** 命令将生成的命令和令牌复制到剪贴板。
     重要： 命令中包含复制到每个导入集群的 pull secret 信息。具有访问导入集群权限的所有用户都可以查看 pull secret 信息。考虑在 https://cloud.redhat.com/ 创建一个二级 pull secret，或创建一个服务帐户来保护个人凭证。

b. 登录到您要导入的受管集群。

c. **对于 Red Hat OpenShift Dedicated 环境：**完成以下步骤：

i. 创建 **open-cluster-management-agent** 和 **open-cluster-management** 命名空间或受管集群上的项目。

ii. 在 OpenShift Container Platform 目录中找到 klusterlet Operator。

iii. 在 **open-cluster-management** 命名空间中或您创建的项目中安装它。
重要： 不要在 **open-cluster-management-agent** 命名空间中安装 Operator。

iv. 通过完成以下步骤，从导入命令中提取 bootstrap secret：

A. 生成导入命令：

I. 从 Red Hat Advanced Cluster Management 控制台主导航中选择 **Infrastructure** > **Clusters**。

II. 选择 **Add a cluster** > **Import an existing cluster**。

III. 添加集群信息，选择 **Save import and generate code**。

B. 复制导入命令。

C. 将导入命令粘贴到您创建的名为 **import-command** 的文件中。

D. 运行以下命令以将内容插入新文件中：

```
cat import-command | awk '{split($0,a,"&&"); print a[3]}' | awk '{split($0,a,"|"); print a[1]}' | sed -e "s/^ echo //" | base64 -d
```

E. 在输出中找到并复制名为 **bootstrap-hub-kubeconfig** 的 secret。

F. 将 secret 应用到受管集群上的 **open-cluster-management-agent** 命名空间。

G. 使用安装的 Operator 中的示例创建 klusterlet 资源，clusterName 应该与导入过程中设置的集群名称相同。
注： 当 **managedcluster** 资源在 hub 中成功注册时，会安装两个 klusterlet operator。一个 klusterlet operator 位于 **open-cluster-management** 命名空间中，另一个位于 **open-cluster-management-agent** 命名空间中。多个 operator 并不会影响 klusterlet 的功能。

d. **对于不在 Red Hat OpenShift Dedicated 环境中的集群导入，**请完成以下步骤：

i. 如有必要，为您的受管集群配置 **kubectl** 命令。
请参阅支持的供应商以了解如何配置 **kubectl** 命令行界面。

ii. 要将 **open-cluster-management-agen-addon** 部署到受管集群，请运行您复制的命令和令牌。

e. 选择 **View cluster** 在 *Overview* 页面中查看集群概述。

- **为现有集群输入服务器 URL 和 API 令牌**：提供您要导入的集群的服务器 URL 和 API 令牌。
- **kubeconfig** ：复制并粘贴您要导入的集群 **kubeconfig** 文件的内容。

4. **可选：** 在集群详情页中配置 **Cluster API address**，它为运行 **oc get managedcluster** 命令时在表中显示的 URL。

   a. 使用具有 **cluster-admin** 权限的 ID 登录到 hub 集群。

   b. 为目标受管集群配置 **kubectl**。
   请参阅支持的供应商了解如何配置 **kubectl**。

   c. 输入以下命令编辑您要导入的集群的受管集群条目：

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

   使用受管集群的名称替换 ***cluster-name***。

   d. 在 YAML 文件中的 **ManagedCluster** spec 中添加 **ManagedClusterClientConfigs**，如下例所示：

   ```
   spec:
     hubAcceptsClient: true
     managedClusterClientConfigs:
     - url: https://multicloud-console.apps.new-managed.dev.redhat.com
   ```

   将 URL 值替换为提供对您要导入的受管集群的外部访问的 URL。

集群已导入。您可以选择 **Import another** 来导入另一个。

1.7.1.3. 删除导入的集群

完成以下步骤以删除导入的集群以及在受管集群上创建的 **open-cluster-management-agent-addon**。

在 *Clusters* 页面上，点 **Actions** > **Detach cluster** 从管理中删除集群。

注意： 如果您试图分离名为 **local-cluster** 的 hub 集群，请注意 **disableHub selfManagement** 的默认设置为 **false**。此设置会导致 hub 集群在分离时会重新导入自己并管理自己，并协调 **MultiClusterHub** 控制器。hub 集群可能需要几小时时间来完成分离过程并重新导入。如果要在等待进程完成后重新导入 hub 集群，您可以输入以下命令来重启 **multiclusterhub-operator** pod 并更快地重新导入：

```
oc delete po -n open-cluster-management `oc get pod -n open-cluster-management | grep multiclusterhub-operator| cut -d' ' -f1`
```

您可以通过将 **disableHubSelfManagement** 值改为 **true** 来更改 hub 集群的值，使其不会自动导入。如需更多信息，请参阅 disableHubSelfManagement 主题。

1.7.2. 使用 CLI 导入受管集群

安装 Red Hat Advanced Cluster Management for Kubernetes 后，就可以使用 Red Hat OpenShift Container Platform CLI 导入集群来管理。您可以使用您要导入的集群的 **kubeconfig** 文件导入集群，也可以在要导入的集群中手动运行导入命令。这两个流程都已被记录在文档中。

- 先决条件
- 支持的构架
- 准备导入

- 使用自动导入 secret 导入集群
- 使用手动命令导入集群
- 导入 klusterlet 附加组件

重要： hub 集群无法管理不同的 hub 集群。hub 集群被设置为自动导入并管理自己。您不必手动导入 hub 集群来自己管理。

但是，如果您删除 hub 集群并尝试再次导入它，则需要添加 **local-cluster:true** 标签。

1.7.2.1. 先决条件

- 您需要一个已部署的 Red Hat Advanced Cluster Management for Kubernetes hub 集群。如果要导入裸机集群，则必须在 Red Hat OpenShift Container Platform 版本 4.6 或更高版本上安装了 hub 集群。
- 您需要一个需要管理的独立集群，且具有互联网连接。
- 您需要 Red Hat OpenShift Container Platform CLI 版本 4.6 或更高版本来运行 **oc** 命令。如需有关安装和配置 Red Hat OpenShift Container Platform CLI **oc** 的信息，请参阅 OpenShift CLI 入门。
- 您需要安装 Kubernetes CLI **kubectl**。要安装 **kubectl**，请参阅 Kubernetes 文档中的*安装和设置 kubectl*。
  注： 通过控制台下载 CLI 工具的安装文件。
- 如果您导入不是由 OpenShift Container Platform 创建的集群，则需要定义一个 **multiclusterhub.spec.imagePullSecret**。安装 Red Hat Advanced Cluster Management for Kubernetes 时可能已创建此 secret。如需有关定义 secret 的更多信息，请参阅自定义 Image Pull Secret。

1.7.2.2. 支持的构架

- Linux (x86_64, s390x, ppc64le)
- macOS

1.7.2.3. 准备导入

1. 运行以下命令登录到您的 *hub 集群* ：

   ```
   oc login
   ```

2. 在 hub 集群中运行以下命令以创建项目和命名空间： **请注意： CLUSTER_NAME** 中定义的集群名称也用作 YAML 文件和命令中的集群命名空间：

   ```
   oc new-project ${CLUSTER_NAME}
   ```

   重要： cluster.open-cluster-management.io/managedCluster 标签会自动添加到受管集群命名空间中并从中删除。不要手动将其添加到受管集群或从受管集群中删除。

3. 使用以下示例内容，创建一个名为 **managed-cluster.yaml** 的文件：

   ```
   apiVersion: cluster.open-cluster-management.io/v1
   ```

```
kind: ManagedCluster
metadata:
  name: ${CLUSTER_NAME}
  labels:
    cloud: auto-detect
    vendor: auto-detect
spec:
  hubAcceptsClient: true
```

当 **cloud** 和 **vendor** 的值被设置为 **auto-detect** 时，Red Hat Advanced Cluster Management 会检测您要导入的集群的云和厂商类型。您可以选择将 **auto-detect** 的值替换为集群的 cloud 和 vendor 值。请参见以下示例：

```
cloud: Amazon
vendor: OpenShift
```

4. 输入以下命令将 YAML 文件应用到 **ManagedCluster** 资源：

```
oc apply -f managed-cluster.yaml
```

使用自动导入 secret 或使用手动命令导入集群继续导入集群。

1.7.2.4. 使用自动导入 secret 导入集群

要使用自动导入 secret，您必须创建一个 secret，其中包含集群的 **kubeconfig** 文件或 kube API 服务器和集群的令牌对。

1. 检索您要导入的集群的 **kubeconfig** 文件或 kube API 服务器和令牌。请参阅 Kubernetes 集群的文档，了解在哪里可以找到您的 **kubeconfig** 文件或 kube API 服务器和令牌。

2. 在 ${CLUSTER_NAME} 命名空间中创建 **auto-import-secret.yaml** 文件。

   a. 创建名为 **auto-import-secret.yaml** 的 YAML 文件，其中包含类似以下模板的内容：

```
apiVersion: v1
kind: Secret
metadata:
  name: auto-import-secret
  namespace: <cluster_name>
stringData:
  autoImportRetry: "5"
  # If you are using the kubeconfig file, add the following value for the kubeconfig file
  # that has the current context set to the cluster to import:
  kubeconfig: |- <kubeconfig_file>
  # If you are using the token/server pair, add the following two values instead of
  # the kubeconfig file:
  token: <Token to access the cluster>
  server: <cluster_api_url>
type: Opaque
```

   b. 使用以下命令应用 ${CLUSTER_NAME} 命名空间中的 YAML 文件：

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

注：默认情况下，自动导入 secret 只使用一次，在导入过程完成后会被删除。如果要保留自动导入 secret，请将 **managedcluster-import-controller.open-cluster-management.io/keeping-auto-import-secret** 添加到 secret。您可以运行以下命令来添加它：

```
oc -n <cluster_name> annotate secrets auto-import-secret managedcluster-import-controller.open-cluster-management.io/keeping-auto-import-secret=""
```

3. 验证您的导入集群的 **JOINED** 和 **AVAILABLE** 状态。在 hub 集群中运行以下命令：

```
oc get managedcluster ${CLUSTER_NAME}
```

4. 在受管集群中运行以下命令来登录到受管集群：

```
oc login
```

5. 运行以下命令，以验证您要导入的集群中的 pod 状态：

```
oc get pod -n open-cluster-management-agent
```

继续 导入 klusterlet 附加组件。

1.7.2.5. 使用手动命令导入集群

重要： 导入命令包含复制到每个导入集群的 pull secret 信息。具有访问导入集群权限的所有用户都可以查看 pull secret 信息。

1. 运行以下命令，获取由导入控制器在 hub 集群上生成的 **klusterlet-crd.yaml** 文件：

```
oc get secret ${CLUSTER_NAME}-import -n ${CLUSTER_NAME} -o jsonpath={.data.crds\\.yaml} | base64 --decode > klusterlet-crd.yaml
```

2. 运行以下命令，获取导入控制器在 hub 集群上生成的 **import.yaml** 文件：

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

在要导入的集群中执行以下步骤：

3. 输入以下命令登录到您导入的受管集群：

```
oc login
```

4. 运行以下命令应用您在第 1 步中生成的 **klusterlet-crd.yaml** ：

```
oc apply -f klusterlet-crd.yaml
```

5. 运行以下命令应用您之前生成的 **import.yaml** 文件：

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

6. 验证您要导入的集群的 **JOINED** 和 **AVAILABLE** 状态。在 hub 集群中运行以下命令：

```
oc get managedcluster ${CLUSTER_NAME}
```

继续 导入 klusterlet 附加组件。

1.7.2.6. 导入 klusterlet 附加组件

您可以通过完成以下步骤来创建并应用 klusterlet 附加组件配置文件：

1. 创建一个类似以下示例的 YAML 文件：

```
apiVersion: agent.open-cluster-management.io/v1
kind: KlusterletAddonConfig
metadata:
  name: <cluster_name>
  namespace: <cluster_name>
spec:
  applicationManager:
    enabled: true
  certPolicyController:
    enabled: true
  iamPolicyController:
    enabled: true
  policyController:
    enabled: true
  searchCollector:
    enabled: true
```

2. 将文件保存为 **klusterlet-addon-config.yaml**。

3. 运行以下命令来应用 YAML：

```
oc apply -f klusterlet-addon-config.yaml
```

ManagedCluster-Import-Controller 将生成一个名为 **${CLUSTER_NAME}-import** 的 secret。**${CLUSTER_NAME}-import** secret 包括 **import.yaml**，用户会把它应用到一个受管集群来安装 klusterlet.

附加组件安装在您导入的集群后为 **AVAILABLE**。

4. 运行以下命令，验证您要导入的集群上附加组件的 pod 状态：

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

集群现已导入。

1.7.2.7. 使用 CLI 删除导入的集群

要删除集群，请运行以下命令：

```
oc delete managedcluster ${CLUSTER_NAME}
```

将 **cluster_name** 替换为集群的名称。

您的集群现已被删除。

1.7.3. 使用自定义 ManagedClusterImageRegistry CRD 导入集群

有时您可能需要覆盖您要导入的受管集群中的镜像 registry。您可以通过创建一个 **ManagedClusterImageRegistry** 自定义资源定义(CRD)来完成此操作。

ManagedClusterImageRegistry CRD 是一个命名空间范围的资源。

ManagedClusterImageRegistry CRD 为要选择的放置指定一组受管集群，但需要与自定义镜像 registry 不同的镜像。使用新镜像更新受管集群后，会在每个受管集群中添加以下标签进行识别： **open-cluster-management.io/image-registry=<namespace>.<managedClusterImageRegistryName>**。

以下示例显示了 **ManagedClusterImageRegistry** CRD:

```
apiVersion: imageregistry.open-cluster-management.io/v1alpha1
kind: ManagedClusterImageRegistry
metadata:
  name: <imageRegistryName>
  namespace: <namespace>
spec:
  placementRef:
    group: cluster.open-cluster-management.io
    resource: placements
    name: <placementName>
  pullSecret:
    name: <pullSecretName>
  registries:
  - mirror: <mirrored-image-registry-address>
    source: <image-registry-address>
  - mirror: <mirrored-image-registry-address>
    source: <image-registry-address>
```

在 **spec** 部分中：

- 将 **placementName** 替换为选择一组受管集群的放置名称。
- 将 **pullSecretName** 替换为用于从自定义镜像 registry 中拉取镜像的 pull secret 名称。
- 列出每个 **source** 和 **mirror** registry 的值。将 **mirrored-image-registry-address** 和 **image-registry-address** 替换为每个 registry 的 **mirror** 和 **source** 的值。
  - 示例 1：要将名为 **registry.redhat.io/rhacm2** 的源 registry 替换为 **localhost:5000/rhacm2**，并将 **registry.redhat.io/multicluster-engine** 替换为 **localhost:5000/multicluster-engine**，请使用以下示例：

    ```
    registries:
    - mirror: localhost:5000/rhacm2/
        source: registry.redhat.io/rhacm2
    - mirror: localhost:5000/multicluster-engine
        source: registry.redhat.io/multicluster-engine
    ```

  - 示例 2： 要将源镜像 **registry.redhat.io/rhacm2/registration-rhel8-operator** 替换为 **localhost:5000/rhacm2-registration-rhel8-operator**，请使用以下示例：

    ```
    registries:
    - mirror: localhost:5000/rhacm2-registration-rhel8-operator
        source: registry.redhat.io/rhacm2/registration-rhel8-operator
    ```

1.7.3.1. 使用 ManagedClusterImageRegistry CRD 导入集群

完成以下步骤，使用 ManagedClusterImageRegistry CRD 导入集群：

1. 在您要导入集群的命名空间中创建 pull secret。对于这些步骤，是 **myNamespace**。

```
$ kubectl create secret docker-registry myPullSecret \
  --docker-server=<your-registry-server> \
  --docker-username=<my-name> \
  --docker-password=<my-password>
```

2. 在您创建的命名空间中创建一个放置。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: myPlacement
  namespace: myNamespace
spec:
  clusterSets:
  - myClusterSet
  tolerations:
  - key: "cluster.open-cluster-management.io/unreachable"
    operator: Exists
```

注： 需要 **unreachable** 容限才能使配置来选择集群。

3. 创建一个 **ManagedClusterSet** 资源，并将其绑定到命名空间。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: ManagedClusterSet
metadata:
  name: myClusterSet

---
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: ManagedClusterSetBinding
metadata:
  name: myClusterSet
  namespace: myNamespace
spec:
  clusterSet: myClusterSet
```

4. 在命名空间中创建 **ManagedClusterImageRegistry** CRD。

```
apiVersion: imageregistry.open-cluster-management.io/v1alpha1
kind: ManagedClusterImageRegistry
metadata:
  name: myImageRegistry
  namespace: myNamespace
spec:
  placementRef:
    group: cluster.open-cluster-management.io
    resource: placements
    name: myPlacement
```

```
  pullSecret:
    name: myPullSecret
  registry: myRegistryAddress
```

5. 从 Red Hat Advanced Cluster Management 控制台导入受管集群，并将其添加到受管集群集中。

6. 在标签 **open-cluster-management.io/image-registry=myNamespace.myImageRegistry** 添加到受管集群后，在受管集群中复制并运行导入命令。

1.7.4. 修改集群的 klusterlet 附加组件设置

您可以使用 hub 集群修改 **KlusterletAddonConfig** 的设置，以更改您的配置。

KlusterletAddonConfig 控制器根据 **klusterletaddonconfigs.agent.open-cluster-management.io** Kubernetes 资源中的设置管理启用和禁用的功能。查看以下 **KlusterletAddonConfig** 示例：

```
apiVersion: agent.open-cluster-management.io/v1
kind: KlusterletAddonConfig
metadata:
  name: <cluster-name>
  namespace: <cluster-name>
spec:
  clusterName: <cluster-name>
  clusterNamespace: <cluster-name>
  clusterLabels:
    cloud: auto-detect
    vendor: auto-detect
  applicationManager:
    enabled: true
  certPolicyController:
    enabled: true
  iamPolicyController:
    enabled: true
  policyController:
    enabled: true
  searchCollector:
    enabled: false
  version: 2.5.0
```

1.7.4.1. klusterlet 附加组件设置描述

以下设置可以在 **klusterletaddonconfigs.agent.open-cluster-management.io** Kubernetes 资源中更新：

表 1.2. klusterlet 附加组件设置列表

设置名称	值	描述
applicationmanager	**true** 或 **false**	此控制器在受管集群中管理应用程序订阅生命周期。
certPolicyController	**true** 或 **false**	此控制器在受管集群中强制实施基于证书的策略。

设置名称	值	描述
iamPolicyController	**true** 或 **false**	此控制器在受管集群上强制实施基于 IAM 的策略生命周期。
policyController	**true** 或 **false**	此控制器在受管集群上强制执行所有其他策略规则。
searchCollector	**true** 或 **false**	此控制器用于定期将资源索引数据推送回 hub 集群。

1.7.4.2. 使用 hub 集群中的控制台进行修改

您可以使用 hub 集群修改 **klusterletaddonconfigs.agent.open-cluster-management.io** 资源设置。完成以下步骤以更改设置：

1. 登录到 hub 集群的 Red Hat Advanced Cluster Management for Kubernetes 控制台。
2. 在 hub 集群控制台的标头菜单中选择 **Search** 图标。
3. 在搜索参数中输入以下值：**kind:klusterletaddonconfigs**
4. 选择您要更新的端点资源。
5. 找到 **spec** 部分并选择 **Edit** 以编辑内容。
6. 修改设置。
7. 选择 **Save** 以应用您的更改。

1.7.4.3. 使用 hub 集群中的命令行进行修改

您必须有权访问 **cluster-name** 命名空间才能使用 hub 集群修改设置。完成以下步骤：

1. 登录到 hub 集群。
2. 输入以下命令以编辑资源：

```
kubectl edit klusterletaddonconfigs.agent.open-cluster-management.io <cluster-name> -n <cluster-name>
```

3. 找到 **spec** 部分。
4. 根据需要修改您的设置。

1.8. 访问集群

要访问由 Red Hat Advanced Cluster Management for Kubernetes 管理的 Red Hat OpenShift Container Platform 集群，请完成以下步骤：

1. 在 Red Hat Advanced Cluster Management for Kubernetes 导航菜单中导航到 **Infrastructure** > **Clusters**，并选择您创建的或想要访问的集群名称。

2. 选择 **Reveal credentials** 来查看集群的用户名和密码。记下这些值以便在登录到集群时使用。
   注： **Reveal credentials** 选项不适用于导入的集群。
3. 选择 **Console URL** 以链接到集群。
4. 使用在第 3 步中找到的用户 ID 和密码登录集群。

1.9. 在代理环境中创建集群

当 hub 集群通过代理服务器连接时，您可以创建 Red Hat OpenShift Container Platform 集群。

要成功创建集群，则必须满足以下情况之一：

- Red Hat Advanced Cluster Management for Kubernetes 与您要创建的受管集群具有私有网络连接，但 Red Hat Advanced Cluster Management 和受管集群使用代理访问互联网。
- 受管集群位于基础架构供应商上，但防火墙端口启用了从受管集群到 hub 集群的通信。

要创建使用代理配置的集群，请完成以下步骤：

1. 通过在 **install-config.yaml** 文件中添加以下信息，在 hub 集群上配置集群范围的代理设置：

```
apiVersion: v1
kind: Proxy
baseDomain: <domain>
proxy:
  httpProxy: http://<username>:<password>@<proxy.example.com>:<port>
  httpsProxy: https://<username>:<password>@<proxy.example.com>:<port>
  noProxy: <wildcard-of-domain>,<provisioning-network/CIDR>,<BMC-address-range/CIDR>
```

   使用代理服务器的用户名替换 **username**。

   使用密码替换 **password** 以访问您的代理服务器。

   将 **proxy.example.com** 替换为代理服务器的路径。

   使用与代理服务器的通信端口替换 **port**。

   将 **wildcard-of-domain** 替换为应当绕过代理的域的条目。

   使用置备网络的 IP 地址和分配的 IP 地址（以 CIDR 表示）替换 **provisioning-network/CIDR**。

   将 **BMC-address-range/CIDR** 替换为 BMC 地址和地址数（以 CIDR 表示）。

   添加前面的值后，设置将应用到集群。

2. 通过完成创建集群的步骤来置备集群。请参阅创建集群以选择您的供应商。

1.9.1. 在现有集群附加组件上启用集群范围代理

您可以配置集群命名空间中的 **KlusterletAddonConfig**，将代理环境变量添加到由 hub 集群管理的 Red Hat OpenShift Container Platform 集群的所有 klusterlet 附加组件 pod 中。

