



Red Hat Advanced Cluster Management for Kubernetes 2.10

安装

安装

安装

法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

摘要

有关在连接和断开连接的网络环境中，对安装、多集群高级配置的要求和推荐信息，以及升级和卸载的信息。

目录

第 1 章 安装和升级	3
1.1. 性能和可扩展性	3
1.2. 在线安装	13
1.3. 在断开连接的网络环境中安装	18
1.4. MULTICLUSTERHUB 高级配置	22
1.5. 升级	28
1.6. 在断开连接的环境中升级	29
1.7. 使用策略升级断开连接的集群	30
1.8. 卸装	40

第 1 章 安装和升级

在安装前，请查看每个产品所需的硬件和系统配置。您可以使用受支持的 Red Hat OpenShift Container Platform 版本在 Linux 上在线安装。

1. 您必须有受支持的 OpenShift Container Platform 版本。例如，您可以在 AWS 或 Red Hat OpenShift Dedicated 上使用 Red Hat OpenShift Service。
2. 您必须为 Kubernetes operator 安装多集群引擎。

FIPS 注意：如果您没有在 `spec.ingress.sslCiphers` 中指定自己的密码，则 `multiclusterhub-operator` 会提供默认密码列表。如果您升级并希望 FIPS 合规性，请从 `multiclusterhub` 资源中删除以下两个密码：`ECDHE-ECDSA-CHACHA20-POLY1305` 和 `ECDHE-RSA-CHACHA20-POLY1305`。

访问 [支持列表](#)，了解 hub 集群和受管集群要求和支持。

安装 Red Hat Advanced Cluster Management for Kubernetes 来设置一个多节点集群生产环境。您可以使用标准配置或高可用性配置安装 Red Hat Advanced Cluster Management for Kubernetes。有关安装过程的更多信息，请参阅以下文档：

- [计划集群大小](#)
- [性能和可扩展性](#)
- [在线安装](#)
- [在断开连接的网络中安装](#)
- [MultiClusterHub 高级配置](#)
- [升级](#)
- [在断开连接的环境中升级](#)
- [使用策略升级断开连接的集群](#)
- [卸装](#)

1.1. 性能和可扩展性

Red Hat Advanced Cluster Management for Kubernetes 已经过测试来决定特定的可扩展性和性能数据。测试的主要领域包括集群可扩展性和搜索性能。您可以在规划环境时使用此信息。

备注：数据基于测试时实验室环境的结果。

Red Hat Advanced Cluster Management 使用裸机上的三个节点 hub 集群进行测试。测试时，有足够的资源容量(CPU、内存和磁盘)来查找软件组件限制。您的具体结果可能会根据您的环境、网络速度和产品更改而有所不同。

- [受管集群的总数](#)
- [搜索可扩展性](#)
- [Observability 可扩展性](#)
- [备份和恢复可扩展性](#)

1.1.1. 受管集群的总数

Red Hat Advanced Cluster Management 可管理的最大集群数量因以下几个因素而有所不同：

- 集群中的资源数量，它取决于部署的策略和应用程序数量等因素。
- hub 集群配置，如使用了多少个 pod 进行扩展。

受管集群是托管在 Red Hat Enterprise Linux hypervisor 上的单节点 OpenShift 虚拟机。虚拟机用于实现测试中每个裸机的高密度计数。sushy-emulator 用于虚拟机 libvirt，以便使用 Redfish API 访问的裸机集群。以下 Operator 是测试安装、Topology Aware Lifecycle Manager、Local Storage Operator 和 Red Hat OpenShift GitOps 的一部分。下表显示了实验环境扩展信息：

表 1.1. 环境扩展表

节点	数量	操作系统	硬件	CPU 内核	内存	磁盘
hub 集群 control plane	3	OpenShift Container Platform	裸机	112	512 GiB	446 GB SSD, 2.9 TB NVMe, 2 x 1.8 TB SSD
受管集群 (managed cluster)	3500	单节点 OpenShift	虚拟机器	8	18 GiB	120 GB