完成以下步骤，配置 **KlusterletAddonConfig**，将 3 个环境变量添加到 klusterlet add-ons 的 pod 中：

1. 打开位于需要添加代理的集群命名空间中的 **KlusterletAddonConfig** 文件。

2. 编辑文件的 **.spec.proxyConfig** 部分，使其类似以下示例：

```
spec
  proxyConfig:
    httpProxy: "<proxy_not_secure>"
    httpsProxy: "<proxy_secure>"
    noProxy: "<no_proxy>"
```

将 **proxy_not_secure** 替换为 **http** 请求的代理服务器的地址。例如：**http://192.168.123.145:3128**。

使用 **https** 请求的代理服务器的地址替换 **proxy_secure**。例如：**https://192.168.123.145:3128**。

使用以逗号分隔的 IP 地址、主机名和域名列表替换 **no_proxy**，其中不会通过代理路由流量。例如：**.cluster.local,.svc,10.128.0.0/14,example.com**。

spec.proxyConfig 是一个可选部分。如果使用在 Red Hat Advanced Cluster Management hub 集群上配置的集群范围内的代理创建 OpenShift Container Platform 集群，则集群范围的代理配置值会在满足以下条件时添加到 klusterlet add-ons 的 pod 中：

- **addon** 部分中的 **.spec.policyController.proxyPolicy** 被启用并设置为 **OCPGlobalProxy**
- **.spec.applicationManager.proxyPolocy** 被启用并设置为 **CustomProxy**。
  注： **addon** 部分中的 **proxyPolicy** 默认值是 **Disabled**。

  请参见以下示例：

```
apiVersion: agent.open-cluster-management.io/v1
   kind: KlusterletAddonConfig
   metadata:
     name: clusterName
     namespace: clusterName
   spec:
     proxyConfig:
       httpProxy: http://pxuser:12345@10.0.81.15:3128
       httpsProxy: http://pxuser:12345@10.0.81.15:3128
       noProxy: .cluster.local,.svc,10.128.0.0/14, example.com
     applicationManager:
       enabled: true
       proxyPolicy: CustomProxy
     policyController:
       enabled: true
       proxyPolicy: OCPGlobalProxy
     searchCollector:
       enabled: true
       proxyPolicy: Disabled
     certPolicyController:
       enabled: true
       proxyPolicy: Disabled
     iamPolicyController:
       enabled: true
       proxyPolicy: Disabled
```

代理在集群附加组件上配置。

重要： 全局代理设置不会影响警报转发。要为使用集群范围代理的 Red Hat Advanced Cluster Management hub 集群设置警报转发，请参阅 转发警报 以了解更多详细信息。

1.10. 启用集群代理附加组件

在某些情况下，受管集群位于防火墙后面，无法由 hub 集群直接访问。要获取访问权限，您可以设置代理附加组件来访问受管集群的 **kube-api** 服务器，以提供更安全的连接。

需要的访问权限 ： Editor

要为 hub 集群和受管集群配置集群代理附加组件，请完成以下步骤：

1. 在 Red Hat Advanced Cluster Management for Kubernetes hub 集群中启用集群代理附加组件。请参阅 高级配置 以了解更多信息。

2. 通过完成以下步骤，配置 **kubeconfig** 文件以访问受管集群 **kube-apiserver** ：

   a. 为受管集群提供有效的访问令牌。您可以使用服务帐户对应的令牌，假设默认服务帐户位于 default 命名空间。

      i. 确保您在使用受管集群的上下文。假设名为 **managed-cluster.kubeconfig** 的文件是受管集群的 **kubeconfig** 文件。**提示：** 带有 **--kubeconfig=managed-cluster.kubeconfig** 的命令在受管集群上运行，此流程中的所有命令都应在同一控制台中运行。不要在不同控制台中运行命令。

      ii. 在服务帐户中添加一个角色，允许它通过运行以下命令来访问 pod：

      ```
      oc create role -n default test-role --verb=list,get --resource=pods --
      kubeconfig=managed-cluster.kubeconfig
      oc create rolebinding -n default test-rolebinding --serviceaccount=default:default --
      role=test-role --kubeconfig=managed-cluster.kubeconfig
      ```

      iii. 运行以下命令来查找服务帐户令牌的 secret：

      ```
      oc get secret -n default --kubeconfig=managed-cluster.kubeconfig | grep default-
      token
      ```

      iv. 运行以下命令复制令牌：

      ```
      export MANAGED_CLUSTER_TOKEN=$(kubectl --kubeconfig=managed-
      cluster.kubeconfig -n default get secret <default-token> -o jsonpath={.data.token} |
      base64 -d)
      ```

      将 **default-token** 替换为您的 secret 的名称。

   b. 在 Red Hat Advanced Cluster Management hub 集群中配置 **kubeconfig** 文件。

      i. 运行以下命令，在 hub 集群中导出当前的 **kubeconfig** 文件：

      ```
      oc config view --minify --raw=true > cluster-proxy.kubeconfig
      ```

      ii. 使用编辑器修改 **server** 文件。本例使用 **sed**。如果您使用 OSX，运行 **alias sed=gsed**。

      ```
      export TARGET_MANAGE_CLUSTER=<cluster1>
      ```

```
export NEW_SERVER=https://$(oc get route -n open-cluster-management cluster-proxy-addon-user -o=jsonpath='{.spec.host}')/$TARGET_MANAGE_CLUSTER

sed -i'' -e '/server:/c\    server: '"$NEW_SERVER"'' cluster-proxy.kubeconfig

export CADATA=$(oc get configmap -n openshift-service-ca kube-root-ca.crt -o=go-template='{{index .data "ca.crt"}}' | base64)

sed -i'' -e '/certificate-authority-data:/c\    certificate-authority-data: '"$CADATA"'' cluster-proxy.kubeconfig
```

使用您要访问的受管集群名称替换 **cluster1**。

iii. 输入以下命令删除原始用户凭证：

```
sed -i'' -e '/client-certificate-data/d' cluster-proxy.kubeconfig
sed -i'' -e '/client-key-data/d' cluster-proxy.kubeconfig
sed -i'' -e '/token/d' cluster-proxy.kubeconfig
```

iv. 添加服务帐户的令牌：

```
sed -i'' -e '$a\    token: '"$MANAGED_CLUSTER_TOKEN"'' cluster-proxy.kubeconfig
```

3. 运行以下命令，列出目标受管集群的目标命名空间中的所有 pod：

```
oc get pods --kubeconfig=cluster-proxy.kubeconfig -n <default>
```

将 **default** 命名空间替换为您要使用的命名空间。

您的 hub 集群现在与受管集群 **的 kube-api** 通信。

1.11. 配置特定的集群管理角色

安装 Red Hat Advanced Cluster Management for Kubernetes 时，默认配置在 Red Hat Advanced Cluster Management hub 集群上提供 **cluster-admin** 角色。此权限允许您在 hub 集群中创建、管理和导入受管集群。在某些情况下，您可能想要限制对由 hub 集群管理的某些受管集群的访问，而不是提供对 hub 集群上所有受管集群的访问。

您可以通过定义集群角色并将其应用到用户或组来限制对特定受管集群的访问。完成以下步骤以配置和应用角色：

1. 通过创建包含以下内容的 YAML 文件来定义集群角色：

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: <clusterrole-name>
rules:
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  resourceNames:
```

```
  - <managed-cluster-name>
  verbs:
  - get
  - list
  - watch
  - update
  - delete
  - deletecollection
  - patch
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  verbs:
  - create
- apiGroups:
  - ""
  resources:
  - namespaces
  resourceNames:
  - <managed-cluster-name>
  verbs:
  - create
  - get
  - list
  - watch
  - update
  - delete
  - deletecollection
  - patch
- apiGroups:
  - register.open-cluster-management.io
  resources:
  - managedclusters/accept
  resourceNames:
  - <managed-cluster-name>
  verbs:
  - update
```

将 ***clusterrole-name*** 替换为您要创建的集群角色的名称。

将 ***managed-cluster-name*** 替换为您希望用户有权访问的受管集群的名称。

2. 输入以下命令应用 **clusterrole** 定义：

```
oc apply <filename>
```

将 ***filename*** 替换为您在上一步中创建的 YAML 文件的名称。

3. 输入以下命令将 **clusterrole** 绑定到指定用户或组：

```
oc adm policy add-cluster-role-to-user <clusterrole-name> <username>
```

将 ***clusterrole-name*** 替换为您在上一步中应用的集群角色的名称。使用您要将集群角色绑定的用户名替换 ***username***。

1.12. 管理集群标签

在集群中添加标签以选择组资源。如需更多信息，请参阅标签和 选择器。

您可以添加新标签、删除现有标签，并为集群编辑现有标签。

要管理标签，请进入到 **Infrastructure** > **Clusters** 并在 *Clusters* 表中查找您的集群。使用集群的 **Options** 菜单选择 **Edit labels**。

- 要添加新标签，在 *Edit labels* 对话框中输入标签。您的输入必须采用以下格式： **Key=Value**。如果要添加多个标签，通过按 **enter**、添加逗号或在标签之间添加空格来分隔标签。只有在点 **Save** 后才会保存标签。
- 要删除现有标签，点您要在列表中删除的标签的 **Remove** 图标。
- 要更新现有标签，可以通过使用具有不同值的同一键添加新标签来为这个键分配一个新值。例如，您可以通过输入 **Key=NewValue** 来更改 **Key=Value**，以更新 **Key** 的值。

提示 ：您还可以从集群详情页面编辑集群标签。在导航菜单中点 **Infrastructure** > **Clusters**。在 Clusters 页面中，点击集群的名称来访问集群的详情页面。选择 *Labels* 部分中的 **Edit** 图标。此时会显示 *Edit labels* 对话框。

1.13. 配置 ANSIBLE TOWER 任务以在受管集群中运行

Red Hat Advanced Cluster Management 与 Ansible Tower 自动化集成，以便您可以创建创建或升级集群之前或之后的 prehook 和 posthook AnsibleJob 实例。为集群销毁配置 prehook 和 posthook 作业，集群扩展操作不被支持。

需要的访问权限：集群管理员

- 先决条件
- 使用控制台将 AnsibleJob 模板配置为在集群中运行
- 创建 AnsibleJob 模板
- 使用标签将 AnsibleJob 模板配置为在受管集群中运行
- 查看 Ansible 作业的状态

1.13.1. 先决条件

您必须满足以下先决条件才能在 Red Hat Advanced Cluster Management 集群中运行 Ansible 模板：

- OpenShift Container Platform 4.6 或更高版本
- 安装 Ansible Automation Platform Resource Operator，将 Ansible 作业连接到 Git 订阅的生命周期。为了获得最佳结果，在使用 AnsibleJob 启动 Ansible Tower 作业时，Ansible Tower 作业模板在运行时应该是等价的。您可以在 OpenShift Container Platform *OperatorHub* 中找到 Ansible Automation Platform Resource Operator。

有关安装和配置 Ansible Tower 自动化的更多信息，请参阅设置 Ansible 任务。

1.13.2. 使用控制台将 *AnsibleJob* 模板配置为在集群中运行

您必须指定在创建集群时要用于集群的 Ansible 作业模板。要在创建集群时指定模板，请在 *Automation* 步骤中选择您要应用到集群的 Ansible 模板。如果没有 Ansible 模板，请单击 **Add Automation template** 来创建。

1.13.3. 创建 *AnsibleJob* 模板

要使用集群安装或升级来启动 Ansible 作业，您必须创建一个 Ansible 作业模板来指定作业何时运行。它们可以配置为在集群安装或升级之前或之后运行。

要指定在创建模板时运行 Ansible 模板的详情，请完成控制台中的步骤：

1. 从 Red Hat Advanced Cluster Management 导航中选择 **Infrastructure** > **Automation**。
2. 选择适用于您的问题单的适用路径：
   - 如果要创建新模板，请单击 **Create Ansible template** 并继续第 3 步。
   - 如果要修改现有模板，请在要修改的模板的 *Options* 菜单中点击 **Edit template**，然后继续第 5 步。
3. 输入模板的唯一名称，其中包含小写字母数字字符或连字符(-)。
4. 选择您要用于新模板的凭据。要将 Ansible 凭证链接到 Ansible 模板，请完成以下步骤：
   a. 在 Red Hat Advanced Cluster Management 导航中选择 **Automation**。任何未链接到凭证的模板列表中的模板都包含可用于将模板链接到现有凭证的 **Link to credential** 图标。仅显示与模板相同的命名空间中的凭证。
   b. 如果没有可以选择的凭证，或者您不想使用现有凭证，请从您要链接的模板的 *Options* 菜单中选择 **Edit template**。
   c. 如果必须创建凭证，点 **Add credentials** 并完成为 Ansible Automation Platform 创建凭证 中的步骤。
   d. 在与模板相同的命名空间中创建凭据后，在编辑模板时，在 *Ansible Automation Platform credential* 字段中选择凭据。
5. 如果要在安装集群前启动任何 Ansible 作业，请在 *Pre-install Ansible job templates* 部分中选择 **Add an Ansible job template**。
6. 选择或输入 prehook 和 posthook Ansible 作业的名称，以添加到集群的安装或升级中。
   注：*Ansible job template name* 需要与 Ansible Tower 中的 Ansible 作业的名称匹配。
7. 如有必要，拖动 Ansible 作业以更改顺序。
8. 对于您要在集群安装后启动的 Ansible 作业模板，重复 *Post-install Ansible job templates* 、*Pre-upgrade Ansible job templates* 以及 *Post-upgrade Ansible job templates* 部分中的第 5 - 7 步。

您的 Ansible 模板已配置为在集群中运行，在指定操作发生时指定此模板。

1.13.4. 使用标签将 AnsibleJob 模板配置为在受管集群中运行

当集群由 Red Hat Advanced Cluster Management for Kubernetes 创建或通过标签导入来由 Red Hat Advanced Cluster Management 管理时，您可以创建一个绑定到集群的 **AnsibleJob**。

完成以下步骤以创建一个 Ansible 作业，并使用尚未由 Red Hat Advanced Cluster Management 管理的集群进行配置：

1. 在应用程序功能支持的一个频道中为 Ansible 任务创建定义文件。只支持 Git 频道。
使用 **AnsibleJob** 作为定义中的 **kind** 值。

   您的定义文件内容可能类似以下示例：

   ```
   apiVersion: apiVersion: tower.ansible.com/v1alpha1
   kind: AnsibleJob
   metadata:
     name: hive-cluster-gitrepo
   spec:
     tower_auth_secret: my-toweraccess
     job_template_name: my-tower-template-name
     extra_vars:
       variable1: value1
       variable2: value2
   ```

   通过将文件存储在 prehook 或 posthook 目录中，它会创建一个与放置规则匹配的集群名称列表。集群名称列表可作为 **extra_vars** 的值传递给 **AnsibleJob kind** 资源。当此值传递给 **AnsibleJob** 资源时，Ansible 作业可以决定新的集群名称并在自动化中使用它。

2. 登录您的 Red Hat Advanced Cluster Management hub 集群。

3. 通过 Red Hat Advanced Cluster Management 控制台，使用 Git 订阅创建一个应用程序，该订阅引用您刚刚创建的定义文件存储频道的频道。有关创建应用程序和订阅的更多信息，请参阅管理应用程序资源。
在创建订阅时，请指定一个标签，您可以在以后创建或导入的集群中添加该订阅以与集群连接。这可以是现有标签，如 **vendor=OpenShift**，也可以是您创建和定义的唯一可用标签。

   注：如果您选择已在使用的标签，Ansible 作业会自动运行。最佳实践是将不属于 prehook 或 posthook 的资源包含在应用程序中。

   当检测到集群时使用与 **AnsibleJob** 标签匹配的标签时，默认放置规则运行作业。如果您希望自动化在由 hub 集群管理的所有正在运行的集群中运行，请将以下内容添加到放置规则中：

   ```
   clusterConditions:
     - type: ManagedClusterConditionAvailable
       status: "True"
   ```

   您可以将其粘贴到放置规则的 YAML 内容中，或者在 Red Hat Advanced Cluster Management 控制台的 *Application create* 页面中选择 *Deploy to all online clusters and local cluster* 集群的选项。

4. 按照创建集群或将目标受管集群分别导入到 hub 集群中 的内容创建或导入集群。
在创建或导入集群时，使用您在创建订阅时使用的相同标签，**AnsibleJob** 会自动配置为在集群中运行。

Red Hat Advanced Cluster Management 会自动将集群名称注入 **AnsibleJob.extra_vars.target_clusters** 路径。您可以动态将集群名称注入到定义中。完成以下步骤，创建一个 AnsibleJob，并使用已经由 Red Hat Advanced Cluster Management 管理的集群进行配置：

1. 在 Git Channel 的 prehook 或 posthook 目录中为 AnsibleJob 创建定义文件。
使用 **AnsibleJob** 作为定义中的 **kind** 值。

   您的定义文件内容可能类似以下示例：

```
apiVersion: tower.ansible.com/v1alpha1
kind: AnsibleJob
metadata:
  name: hive-cluster-gitrepo
spec:
  tower_auth_secret: my-toweraccess
  job_template_name: my-tower-template-name
  extra_vars:
    variable1: value1
    variable2: value2
```

使用访问您的 Ansible Tower 所需的验证 secret 替换 ***my-toweraccess***。

将 ***my-tower-template-name*** 替换为 Ansible Tower 中的模板名称。

每次删除或添加由 Ansible 作业控制的集群时，AnsibleJob 会自动运行和更新 **extra_vars.target_clusters** 变量。此更新提供了通过特定自动化指定集群名称，或将自动化应用到一组集群的功能。

1.13.5. 查看 Ansible 作业的状态

您可以查看正在运行的 Ansible 作业的状态，以确保它启动并在成功运行。要查看正在运行的 Ansible 作业的当前状态，请完成以下步骤：

1. 在 Red Hat Advanced Cluster Management 菜单中，选择 **Infrastructure** > **Clusters** 以访问 *Clusters* 页面。
2. 选择集群名称来查看其详情。
3. 在集群信息上查看 Ansible 作业最后一次运行的状态。该条目显示以下状态之一：
   - 当安装 prehook 或 posthook 任务失败时，集群状态会显示 **Failed**。
   - 当升级 prehook 或 posthook 任务失败时，会在 *Distribution* 字段中显示升级失败的警告信息。
     提示： 如果集群 prehook 或 posthook 失败，您可以从 *Clusters* 页面重试升级。

1.14. 创建和管理 MANAGEDCLUSTERSETS

ManagedClusterSet 是一个受管集群的组。使用受管集群集，您可以一起管理对组中所有受管集群的访问。您还可以创建一个 **ManagedClusterSetBinding** 资源，将 **ManagedClusterSet** 资源绑定到命名空间。

每个受管集群都必须是 **ManagedClusterSet** 的成员。安装 hub 集群时，会创建一个名为 **default** 的 **ManagedClusterSet**。所有没有特别分配给受管集群集的受管集群都会被自动分配给 **default** 受管集群集。为确保默认受管集群集始终可用，您无法删除或更新 **default** 受管集群集。

注： 没有特别添加到 **ManagedClusterSet** 中的集群池不会添加到默认的 **ManagedClusterSet** 中。从集群池中声明受管集群后，如果不明确添加到另一个 **ManagedClusterSet** 中，则会将其添加到默认 **ManagedClusterSet** 中。

- 创建 ManagedClusterSet
- 为 ManagedClusterSet 分配用户或组基于角色的访问控制权限
- 创建 ManagedClusterSetBinding 资源

- 将集群添加到 ManagedClusterSet
- 从 ManagedClusterSet 中删除集群

1.14.1. 创建 ManagedClusterSet

您可以在受管集群集中将受管集群分组在一起，以限制受管集群的用户访问权限。

需要的访问权限： 集群管理员

ManagedClusterSet 是一个集群范围的资源，因此您必须在创建 **ManagedClusterSet** 的集群中具有集群管理权限。受管集群不能包含在多个 **ManagedClusterSet** 中。您可以从 Red Hat Advanced Cluster Management for Kubernetes 控制台或命令行界面创建受管集群集。

1.14.1.1. 使用控制台创建 ManagedClusterSet

完成以下步骤，使用 Red Hat Advanced Cluster Management 控制台创建受管集群集：

1. 在主控制台导航中，选择 **Infrastructure** > **Clusters** 并确保选择了 *Cluster set* 选项卡。
2. 选择 **Create cluster set**，并输入集群集的名称。

1.14.1.2. 使用命令行创建 ManagedClusterSet

将受管集群集的以下定义添加到 **yaml** 文件中，使用命令行创建受管集群集：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: ManagedClusterSet
metadata:
  name: <clusterset1>
```

将 **clusterset1** 替换为受管集集群的名称。

1.14.2. 为 ManagedClusterSet 分配用户或组基于角色的访问控制权限

您可以将用户或组分配给由 hub 集群上配置的身份提供程序提供的集群集合。

需要的访问权限：集群管理员

ManagedClusterSet API 提供了两个级别的 RBAC 权限：

- 集群集 **admin**
  - 对分配给受管集群集的所有集群和集群池资源具有完全访问权限。
  - 创建集群、导入集群和创建集群池的权限。创建受管集群集时，必须将权限分配给受管集群集。
- 集群集 **view**
  - 对分配给受管集群集的所有集群和集群池资源只读权限。
  - 没有创建集群、导入集群或创建集群池的权限。

完成以下步骤，通过 Red Hat Advanced Cluster Management 控制台将用户或组分配给受管集群集：

1. 在控制台的主导航菜单中选择 **Infrastructure** > **Clusters**。
2. 选择 *Cluster sets* 选项卡。
3. 选择您的目标集群集。
4. 选择 *Access management* 选项卡。
5. 选择 **Add user or group**。
6. 搜索，然后选择您要提供访问权限的用户和组。
7. 选择 **Cluster set admin** 或 **Cluster set view** 角色，赋予所选用户或用户组。如需有关角色权限的更多信息，请参阅角色概述。
8. 选择 **Add** 以提交更改。

表中会显示您的用户或组。可能需要几秒钟后，分配到所有受管集群设置的资源的权限才会被传播到您的用户或组。

有关基于角色的操作的更多信息，请参阅基于角色的访问控制。

如需放置信息，请参阅使用 ManagedClusterSets with Placement。

1.14.2.1. 创建 ManagedClusterSetBinding 资源

创建一个 **ManagedClusterSetBinding** 资源，将 **ManagedClusterSet** 资源绑定到命名空间。在同一命名空间中创建的应用程序和策略只能访问包含在绑定受管集群集资源的受管集群。

命名空间的访问权限会自动应用到绑定到该命名空间的受管集群集。如果您有访问权限来访问受管集群设置的命名空间，则会自动具有访问绑定到该命名空间的任何受管集群集的权限。但是，如果您只拥有访问受管集群集的权限，则没有访问命名空间中其他受管集群集的权限。如果没有看到受管集群集，则可能没有查看它所需的权限。

您可以使用控制台或命令行创建受管集群集绑定。

1.14.2.1.1. 使用控制台创建 ManagedClusterSetBinding

完成以下步骤，使用 Red Hat Advanced Cluster Management 控制台从受管集群集中删除集群：

1. 在主导航中选择 **Infrastructure** > **Clusters** 并选择 *Cluster set* 选项卡来访问集群页面。
2. 选择您要为创建绑定的集群集的名称，以查看集群集详情。
3. 选择 **Actions** > **Edit namespace bindings**。
4. 在 *Edit namespace bindings* 页面中，从下拉菜单中选择您要将集群集绑定到的命名空间。已选择现有的与集群集绑定的命名空间。

1.14.2.1.2. 使用命令行创建 ManagedClusterSetBinding

要使用命令行创建受管集群集绑定，请完成以下步骤：

1. 在 **yaml** 文件中创建 **ManagedClusterSetBinding** 资源。在创建受管集群集绑定时，受管集群集绑定的名称必须与要绑定的受管集群集的名称匹配。您的 **ManagedClusterSetBinding** 资源可能类似以下信息：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: ManagedClusterSetBinding
metadata:
  namespace: project1
  name: clusterset1
spec:
  clusterSet: clusterset1
```

2. 确保目标受管集群集有绑定权限。查看以下 **ClusterRole** 资源示例，其中包含允许用户绑定到 **clusterset1** 的规则：

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: clusterrole1
rules:
  - apiGroups: ["cluster.open-cluster-management.io"]
    resources: ["managedclustersets/bind"]
    resourceNames: ["clusterset1"]
    verbs: ["create"]
```

1.14.3. 将集群添加到 ManagedClusterSet

创建 **ManagedClusterSet** 后，您必须添加一个或多个受管集群。您可以使用控制台或命令行将受管集群添加到受管集群集。

1.14.3.1. 使用控制台将集群添加到 ManagedClusterSet

完成以下步骤，使用 Red Hat Advanced Cluster Management 控制台将集群添加到受管集群集中：

1. 如果您只创建了受管集群集，选择 **Manage resource assignments** 以直接进入 *Manage 资源分配*页面。继续执行此流程的第 6 步。
2. 如果集群已存在，在主导航中选择 **Infrastructure** > **Clusters** 来访问**集群**页面。
3. 选择 **Cluster set** 选项卡来查看可用的集群集。
4. 选择您要添加到受管集群集的集群集的名称，以查看集群设置详情。
5. 选择 **Actions** > **Manage resource assignments**。
6. 在 *Manage resource assignments* 页面上，选中您要添加到集群集的资源的复选框。
7. 选择 **Review** 查看您的更改。
8. 选择 **Save** 保存您的更改。
   注：如果将受管集群从一个受管集群集移到另一个受管集群，则必须在两个受管集群集中都有所需的 RBAC 权限。

1.14.3.2. 使用命令行将集群添加到 ManagedClusterSet

完成以下步骤，使用命令行将集群添加到受管集群集中：

1. 确保有一个 RBAC **ClusterRole** 条目，供您在 **managedclustersets/join** 的虚拟子资源中创建。没有这个权限，就无法将受管集群分配给 **ManagedClusterSet**。

如果此条目不存在，请将其添加到您的 **yaml** 文件中。示例条目类似以下内容：

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: clusterrole1
rules:
  - apiGroups: ["cluster.open-cluster-management.io"]
    resources: ["managedclustersets/join"]
    resourceNames: ["<clusterset1>"]
    verbs: ["create"]
```

将 **clusterset1** 替换为 **ManagedClusterSet** 的名称。

注：如果要将受管集群从一个 **ManagedClusterSet** 移到另一个，则必须在两个受管集群集中都有该权限。

2. 在 **yaml** 文件中查找受管集群的定义。在受管集群定义的这个部分添加与以下内容类似的标签：

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
spec:
  hubAcceptsClient: true
```

在本例中，**cluster1** 是受管集群的名称。

3. 添加一个标签，指定 **ManagedClusterSet**，格式为：**cluster.open-cluster-management.io/clusterset: clusterset1**。
您的代码类似以下示例：

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
  labels:
    cluster.open-cluster-management.io/clusterset: clusterset1
spec:
  hubAcceptsClient: true
```

在本例中，**cluster1** 是添加到受管集群集 **clusterset1** 的集群 。

注： 如果受管集群之前被分配给已删除的受管集群集，受管集群可能会有一个受管集群集已被指定到不存在的集群集合。如果出现这种情况，请用新名称替换。

1.14.4. 从 ManagedClusterSet 中删除受管集群

您可能希望从受管集群集中删除受管集群，将其移到不同的受管集群集，或者从集合的管理设置中删除。您可以使用控制台或命令行界面从受管集群集中删除受管集群。

注：每个受管集群都必须分配到一个受管集群集。如果您从 **ManagedClusterSet** 中删除受管集群，且不将其分配给不同的 **ManagedClusterSet**，则会自动添加到 **default** 受管集群集中。

1.14.4.1. 使用控制台从 ManagedClusterSet 中删除受管集群

完成以下步骤，使用 Red Hat Advanced Cluster Management 控制台从受管集群集中删除集群：

1. 如果您只创建了受管集群集，选择 **Manage resource assignments** 以直接进入 *Manage 资源分配*页面。继续执行此流程的第 5 步。
2. 如果集群已存在，在主导航中选择 **Infrastructure** > **Clusters** 来访问集群页面，并确保选择了 *Cluster sets* 选项卡。
3. 选择您要从受管集群集中删除的集群集的名称，以查看集群设置详情。
4. 选择 **Actions** > **Manage resource assignments**。
5. 在 *Manage resource assignments* 页面上，选中您要从集群集合中删除的资源的复选框。此步骤删除已是集群集成员的资源，或添加尚未成为集群集成员的资源。您可以通过查看受管集群的详情来查看资源是否已是集群集的成员。

注： 如果要将受管集群从一个受管集群集移到另一个受管集群，则必须在两个受管集群集中都有所需的 RBAC 权限。

1.14.4.2. 使用命令行从 ManagedClusterSet 中删除集群

要使用命令行从受管集群集中删除受管集群，请完成以下步骤：

1. 运行以下命令在受管集群集中显示受管集群列表：

   ```
   oc get managedclusters -l cluster.open-cluster-management.io/clusterset=<clusterset1>
   ```

   将 **clusterset1** 替换为受管集群集的名称。
2. 找到您要删除的集群条目。
3. 从您要删除的集群的 **yaml** 条目中删除该标签。参阅以下标签代码示例：

   ```
   labels:
     cluster.open-cluster-management.io/clusterset: clusterset1
   ```

注： 如果要将受管集群从一个受管集群集移到另一个受管集群集，则必须在两个受管集群集中都有所需的 RBAC 权限。

1.14.5. 在放置中使用 ManagedClusterSet

Placement 资源是一个命名空间范围的资源，它定义了一个规则来从 **ManagedClusterSets** 中选择 **ManagedClusters** 集合，它们绑定到放置命名空间。

需要的访问权限： Cluster administrator，Cluster set administrator

1.14.5.1. 放置概述

参阅以下有关使用受管集群放置的信息：

- Kubernetes 集群在 hub 集群中注册，作为集群范围的 **ManagedClusters**。
- **ManagedClusters** 被组织到集群范围的 **ManagedClusterSets** 中。

- **ManagedClusterSets** 与工作负载命名空间绑定。
- 命名空间范围的**放置**指定 **ManagedClusterSet** 的一个部分，用于选择潜在 **ManagedClusters** 的工作集合。
- 使用标签和声明选择器从该工作集中选择**放置**。
  重要：如果没有绑定到放置命名空间的 **ManagedClusterSet** 绑定，**Placement** 不会选择 **ManagedCluster**。
- **ManagedClusters** 的放置可以使用污点和容限控制。如需更多信息，请参阅使用污点和容限放置受管集群。

Placement 规格包括以下字段：

- **ClusterSets** 代表从中选择 **ManagedClusters** 的 **ManagedClusterSets**。
  - 如果没有指定，则从绑定到放置命名空间的 **ManagedClusterSets** 中选择 **ManagedClusters**。
  - 如果指定，**ManagedClusters** 会从这个集合的交集和绑定到放置命名空间的 **ManagedClusterSets** 中选择。
- **NumberOfClusters** 代表要选择的可以满足放置要求的 **ManagedClusters** 数量。
  如果没有指定，则会选择满足放置要求的所有 **ManagedClusters**。
- **Predicates** 代表一个 predicates 片段，用于选择带有标签和声明选择器的 **ManagedClusters**。predicate 是 ORed。
- **priorityrPolicy** 代表优先级器的策略。
  - **mode** 可以是 **Exact**, **Additive**, 或 **""**，其中 **""** 默认为 **Additive**。
    - 在 **Additive** 模式中，未特别提供配置值的任何优先级程序均通过其默认配置启用。在当前的默认配置中，*Steady* 和 *Balance* 的权重为 1，而其他优先级的权重为 0。默认配置将来可能会改变，这可能会改变优先级。**additive** 模式不要求您配置所有优先级别。
    - 在 **Exact** 模式中，没有特别提供的配置值的任何优先级优先级的权重为零。**Exact** 模式要求您输入您想要的完整一组优先级程序，但可避免版本之间的行为改变。
  - **configurations** 表示优先级器的配置。
    - scoreCoordinate 代表优先级和分数源的配置。
      - **type** 定义优先级分数的类型。类型是 **BuiltIn**, **AddOn**, 或 **" "**，其中 **" "** 默认为 **BuiltIn**。当类型为 **BuiltIn** 时，必须指定 prioritizer 的名称。当类型为 **AddOn** 时，您需要在 **AddOn** 中配置分数源。
      - **builtIn** 定义 BuiltIn prioritizer 的名称。以下列表包含有效的 **BuiltIn** prioritizer 名称：
        - Balance: 平衡集群之间的决策。
        - Steady：确保现有决策稳定。
        - ResourceAllocatableCPU 和 ResourceAllocatableMemory: 根据可分配资源排序集群。

- **addOn** 定义资源名称和分数名称。引入了 **AddOnPlacementScore** 来描述附加组件分数。请参阅可扩展调度以了解更多信息。
  - **resourceName** 定义 **AddOnPlacementScore** 的资源名称。放置 prioritizer 根据此名称选择 **AddOnPlacementScore** 自定义资源。
  - **scoreName** 定义 **AddOnPlacementScore** 中的分数名称。**AddOnPlacementScore** 包含分数名称和分数值的列表。**scoreName**，指定 prioritizer 要使用的分数。

- **weight** 定义优先级优先级的权重。该值必须在 [-10,10] 范围内。每个优先级器计算一个集群在 [-100, 100] 范围内的整数分数。集群的最终分数由以下公式 **sum(weight * prioritizer_score)** 决定。权重越高，表示优先级函数在集群选择中获得更高的权重，而 0 个权重表示已禁用优先级优先级。负权重表示它是选定的最后一个。

注： **configure.name** 文件将在 v1beta1 中删除，并替换为 **scoreCoordinate.builtIn** 文件。如果同时定义了 **name** 和 **scoreCoordinate.builtIn**，则使用 **scoreCoordinate.builtIn** 的值来确定选择。

1.14.5.2. 放置示例

您需要通过在该命名空间中创建一个 **ManagedClusterSetBinding** 来最少将一个 **ManagedClusterSet** 绑定到一个命名空间。**注：**您需要在 **managedclustersets/bind** 的虚拟子资源上对 **CREATE** 进行基于角色的访问权限。请参见以下示例：

- 您可以使用 **labelSelector** 选择 **ManagedClusters**。请参阅以下示例，其中 **labelSelector** 仅与带有标签 **vendor: OpenShift** 的集群匹配：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement1
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
        labelSelector:
          matchLabels:
            vendor: OpenShift
```

- 您可以使用 **claimSelector** 选择 **ManagedClusters**。请参阅以下示例，其中 **claimSelector** 仅与带有 **us-west-1** 的 **region.open-cluster-management.io** 的集群匹配：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement2
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
        claimSelector:
          matchExpressions:
            - key: region.open-cluster-management.io
              operator: In
              values:
                - us-west-1
```

- 您可以从特定的 **clusterSets** 中选择 **ManagedClusters**。请参阅以下示例，其中 **claimSelector** 仅与 **clusterSets: clusterset1 clusterset2** 匹配：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement3
  namespace: ns1
spec:
  clusterSets:
    - clusterset1
    - clusterset2
  predicates:
    - requiredClusterSelector:
        claimSelector:
          matchExpressions:
            - key: region.open-cluster-management.io
              operator: In
              values:
                - us-west-1
```

- 选择所需的 **ManagedClusters** 数量。请参阅以下示例，其中 **numberOfClusters** 为 **3** ：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement4
  namespace: ns1
spec:
  numberOfClusters: 3
  predicates:
    - requiredClusterSelector:
        labelSelector:
          matchLabels:
            vendor: OpenShift
        claimSelector:
          matchExpressions:
            - key: region.open-cluster-management.io
              operator: In
              values:
                - us-west-1
```

- 选择具有最大可分配内存的集群。
  注： 与 Kubernetes Node Allocatable 类似，"可分配"这里定义为每个集群中 pod 可用的计算资源数量。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement6
  namespace: ns1
spec:
  numberOfClusters: 1
  prioritizerPolicy:
```

```
    configurations:
      - scoreCoordinate:
          builtIn: ResourceAllocatableMemory
```

- 选择具有最大可分配 CPU 和内存的集群，并区分资源更改。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement7
  namespace: ns1
spec:
  numberOfClusters: 1
  prioritizerPolicy:
    configurations:
      - scoreCoordinate:
          builtIn: ResourceAllocatableCPU
        weight: 2
      - scoreCoordinate:
          builtIn: ResourceAllocatableMemory
        weight: 2
```

- 选择具有最大可分配内存的两个集群，以及最大附加组件分数 cpu 比例，并固定放置决策。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement8
  namespace: ns1
spec:
  numberOfClusters: 2
  prioritizerPolicy:
    mode: Exact
    configurations:
      - scoreCoordinate:
          builtIn: ResourceAllocatableMemory
      - scoreCoordinate:
          builtIn: Steady
        weight: 3
      - scoreCoordinate:
          type: AddOn
          addOn:
            resourceName: default
            scoreName: cpuratio
```

1.14.5.3. 放置决定

将创建一个或多个带有标签 **cluster.open-cluster-management.io/placement={placement name}** 的 **PlacementDecisions** 来代表由一个 **Placement** 选择的 **ManagedClusters**。

如果选择了 **ManagedCluster** 并添加到 **PlacementDecision** 中，消耗此**放置**的组件可能会在这个 **ManagedCluster** 上应用工作负载。当 **ManagedCluster** 不再被选择后，它会从 **PlacementDecisions** 中删除后，在此 **ManagedCluster** 上应用的工作负载也会被相应地删除。

请参阅以下 **PlacementDecision** 示例：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: PlacementDecision
metadata:
  labels:
    cluster.open-cluster-management.io/placement: placement1
  name: placement1-kbc7q
  namespace: ns1
  ownerReferences:
    - apiVersion: cluster.open-cluster-management.io/v1beta1
      blockOwnerDeletion: true
      controller: true
      kind: Placement
      name: placement1
      uid: 05441cf6-2543-4ecc-8389-1079b42fe63e
status:
  decisions:
    - clusterName: cluster1
      reason: ''
    - clusterName: cluster2
      reason: ''
    - clusterName: cluster3
      reason: ''
```

1.14.5.4. 附加组件状态

您可能希望根据部署在其上的附加组件的状态，为放置选择受管集群。例如，只有在集群中启用了特定的附加组件时，才会为放置选择受管集群。

您可以为附加组件指定标签，并在创建放置时根据需要指定其状态（如果需要）。如果集群中启用了附加组件，则会自动在 **ManagedCluster** 资源上创建一个标签。如果禁用了附加组件，则会自动删除该标签。

每个附加组件都由一个标签表示，格式为 **feature.open-cluster-management.io/addon-<addon_name>=<status_of_addon>**。

使用要在要选择的受管集群中启用的附加组件名称替换 **addon_name**。

将 **status_of_addon** 替换为在选择集群时该附加组件应具有的状态。**status_of_addon** 的可能值位于以下列表中：

- **available** ：附加组件已启用并可用。
- **unhealthy**：附加组件已启用，但租期不会持续更新。
- **unreachable** ：附加组件已启用，但没有为其找到租用。也可以在受管集群离线时导致这个问题。

例如，可用的 **application-manager** 附加组件由受管集群中的标签表示：

```
feature.open-cluster-management.io/addon-application-manager: available
```

请参阅以下基于附加组件及其状态创建放置的示例：

- 您可以通过添加以下 YAML 内容来创建放置，其中包含启用了 **application-manager** 的所有受管集群：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement1
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
        labelSelector:
          matchExpressions:
            - key: feature.open-cluster-management.io/addon-application-manager
              operator: Exists
```

- 您可以通过添加以下 YAML 内容来创建放置，其中包含启用了 **application-manager** 且具有 **available** 状态的所有受管集群：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement2
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
        labelSelector:
          matchLabels:
            "feature.open-cluster-management.io/addon-application-manager": "available"
```

- 您可以通过添加以下 YAML 内容来创建包含禁用 **application-manager** 的所有受管集群的放置：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement3
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
        labelSelector:
          matchExpressions:
            - key: feature.open-cluster-management.io/addon-application-manager
              operator: DoesNotExist
```

1.14.5.5. 可扩展调度

在基于放置资源的调度中，prioritizer 需要比 **MananagedCluster** 资源提供的默认值更多的数据来计算受管集群分数。例如，根据通过监控系统获取的集群 CPU 或内存使用情况数据来调度集群。

API **AddOnPlacementScore** 支持根据自定义分数进行调度的更可扩展方式。

- 您可以指定 **placement.yaml** 文件中的分数来选择集群。
- 作为分数供应商，第 3 方控制器可以在 hub 集群或受管集群上运行，以便维护 **AddOnPlacementScore** 生命周期，并在其中更新分数。

请参阅 **open-cluster-management** 存储库中的放置可扩展调度增强，以了解更多信息。

1.14.6. 使用污点和容限来放置受管集群

您可以使用污点和容限控制受管集群或受管集群集的放置。污点和容限提供了一种防止为特定放置选择受管集群的方法。如果要阻止某些受管集群包含在某些放置中，这个控制会很有用。您可以向受管集群添加污点，并为放置添加容限。如果污点和容限不匹配，则不会为该放置选择受管集群。

1.14.6.1. 将污点添加到受管集群

污点在受管集群的属性中指定，并允许放置来重新放置受管集群或一组受管集群。您可以通过输入类似以下示例的命令，为受管集群添加污点：

```
kubectl taint ManagedCluster <managed_cluster_name> key=value:NoSelect
```

污点的规格包括以下字段：

- **必需**键 - 应用到集群的污点键。这个值必须与受管集群的容限中的值匹配，以满足添加到该放置的条件。您可以确定这个值。例如，这个值可以是 **bar** 或 **foo.example.com/bar**。
- **可选**值 - 污点键的污点值。这个值必须与受管集群的容限中的值匹配，以满足添加到该放置的条件。例如，这个值可以是 **value**。
- **必需**效果 - 污点对不容许污点的放置效果，或者在污点和放置容限不匹配时发生什么。effect 的值必须是以下值之一：
  - **NoSelect** - 除非容许这个污点，否则不允许放置来选择集群。如果在设置污点前放置选择了集群，则会从放置决定中移除集群。
  - **NoSelectIfNew** - 如果是新集群，调度程序就无法选择该集群。只有容许污点且已在其集群决策中拥有集群，放置才可以选择集群。
- **必需 TimeAdded** - 添加污点的时间。这个值会自动设置。

1.14.6.2. 识别内置污点以反映受管集群的状态

当无法访问受管集群时，您不希望集群添加到放置中。以下污点会自动添加到无法访问的受管集群：

- **cluster.open-cluster-management.io/unavailable** - 当集群有 **ManagedClusterConditionAvailable** 条件为 **False** 时，这个污点被添加到受管集群。污点具有 **NoSelect** 和空值的效果，以防止调度不可用的集群。以下内容中提供了此污点的示例：

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
 name: cluster1
spec:
 hubAcceptsClient: true
 taints:
   - effect: NoSelect
     key: cluster.open-cluster-management.io/unavailable
     timeAdded: '2022-02-21T08:11:54Z'
```

- **cluster.open-cluster-management.io/unreachable** - 当 **ManagedClusterConditionAvailable** 条件的状态为 **Unknown** 或没有条件时，此污点被添加到受管集群。污点对 **NoSelect** 和一个空值的影响，以防止调度无法访问的集群。以下内容中提供了此污点的示例：

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
spec:
  hubAcceptsClient: true
  taints:
    - effect: NoSelect
      key: cluster.open-cluster-management.io/unreachable
      timeAdded: '2022-02-21T08:11:06Z'
```

1.14.6.3. 为放置添加容限

容限应用到放置，并允许放置没有与放置容限匹配的污点的受管集群。容限的规格包括以下字段：

- **可选** 键 - 密钥与 taint 键匹配以允许放置。
- **可选** 值 - 容限中的值必须与容限的污点值匹配，以允许放置。
- **可选** Operator - Operator 代表键和值之间的关系。有效的操作符是 **equal** 和 **exists**。默认值为 **equal**。当键相同时，容限与污点匹配，影响相同，运算符是以下值之一：
  - **equal** - 运算符 **equal**，值在污点和容忍度中是相同的。
  - **exists** - 值的通配符，因此放置可以容限特定类别的所有污点。
- **可选** 效果 - 要匹配的污点效果。当留空时，它将匹配所有污点效果。指定后允许的值为 **NoSelect** 或 **NoSelectIfNew**。
- **可选** TolerationSeconds - 将受管集群移至新放置前容许污点的时间长度（以秒为单位）。如果 effect 值不是 **NoSelect** 或 **PreferNoSelect**，会忽略此字段。默认值为 **nil**，这表示没有时间限制。**TolerationSeconds** 的开始时间自动列为污点中的 **TimeAdded** 值，而不是在集群调度时间或者 **TolerationSeconds** 添加时间。

以下示例演示了如何配置容许具有污点的集群的容限：

- 受管集群中的污点，例如：

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
spec:
  hubAcceptsClient: true
  taints:
    - effect: NoSelect
      key: gpu
      value: "true"
      timeAdded: '2022-02-21T08:11:06Z'
```

- 允许容许污点的放置上的容限

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement1
  namespace: default
spec:
  tolerations:
    - key: gpu
      value: "true"
      operator: Equal
```