1.1.2. 搜索可扩展性

Search 组件的可扩展性取决于数据存储的性能。在分析搜索性能时，查询运行时间是一个重要的变量。

1.1.2.1. 查询运行时注意事项

有些事情可能会影响查询运行时间以及返回的结果。在计划和配置环境时请考虑以下方面：

- 搜索关键字效率不高。
如果您搜索 **RedHat** 且管理大量集群，可能需要更长的时间来接收搜索结果。
- 第一次搜索需要更长的时间，因为收集基于用户的访问控制规则需要额外的时间。
- 完成请求的时间长度与用户有权访问的命名空间和资源数量成比例。
注：如果您与另一个用户保存并共享搜索查询，返回的结果会根据用户的访问级别而不同。如需有关角色访问权限的更多信息，请参阅 OpenShift Container Platform [文档中的使用 RBAC 定义和应用权限](#)。
- 经过观察，当一个有访问所有命名空间或所有受管集群权限的、非管理员用户发出的请求性能最差。

1.1.3. Observability 可扩展性

如果要启用和使用可观察性（observability）服务，则需要规划您的环境。之后的资源消耗是用于安装可观察性组件的 OpenShift Container Platform 项目。您计划使用的值为所有可观察性组件的总和。

备注：数据基于测试时实验室环境的结果。您的具体结果可能会根据您的环境、网络速度和产品更改而有所不同。

1.1.3.1. 可观察性环境示例

在示例环境中，hub 集群和受管集群位于 Amazon Web Services 云平台中，并具有以下拓扑和配置：

节点	Flavor	vCPU	RAM (GiB)	磁盘类型	磁盘大小 (GiB)	数量	区域
Master 节点	m5.4xlarge	16	64	gp2	100	3	sa-east-1
Worker 节点	m5.4xlarge	16	64	gp2	100	3	sa-east-1

可观察性部署是为高可用性环境配置的。使用高可用性环境，每个 Kubernetes 部署都有两个实例，每个有状态集 (StatefulSet) 都有三个实例。

在示例测试过程中，会模拟不同的受管集群来推送指标，每次测试会持续 24 小时。请参见以下吞吐量：

1.1.3.2. 写入吞吐量

Pods	间隔 (分钟)	每分钟的时间系列
400	1	83000

1.1.3.3. CPU 用量 (millicores)

在测试过程中，CPU 用量是稳定的：

Size	CPU 用量
10 个集群	400
20 个集群	800

1.1.3.4. RSS 和工作集合内存

查看以下 RSS 和工作集合内存描述：

- **内存用量 RSS:** 来自 metrics `container_memory_rss`，在测试过程中保持稳定状态。
- **内存用量工作集：** 来自 metrics `container_memory_working_set_bytes`，随着测试会增加。

以下来自于一个 24 小时测试的结果：

Size	内存用量 RSS	内存用量工作集
10 个集群	9.84	4.93

Size	内存用量 RSS	内存用量工作集
20 个集群	13.10	8.76

1.1.3.5. 用于 thanos-receive 组件的持久性卷

重要：指标数据存储于 **thanos-receive** 中，直到达到了保留时间（四天）为止。其他组件不需要与 **thanos-receive** 组件一样多的卷。

磁盘用量随着测试会增加。数据代表一天后的磁盘用量，因此最终的磁盘用量要乘以 4。

请查看以下磁盘用法：

Size	磁盘用量 (GiB)
10 个集群	2
20 个集群	3

1.1.3.6. 网络传输

在测试过程中，网络传输提供了稳定性。查看大小和网络传输值：

Size	入站网络传输	出站网络传输
10 个集群	每秒 6.55 MBs	每秒 5.80 MBs
20 个集群	每秒 13.08 MBs	每秒 10.9 MBs