在定义了容限示例后，放置可以选择 **cluster1**，因为 **key: gpu** 和 **value: "true"** 匹配。

注： 受管集群不能保证放在包含污点容限的放置上。如果其他放置包含相同的容限，受管集群可能会放置到其中一个放置上。

1.14.6.4. 指定临时容限

TolerationSeconds 的值指定容限容许污点的时间期限。当受管集群离线时，这个临时容限会很有用，您可以将在此集群中部署的应用程序传送到另一个受管集群中，以便容忍的时间。

例如，具有以下污点的受管集群将无法访问：

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
spec:
  hubAcceptsClient: true
  taints:
    - effect: NoSelect
      key: cluster.open-cluster-management.io/unreachable
      timeAdded: '2022-02-21T08:11:06Z'
```

如果您使用 **TolerationSeconds** 的值定义放置，如下例所示，工作负载在 5 分钟后传输到另一个可用的受管集群。

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: Placement
metadata:
  name: demo4
  namespace: demo1
spec:
  tolerations:
    - key: cluster.open-cluster-management.io/unreachable
      operator: Exists
      tolerationSeconds: 300
----
```

受管集群在无法访问 5 分钟后，应用程序会被移到另一个受管集群。

1.15. 管理集群池（技术预览）

集群池提供对按需和规模配置的 Red Hat OpenShift Container Platform 集群的快速、经济的访问。集群

池在 Amazon Web Services、Google Cloud Platform 或 Microsoft Azure 上置备可配置且可扩展的 OpenShift Container Platform 集群，在需要时可以声明。在为开发、持续集成和生产环境提供或替换集群环境时，它们特别有用。您可以指定多个集群来保持运行，以便可以立即声明它们，而集群的剩余部分将保留在休眠状态，以便在几分钟后恢复并声明。

ClusterClaim 资源用于从集群池中签出集群。创建集群声明时，池会为它分配一个正在运行的集群。如果没有正在运行的集群可用，则会恢复休眠集群来提供集群或新的集群。集群池自动创建新集群，并恢复休眠集群来维护池中的指定大小和可用集群的数量。

注： 当从集群池中声明的集群不再需要并销毁时，资源会被删除。集群不会返回到集群池。

需要的访问权限：Administrator

- 先决条件
- 创建集群池
- 从集群池中声明集群
- 扩展集群池
- 更新集群池发行镜像
- 销毁一个集群池

创建集群池的过程与创建集群的步骤类似。在集群池中创建的集群不会供立即使用。

1.15.1. 先决条件

在创建集群池前，请查看以下先决条件：

- 您需要部署一个 Red Hat Advanced Cluster Management for Kubernetes hub 集群。
- 您需要互联网访问 Red Hat Advanced Cluster Management for Kubernetes hub 集群，以便它在供应商环境中创建 Kubernetes 集群。
- 您需要一个 AWS、GCP 或 Microsoft Azure 供应商凭证。如需更多信息，请参阅 管理凭证概述。
- 您需要供应商环境中配置的域。有关如何配置域的说明，请参阅您的供应商文档。
- 您需要供应商登录凭证。
- 您需要 OpenShift Container Platform 镜像 pull secret。请参阅使用镜像 pull secret。

注： 使用这个流程添加集群池，以便在从池中声明集群时自动导入由 Red Hat Advanced Cluster Management 管理的集群。如果要创建一个没有自动导入使用集群声明的集群池，请将以下注解添加到 **clusterClaim** 资源中：

```
kind: ClusterClaim
metadata:
  annotations:
    cluster.open-cluster-management.io/createmanagedcluster: "false"
```

"false" 必须用引号括起来，表示它是一个字符串。

1.15.2. 创建集群池

要创建集群池，在导航菜单中选择 **Infrastructure** > **Clusters**。*Cluster pool* 选项卡列出您可以访问的集群池。选择 **Create cluster pool** 并完成控制台中的步骤。

如果您没有要用于集群池的基础架构凭证，可以通过选择 **Add credential** 来创建一个。

您可以从列表中选择现有命名空间，或者键入要创建新命名空间的名称。集群池不必与集群位于同一个命名空间中。

当您在集群集合中创建集群池时，**namespace admin** 权限将应用到您添加集群池的命名空间中具有 **clusterset admin** 权限的所有用户。同样，**namespace view** 权限也会应用到具有 **clusterset view** 权限的用户。

如果您希望集群池的 RBAC 角色共享现有集群集合的角色分配，请选择集群设置名称。只有创建集群池时，才能设置集群池中的集群设置。您无法在创建集群池后更改集群池或集群池中集群集的关联。从集群池中声明的任何集群都会自动添加到与集群池相同的集群集合中。

注：如果没有 **cluster admin** 权限，则必须选择一个集群集。如果您在此情形中没有包含集群集的名称，则创建集群集的请求将被拒绝，并带有禁止的错误。如果没有集群集可供选择，请联络您的集群管理员来创建集群集，并为您提供 **clusterset admin** 权限。

集群池大小指定您在集群池中置备的集群数量，而集群池运行计数指定池运行的集群数量并准备好立即使用。

此过程与创建集群的步骤非常相似。

有关您的供应商所需信息的详情，请查看以下信息：

- 在 Amazon Web Services 上创建集群
- 在 Google Cloud Platform 上创建集群
- 在 Microsoft Azure 上创建集群

1.15.3. 从集群池中声明集群

ClusterClaim 资源用于从集群池中签出集群。当集群正在运行并位于集群池中时，会出现一个声明。集群池自动在集群池中创建新运行和休眠集群，以维护为集群池指定的要求。

注： 当从集群池中声明的集群不再需要并销毁时，资源会被删除。集群不会返回到集群池。

需要的访问权限：Administrator

1.15.3.1. 前提条件

在从集群池中声明集群前，您必须有以下可用：

具有或没有可用集群的集群池。如果集群池中存在可用的集群，则会声明可用的集群。如果集群池中没有可用的集群，则会创建一个集群来满足这个声明。如需有关如何创建集群池的信息，请参阅创建集群池。

1.15.3.2. 从集群池中声明集群

在创建集群声明时，您可以从集群池请求新集群。当集群可用时，从池中签出集群。声明的集群自动导入为其中一个受管集群，除非您禁用了自动导入。

完成以下步骤以声明集群：

1. 在导航菜单中点 **Infrastructure** > **Clusters**，然后选择 *Cluster pool* 选项卡。
2. 从中查找您要声明集群的集群池的名称，然后选择 **Claim cluster**。

如果集群可用，它将被声明并立即出现在 *Managed cluster* 选项卡中。如果没有可用的集群，可能需要几分钟时间来恢复休眠集群或置备新集群。在这个时间中，声明状态为 **pending**。扩展集群池，以查看或删除待处理的声明。

当它在集群池中，声明的集群会保留作为与它关联的集群集的成员。在声明集群时，您无法更改声明的集群集合。

1.15.4. 扩展集群池

您可以通过增加或减少集群池大小中的集群数量来更改集群池中的集群数量。

需要的访问权限： 集群管理员

完成以下步骤以更改集群池中的集群数量：

1. 在导航菜单中点 **Infrastructure** > **Clusters**。
2. 选择 *Cluster pools* 选项卡。
3. 在您要更改的集群池的 *Options* 菜单中，选择 **Scale cluster pool**。
4. 更改池大小的值。
5. 另外，您还可以更新正在运行的集群数量，以便在声明它们时立即可用的集群数量。

集群池已扩展，以反映您的新值。

1.15.5. 更新集群池发行镜像

当集群中的集群保留在休眠状态一段时间时，集群的 Red Hat OpenShift Container Platform 发行镜像可能会变为 backlevel。如果发生这种情况，您可以升级集群池中集群的发行镜像版本。

需要的访问权限 ： Edit

完成以下步骤，为集群池中的集群更新 OpenShift Container Platform 发行镜像：

注： 此步骤不会从集群池中已声明的集群池中更新集群。完成此步骤后，对发行镜像的更新仅适用于与集群池相关的以下集群：

- 使用此流程更新发行镜像后，由集群池创建的集群。
- 在集群池中休眠的集群。使用旧发行镜像的现有休眠集群将被销毁，新集群使用新的发行镜像替换它们。

1. 在导航菜单中点 **Infrastructure** > **Clusters**。
2. 选择 *Cluster pools* 选项卡。
3. 在 *Cluster pool* 表中找到您要更新的集群池的名称。
4. 点表中的 *Cluster pools* 的 *Options* 菜单，然后选择 **Update release image**。

5. 从这个集群池中选择一个新的发行镜像，用于将来的集群创建。

集群池发行镜像已更新。

提示：您可以通过选择一个操作来更新多个集群池的发行镜像，方法是选择每个集群池的方框并使用 *Actions* 菜单来更新所选集群池的发行镜像。

1.15.6. 销毁一个集群池

如果您创建了集群池，并确定您不再需要它，您可以销毁集群池。当您销毁集群池时，所有未声明的休眠集群都会被销毁，并释放其资源。

需要的访问权限： 集群管理员

要销毁集群池，请完成以下步骤：

1. 在导航菜单中点 **Infrastructure** > **Clusters**。
2. 选择 *Cluster pools* 选项卡。
3. 在您要删除的集群池的 *Options* 菜单中，选择 **Destroy cluster pool**。集群池中的任何未声明的集群都会被销毁。可能需要一些时间才能删除所有资源，集群池会在控制台中可见，直到所有资源都被删除为止。
   包含 ClusterPool 的命名空间不会被删除。删除命名空间将销毁 ClusterPool 声明的任何集群，因为这些集群的 ClusterClaim 资源会在同一命名空间中创建。

提示：您可以通过一个操作来销毁多个集群池，只需选择每个集群池的复选框，并使用 *Actions* 菜单销毁所选集群池。

1.16. CLUSTERCLAIMS

ClusterClaim 是受管集群中的一个集群范围的自定义资源定义（CRD）。ClusterClaim 代表受管集群声明的一个信息片段。以下示例显示了 YAML 文件中标识的声明：

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ClusterClaim
metadata:
  name: id.openshift.io
spec:
  value: 95f91f25-d7a2-4fc3-9237-2ef633d8451c
```

下表显示了 Red Hat Advanced Cluster Management for Kubernetes 管理的集群中可能会定义的 ClusterClaims：

声明名称	保留	可变	描述
id.k8s.io	true	false	在上游社区定义的 ClusterID
kubeversion.open-cluster-management.io	true	true	Kubernetes 版本

声明名称	保留	可变	描述
platform.open-cluster-management.io	true	false	运行受管集群的平台，如 AWS、GCE 和 Equinix Metal
product.open-cluster-management.io	true	false	产品名称，如 OpenShift、Anthos、EKS 和 GKE
id.openshift.io	false	false	OpenShift Container Platform 外部 ID，它仅适用于 OpenShift Container Platform 集群
consoleurl.openshift.io	false	true	管理控制台的 URL，仅适用于 OpenShift Container Platform 集群
version.openshift.io	false	true	OpenShift Container Platform 版本，它仅适用于 OpenShift Container Platform 集群

如果在受管集群中删除或更新之前的声明，它们会自动恢复或回滚到上一版本。

在受管集群加入 hub 后，在受管集群上创建的 ClusterClaims 与 hub 上的 **ManagedCluster** 资源的状态同步。带有 ClusterClaims 的受管集群可能类似以下示例：

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  labels:
    cloud: Amazon
    clusterID: 95f91f25-d7a2-4fc3-9237-2ef633d8451c
    installer.name: multiclusterhub
    installer.namespace: open-cluster-management
    name: cluster1
    vendor: OpenShift
  name: cluster1
spec:
  hubAcceptsClient: true
  leaseDurationSeconds: 60
status:
  allocatable:
    cpu: '15'
    memory: 65257Mi
  capacity:
    cpu: '18'
    memory: 72001Mi
  clusterClaims:
```

```
    - name: id.k8s.io
      value: cluster1
    - name: kubeversion.open-cluster-management.io
      value: v1.18.3+6c42de8
    - name: platform.open-cluster-management.io
      value: AWS
    - name: product.open-cluster-management.io
      value: OpenShift
    - name: id.openshift.io
      value: 95f91f25-d7a2-4fc3-9237-2ef633d8451c
    - name: consoleurl.openshift.io
      value: 'https://console-openshift-console.apps.xxxx.dev04.red-chesterfield.com'
    - name: version.openshift.io
      value: '4.5'
  conditions:
    - lastTransitionTime: '2020-10-26T07:08:49Z'
      message: Accepted by hub cluster admin
      reason: HubClusterAdminAccepted
      status: 'True'
      type: HubAcceptedManagedCluster
    - lastTransitionTime: '2020-10-26T07:09:18Z'
      message: Managed cluster joined
      reason: ManagedClusterJoined
      status: 'True'
      type: ManagedClusterJoined
    - lastTransitionTime: '2020-10-30T07:20:20Z'
      message: Managed cluster is available
      reason: ManagedClusterAvailable
      status: 'True'
      type: ManagedClusterConditionAvailable
  version:
    kubernetes: v1.18.3+6c42de8
```

1.16.1. 列出现有 ClusterClaims

您可以使用 **kubectl** 命令列出应用到受管集群的 ClusterClaims。当您要将 ClusterClaim 与错误消息进行比较时，这很有用。

备注：您需要具有 **clusterclaims.cluster.open-cluster-management.io** 资源的 **list** 权限。

运行以下命令列出受管集群中的所有现有 ClusterClaims：

```
kubectl get clusterclaims.cluster.open-cluster-management.io
```

1.16.2. 创建自定义 ClusterClaims

您可以使用受管集群上的自定义名称创建 ClusterClaims，这样可更轻松地识别它们。自定义 ClusterClaims 会与 hub 集群上的 **ManagedCluster** 资源进行同步。以下内容显示了自定义 **ClusterClaim** 的定义示例：

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ClusterClaim
metadata:
```

```
  name: <custom_claim_name>
spec:
  value: <custom_claim_value>
```

字段 **spec.value** 的最大长度为 1024。创建 ClusterClaim 需要资源 **clusterclaims.cluster.open-cluster-management.io** 的 **create** 权限。

1.17. 使用托管的 CONTROL PLANE 集群（技术预览）

带有 multicluster engine operator 2.0 的 Red Hat Advanced Cluster Management for Kubernetes 版本 2.5 可以使用两个不同的 control plane 配置来部署 Red Hat OpenShift Container Platform 集群。独立配置使用多个专用虚拟机或物理机器来托管 OpenShift Container Platform control plane。您可以置备托管的 control plane，将 OpenShift Container Platform control plane 置备为托管服务集群中的 pod，而无需为每个 control-plane 专用物理机器。

注： 此功能还可在没有 Red Hat Advanced Cluster Management for Kubernetes 的情况下与多集群引擎 operator 2.0 一同使用。

对于 Red Hat Advanced Cluster Management，Amazon Web Services 作为技术预览提供。您可以托管 Red Hat OpenShift Container Platform 版本 4.10.7 及更新版本的 control plane。

control plane 作为单一命名空间中包含的 pod 运行，并与托管的 control plane 集群关联。当 OpenShift Container Platform 置备这种类型的托管集群时，它会置备一个独立于 control plane 的 worker 节点。

查看托管 control plane 集群的以下优点：

- 通过删除对专用 control plane 节点的需求来降低成本
- 引入 control plane 和工作负载的隔离，改进了隔离并减少可能需要更改的配置错误
- 通过删除 control-plane 节点 bootstrap 的要求来显著减少集群置备时间
- 支持 turn-key 部署或完全自定义的 OpenShift Container Platform 置备

在以下产品文档中，请参阅使用托管的 control plane 的更多信息：

- 配置托管的 control plane
- 禁用托管的 control plane 资源

1.17.1. 配置托管的 control plane

配置托管的 control plane 需要托管服务集群和一个托管的集群。通过在现有集群中部署 HyperShift Operator，您可以让该集群部署到托管服务集群中，并开始创建托管集群。

托管 control plane 是一个技术预览功能，因此相关组件默认是禁用的。通过编辑 **multiclusterengine** 自定义资源来启用该功能，将 **spec.overrides.components[?(@.name=='hypershift-preview')]** 设置为 **true**。

输入以下命令来确保启用托管 control planes 功能：

```
oc patch mce multiclusterengine-sample--type=merge -p '{"spec":{"overrides":{"components":
[{"name":"hypershift-preview","enabled": true}]}}}'
```

1.17.1.1. 配置托管服务集群

您可以通过将现有集群配置为充当托管服务集群来部署托管 control plane。托管的服务集群是托管 control plane 的 OpenShift Container Platform 集群，可以是 hub 集群或一个 OpenShift Container Platform 受管集群。

1.17.1.1.1. 先决条件

您必须有以下先决条件才能配置托管服务集群：

- 至少一个由 Red Hat OpenShift Container Platform 管理的集群上安装 multicluster engine operator。在安装 Red Hat Advanced Cluster Management 版本 2.5 及更新的版本时，多集群引擎 Operator 会被自动安装，也可以在没有 Red Hat Advanced Cluster Management 作为 OpenShift Container Platform OperatorHub 中的 Operator 的情况下安装。
- 如果您希望 Red Hat Advanced Cluster Management hub 集群作为托管服务集群，则必须通过完成以下步骤将 **local-cluster** 配置为您的托管服务集群：

  1. 创建名为 **import-hub.yaml** 的 YAML 文件，类似以下示例：

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

  2. 使用以下命令应用该文件：

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

  由自身管理的 hub 集群在集群列表中被指定为 **local-cluster**。

1.17.1.1.2. 配置托管服务集群

在安装 multicluster engine operator 的集群上完成以下步骤，将 OpenShift Container Platform 受管集群启用为托管服务集群：

1. 如果您计划在 AWS 上创建和管理托管集群，请为 HyperShift operator 创建一个名为 **hypershift-operator-oidc-provider-s3-credentials** 的 OIDC S3 凭证 secret。将 secret 保存到受管集群命名空间中，这是用作托管服务集群的受管集群的命名空间。如果您使用 **local-cluster**，请在 **local-cluster** 命名空间中创建 secret
   secret 必须包含 3 个字段。**bucket** 字段包含一个 S3 存储桶，它可以访问您的 HyperShift 集群的主机 OIDC 发现文档。**credentials** 字段是对文件的引用，其中包含可访问存储桶的**默认**配置集凭证。默认情况下，HyperShift 仅使用 **default** 配置集来运行 **bucket**。**region** 字段指定 S3 存储桶的区域。

   有关 secret 的更多信息，请参阅 HyperShift 文档中的 Getting started。以下示例显示了 AWS secret 模板示例：

   ```
   oc create secret generic hypershift-operator-oidc-provider-s3-credentials --from-file=credentials=$HOME/.aws/credentials --from-literal=bucket=<s3-bucket-for-hypershift> --from-literal=region=<region> -n <hypershift-hosting-service-cluster>
   ```

注：不会自动启用 secret 的恢复备份。运行以下命令添加启用 **hypershift-operator-oidc-provider-s3-credentials** secret 的标签，以便为灾难恢复进行备份：

```
oc label secret hypershift-operator-oidc-provider-s3-credentials -n <hypershift-hosting-service-cluster> cluster.open-cluster-management.io/backup=""
```

2. 安装 HyperShift 附加组件。
托管 HyperShift 操作器的集群是托管服务集群。此步骤使用 **hypershift-addon** 在受管集群上安装 HyperShift operator。

   a. 通过创建一个类似以下示例的文件来创建 **ManagedClusterAddon** HyperShift 附加组件：

   ```
   apiVersion: addon.open-cluster-management.io/v1alpha1
   kind: ManagedClusterAddOn
   metadata:
     name: hypershift-addon
     namespace: <managed-cluster-name>
   spec:
     installNamespace: open-cluster-management-agent-addon
   ```

   将 **managed-cluster-name** 替换为您要在其中安装 HyperShift 命名空间的受管集群的名称。如果要在 Red Hat Advanced Cluster Management hub 集群上安装，请使用 **local-cluster** 作为这个值。

   b. 运行以下命令来应用该文件：

   ```
   oc apply -f <filename>
   ```

   使用您创建的文件的名称替换 **filename**。

3. 运行以下命令确认已安装 **hypershift-addon**:

   ```
   oc get managedclusteraddons -n <hypershift-hosting-service-cluster> hypershift-addon
   ```

   当安装了附加组件时，输出类似以下示例：

   ```
   NAME              AVAILABLE   DEGRADED   PROGRESSING
   hypershift-addon   True
   ```

您的 HyperShift 附加组件已经安装，且托管服务集群可用于管理 HyperShift 集群。

1.17.1.2. 部署托管集群

安装 HyperShift operator 并将现有集群启用为托管服务集群后，您可以通过创建一个 **HypershiftDeployment** 自定义资源来置备 HyperShift 托管的集群。

1. 使用控制台或文件添加将云供应商 secret 创建为凭证。您必须具有为集群创建基础架构资源的权限，如 VPC、子网和 NAT 网关。帐户也必须与您的客户机集群的帐户对应，其中您的 worker 处于活动状态。有关所需权限的更多信息，请参阅 HyperShift 文档中的创建 AWS 基础架构和 IAM 资源。
   以下示例显示了 AWS 的格式：

   ```
   apiVersion: v1
   metadata:
   ```

```
  name: my-aws-cred
  namespace: default      # Where you create HypershiftDeployment resources
type: Opaque
kind: Secret
stringData:
  ssh-publickey:         # Value
  ssh-privatekey:        # Value
  pullSecret:            # Value, required
  baseDomain:            # Value, required
  aws_secret_access_key:  # Value, required
  aws_access_key_id:      # Value, required
```

- 要使用控制台创建此 secret，请通过访问导航菜单中的 **Credentials** 来创建步骤。
- 要使用命令行创建 secret，请运行以下命令：

```
oc create secret generic <my-secret> -n <hypershift-deployment-namespace> --from-literal=baseDomain='your.domain.com' --from-literal=aws_access_key_id='your-aws-access-key' --from-literal=aws_secret_access_key='your-aws-secret-key' --from-literal=pullSecret='your-quay-pull-secret' --from-literal=ssh-publickey='your-ssh-publickey' --from-literal=ssh-privatekey='your-ssh-privatekey'
```

注：不会自动启用 secret 的恢复备份。运行以下命令来添加可备份 secret 的标签：