1.1.3.7. Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3) 中的总使用量会增加。指标数据会存储在 S3 中，直到达到默认的保留时间（5 天）。请查看以下磁盘用法：

Size	磁盘用量 (GiB)
10 个集群	16.2
20 个集群	23.8

1.1.4. 备份和恢复可扩展性

在大规模环境中执行的测试显示了备份和恢复的以下数据：

表 1.2. 受管集群备份的运行时间表

Backups	Duration	资源数量	备份内存
credentials	2m5s	18272 资源	55MiB 备份大小
受管集群	3m22s	58655 资源	38MiB 备份大小
resources	1m34s	1190 资源	1.7MiB 备份大小
generic/user	2m56s	0 个资源	16.5KiB 备份大小

备份总时间为 **10m**。

表 1.3. 恢复被动 hub 集群的运行时间表

Backups	Duration	资源数量
redentials	47m8s	18272 资源
resources	3m10s	1190 资源
generic/user 备份	0m	0 个资源

总恢复时间为 **50m18s**。

在创建 **BackupSchedule** 时，使用 **veleroTtl** 参数选项集来修剪备份文件数量。任何创建时间超过指定 TTL（生存时间）的备份都会过期，并自动从 Velero 的存储位置中删除。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name:schedule-acm
  namespace:open-cluster-management-backup
spec:
  veleroSchedule:0 */1 * * *
  veleroTtl:120h
```

1.1.5. 计划集群大小

每个 Red Hat Advanced Cluster Management for Kubernetes 集群都是唯一的，以下指南为您提供部署大小示例。根据大小和目的对推荐进行分类。

Red Hat Had Advanced Cluster Management 应用以下部分来调整支持服务的大小和位置：

- **可用域 (Availability Zone)** 用来在集群中分离潜在的故障域。典型的集群在三个或更多可用区中具有接近 worker 节点容量。
- **vCPU 保留 (reservation) 和限制 (limit)** 在 worker 节点上建立 vCPU 容量以分配给一个容器。vCPU 等于 Kubernetes 计算单元。如需更多信息，请参阅 Kubernetes 中 [CPU 的意义](#)。
- **内存保留和限制** 会在 worker 节点上建立内存容量，以便分配给容器。

- 持久性数据由产品管理，并存储在 Kubernetes 使用的 etcd 集群中。

重要: 对于 OpenShift Container Platform，在三个可用区间分发集群的主节点。

1.1.5.1. 产品环境

注：以下要求不是最低要求。

表 1.4. 产品环境

节点类型	可用区	etcd	保留内存总量	保留 CPU 总量
Master	3	3	OpenShift Container Platform 大小指南	OpenShift Container Platform 大小指南
worker 或基础架构	3	1	12 GB	6

除了 Red Hat Advanced Cluster Management，OpenShift Container Platform 集群还运行其他服务来支持集群功能。详情请参阅在[基础架构节点上安装 Red Hat Advanced Cluster Management hub 集群](#)部分。

1.1.5.1.1. OpenShift Container Platform on additional services

可用域 (Availability Zone) 用来在集群中分离潜在的故障域。

表 1.5. Additional services

服务	节点数	可用区	实例大小	vCPU	memory	存储大小	Resources
Amazon Web Services 上的 OpenShift Container Platform	3	3	m5.xlarge	4	16 GB	120 GB	<p>如需更多信息，请参阅 OpenShift Container Platform 产品文档中的使用自定义在 AWS 上安装集群。</p> <p>同时还可以参阅 与机器类型相关的详细信息。</p>

服务	节点数	可用区	实例大小	vCPU	memory	存储大小	Resources
Google Cloud Platform 上的 OpenShift Container Platform	3	3	N1-standard-4 (0.95–6.5 GB)	4	15 GB	120 GB	<p>有关配额的更多信息，请参阅 Google Cloud Platform 产品文档。</p> <p>同时还可以参阅与机器类型相关的详细信息。</p>
OpenShift Container Platform on Microsoft Azure	3	3	Standard_D4_v3	4	16 GB	120 GB	如需了解更多详细信息，请参阅 OpenShift Container Platform 文档中的配置 Azure 帐户 。
OpenShift Container Platform on VMware vSphere	3	3		4（每个插槽 2 个内核）	16 GB	120 GB	详情请查看以下 产品文档 。
IBM Z 系统的 OpenShift Container Platform	3	3		10	16 GB	100 GB	<p>如需更多信息，请参阅 OpenShift Container Platform 文档中的在 IBM Z 系统上安装集群。</p> <p>IBM Z 系统提供配置并发多线程 (SMT) 的</p>

服务	节点数	可用区	实例大小	vCPU	memory	存储大小	功能, 可扩展每个内核上运行的 vCPU 数量。如果您配置了 SMT, 则一个物理内核 (IFL) 提供两个逻辑内核 (线程)。管理程序可以提供两个或多个 vCPU。 当未启用并发多线程 (SMT) 或超线程时, 一个 vCPU 等于一个物理内核。启用后, 使用以下公式来计算对应的比例: (每个内核数的线程) × sockets = vCPU。 有关 SMT 的更多信息, 请参阅 Simultaneous 多线程 。
IBM Power 系统上的 OpenShift Container Platform	3	3		16	16 GB	120 GB	如需更多信息, 请参阅 OpenShift Container Platform 文档中的在 Power 系统上安装集群 。

服务	节点数	可用区	实例大小	vCPU	memory	存储大小	IBM Power Systems 系统提供配置并发多线程 (SMT) 的功能, 可扩展每个内核上运行的 vCPU 数量。如果您配置了 SMT, 则您的 SMT 级别决定如何满足 16 个 vCPU 的要求。最常见的配置有: 在 SMT-8 上运行的两个内核 (运行 IBM Power 虚拟机的系统的默认配置) 提供了所需的 16 个 vCPU。 在 SMT-4 上运行的四个内核提供所需的 16 个 vCPU。 有关 SMT 的更多信息, 请参阅 Simultaneous 多线程 。

服务	节点数	可用区	实例大小	vCPU	memory	存储大小	Resources
OpenShift Container Platform 内部	3			4	16 GB	120 GB	<p>详情请查看以下 产品文档。</p> <p>OpenShift Container Platform 裸机上可安装并支持 Red Hat Advanced Cluster Management for Kubernetes hub 集群。hub 集群可以在紧凑的裸机拓扑上运行，其中有 3 个可调度的 control plane 节点，以及 0 个额外的 worker。</p>

1.1.5.1.2. 创建和管理单一节点 OpenShift Container Platform 集群

查看 [在单个节点上安装](#) 以了解要求。由于每个集群都是唯一的，因此以下准则只提供了按大小和目的分类的部署要求示例。

可用域 (Availability Zone) 用来在集群中分离潜在的故障域。典型的集群在三个或更多可用区中具有相等的 worker 节点容量。不支持高可用性。

重要: 对于 OpenShift Container Platform，在三个可用区间分发集群的主节点。

请参阅 [创建和管理 3500 个单节点 OpenShift Container Platform 集群的示例要求](#)。请参阅 [使用 Red Hat Advanced Cluster Management 创建单节点 OpenShift 集群的最低要求\(230 和更多置备\)](#)，以及 [使用 hub 集群管理这些单节点 OpenShift 集群](#)：

表 1.6. Master (可调度)

节点数	内存（集群使用情况）	内存（单一节点 min-max）	CPU 集群	CPU 单一节点
3	289 GB	64 GB - 110 GB	90	44

1.2. 在线安装

您可以通过 Operator Lifecycle Manager 安装 Red Hat Advanced Cluster Management for Kubernetes，它管理安装、升级和删除包含 Red Hat Advanced Cluster Management hub 集群的组件。

需要的访问权限：集群管理员。**OpenShift Container Platform Dedicated 环境需要访问权限：**您必须具有 **cluster-admin** 权限。默认情况下，**dedicated-admin** 角色没有在 OpenShift Container Platform Dedicated 环境中创建命名空间所需的权限。