```
oc label secret <my-secret> -n <hypershift-deployment-namespace> cluster.open-cluster-management.io/backup=""
```

2. 在云供应商 secret 命名空间中创建 **HypershiftDeployment** 自定义资源文件。**HypershiftDeployment** 自定义资源在供应商帐户中创建基础架构，配置创建的基础架构计算容量，置备使用托管控制平面的 **nodePools**，并在托管服务集群中创建托管控制平面。

   a. 创建包含类似以下示例的信息的文件：

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: HypershiftDeployment
metadata:
  name: <cluster>
  namespace: default
spec:
  hostingCluster: <hosting-service-cluster>
  hostingNamespace: clusters
  hostedClusterSpec:
    networking:
      machineCIDR: 10.0.0.0/16    # Default
      networkType: OpenShiftSDN
      podCIDR: 10.132.0.0/14      # Default
      serviceCIDR: 172.31.0.0/16  # Default
    platform:
      type: AWS
    pullSecret:
      name: <cluster>-pull-secret    # This secret is created by the controller
    release:
      image: quay.io/openshift-release-dev/ocp-release:4.10.15-x86_64  # Default
    services:
    - service: APIServer
```

```
    servicePublishingStrategy:
      type: LoadBalancer
  - service: OAuthServer
    servicePublishingStrategy:
      type: Route
  - service: Konnectivity
    servicePublishingStrategy:
      type: Route
  - service: Ignition
    servicePublishingStrategy:
      type: Route
  sshKey: {}
nodePools:
- name: <cluster>
  spec:
    clusterName: <cluster>
    management:
      autoRepair: false
      replace:
        rollingUpdate:
          maxSurge: 1
          maxUnavailable: 0
        strategy: RollingUpdate
      upgradeType: Replace
    platform:
      aws:
        instanceType: m5.large
      type: AWS
    release:
      image: quay.io/openshift-release-dev/ocp-release:4.10.15-x86_64 # Default
    replicas: 2
infrastructure:
  cloudProvider:
    name: <my-secret>
  configure: True
  platform:
    aws:
      region: <region>
```

将 **cluster** 替换为集群的名称。

将 **hosting-service-cluster** 替换为托管 HyperShift operator 的集群的名称。

将 **my-secret** 替换为用于访问您的云供应商的 secret。

将 **region** 替换为云供应商的区域。

b. 输入以下命令应用该文件：

```
oc apply -f <filename>
```

您可以参考 API 的字段定义，以确保它们正确。

3. 运行以下命令，检查 **HypershiftDeployment** 状态：

```
oc get hypershiftdeployment -n default hypershift-demo -w
```

4. 在创建了托管集群后，它会自动导入到 hub。您可以在 Red Hat Advanced Cluster Management 控制台中查看集群列表或运行以下命令来验证它：

```
oc get managedcluster <hypershiftDeployment.Spec.infraID>
```

1.17.1.3. 访问托管服务集群

现在，您可以访问集群。访问 secret 存储在 **hypershift-hosting-service-cluster** 命名空间中。此命名空间与托管服务集群的名称相同。了解以下格式：

- **kubeconfig** secret: **<hypershiftDeployment.Spec.hostingNamespace>-<hypershiftDeployment.Name>-admin-kubeconfig** (clusters-hypershift-demo-admin-kubeconfig)
- **kubeadmin** password secret: **<hypershiftDeployment.Spec.hostingNamespace>-<hypershiftDeployment.Name>-kubeadmin-password** (clusters-hypershift-demo-kubeadmin-password)

1.17.2. 禁用托管的 control plane 资源

禁用托管的 control plane 集群功能时，您必须销毁 HyperShift 受管集群并卸载 HyperShift Operator。

1.17.2.1. 销毁 HyperShift 托管的集群

要销毁 HyperShift 托管集群，请运行以下命令删除 **HypershiftDeployment** 资源：

```
oc delete -f <HypershiftDeployment_yaml_file_name>
```

或者

```
oc delete hd -n <HypershiftDeployment_namespace> <HypershiftDeployment_resource_name>
```

1.17.2.2. 卸载 HyperShift Operator

要从管理或托管服务集群中卸载 HyperShift Operator，请运行以下命令从管理集群中删除 **hypershift-addon ManagedClusterAddon** ：

```
oc delete managedclusteraddon -n <hypershift-management-cluster> hypershift-addon
```

1.18. 发现服务简介

您可以发现 OpenShift Cluster Manager 可用的 OpenShift 4 集群。发现后，您可以导入集群进行管理。发现服务使用 Discover Operator 进行后端和控制台使用。

您必须具有 OpenShift Cluster Manager 凭据。如果需要创建凭证，请参阅为 Red Hat OpenShift Cluster Manager 创建凭证。

需要的访问权限：Administrator

- 使用控制台配置发现
- 使用 CLI 配置发现

1.18.1. 使用控制台配置发现

使用产品控制台启用发现。

需要的访问权限：访问创建凭证的命名空间。

1.18.1.1. 先决条件

- 您需要一个凭证。请参阅为 Red Hat OpenShift Cluster Manager 创建凭证 以连接到 OpenShift Cluster Manager。

1.18.1.2. 配置发现

在控制台中配置 Discovery 以查找集群。您可以使用单独的凭证创建多个 **DiscoveryConfig** 资源。按照控制台中的说明操作。

1.18.1.3. 查看发现的集群

在设置凭证并发现集群以导入后，您可以在控制台中查看它们。

1. 点 **Clusters** > **Discovered cluster**
2. 使用以下信息查看填充的表：
   - *Name* 是 OpenShift Cluster Manager 中指定的显示名称。如果集群没有显示名称，则会显示基于集群控制台 URL 生成的名称。如果 OpenShift Cluster Manager 缺少控制台URL，或手动修改了控制台 URL，则会显示集群外部 ID。
   - *Namespace* 是您创建凭证和发现集群的命名空间。
   - *Type* 是发现的集群 Red Hat OpenShift 类型。
   - *Distribution version* 是发现的集群 Red Hat OpenShift 版本。
   - *基础架构供应商*是已发现集群的云供应商。
   - *最后活跃*是发现的集群最后一次活跃的时间。
   - 当发现的集群被创建时为 *Created* 。
   - 当发现的集群被发现时为 *Discovered*。
3. 您还可以搜索表中的任何信息。例如，要只显示特定命名空间中的*发现集群*，请搜索该命名空间。
4. 现在，您可以点 **Import cluster** 创建受管集群。请参阅导入发现的集群。

1.18.1.4. 导入发现的集群

发现集群后，您可以导入控制台的 *Discovered clusters* 选项卡中出现的集群。

1.18.1.5. 先决条件

您需要访问用于配置 Discovery 的命名空间。

1.18.1.6. 导入发现的集群

1. 进入到现有 *Clusters* 页面并点 *Discovered clusters* 选项卡。
2. 在*发现的集群*表中找到您要导入的集群。
3. 在选项菜单中选择 **Import cluster**。
4. 对于发现的集群，您可以使用文档手动导入，或者您可以自动选择导入集群。
5. 要使用凭证或 Kubeconfig 文件自动导入，请复制并粘贴内容。
6. 点 **Import**。

1.18.2. 使用 CLI 启用发现

使用 CLI 启用发现以查找 Red Hat OpenShift Cluster Manager 可用的集群。

需要的访问权限：Administrator

1.18.2.1. 先决条件

- 创建用于连接到 Red Hat OpenShift Cluster Manager 的凭证。

1.18.2.2. 发现设置和进程

注： DiscoveryConfig 必须命名为 **discovery**，且必须与所选**凭证**在同一命名空间中创建。请参见以下 **DiscoveryConfig** 示例：

```
apiVersion: discovery.open-cluster-management.io/v1
kind: DiscoveryConfig
metadata:
  name: discovery
  namespace: <NAMESPACE_NAME>
spec:
  credential: <SECRET_NAME>
  filters:
    lastActive: 7
    openshiftVersions:
    - "4.10"
    - "4.9"
    - "4.8"
```

1. 将 **SECRET_NAME** 替换为之前设置的凭证。
2. 将 **NAMESPACE_NAME** 替换为 **SECRET_NAME** 的命名空间。
3. 输入集群最后一次活动（以天为单位）进行发现的最大时间。例如，带有 **lastActive: 7** 的集群，最后 7 天内活跃的集群会被发现。
4. 输入要作为字符串列表发现的 Red Hat OpenShift 集群的版本。**注： openshiftVersions** 列表中的每个条目都指定了一个 OpenShift 主版本和次版本。例如，指定 **"4.9"** 将包括 OpenShift 版本 **4.9** 的所有补丁版本，如 **4.9.1**、**4.9.2**。

1.18.2.3. 查看发现的集群

通过运行 **oc get discoveredclusters -n <namespace>**（其中 **namespace** 是发现凭证存在的命名空间）来查看发现的集群。

1.18.2.3.1. DiscoveredClusters

对象由 Discovery 控制器创建。这些 **DiscoveredClusters** 使用在 **DiscoveryConfig** **discoveredclusters.discovery.open-cluster-management.io** API 中指定的过滤器和凭证来代表 OpenShift Cluster Manager 中找到的集群。**name** 的值是集群外部 ID：

```
apiVersion: discovery.open-cluster-management.io/v1
kind: DiscoveredCluster
metadata:
  name: fd51aafa-95a8-41f7-a992-6fb95eed3c8e
  namespace: <NAMESPACE_NAME>
spec:
  activity_timestamp: "2021-04-19T21:06:14Z"
  cloudProvider: vsphere
  console: https://console-openshift-console.apps.qe1-vmware-pkt.dev02.red-chesterfield.com
  creation_timestamp: "2021-04-19T16:29:53Z"
  credential:
    apiVersion: v1
    kind: Secret
    name: <SECRET_NAME>
    namespace: <NAMESPACE_NAME>
  display_name: qe1-vmware-pkt.dev02.red-chesterfield.com
  name: fd51aafa-95a8-41f7-a992-6fb95eed3c8e
  openshiftVersion: 4.10
  status: Stale
```

1.19. 升级集群

创建要通过 Red Hat Advanced Cluster Management for Kubernetes 管理的 Red Hat OpenShift Container Platform 集群后，您可以使用 Red Hat Advanced Cluster Management for Kubernetes 控制台将这些集群升级到受管集群使用的版本频道中可用的最新次版本。

在连接的环境中，会在 Red Hat Advanced Cluster Management 控制台中为需要升级的每个集群自动识别更新。

重要： 在断开连接的环境中升级集群的过程需要一些额外的步骤来配置和镜像所需的发行镜像。它使用 Red Hat OpenShift Update Service 的 operator 来识别升级。如果您位于断开连接的环境中，请参阅 升级断开连接的集群以了解所需步骤。

备注：

要升级到一个主要版本，您必须确定是否满足升级到该版本的所有先决条件。在可以使用控制台升级集群前，您必须更新受管集群上的版本频道。

更新受管集群上的版本频道后，Red Hat Advanced Cluster Management for Kubernetes 控制台会显示可用于升级的最新版本。

此升级方法只适用于处于 *Ready* 状态的 OpenShift Container Platform 受管集群。

重要：您无法使用 Red Hat Advanced Cluster Management for Kubernetes 控制台在 Red Hat OpenShift Dedicated 上升级 Red Hat OpenShift Kubernetes Service 受管集群或 OpenShift Container Platform 受管集群。

要在连接的环境中升级集群，请完成以下步骤：

1. 通过导航菜单进入 **Infrastructure** > **Clusters**。如果有可用的升级，会在 *Distribution version* 列中显示。
2. 选择您要升级的 *Ready* 状态的集群。集群必须是 OpenShift Container Platform 集群才能使用控制台升级。
3. 选择 **Upgrade**。
4. 选择每个集群的新版本。
5. 选择 **Upgrade**。

如果集群升级失败，Operator 通常会重试升级，停止并报告故障组件的状态。在某些情况下，升级过程会一直通过尝试完成此过程进行循环。不支持在失败的升级后将集群还原到以前的版本。如果您的集群升级失败，请联系红帽支持以寻求帮助。

1.19.1. 选择一个频道

您可以使用 Red Hat Advanced Cluster Management 控制台为 OpenShift Container Platform 版本 4.6 或更高版本上的集群升级选择一个频道。选择频道后，会自动提醒两个勘误版本可用的集群升级（4.8.1 > 4.8.2 > 4.8.3 等）和发行版本（4.8 > 4.9 等）。

要为集群选择频道，请完成以下步骤：

1. 在 Red Hat Advanced Cluster Management 导航中选择 **Infrastructure** > **Clusters**。
2. 选择要更改的集群名称来查看 *Cluster details* 页面。如果集群有一个不同的频道，则 *Channel* 字段中会显示一个编辑图标。
3. 点编辑图标，以修改字段中的设置。
4. 在 *New channel* 字段中选择一个频道。

您可以在集群的 *Cluster details* 页中找到有关可用频道更新的提示信息。

1.19.2. 升级断开连接的集群

您可以将 Red Hat OpenShift Update Service 与 Red Hat Advanced Cluster Management for Kubernetes 搭配使用，以便在断开连接的环境中升级您的集群。

在某些情况下，安全性考虑会阻止集群直接连接到互联网。这使得您很难知道什么时候可以使用升级，以及如何处理这些升级。配置 OpenShift Update Service 可能会有所帮助。

OpenShift Update Service 是一个独立的操作对象，它监控受管集群在断开连接的环境中的可用版本，并使其可用于在断开连接的环境中升级集群。配置 OpenShift Update Service 后，它可以执行以下操作：

1. 监测何时有适用于断开连接的集群的升级。
2. 使用图形数据文件识别哪些更新需要被镜像到您的本地站点进行升级。
3. 使用 Red Hat Advanced Cluster Management 控制台通知您的集群可以使用升级。

- 先决条件
- 准备断开连接的镜像 registry

- 为 OpenShift Update Service 部署 Operator
- 构建图形数据 init 容器
- 为已镜像的 registry 配置证书
- 部署 OpenShift Update Service 实例
- 部署策略以覆盖默认 registry（可选）
- 部署策略来部署断开连接的目录源
- 部署策略以更改受管集群参数
- 查看可用升级
- 选择一个频道
- 升级集群

1.19.2.1. 先决条件

您必须满足以下先决条件，才能使用 OpenShift Update Service 升级断开连接的集群：

- 已部署了一个在 Red Hat OpenShift Container Platform 版本 4.6 或更高版本上运行的 Red Hat Advanced Cluster Management hub 集群，并配置了受限 OLM。如需了解如何配置受限 OLM 的详细信息，请参阅在受限网络中使用 Operator Lifecycle Manager。
  提示： 配置受限 OLM 时请记录目录源镜像。
- 由 Red Hat Advanced Cluster Management hub 集群管理的 OpenShift Container Platform 集群
- 访问您可以镜像集群镜像的本地存储库的凭证。如需有关如何创建此软件仓库的更多信息，请参阅断开连接的安装镜像。
  注： 您升级的集群当前版本的镜像必须始终作为镜像的一个镜像可用。如果升级失败，集群会在试图升级时恢复到集群的版本。

1.19.2.2. 准备断开连接的镜像 registry

您必须镜像要升级到的镜像，以及您要从本地镜像 registry 升级到的当前镜像。完成以下步骤以镜像镜像：

1. 创建一个包含类似以下示例内容的脚本文件：

```
UPSTREAM_REGISTRY=quay.io
PRODUCT_REPO=openshift-release-dev
RELEASE_NAME=ocp-release
OCP_RELEASE=4.5.2-x86_64
LOCAL_REGISTRY=$(hostname):5000
LOCAL_SECRET_JSON=/path/to/pull/secret

oc adm -a ${LOCAL_SECRET_JSON} release mirror \
--
from=${UPSTREAM_REGISTRY}/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE} \
--to=${LOCAL_REGISTRY}/ocp4 \
--to-release-image=${LOCAL_REGISTRY}/ocp4/release:${OCP_RELEASE}
```

将 **/path/to/pull/secret** 替换为 OpenShift Container Platform pull secret 的路径。

2. 运行该脚本来对镜像进行镜像、配置设置并将发行镜像与发行内容分开。

提示： 在创建 **ImageContentSourcePolicy** 时，您可以使用此脚本最后一行的输出。

1.19.2.3. 为 OpenShift Update Service 部署 Operator

要在 OpenShift Container Platform 环境中为 OpenShift Update Service 部署 Operator，请完成以下步骤：

1. 在 hub 集群中，访问 OpenShift Container Platform operator hub。
2. 选择 **Red Hat OpenShift Update Service Operator** 来部署 Operator。如果需要，更新默认值。Operator 的部署会创建一个名为 **openshift-cincinnati** 的新项目。
3. 等待 Operator 的安装完成。
   提示： 您可以通过在 OpenShift Container Platform 命令行中输入 **oc get pods** 命令来检查安装的状态。验证 Operator 是否处于 **running** 状态。

1.19.2.4. 构建图形数据 init 容器

OpenShift Update Service 使用图形数据信息来决定可用的升级。在连接的环境中，OpenShift Update Service 会直接从 Cincinnati 图形数据GitHub 仓库中 提取可用于升级的图形数据信息。由于要配置断开连接的环境，所以必须使用 **init 容器**使图形数据在本地存储库中可用。完成以下步骤以创建图形数据 **init 容器**:

1. 输入以下命令克隆 *graph data* Git 存储库：

```
git clone https://github.com/openshift/cincinnati-graph-data
```

2. 创建一个包含您的图形数据 **init** 信息的文件。您可以在 **cincinnati-operator GitHub** 仓库中找到此 Dockerfile 示例。该文件的内容在以下示例中显示：

```
FROM registry.access.redhat.com/ubi8/ubi:8.1

RUN curl -L -o cincinnati-graph-data.tar.gz https://github.com/openshift/cincinnati-graph-data/archive/master.tar.gz

RUN mkdir -p /var/lib/cincinnati/graph-data/

CMD exec /bin/bash -c "tar xvzf cincinnati-graph-data.tar.gz -C /var/lib/cincinnati/graph-data/ --strip-components=1"
```

   在本例中：

   - **FROM** 值是 OpenShift Update Service 查找镜像的外部 registry。
   - **RUN** 命令创建目录并打包升级文件。
   - **CMD** 命令将软件包文件复制到本地库并提取文件进行升级。

3. 运行以下命令来构建 **图形数据 init 容器**:

```
podman build -f <path_to_Dockerfile> -t
```

```
${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-container:latest
podman push ${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-container:latest --authfile=/path/to/pull_secret.json
```

将 *path_to_Dockerfile* 替换为您在上一步中创建文件的路径。

将 *${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-container* 替换为 graph data init 容器的路径。

将 */path/to/pull_secret* 替换为 pull secret 文件的路径。

注： 如果没有安装 **podman**，您也可以将命令中的 **podman** 替换为 **docker**。

1.19.2.5. 为已镜像的 registry 配置证书

如果您使用安全的外部容器 registry 来存储已镜像的 OpenShift Container Platform 发行镜像，OpenShift Update Service 需要访问此 registry 来构建升级图。完成以下步骤以配置您的 CA 证书以用于 OpenShift Update Service Pod:

1. 查找位于 **image.config.openshift.io** 的 OpenShift Container Platform 外部 registry API。这是存储外部 registry CA 证书的位置。
   如需更多信息，请参阅 OpenShift Container Platform 文档中的配置额外的信任存储以访问镜像 registry。

2. 在 **openshift-config** 命名空间中创建 ConfigMap。

3. 在密钥 **updateservice-registry** 中添加您的 CA 证书。OpenShift Update Service 使用此设置来定位您的证书：

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: trusted-ca
data:
  updateservice-registry: |
    -----BEGIN CERTIFICATE-----
    ...
    -----END CERTIFICATE-----
```

4. 编辑 **image.config.openshift.io** API 中的 **cluster** 资源，将 **additionalTrustedCA** 字段设置为您创建的 ConfigMap 的名称。

```
oc patch image.config.openshift.io cluster -p '{"spec":{"additionalTrustedCA":{"name":"trusted-ca"}}}' --type merge
```

将 ***trusted-ca*** 替换为新 ConfigMap 的路径。

OpenShift Update Service Operator 会监视 **image.config.openshift.io** API 和您在 **openshift-config** 命名空间中创建的 ConfigMap 以获取更改，然后在 CA 证书已更改时重启部署。

1.19.2.6. 部署 OpenShift Update Service 实例

当在 hub 集群上完成 OpenShift Update Service 实例部署时，此实例就位于集群升级的镜像镜像位置，并可供断开连接的受管集群使用。完成以下步骤以部署实例：

1. 如果您不想使用 Operator 的默认命名空间（**openshift-cincinnati**），为 OpenShift Update Service 实例创建一个命名空间：

   a. 在 OpenShift Container Platform hub 集群控制台导航菜单中，选择 **Administration** > **Namespaces**。

   b. 点 **Create Namespace**。

   c. 添加命名空间的名称以及您的命名空间的任何其他信息。

   d. 选择 **Create** 来创建命名空间。

2. 在 OpenShift Container Platform 控制台的 *Installed Operators* 部分中，选择 **Red Hat OpenShift Update Service Operator**。

3. 在菜单中选择 **Create Instance**。

4. 粘贴 OpenShift Update Service 实例中的内容。您的 YAML 实例可能类似以下清单：

```
apiVersion: cincinnati.openshift.io/v1beta2
kind: Cincinnati
metadata:
  name: openshift-update-service-instance
  namespace: openshift-cincinnati
spec:
  registry: <registry_host_name>:<port>
  replicas: 1
  repository: ${LOCAL_REGISTRY}/ocp4/release
  graphDataImage: '<host_name>:<port>/cincinnati-graph-data-container'
```

   将 **spec.registry** 值替换为镜像的本地断开连接 registry 的路径。

   将 **spec.graphDataImage** 值替换为图形数据 init 容器的路径。**提示：** 这与运行 **podman push** 命令来推送图形数据 init 容器时使用的值相同。

5. 选择 **Create** 来创建实例。

6. 在 hub 集群 CLI 中输入 **oc get pods** 命令来查看实例创建的状态。它可能需要一段时间，但当命令结果显示实例和运算符正在运行时，进程就会完成。

1.19.2.7. 部署策略以覆盖默认 registry（可选）

注： 本节中的步骤只在将发行版本镜像到您的镜像 registry 时才应用。

OpenShift Container Platform 具有一个默认的镜像 registry 值，用于指定它找到升级软件包的位置。在断开连接的环境中，您可以创建一个策略来替换该值，并将该值替换为您对发行版本镜像进行镜像的本地镜像 registry 的路径。

对于这些步骤，策略名为 *ImageContentSourcePolicy*。完成以下步骤以创建策略：

1. 登录到 hub 集群的 OpenShift Container Platform 环境。

2. 在 OpenShift Container Platform 导航中，选择 **Administration** > **Custom Resource Definitions**。

3. 选择 *Instances* 选项卡。

4. 选择设置断开连接的 OLM 时创建的 *ImageContentSourcePolicy* 名称，以查看其内容。
5. 选择 *YAML* 选项卡以 YAML 格式查看内容。
6. 复制 ImageContentSourcePolicy 的整个内容。
7. 在 Red Hat Advanced Cluster Management 控制台中选择 **Governance** > **Create policy**。
8. 将 **YAML** 开关设置为 *On* 以查看策略的 YAML 版本。
9. 删除 YAML 代码中的所有内容。
10. 将以下 YAML 内容粘贴到窗口以创建自定义策略：

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards: ""
    policy.open-cluster-management.io/categories: ""
    policy.open-cluster-management.io/controls: ""
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
          namespace: default
        spec:
          remediationAction: inform
          severity: low
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: operator.openshift.io/v1alpha1
                kind: ImageContentSourcePolicy
                metadata:
                  name: <your-local-mirror-name>
                spec:
                  repositoryDigestMirrors:
                    - mirrors:
                        - <your-registry>
                      source: registry.redhat.io
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
```

```
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      []  # selects all clusters if not specified
```

11. 将模板 **objectDefinition** 部分中的内容替换为内容，并添加 **ImageContentSourcePolicy** 的设置。将 **path-to-local-mirror** 替换为本地镜像存储库的路径。
提示： 您可以通过输入 **oc adm release mirror** 命令来查找到本地镜像的路径。

12. 选择 **Enforce if supported**。

13. 选择 **Create** 来创建策略。

1.19.2.8. 部署策略来部署断开连接的目录源

将 *Catalogsource* 策略推送到受管集群，将默认位置从连接的位置更改为您断开连接的本地 registry。

1. 在 Red Hat Advanced Cluster Management 控制台中，选择 **Infrastructure** > **Clusters**。

2. 在集群列表中找到要接收策略的受管集群。

3. 记录下受管集群 **name** 标签的值。标签格式为 **name=managed-cluster-name**。该值会在推送策略时使用。

4. 在 Red Hat Advanced Cluster Management 控制台菜单中，选择 **Governance** > **Create policy**。

5. 将 **YAML** 切换设置为 *On* 以查看策略的 YAML 版本。

6. 删除 **YAML** 代码中的所有内容。

7. 将以下 **YAML** 内容粘贴到窗口以创建自定义策略：

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
```

```
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
        spec:
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: v1
                kind: Pod
                metadata:
                  name: sample-nginx-pod
                  namespace: default
                status:
                  phase: Running
          remediationAction: inform
          severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      [] # selects all clusters if not specified
```

8. 在策略中添加以下内容：

```
apiVersion: config.openshift.io/vi
kind: OperatorHub
metadata:
```

```
 name: cluster
spec:
 disableAllDefaultSources: true
```

9. 添加以下内容：

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: my-operator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image: <registry_host_name>:<port>/olm/redhat-operators:v1
  displayName: My Operator Catalog
  publisher: grpc
```

将 *spec.image* 值替换为本地受限目录源镜像的路径。

10. 在 Red Hat Advanced Cluster Management 控制台导航中，选择 **Infrastructure** > **Clusters** 以检查受管集群的状态。应用策略时，集群状态为 **ready**。

1.19.2.9. 部署策略以更改受管集群参数

将 *ClusterVersion* 策略推送到受管集群，以更改其检索升级的默认位置。

1. 在受管集群中，输入以下命令确认 *ClusterVersion* upstream 参数目前是默认公共 OpenShift Update Service 操作对象：

```
oc get clusterversion -o yaml
```

返回的内容可能类似以下内容：

```
apiVersion: v1
items:
- apiVersion: config.openshift.io/v1
  kind: ClusterVersion
[..]
  spec:
    channel: stable-4.4
    upstream: https://api.openshift.com/api/upgrades_info/v1/graph
```

2. 在 hub 集群中，输入以下命令来识别到 OpenShift Update Service 操作对象：**oc get routes**。**提示：** 记录这个值以便在以后的步骤中使用。

3. 在 hub 集群 Red Hat Advanced Cluster Management 控制台菜单中，选择 **Governance** > **Create a policy**。

4. 将 **YAML** 切换设置为 *On* 以查看策略的 YAML 版本。

5. 删除 **YAML** 代码中的所有内容。

6. 将以下 **YAML** 内容粘贴到窗口以创建自定义策略：

```
apiVersion: policy.open-cluster-management.io/v1
```

```
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
        spec:
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: v1
                kind: Pod
                metadata:
                  name: sample-nginx-pod
                  namespace: default
                status:
                  phase: Running
          remediationAction: inform
          severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
```

```
  clusterSelector:
    matchExpressions:
      []  # selects all clusters if not specified
```

7. 将以下内容添加到 *policy* 项中的 **policy.spec** 部分：

```
apiVersion: config.openshift.io/v1
  kind: ClusterVersion
  metadata:
    name: version
  spec:
    channel: stable-4.4
    upstream: https://example-cincinnati-policy-engine-uri/api/upgrades_info/v1/graph
```

将 *spec.upstream* 值替换为 hub 集群 OpenShift Update Service 操作对象的路径。

提示： 您可以完成以下步骤以确定操作对象的路径：

a. 在 hub 集群上运行 **oc get get routes -A** 命令。

b. 找到到 **cincinnati** 的路由。+ 到操作对象的路径是 **HOST/PORT** 字段的值。

8. 在受管集群 CLI 中，使用以下命令确认 **ClusterVersion** 中的上游参数已使用本地 hub 集群 OpenShift Update Service URL 更新：

```
oc get clusterversion -o yaml
```

结果应该类似以下内容：

```
apiVersion: v1
items:
- apiVersion: config.openshift.io/v1
  kind: ClusterVersion
[..]
  spec:
    channel: stable-4.4
    upstream: https://<hub-cincinnati-uri>/api/upgrades_info/v1/graph
```

1.19.2.10. 查看可用升级

您可以通过完成以下步骤来查看受管集群可用升级列表：

1. 登录您的 Red Hat Advanced Cluster Management 控制台。
2. 在导航菜单中选择 **Infrastructure** > **Clusters**。
3. 选择处于 *Ready* 状态的一个集群。
4. 在 **Actions** 菜单中选择 **Upgrade cluster**。
5. 验证可选的升级路径是否可用。
   注： 如果当前版本没有镜像到本地镜像存储库，则不会显示可用的升级版本。

1.19.2.11. 选择一个频道

您可以使用 Red Hat Advanced Cluster Management 控制台为 OpenShift Container Platform 版本 4.6 或更高版本上的集群升级选择一个频道。这些版本必须在镜像 registry 上可用。完成选择频道中的步骤为您的升级指定频道。

1.19.2.12. 升级集群

配置断开连接的 registry 后，Red Hat Advanced Cluster Management 和 OpenShift Update Service 使用断开连接的 registry 来确定升级是否可用。如果没有可用的升级，请确保您有集群当前级别的发行镜像，且至少有一个后续级别的镜像位于本地存储库中。如果集群当前版本的发行镜像不可用，则没有可用的升级。

完成以下步骤进行升级：

1. 在 Red Hat Advanced Cluster Management 控制台中，选择 **Infrastructure** > **Clusters**。
2. 查找您要确定是否有可用的升级的集群。
3. 如果有可用的升级，集群的 **Distribution version** 栏表示有可用的升级可用。
4. 选择集群的 *Options* 菜单，然后选择 **Upgrade cluster**。
5. 为升级选择目标版本，然后选择 **Upgrade**。

受管集群已更新至所选版本。

如果集群升级失败，Operator 通常会重试升级，停止并报告故障组件的状态。在某些情况下，升级过程会一直通过尝试完成此过程进行循环。不支持在失败的升级后将集群还原到以前的版本。如果您的集群升级失败，请联系红帽支持以寻求帮助。

1.20. 从管理中移除集群

当您从管理中删除通过 Red Hat Advanced Cluster Management for Kubernetes 创建的 OpenShift Container Platform 集群时，您可以将其*分离（detach）*或*销毁（destroy）*。分离集群会将其从管理中移除，但不会完全删除。如果要管理它，您可以再次导入它。只有集群处于 *Ready* 状态时方可使用这个选项。

以下流程在以下情况下从管理中删除集群：

- 您已删除集群，并希望从 Red Hat Advanced Cluster Management 中删除已删除的集群。
- 您要从管理中删除集群，但还没有删除集群。

重要：

- 销毁集群会将其从管理中移除，并删除集群的组件。
- 当您分离受管集群时，相关命名空间会被自动删除。不要将自定义资源放在这个命名空间中。
  - 使用控制台删除集群
  - 使用命令行删除集群
  - 删除集群后删除剩余的资源
  - 删除集群后对 etcd 数据库进行碎片整理

1.20.1. 使用控制台删除集群

在导航菜单中导航到 **Infrastructure** > **Clusters**，从您要从管理中删除的集群旁的选项菜单中选择 **Destroy cluster** 或 **Detach cluster**。

+ **提示：** 您可以通过选择要分离或销毁的集群的复选框来分离或销毁多个集群，然后选择 **Detach** 或 **Destroy**。

注： 如果您在管理的 hub 集群（称为 **local-cluster** ）时尝试分离 hub 集群，请检查 **disableHubSelfManagement** 的默认设置是否为 **false**。此设置会导致 hub 集群在分离时重新导入自己并管理自己，并协调 **MultiClusterHub** 控制器。hub 集群可能需要几小时时间来完成分离过程并重新导入。

要在不需要等待进程完成的情况下重新导入 hub 集群，您可以输入以下命令来重启 **multiclusterhub-operator** pod 并更快地重新导入：

```
oc delete po -n open-cluster-management `oc get pod -n open-cluster-management | grep multiclusterhub-operator| cut -d' ' -f1`
```

您可以通过将 **disableHubSelfManagement** 值改为 **true** 来更改 hub 集群的值，如 在线安装所述。

1.20.2. 使用命令行删除集群

要使用 hub 集群的命令行分离受管集群，请运行以下命令：

```
oc delete managedcluster $CLUSTER_NAME
```

要在分离后删除受管集群，请运行以下命令：

```
oc delete clusterdeployment <CLUSTER_NAME> -n $CLUSTER_NAME
```

注意： 如果您试图分离名为 **local-cluster** 的 hub 集群，请注意 **disableHub selfManagement** 的默认设置为 **false** 。此设置会导致 hub 集群在分离时会重新导入自己并管理自己，并协调 **MultiClusterHub** 控制器。hub 集群可能需要几小时时间来完成分离过程并重新导入。如果要在等待进程完成后重新导入 hub 集群，您可以输入以下命令来重启 **multiclusterhub-operator** pod 并更快地重新导入：

```
oc delete po -n open-cluster-management `oc get pod -n open-cluster-management | grep multiclusterhub-operator| cut -d' ' -f1`
```

您可以通过将 **disableHubSelfManagement** 值改为 **true** 来更改 hub 集群的值，如 在线安装所述。

1.20.3. 删除集群后删除剩余的资源

如果受管集群上有剩余的资源，则需要额外的步骤以确保删除所有剩余的组件。需要这些额外步骤的情况，包括以下示例：

- 受管集群在完全创建前会被分离，但 **klusterlet** 等一些组件会保留在受管集群中。
- 在分离受管集群前，管理集群的 hub 丢失或销毁，因此无法从 hub 中分离受管集群。
- 当受管集群被分离后，受管集群将处于非在线状态。

如果其中一个情况适用于您试图分离的受管集群，则有些资源将无法从受管集群中删除。完成以下步骤以分离受管集群：

1. 确保配置了 **oc** 命令行界面。
2. 确保您在受管集群中配置了 **KUBECONFIG**。
   如果运行 **oc get ns | grep open-cluster-management-agent**，您应该看到两个命名空间：

```
open-cluster-management-agent         Active   10m
open-cluster-management-agent-addon   Active   10m
```

3. 运行以下命令以删除剩余的资源：

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

4. 运行以下命令，以确保命名空间和所有打开的集群管理 **crds** 均已被删除：

```
oc get crds | grep open-cluster-management.io | awk '{print $1}'
oc get ns | grep open-cluster-management-agent
```

1.20.4. 删除集群后对 etcd 数据库进行碎片整理

拥有多个受管集群可能会影响 hub 集群中 **etcd** 数据库的大小。在 OpenShift Container Platform 4.8 中，当删除受管集群时，hub 集群中的 **etcd** 数据库不会被自动减小。在某些情况下，**etcd** 数据库可能会耗尽空间。此时会显示一个错误 **etcdserver: mvcc: database space exceeded**。要更正此错误，请通过压缩数据库历史记录并对 **etcd** 数据库进行碎片整理来减小 **etcd** 数据库的大小。

注： 对于 OpenShift Container Platform 版本 4.9 及更新的版本，etcd Operator 会自动清理磁盘并压缩 **etcd** 历史记录。不需要人工干预。以下流程适用于 OpenShift Container Platform 版本 4.8 及更早版本。

通过完成以下步骤，压缩 **etcd** 历史记录并整理 hub 集群中 **etcd** 数据库的碎片。

1.20.4.1. 先决条件

- 安装 OpenShift CLI（**oc**）。
- 以具有 **cluster-admin** 权限的用户身份登录。

1.20.4.2. 流程

1. 压缩 **etcd** 历史记录。
   a. 打开到 **etcd** 成员的远程 shell 会话，例如：

```
$ oc rsh -n openshift-etcd etcd-control-plane-0.example.com etcdctl endpoint status --cluster -w table
```

   b. 运行以下命令来压缩 **etcd** 历史记录：

```
sh-4.4#etcdctl compact $(etcdctl endpoint status --write-out="json" |  egrep -o '"revision":[0-9]*' | egrep -o '[0-9]*' -m1)
```

输出示例

```
$ compacted revision 158774421
```

2. 清理 **etcd** 数据库并清除任何 **NOSPACE** 警报，如分离 **etcd** 数据 中所述。

1.21. 集群备份和恢复 OPERATOR

当 Red Hat Advanced Cluster Management for Kubernetes hub 集群停机并需要重新创建时，集群备份和恢复 Operator 提供灾难恢复解决方案。它运行在 hub 集群中，它依赖于 OADP Operator 安装 Velero，并从 hub 集群创建到存储数据的备份存储位置的连接。Velero 是运行备份和恢复操作的组件。集群备份和恢复 Operator 解决方案为所有 Red Hat Advanced Cluster Management hub 集群资源（如受管集群、应用程序、策略和裸机资产）提供备份和恢复支持。

它支持备份扩展 hub 集群安装的任何第三方资源。使用这个备份解决方案，您可以定义基于 cron 的备份计划，这些计划在指定时间段内运行。当 hub 集群停机时，可以部署新的 hub 集群，并将备份的数据移到新的 hub 集群中。

集群备份和恢复 Operator 不会被自动安装。通过在 **MultiClusterHub** 资源中将 **cluster-backup** 参数设置为 **true** 来启用备份组件。集群备份 Operator 安装在安装了 Red Hat Advanced Cluster Management 的 **open-cluster-management-backup** 命名空间中。安装集群备份 Operator 时，也会自动安装 OADP Operator。

备注:

- OADP Operator 1.0 已禁用构建多架构构建，仅为官方版本生成 **x86_64** 构建。这意味着，如果您使用 **x86_64** 以外的构架，由备份组件安装的 OADP Operator 必须替换为正确的版本。在这种情况下，卸载 OADP Operator 并找到与您的架构匹配的 Operator，然后安装它。
- 如果您之前已在 hub 集群上安装并使用 OADP Operator，请卸载这个版本，因为备份组件现在可以在组件命名空间中安装 OADP。对安装有备份组件的 OADP Operator 拥有的 **DataProtectionApplication** 资源使用相同的存储位置，它会访问与之前 Operator 相同的备份数据。Velero 备份资源现在在此 hub 集群上的新 OADP Operator 命名空间中载入。

Velero 在 Red Hat Advanced Cluster Management hub 集群上安装 OADP Operator。Velero 用于备份和恢复 Red Hat Advanced Cluster Management hub 集群资源。

有关 Velero 支持的存储供应商列表，请参阅 S3-Compatible 对象存储供应商。

- 前提条件
- 备份和恢复 Operator 架构
  - 备份的资源
    - 扩展备份数据
    - 在受管集群激活时恢复的资源
  - 资源请求和限值自定义
  - 使用服务器端加密保护数据
  - 调度集群备份

- 恢复备份
  - 准备新的 hub 集群
  - 在恢复前清理 hub 集群
  - 资源在受管激活时恢复
  - 恢复被动资源
  - 在检查备份时恢复被动资源
  - 恢复导入的受管集群
  - 恢复激活资源
- 主动被动配置
  - 受管集群激活数据
- 灾难恢复
- 使用策略备份验证

1.21.1. 先决条件

- 确保完成为保存备份的云存储创建凭证 secret 的步骤。secret 资源必须在 OADP operator 命名空间中创建，后者是 **open-cluster-management-backup** 命名空间。
- 在创建 **DataProtectionApplication** 资源时，使用创建的 secret。
  完成以下步骤以创建 **DataProtectionApplication** 资源实例：

  1. 在 Red Hat OpenShift Container Platform 控制台中选择 **Operators** > **Installed Operators**。
  2. 在 DataProtectionApplication 下点 **Create instance**。
  3. 使用 {ocp-short）控制台或使用 **DataProtectionApplication** 示例中所述的 YAML 文件选择配置来创建 Velero 实例。
  4. 将 **DataProtectionApplication** namespace 设置为 **open-cluster-management-backup**。
  5. 为 **DataProtectionApplication** 资源正确设置规格(**spec:**)值。然后点**创建**。
     提到的资源值是为了便于使用。如果您打算使用默认的备份存储位置，请在 **backupStorageLocations** 部分中设置以下值 **default: true**。您的 **DataProtectionApplication** 资源可能类似以下 YAML 文件：

     您的 **DataProtectionApplication** 资源可能类似以下 YAML 文件：

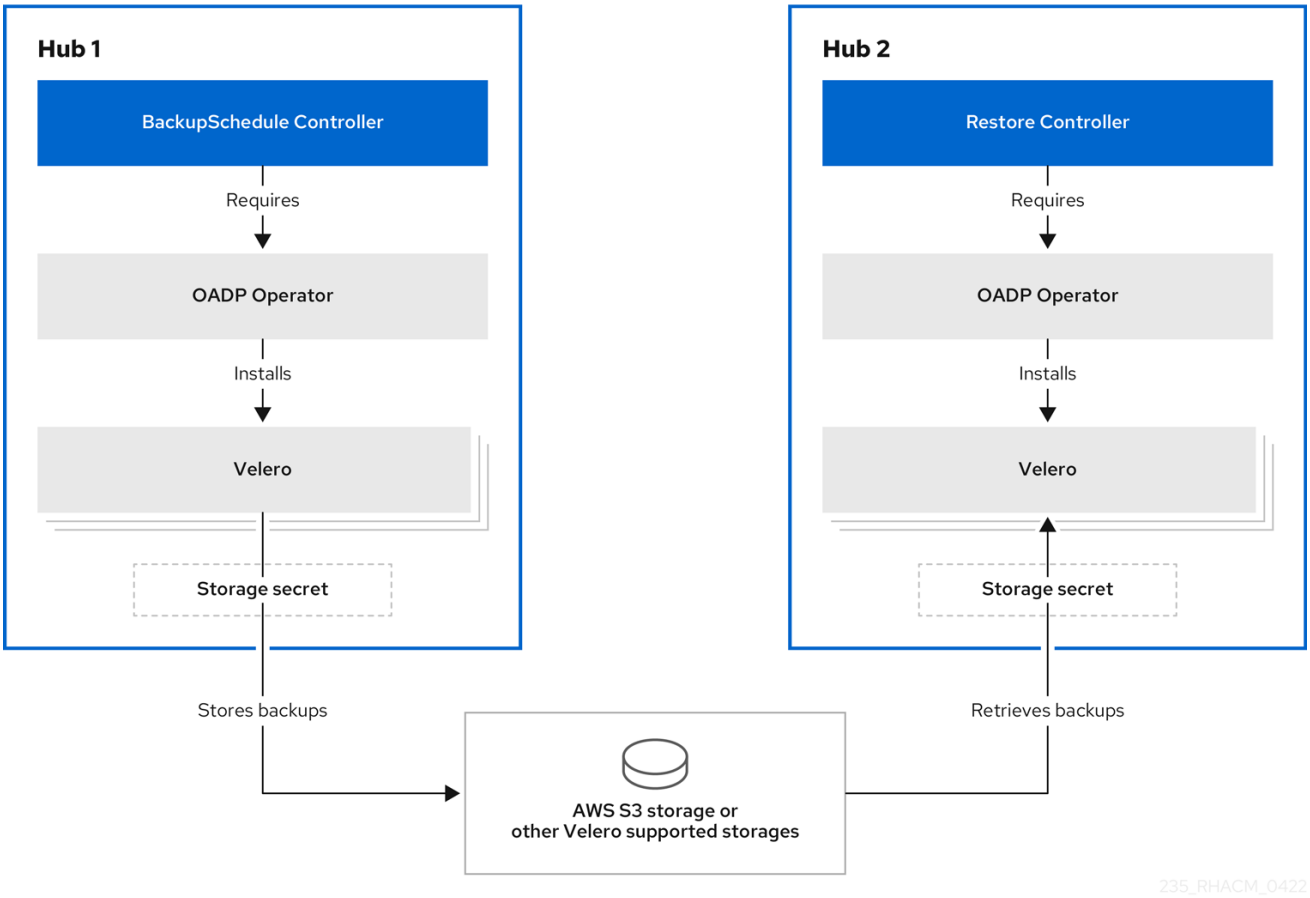
```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-sample
spec:
  configuration:
    velero:
      defaultPlugins:
```

```
    - openshift
    - aws
  restic:
    enable: true
backupLocations:
  - name: default
    velero:
      provider: aws
      default: true
      objectStorage:
        bucket: my-bucket
        prefix: my-prefix
      config:
        region: us-east-1
        profile: "default"
      credential:
        name: cloud-credentials
        key: cloud
snapshotLocations:
  - name: default
    velero:
      provider: aws
      config:
        region: us-west-2
        profile: "default"
```

请参阅创建 **DataProtectionApplication** 资源的示例。

1.21.2. 备份和恢复 Operator 架构

Operator 定义 **backupSchedule.cluster.open-cluster-management.io** 资源，用于设置 Red Hat Advanced Cluster Management 备份计划，以及 **restore.cluster.open-cluster-management.io** 资源，该资源用于处理和恢复这些备份。Operator 会创建对应的 Velero 资源，并定义备份远程集群和需要恢复的任何其他 hub 集群资源所需的选项。查看以下示意图：



1.21.2.1. 备份的资源

集群备份和恢复 Operator 解决方案为所有 hub 集群资源（如受管集群、应用程序、策略和裸机资产）提供备份和恢复支持。您可以使用解决方案备份任何扩展基本 hub 集群安装的第三方资源。使用这个备份解决方案，您可以定义一个基于 cron 的备份调度，该调度在指定时间段内运行，并持续备份 hub 集群内容的最新版本。

当 hub 集群需要替换或处于灾难情况下，当 hub 集群停机时，可以部署新的 hub 集群并备份数据被移到新的 hub 集群中。

查看以下用于识别备份数据的集群备份和恢复过程的排序列表：

- 排除 **MultiClusterHub** 命名空间中的所有资源。这是为了避免备份链接到当前 hub 集群身份的安装资源，不应该备份。
- 使用 **.open-cluster-management.io** 后缀的 API 版本备份所有 CRD。这个后缀表示所有 Red Hat Advanced Cluster Management 资源都已备份。
- 从以下 API 组备份所有 CRD：**argoproj.io**,**app.k8s.io**,**core.observatorium.io**,**hive.openshift.io**。
- 排除以下 API 组中的所有 CRD: **admission.cluster.open-cluster-management.io**, **admission.work.open-cluster-management.io**, **internal.open-cluster-management.io**, **operator.open-cluster-management.io**, **work.open-cluster-management.io**, **search.open-cluster-management.io**, **admission.hive.openshift.io**, **velero.io**。
- 排除以下 CRD，它们是包含的 API 组的一部分，但并不需要或被所有者资源（也会被备份）所替代：**clustermanagementaddon**, **observabilityaddon**, **applicationmanager**, **certpolicycontroller**, **iampolicycontroller**, **policycontroller**, **searchcollector**, **workmanager**,

backupschedule, **restore**, **clusterclaim.cluster.open-cluster-management.io**。

- 使用以下标签之一备份 secret 和 ConfigMap：**cluster.open-cluster-management.io/type**,**hive.openshift.io/secret-type**,**cluster.open-cluster-management.io/backup**。
- 对于您要备份的任何其他资源，使用 **cluster.open-cluster-management.io/backup** 标签，且不包含在前面提到的条件中。请参见以下示例：

```
apiVersion: my.group/v1alpha1
kind: MyResource
metadata:
  labels:
    cluster.open-cluster-management.io/backup: ""
```

注：需要备份 **hive.openshift.io.ClusterDeployment** 资源使用的 secret，并仅在使用控制台创建集群时使用 **cluster.open-cluster-management.io/backup** 标签自动标注。如果使用 GitOps 部署 Hive 集群，则必须手动将 **cluster.open-cluster-management.io/backup** 标签添加到 **ClusterDeployment** 使用的 secret 中。

- 排除您不想备份的特定资源。例如，请查看以下从备份过程中排除 Velero 资源的示例：

```
apiVersion: my.group/v1alpha1
kind: MyResource
metadata:
  labels:
    velero.io/exclude-from-backup: "true"
```

==== 扩展备份数据

您可以通过在资源中添加 **cluster.open-cluster-management.io/backup** 标签来备份集群备份和恢复的第三方资源。标签的值可以是任意字符串，包括空字符串。使用一个可以帮助您识别要备份的组件的值。例如，如果组件是由 IDP 解决方案提供，请使用 **cluster.open-cluster-management.io/backup: idp** 标签。

注： 如果您希望在受管集群激活资源时恢复资源，请使用 **cluster.open-cluster-management.io/backup** 标签的 **cluster-activation** 值。恢复受管集群激活资源会导致受管集群活跃由 hub 集群（在启动恢复的位置）主动管理。

1.21.2.1.1. 在受管集群激活时恢复的资源

当您将 **cluster.open-cluster-management.io/backup** 标签添加到资源时，资源会在 **acm-resources-generic-schedule** 备份中自动备份。如果需要恢复任何资源，则必须将标签值设置为 **cluster-activation**，仅在受管集群移到新的 hub 集群后，并在恢复的资源中使用 **veleroManagedClustersBackupName:latest**。这样可确保资源不会被恢复，除非受管集群激活被调用。查看以下示例：

```
apiVersion: my.group/v1alpha1
kind: MyResource
metadata:
  labels:
    cluster.open-cluster-management.io/backup: cluster-activation
```

除了使用 **cluster.open-cluster-management.io/backup: cluster-activation** 标签外

除了使用 **cluster.open-cluster-management.io/backup: cluster-activation** 标签并由 **acm-resources-generic-schedule** 备份存储的激活数据资源外，集群备份和恢复 Operator 还默认在激活集合中包括一些资源。以下资源由 **acm-managed-clusters-schedule** 备份备份：

- **managedcluster.cluster.open-cluster-management.io**
- **managedcluster.clusterview.open-cluster-management.io**
- **klusterletaddonconfig.agent.open-cluster-management.io**
- **managedclusteraddon.addon.open-cluster-management.io**
- **managedclusterset.cluster.open-cluster-management.io**
- **managedclusterset.clusterview.open-cluster-management.io**
- **managedclustersetbinding.cluster.open-cluster-management.io**
- **clusterpool.hive.openshift.io**
- **clusterclaim.hive.openshift.io**
- **clustercurator.cluster.open-cluster-management.io**

1.21.2.2. 资源请求和限值自定义

最初安装 Velero 时，Velero pod 会被设置为默认 CPU 和内存限值，如下例所示：

```
resources:
 limits:
   cpu: "1"
   memory: 256Mi
 requests:
   cpu: 500m
   memory: 128Mi
```

以上示例中的限制在某些情况下可以正常工作，但可能会在集群备份大量资源时进行更新。例如，当备份在管理 2000 集群的 hub 集群上运行时，Velero pod 会因为内存不足错误(OOM)崩溃。对于这种情况，以下配置允许备份完成：

```
  limits:
    cpu: "2"
    memory: 1Gi
  requests:
    cpu: 500m
    memory: 256Mi
```

要更新 Velero pod 资源的限制和请求，您需要更新 **DataProtectionApplication** 资源，并为 Velero pod 插入 **resourceAllocation** 模板。查看以下示例：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero
  namespace: open-cluster-management-backup
spec:
```

```
...
  configuration:
...
    velero:
      podConfig:
        resourceAllocations:
          limits:
            cpu: "2"
            memory: 1Gi
          requests:
            cpu: 500m
            memory: 256Mi
```

请参阅 Velero 资源请求和限值自定义，以了解更多有关 **DataProtectionApplication** 参数的信息。

1.21.2.3. 使用服务器端加密保护数据

服务器端加密是应用程序或服务的数据加密，在存储位置接收数据。在传输过程中，备份机制本身不会加密数据（因为它会前往备份存储位置），或其他人（同时存储在备份存储位置的磁盘上）。它依赖于对象和快照系统中的原生机制。

最佳实践 ：使用可用的备份存储服务器端加密数据。备份包含资源，如在 hub 集群外存储需要加密的凭证和配置文件。

您可以使用 **serverSideEncryption** 和 **kmsKeyId** 参数为存储在 Amazon S3 中的备份启用加密。如需了解更多详细信息，请参阅备份存储位置 YAML。以下示例指定在设置 **DataProtectionApplication** 资源时的 AWS KMS 密钥 ID：

```
spec:
  backupLocations:
    - velero:
        config:
          kmsKeyId: 502b409c-4da1-419f-a16e-eif453b3i49f
          profile: default
          region: us-east-1
```

请参阅 Velero 支持的存储供应商，以找出其它存储供应商的所有可配置参数。

1.21.2.4. 调度集群备份

在创建 **backupschedule.cluster.open-cluster-management.io** 资源时，会激活备份调度。查看以下 **backupschedule.cluster.open-cluster-management.io** 示例：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name: schedule-acm
spec:
  veleroSchedule: 0 */2 * * *
  veleroTtl: 120h
```

创建 **backupschedule.cluster.open-cluster-management.io** 资源后，运行以下命令来获取调度的集群备份的状态：

```
oc get bsch -n <oadp-operator-ns>
```

上一命令中的 **<oadp-operator-ns>** 参数是创建 **BackupSchedule** 的命名空间，它与安装 OADP Operator 的命名空间相同。**backupschedule.cluster.open-cluster-management.io** 资源会创建六个 **schedule.velero.io** 资源，用于生成备份。运行以下命令查看调度的备份列表：

```
os get schedules -A | grep acm
```

资源在以下组中单独备份：

- *凭证备份*，其中包含 Hive、Red Hat Advanced Cluster Management 和用户创建凭证的三个备份文件。
- *资源备份*，其中包含 Red Hat Advanced Cluster Management 资源的一个备份，另一个用于通用资源。这些资源使用以下标签 **cluster.open-cluster-management.io/backup**。
- *受管集群备份*，其中仅包含激活与 hub 集群连接的资源，在恢复备份时。

注 ： *资源备份*文件包含特定于受管集群的资源，但不包含将受管集群连接到 hub 集群的资源子集。连接受管集群的资源称为激活资源，并包含在受管集群备份中。当您只在新 hub 集群中为 *凭证*和 *资源*备份中恢复备份时，新的 hub 集群将所有使用 Hive API 创建的受管集群处于分离状态。但是，只有在 passive hub 集群上恢复激活数据时，使用导入操作在主 hub 集群上导入的受管集群才会出现。目前，受管集群仍然连接到创建备份文件的原始 hub 集群。

当恢复激活数据时，只有使用 Hive API 创建的受管集群才会与新的 hub 集群自动连接。所有其他受管集群都显示为 *Pending* 状态，必须手动重新附加到新集群。

1.21.3. 恢复备份

在进行一般的恢复时，运行备份的 hub 集群变得不可用，备份的数据需要移到一个新的 hub 集群。这可以通过在新的 hub 集群上运行集群恢复操作来完成。在这种情况下，恢复操作会在创建备份的不同 hub 集群中运行。

有些情况下，您要在收集备份的同一 hub 集群中恢复数据，以便恢复来自以前快照的数据。在这种情况下，恢复和备份操作都在同一 hub 集群中运行。

在 hub 集群中创建 **restore.cluster.open-cluster-management.io** 资源后，您可以运行以下命令来获取恢复操作的状态：**oc get restore -n <oadp-operator-ns>**。您还应能够验证是否已创建备份文件中包含的已备份资源。

注 ：restore.cluster.open-cluster-management.io 资源运行一次。如果要在恢复操作完成后再次运行相同的恢复操作，您必须使用相同的 **spec** 选项创建新的 **restore.cluster.open-cluster-management.io** 资源。

restore 操作用于恢复备份操作创建的所有三种备份类型。但是，您可以选择只安装特定类型的备份（仅限受管集群、仅用户凭证或只安装 hub 集群资源）。

恢复定义以下三个必要的 **spec** 属性，其中为备份文件类型定义了恢复逻辑：

- **veleroManagedClustersBackupName** 用于定义受管集群激活资源的恢复选项。
- **veleroCredentialsBackupName** 用于为用户凭证定义 restore 选项。
- **veleroResourcesBackupName** 用于定义 hub 集群资源的 restore 选项（**Applications**、**Policy** 及其他 hub 集群资源，如受管集群被动数据）。
  前面提到的属性的有效选项有以下值：

- **latest** - 此属性恢复此类型的备份文件。
- **skip** - 此属性不会尝试使用当前恢复操作恢复这种类型的备份。
- **<backup_name>** - 此属性按名称恢复指向它的指定的备份。

由 **restore.cluster.open-cluster-management.io** 创建的 **restore.velero.io** 资源的名称遵循以下模版规则 **<restore.cluster.open-cluster-management.io name>-<velero-backup-resource-name>**.查看以下描述：

- **restore.cluster.open-cluster-management.io 名称** 是当前 **restore.cluster.open-cluster-management.io** 资源的名称，该资源用于启动恢复。
- **velero-backup-resource-name** 是 Velero 备份文件的名称，用于恢复数据。例如，名为 **restore-acm** 的 **restore.cluster.open-cluster-management.io** 资源创建 **restore.velero.io** 恢复资源。查看以下格式示例：
  - **restore-acm-acm-managed-clusters-schedule-20210902205438** 可用于恢复受管集群激活数据备份。在本例中，用于恢复资源的 **backup.velero.io** 备份名称为 **acm-managed-clusters-schedule-20210902205438**。
  - **restore-acm-acm-credentials-schedule-20210902206789** 用于恢复凭据备份。在本例中，用于恢复资源的 **backup.velero.io** 备份名称为 **acm-managed-clusters-schedule-20210902206789**。
  - **restore-acm-acm-resources-schedule-20210902201234** 用于恢复应用程序、策略和其他 hub 集群资源，如受管集群被动数据备份。在这个示例中，用于恢复资源的 **backup.velero.io** 备份名称为 **acm-managed-clusters-schedule-20210902201234**。

注 ：备份类型为 **skip**，则不会创建 **restore.velero.io**。

查看以下集群 **Restore** 资源的 YAML 示例。在这个示例中，使用最新可用的备份文件恢复所有三种备份文件：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
spec:
  veleroManagedClustersBackupName: latest
  veleroCredentialsBackupName: latest
  veleroResourcesBackupName: latest
```

Note:当来自受管集群备份的 **acm-managed-clusters** 备份被恢复到另外一个 hub 集群时，只有 Hive API 创建的受管集群会自动连接到新的 hub 集群。所有其他受管集群都处于 **Pending Import** 状态，且必须重新导入到新的 hub 集群中。如需更多信息，请参阅 恢复导入的受管集群（技术预览）。

1.21.3.1. 准备新的 hub 集群

在新 hub 集群上运行恢复操作前，您需要手动配置 hub 集群，并在初始 hub 集群上安装相同的 Operator。您必须在与初始 hub 集群相同的命名空间中安装 Red Hat Advanced Cluster Management Operator，创建 **DataProtectionApplication** 资源，然后连接到之前备份了数据的同一存储位置。

例如，如果初始 hub 集群安装了任何其他 Operator，如 Ansible Automation Platform、Red Hat OpenShift GitOps、**cert-manager**，则必须在运行恢复操作前安装它们。这样可确保配置新的 hub 集群与初始 hub 集群相同。

1.21.3.2. 在恢复前清理 hub 集群

Velero 目前跳过 hub 集群中现有的备份资源。这会限制在新 hub 集群上恢复 hub 集群数据时可以使用的场景。如果使用新的 hub 集群并且应用了多次恢复，则不建议使用 hub 集群作为被动配置，除非在运行恢复前清理数据。新 hub 集群中的数据没有反映恢复的资源的可用数据。

创建 **restore.cluster.open-cluster-management.io** 资源时，集群备份和恢复 Operator 将运行一组步骤，以便在 Velero 恢复开始前清理 hub 集群来准备恢复。

cleanup 选项使用 **cleanupBeforeRestore** 属性来识别要清理的对象子集。您可以为这个清理设置三个选项：

- **None**: 不需要清理，只开始 Velero 恢复。这在全新的 hub 集群中使用。
- **CleanupRestored** ：清理以前 Red Hat Advanced Cluster Management 恢复创建的所有资源。建议使用此属性，因为它比 **CleanupAll** 属性小。
- **CleanupAll** ：清理 hub 集群上的所有资源，它可以作为 Red Hat Advanced Cluster Management 备份的一部分，即使资源没有因为恢复操作而创建。当在 hub 集群中创建额外的内容时，会使用它，这需要清理。请谨慎使用这个选项，因为这个选项会在由用户创建的 hub 集群上清理资源，而不是之前备份。强烈建议您使用 **CleanupRestored** 选项，并在 hub 集群指定为灾难情况下，禁止手动更新 hub 集群内容。使用 **CleanupAll** 选项作为最后一个替代方案。

备注:

- 如果恢复的备份没有资源，Velero 会为 velero 恢复资源设置状态 **PartiallyFailed**。这意味着，如果任何创建的 **restore.velero.io** 资源没有恢复任何资源，则 **restore.cluster.open-cluster-management.io** 资源可能会处于 **PartiallyFailed** 状态。
- **restore.cluster.open-cluster-management.io** 资源会运行一次，除非您使用 **syncRestoreWithNewBackups:true** 来在新的备份可用时恢复被动数据。在这种情况下，请按照使用同步示例的恢复被动操作。请参阅在检查备份时恢复被动资源。完成恢复操作后，您想要在同一 hub 集群上运行另一个恢复操作，您必须创建一个新的 **restore.cluster.open-cluster-management.io** 资源。
- 虽然您可以创建多个 **restore.cluster.open-cluster-management.io** 资源，但在任何时间点上只能有一个。

1.21.3.3. 恢复激活资源

当您希望 hub 集群管理集群时，请使用 **restore-passive-activate** 示例。在这种情况下，假设其它数据已在使用被动资源的 hub 集群上恢复。

1.21.3.4. 恢复被动资源

被动数据是备份数据，如 secret、ConfigMap、应用程序、策略以及所有受管集群自定义资源，它不在受管集群和 hub 集群之间激活连接。备份资源通过凭证备份和恢复资源在 hub 集群上恢复。

1.21.3.5. 在检查备份时恢复被动资源

使用 **restore-passive-sync** 示例恢复被动数据，同时继续检查新的备份是否可用并自动恢复它们。要自动恢复新的备份，您必须将 **syncRestoreWithNewBackups** 参数设置为 **true**。您还必须仅恢复最新的被动数据。

将 **VeleroResourcesBackupName** 和 **VeleroCredentialsBackupName** 参数设置为 **latest**，**VeleroManagedClustersBackupName** 参数为 **skip**。当将

VeleroManagedClustersBackupName 设置为 **latest** 后，受管集群会在新的 hub 集群中激活，现在是主 hub 集群。

当激活的受管集群变为主 hub 集群时，恢复资源被设置为 **Finished**，并且 **syncRestoreWithNewBackups** 会被忽略，即使设置为 **true**。

默认情况下，当 **syncRestoreWithNewBackups** 设为 **true** 时，控制程序会每 30 分钟检查新的备份。如果找到新的备份，它会恢复备份的资源。您可以通过更新 **restoreSyncInterval** 参数来更改检查的持续时间。

例如，以下资源每 10 分钟检查备份：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm-passive-sync
spec:
  syncRestoreWithNewBackups: true # restore again when new backups are available
  restoreSyncInterval: 10m # check for new backups every 10 minutes
  cleanupBeforeRestore: CleanupRestored
  veleroManagedClustersBackupName: skip
  veleroCredentialsBackupName: latest
  veleroResourcesBackupName: latest
```

1.21.3.6. 恢复导入的受管集群

只有使用 Hive API 与主 hub 集群连接的受管集群会自动连接到新的 hub 集群，其中恢复激活数据。这些集群已在主 hub 集群上使用 **Clusters** 选项卡的 **Create cluster** 按钮创建。当激活数据被恢复时，使用 **Import cluster** 按钮连接到初始 hub 集群的受管集群显示为 **Pending Import**，需要在新的 hub 集群中重新导入。

Hive 受管集群可以与新的 hub 集群连接，因为 Hive 将受管集群 **kubeconfig** 存储在 hub 集群上的受管集群命名空间中。这在新的 hub 集群上备份和恢复。然后，导入控制器使用恢复的配置更新受管集群上的 bootstrap **kubeconfig**，该配置仅适用于使用 Hive API 创建的受管集群。导入的集群不可用。

要在新 hub 集群上重新连接导入的集群，请在启动恢复操作后手动创建 **auto-import-secret** 资源。如需了解更多详细信息，请参阅使用自动导入 secret 导入集群。

在受管集群命名空间中创建 **auto-import-secret** 资源，每个集群处于 **Pending Import** 状态。使用具有足够权限的 **kubeconfig** 或令牌，以便导入组件在新 hub 集群上启动自动导入。您必须使用令牌连接到受管集群，为每个受管集群具有访问权限。令牌必须具有 **klusterlet** 角色绑定或具有相同权限的角色。

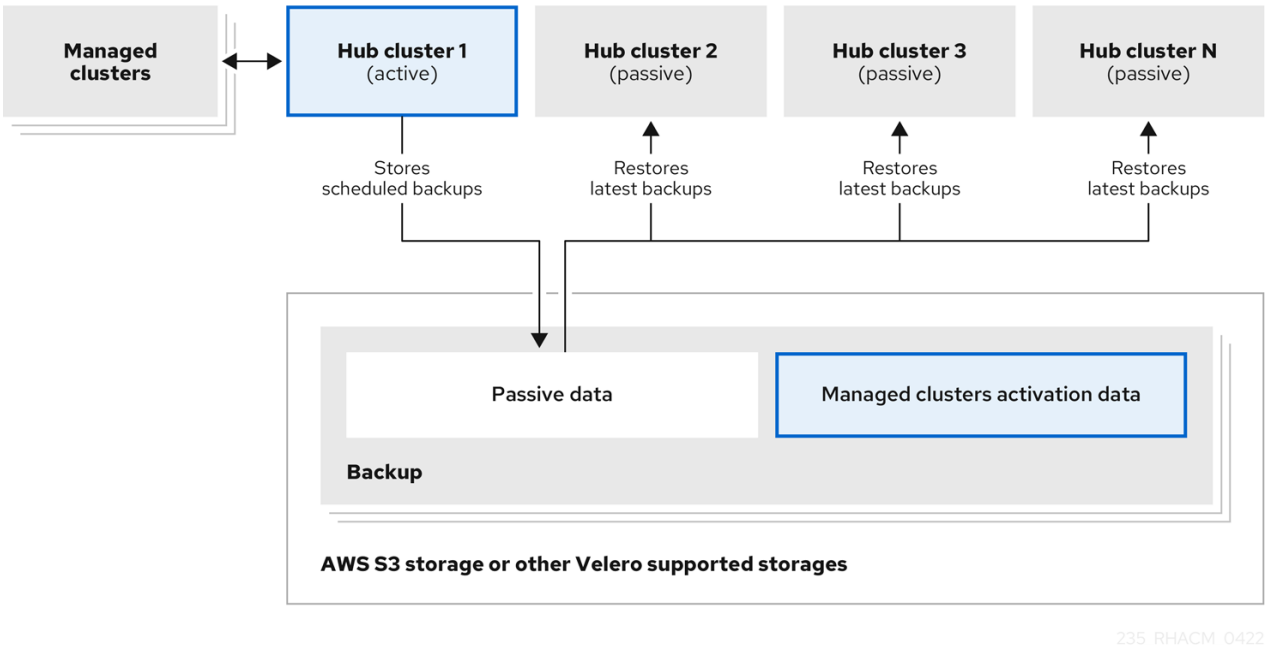
1.21.4. 主动被动配置

在主动被动配置中，有一个主动 hub 集群和被动 hub 集群。一个活跃 hub 集群也被视为主 hub 集群，它使用 **BackupSchedule.cluster.open-cluster-management.io** 资源以定义的时间间隔管理集群并备份资源。

被动 hub 集群会持续检索最新的备份并恢复被动数据。当有新的备份数据时，被动 hub 使用 **Restore.cluster.open-cluster-management.io** 资源从主 hub 集群恢复被动数据。当主 hub 集群停机时，这些 hub 集群处于备用状态，成为主 hub 集群。

主动和被动 hub 集群连接到相同的存储位置，主 hub 集群备份 被动 hub 集群的数据，以访问主 hub 集群。有关如何设置这个自动恢复配置的详情，请参阅 Restore 被动资源，同时检查备份部分。

在以下图中，活跃 hub 集群会管理本地集群并定期备份 hub 集群数据：



被动 hub 集群恢复这个数据，但受管集群激活数据除外，后者将受管集群移到 passive hub 集群。被动 hub 集群可以持续恢复被动数据，请参阅 Restore 被动资源，同时检查备份 部分。被动 hub 集群可以将被动数据恢复为一次性操作，请参阅 Restore passive resources 部分以了解更多详细信息。

1.21.4.1. 受管集群激活数据

受管集群激活数据或其他激活数据是一个备份资源。当在新的 hub 集群上恢复激活数据时，受管集群就会由运行恢复的 hub 集群主动管理。当使用 **cluster.open-cluster-management.io/backup: cluster-activation** 标签时，激活数据资源由受管集群备份和 resource-generic 备份存储。

1.21.4.2. 资源在受管激活时恢复

将 **cluster.open-cluster-management.io/backup: cluster-activation** 标签添加到资源时，资源会在 **acm-resources-generic-schedule** 备份资源中自动备份。当您在恢复资源中设置 **veleroManagedClustersBackupName:latest** 标签值时，通常需要恢复资源。如果在受管集群移动到新 hub 集群时需要恢复任何这些资源，请将 **veleroManagedClustersBackupName:latest** 标签值设置为 **cluster-activation**。这样可确保不会恢复资源，除非受管集群激活启动。

您的资源可能类似以下示例：

```
apiVersion: my.group/v1alpha1
kind: MyResource
metadata:
  labels:
    cluster.open-cluster-management.io/backup: cluster-activation
```

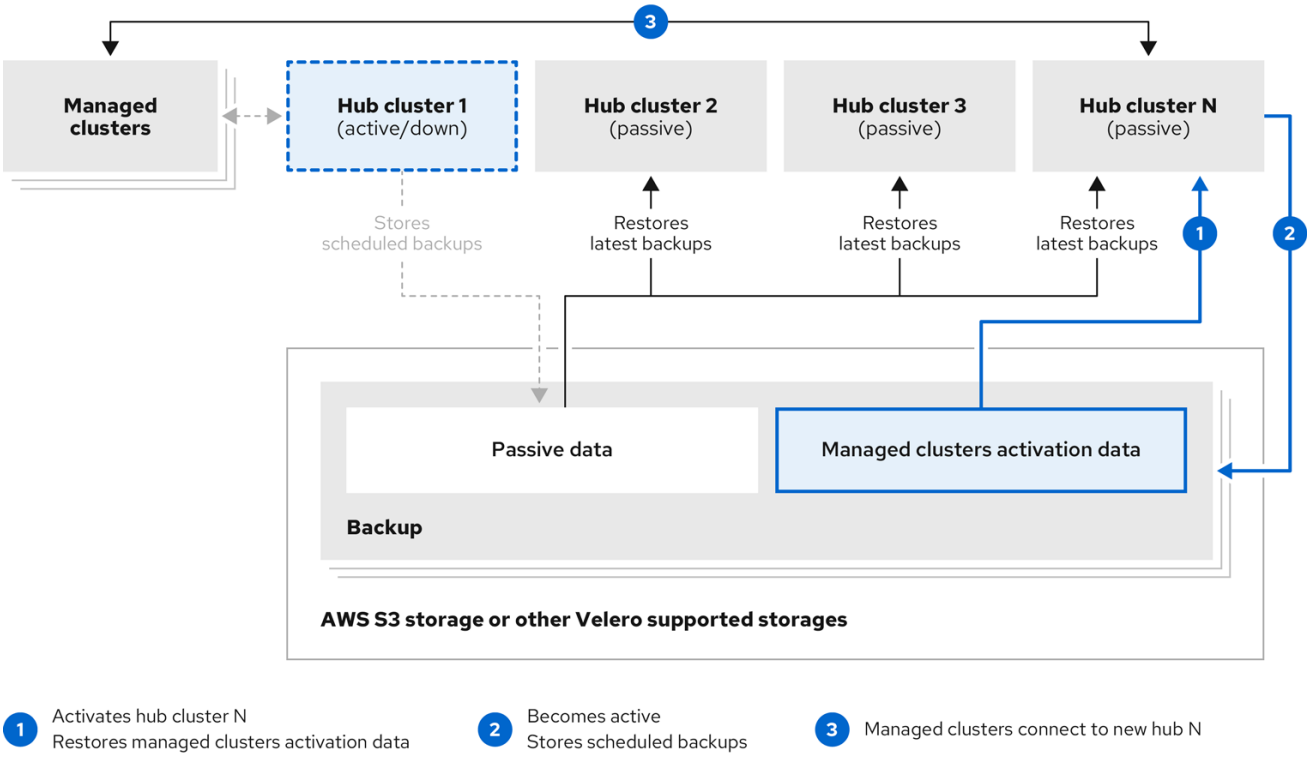
也支持由 **acm-managed-clusters-schedule** 资源备份的激活集合中也会有默认资源。查看由 **acm-managed-clusters-schedule** 资源恢复的以下默认资源：

- **managedcluster.cluster.open-cluster-management.io**
- **managedcluster.clusterview.open-cluster-management.io**

- **klusterletaddonconfig.agent.open-cluster-management.io**
- **managedclusteraddon.addon.open-cluster-management.io**
- **clusterpool.hive.openshift.io**
- **clusterclaim.hive.openshift.io**
- **clustercurator.cluster.open-cluster-management.io**
- **clustersync.hiveinternal.openshift.io**
- **baremetalhost.metal3.io**
- **bmceventsubscription.metal3.io**
- **hostfirmwaresettings.metal3.io**

1.21.5. 灾难恢复

当主 hub 集群停机时，管理员选择其中一个被动 hub 集群来接管受管集群。在以下镜像中，管理员决定将 *Hub 集群 N* 用作新的主 hub 集群：



hub 集群 N 恢复受管集群激活数据。此时，受管集群与 *Hub 集群 N* 连接。管理员通过在新的主 hub 集群（*Hub 集群 N*）上激活备份，方法是创建一个 **BackupSchedule.cluster.open-cluster-management.io** 资源，并将备份存储在与初始主 hub 集群相同的存储位置。

所有其他被动 hub 集群现在使用由新主 hub 集群创建的备份数据恢复被动数据。*Hub N* 现在是主 hub 集群，管理集群和备份数据。

1.21.6. 使用策略备份验证

集群备份和恢复 Operator Helm chart (**cluster-backup-chart**) 在 hub 集群上安装 **backup-restore-enabled** 策略，用于告知您备份和恢复组件的问题。**backup-restore-enabled** 策略包括一组用于检查以下限制的模板：

- Pod 验证
  以下模板检查备份组件和依赖项的 pod 状态：
  - **acm-backup-pod-running** 模板检查备份和恢复 operator pod 是否在运行。
  - **oadp-pod-running** 模板检查 OADP operator pod 是否在运行。
  - **velero-pod-running** 模板检查 Velero pod 是否在运行。
- **数据保护应用程序验证**
  - **data-protection-application-available** 模板检查是否创建了 **DataProtectioApplicatio.oadp.openshift.io** 资源。这个 OADP 资源设置 Velero 配置。
- **备份存储验证**
  - **backup-storage-location-available** 模板检查 **BackupStorageLocation.velero.io** 资源是否已创建以及状态值是否为 **Available**。这意味着与备份存储的连接有效。
- BackupSchedule **冲突验证**
  - 如果当前 hub 集群上存在 **BackupSchedule.cluster.open-cluster-management.io**，则 **acm-backup-clusters-collision-report** 模板会验证状态不是 **BackupCollision**。这会在将备份数据写入存储位置时，验证当前 hub 集群与其它 hub 集群不冲突。
    对于 **BackupCollision** 状态的定义，请参阅 Backup Collisions 部分 。
- BackupSchedule **和恢复状态验证**
  - **acm-backup-phase-validation** 模板检查当前集群中是否存在 **BackupSchedule.cluster.open-cluster-management.io**，则检查状态为 **Failed** 或 **Empty** 状态。这样可确保如果此集群是主 hub 集群，并正在生成备份，则 **BackupSchedule.cluster.open-cluster-management.io** 状态是健康。
  - 如果当前集群中存在 **Restore.cluster.open-cluster-management.io**，则相同的模板会检查当前集群中没有处于 **Failed** 或 **Empty** 状态的状态。这样可确保如果这个集群是二级 hub 集群，且被恢复备份，**Restore.cluster.open-cluster-management.io** 状态是健康。
- **备份存在验证**
  - **acm-managed-clusters-schedule-backups-available** 模板检查 **Backup.velero.io** 资源是否位于 **BackupStorageLocation.velero.io** 指定的位置上，以及备份是否由 **BackupSchedule.cluster.open-cluster-management.io** 资源创建。这验证了备份已至少运行一次，使用备份和恢复 Operator。
- **备份完成**
  - **acm-backup-in-progress-report** 模板检查 **Backup.velero.io** 资源是否处于 **InProgress** 状态。这个验证会被添加，因为带有大量资源，velero pod 会作为备份运行重启，备份会停留在不继续完成状态。在正常备份过程中，备份资源会在某一时间点进行，但不会被卡住并在完成运行。正常情况下，在调度运行时报告 **acm-backup-in-progress-report** 模板会在调度运行时报告警告并备份正在进行。
- **主动作为 cron 作业运行的备份**

- **BackupSchedule.cluster.open-cluster-management.io** 主动运行并在存储位置保存新的备份。此验证通过 **backup-schedule-cron-enabled** 策略模板来完成。模板检查是否有带有 **velero.io/schedule-name: acm-validation-policy-schedule** 标签的 **Backup.velero.io**。**acm-validation-policy-schedule** 备份设置为在为备份 cron 调度设定时间后过期。如果没有创建备份的 cron 作业，旧的 **acm-validation-policy-schedule** 备份将被删除，因为它过期且没有创建新的备份。因此，如果在任何时间点上没有 **acm-validation-policy-schedule backups**，这代表没有活跃的 cron 作业生成备份。

  此策略旨在帮助在 hub 集群活跃并生成或恢复备份时通知 hub 集群管理员。

了解如何启用和管理集群备份和恢复 Operator，请参阅管理 备份和恢复 Operator。

1.21.7. 管理备份和恢复 Operator

启用集群备份和恢复操作器，为集群资源调度备份和恢复。

需要的访问权限： 集群管理员

- 前提条件
- 启用备份和恢复 Operator
- 使用备份和恢复 Operator
- 查看恢复事件

1.21.7.1. 先决条件

对于主动和被动 hub 集群：

- 在 Red Hat OpenShift Container Platform 集群中，安装 Red Hat Advanced Cluster Management for Kubernetes operator 版本 2.5.x。安装 Red Hat Advanced Cluster Management 时会自动创建 **MultiClusterHub** 资源，并显示以下状态： **Running**。
- 集群备份和恢复 Operator 必须手动安装。启用集群备份和恢复 Operator (**cluster-backup**)。通过将 **cluster-backup** 参数设置为 **true** 来编辑 **MultiClusterHub** 资源。这将使用 **cluster-backup** 资源在同一命名空间中安装 OADP operator。

对于被动 hub 集群：

- 在被动 hub 集群上运行恢复操作前，您必须手动配置 hub 集群，并在活跃 hub 集群上安装所有 Operator，以及与活跃 hub 集群相同的命名空间中。
- 确保 Red Hat Advanced Cluster Management Operator 安装在与初始 hub 集群相同的命名空间中。然后，创建 **DataProtectionApplication** 资源，并连接到初始 hub 集群备份数据的相同存储位置。查看以下 **DataProtectionApplication** 资源示例：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-sample
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
```

```
      - aws
    restic:
      enable: true
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: my-bucket
          prefix: my-prefix
        config:
          region: us-east-1
          profile: "default"
        credential:
          name: cloud-credentials
          key: cloud
  snapshotLocations:
    - name: default
      velero:
        provider: aws
        config:
          region: us-west-2
          profile: "default"
```

- 在运行恢复操作前，请验证已安装了其他 Operator，如 Ansible Automation Platform、Red Hat OpenShift Container Platform GitOps 或证书管理器。这样可确保新 hub 集群配置与初始 hub 集群相同。

- 安装备份和恢复 Operator 时，被动 hub 集群必须使用与初始 hub 集群相同的命名空间名称，以及在之前的 hub 集群上配置的任何其他 Operator。

1.21.7.2. 启用备份和恢复 Operator

当第一次创建 **MultiClusterHub** 资源时，可以启用集群备份和恢复 Operator。**cluster-backup** 参数设为 **true**。启用 Operator 后，会安装 operator 资源。

如果已创建了 **MultiClusterHub** 资源，您可以通过编辑 **MultiClusterHub** 资源来安装或卸载集群备份 Operator。如果要卸载集群备份 Operator，将 **cluster-backup** 设置为 **false**。

启用备份和恢复 Operator 时，**MultiClusterHub** 资源可能类似以下 YAML 文件：

```
apiVersion: operator.open-cluster-management.io/v1
  kind: MultiClusterHub
  metadata:
    name: multiclusterhub
    namespace: open-cluster-management
  spec:
    availabilityConfig: High
    enableClusterBackup: false
    imagePullSecret: multiclusterhub-operator-pull-secret
    ingress:
      sslCiphers:
        - ECDHE-ECDSA-AES256-GCM-SHA384
        - ECDHE-RSA-AES256-GCM-SHA384
        - ECDHE-ECDSA-AES128-GCM-SHA256
```

```
    - ECDHE-RSA-AES128-GCM-SHA256
overrides:
  components:
    - enabled: true
      name: multiclusterhub-repo
    - enabled: true
      name: search
    - enabled: true
      name: management-ingress
    - enabled: true
      name: console
    - enabled: true
      name: insights
    - enabled: true
      name: grc
    - enabled: true
      name: cluster-lifecycle
    - enabled: true
      name: volsync
    - enabled: true
      name: multicluster-engine
    - enabled: false
      name: cluster-proxy-addon
    - enabled: true <<<<<<<<
      name: cluster-backup
separateCertificateManagement: false
```

1.21.7.3. 使用备份和恢复 Operator

完成以下步骤以调度和恢复备份：

1. 使用备份和恢复 operator，**backupschedule.cluster.open-cluster-management.io** 和 **restore.cluster.open-cluster-management.io** 资源，使用 **cluster_v1beta1_backupschedule.yaml** 示例文件创建 **backupschedule.cluster.open-cluster-management.io** 资源。请参阅 **cluster-backup-operator** 示例。运行以下命令，使用 **cluster_v1beta1_backupschedule.yaml** 示例文件创建 **backupschedule.cluster.open-cluster-management.io** 资源：

   ```
   kubectl create -n <oadp-operator-ns> -f
   config/samples/cluster_v1beta1_backupschedule.yaml
   ```

   您的资源可能类似以下文件：

   ```
   apiVersion: cluster.open-cluster-management.io/v1beta1
   kind: BackupSchedule
   metadata:
     name: schedule-acm
   spec:
     veleroSchedule: 0 */6 * * * # Create a backup every 6 hours
     veleroTtl: 72h # deletes scheduled backups after 72h; optional, if not specified, the
   maximum default value set by velero is used - 720h
   ```

   查看 **backupschedule.cluster.open-cluster-management.io spec** 属性的描述：

   - **veleroSchedule** 是必需属性，定义用于调度备份的 cron 作业。

- **veleroTtl** 是可选属性，定义调度的备份资源的过期时间。如果没有指定，则使用 Velero 设置的最大默认值，即 **720h**。

2. 检查 **backupschedule.cluster.open-cluster-management.io** 资源的状态，这会显示三个 **schedule.velero.io** 资源的定义。运行以下命令：

```
oc get bsch -n <oadp-operator-ns>
```

3. 提醒，恢复操作在不同的 hub 集群上运行，用于恢复场景。要启动恢复操作，请在要恢复备份的 hub 集群中创建一个 **restore.cluster.open-cluster-management.io** 资源。
您可以使用集群备份和恢复 Operator，**backupschedule.cluster.open-cluster-management.io** 和 **restore.cluster.open-cluster-management.io** 资源来创建备份或恢复资源。请参阅 **cluster-backup-operator** 示例。

4. 运行以下命令，使用 **cluster_v1beta1_restore.yaml** 示例文件创建 **restore.cluster.open-cluster-management.io** 资源。确保将 **oadp-operator-ns** 替换为用于安装 OADP Operator 的命名空间名称。OADP Operator 安装命名空间的默认值为 **oadp-operator** ：

```
kubectl create -n <oadp-operator-ns> -f config/samples/cluster_v1beta1_restore.yaml
```

您的资源可能类似以下文件：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
spec:
  veleroManagedClustersBackupName: latest
  veleroCredentialsBackupName: latest
  veleroResourcesBackupName: latest
```

查看以下 **restore.cluster.open-cluster-management.io** 的三个所需 **spec** 属性的描述：

- **veleroManagedClustersBackupName** 用于定义受管集群协调数据的恢复选项。
- **veleroCredentialsBackupName** 用于为用户凭证定义 restore 选项。
- **veleroResourcesBackupName** 用于定义 hub 集群资源的 restore 选项（**Applications**、**Policy** 及其他 hub 资源，如受管集群被动数据）。
前面提到的属性的有效选项有以下值：
- **latest** - 此属性恢复此类型的备份文件。
- **skip** - 此属性不会尝试使用当前恢复操作恢复这种类型的备份。
- **backup_name** - 此属性通过引用名称来恢复指定的备份。

5. 运行以下命令来查看 Velero **Restore** 资源：

```
oc get restore.velero.io -n <oadp-operator-ns>
```

查看以下 YAML 示例以恢复不同类型的备份文件：

- 恢复所有三种备份资源：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
spec:
  veleroManagedClustersBackupSchedule: latest
  veleroCredentialsBackupSchedule: latest
  veleroResourcesBackupSchedule: latest
```

- 仅恢复受管集群资源：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
spec:
  veleroManagedClustersBackupName: latest
  veleroCredentialsBackupName: skip
  veleroResourcesBackupName: skip
```

- 使用 **acm-managed-clusters-schedule-20210902205438** 备份只为受管集群恢复资源：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
spec:
  veleroManagedClustersBackupName: acm-managed-clusters-schedule-
20210902205438
  veleroCredentialsBackupName: skip
  veleroResourcesBackupName: skip
```

备注:

  - **restore.cluster.open-cluster-management.io** 资源运行一次。恢复操作完成后，您可以选择在同一 hub 集群中运行另一个恢复操作。您必须创建新的 **restore.cluster.open-cluster-management.io** 资源才能运行新的恢复操作。
  - 您可以创建多个 **restore.cluster.open-cluster-management.io**，但在任何时候都只能运行一个。

1.21.7.4. 查看恢复事件

使用以下命令获取有关恢复事件的信息：

```
oc describe -n <oadp-n> <restore-name>
```

您的事件列表可能类似以下示例：

```
Spec:
  Cleanup Before Restore:             CleanupRestored
  Restore Sync Interval:              4m
  Sync Restore With New Backups:      true
  Velero Credentials Backup Name:     latest
```

```
  Velero Managed Clusters Backup Name:  skip
  Velero Resources Backup Name:         latest
Status:
  Last Message:                 Velero restores have run to completion, restore will continue to sync
with new backups
  Phase:                        Enabled
  Velero Credentials Restore Name:  example-acm-credentials-schedule-20220406171919
  Velero Resources Restore Name:    example-acm-resources-schedule-20220406171920
Events:
  Type    Reason                  Age   From              Message
  ----    ------                  ----  ----              -------
  Normal  Prepare to restore:     76m   Restore controller  Cleaning up resources for backup acm-
credentials-hive-schedule-20220406155817
  Normal  Prepare to restore:     76m   Restore controller  Cleaning up resources for backup acm-
credentials-cluster-schedule-20220406155817
  Normal  Prepare to restore:     76m   Restore controller  Cleaning up resources for backup acm-
credentials-schedule-20220406155817
  Normal  Prepare to restore:     76m   Restore controller  Cleaning up resources for backup acm-
resources-generic-schedule-20220406155817
  Normal  Prepare to restore:     76m   Restore controller  Cleaning up resources for backup acm-
resources-schedule-20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-credentials-schedule-
20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-resources-generic-
schedule-20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-resources-schedule-
20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-credentials-cluster-
schedule-20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-credentials-hive-schedule-
20220406155817
  Normal  Prepare to restore:     64m   Restore controller  Cleaning up resources for backup acm-
resources-schedule-20220406165328
  Normal  Prepare to restore:     62m   Restore controller  Cleaning up resources for backup acm-
credentials-hive-schedule-20220406165328
  Normal  Prepare to restore:     62m   Restore controller  Cleaning up resources for backup acm-
credentials-cluster-schedule-20220406165328
  Normal  Prepare to restore:     62m   Restore controller  Cleaning up resources for backup acm-
credentials-schedule-20220406165328
  Normal  Prepare to restore:     62m   Restore controller  Cleaning up resources for backup acm-
resources-generic-schedule-20220406165328
  Normal  Velero restore created:  61m   Restore controller  example-acm-credentials-cluster-
schedule-20220406165328
  Normal  Velero restore created:  61m   Restore controller  example-acm-credentials-schedule-
20220406165328
  Normal  Velero restore created:  61m   Restore controller  example-acm-resources-generic-
schedule-20220406165328
  Normal  Velero restore created:  61m   Restore controller  example-acm-resources-schedule-
20220406165328
  Normal  Velero restore created:  61m   Restore controller  example-acm-credentials-hive-schedule-
20220406165328
  Normal  Prepare to restore:     38m   Restore controller  Cleaning up resources for backup acm-
resources-generic-schedule-20220406171920
  Normal  Prepare to restore:     38m   Restore controller  Cleaning up resources for backup acm-
resources-schedule-20220406171920
  Normal  Prepare to restore:     36m   Restore controller  Cleaning up resources for backup acm-
```

```
credentials-hive-schedule-20220406171919
  Normal  Prepare to restore:     36m   Restore controller  Cleaning up resources for backup acm-
credentials-cluster-schedule-20220406171919
  Normal  Prepare to restore:     36m   Restore controller  Cleaning up resources for backup acm-
credentials-schedule-20220406171919
  Normal  Velero restore created:  36m   Restore controller  example-acm-credentials-cluster-
schedule-20220406171919
  Normal  Velero restore created:  36m   Restore controller  example-acm-credentials-schedule-
20220406171919
  Normal  Velero restore created:  36m   Restore controller  example-acm-resources-generic-
schedule-20220406171920
  Normal  Velero restore created:  36m   Restore controller  example-acm-resources-schedule-
20220406171920
  Normal  Velero restore created:  36m   Restore controller  example-acm-credentials-hive-schedule-
20220406171919
```

有关所需规格属性和有效选项的描述，请参阅 恢复备份。