- 默认情况下，hub 集群组件安装在 OpenShift Container Platform 集群的 worker 节点上，无需额外配置。您可以使用 OpenShift Container Platform OperatorHub Web 控制台界面或使用 OpenShift Container Platform CLI 将 hub 集群安装到 worker 节点上。
- 如果您使用基础架构节点配置了 OpenShift Container Platform 集群，您可以使用带有其他资源参数的 OpenShift Container Platform CLI 将 hub 集群安装到这些基础架构节点上。详情请参阅 [在基础架构节点上安装 Red Hat Advanced Cluster Management hub 集群](#) 部分。
- 如果您计划导入不是由 OpenShift Container Platform 或 Red Hat Advanced Cluster Management 创建的 Kubernetes 集群，则需要配置镜像 pull secret。

有关如何配置高级配置的详情，请查看文档中的 [MultiClusterHub 高级配置](#) 部分中的选项。

- [先决条件](#)
- [确认 OpenShift Container Platform 安装](#)
- [从 OperatorHub Web 控制台界面安装](#)
- [通过 OpenShift Container Platform CLI 安装](#)
- [在基础架构节点上安装 Red Hat Advanced Cluster Management hub 集群](#)

1.2.1. 先决条件

在安装 Red Hat Advanced Cluster Management 前，请查看以下要求：

- 您的 Red Hat OpenShift Container Platform 集群必须通过 OpenShift Container Platform 控制台访问 OperatorHub 目录中的 Red Hat Advanced Cluster Management operator。
- 您需要访问 catalog.redhat.com。
- OpenShift Container Platform 版本 4.13 或更高版本必须部署到您的环境中，且必须通过 OpenShift Container Platform CLI 登录。如需了解 OpenShift Container Platform 的以下安装文档，并根据需要更改为更早的版本，[OpenShift Container Platform 版本 4.13](#)
- 您的 OpenShift Container Platform 命令行界面（CLI）被配置为运行 **oc** 命令。如需有关安装和配置 OpenShift Container Platform CLI 的信息，[请参阅 CLI 入门](#)。
- OpenShift Container Platform 权限必须允许创建命名空间。如果没有命名空间，安装将失败。

- 需要有一个互联网连接来访问 Operator 的依赖项。
- **重要**：要在 OpenShift Container Platform Dedicated 环境中安装，请参阅以下要求：
 - 您必须已配置并运行了 OpenShift Container Platform Dedicated 环境。
 - 您必须在要安装 hub 集群的 OpenShift Container Platform Dedicated 环境中具有 **cluster-admin** 授权。
 - 要导入，您必须为 2.10 使用 klusterlet operator 的 **stable-2.0** 频道。

1.2.2. 确认 OpenShift Container Platform 安装

您必须有一个受支持的 OpenShift Container Platform 版本，包括 registry 和存储服务，并可以正常工作。有关安装 OpenShift Container Platform 的更多信息，请参阅 OpenShift Container Platform 文档。

1. 验证 OpenShift Container Platform 集群中是否尚未安装 Red Hat Advanced Cluster Management hub 集群。Red Hat Advanced Cluster Management 只允许在每个 OpenShift Container Platform 集群上安装一个 Red Hat Advanced Cluster Management hub 集群。如果没有安装 Red Hat Advanced Cluster Management hub 集群，请继续执行以下步骤。
2. 要确保正确设置 OpenShift Container Platform 集群，请使用以下命令访问 OpenShift Container Platform Web 控制台：

```
kubectl -n openshift-console get route
```

请参见以下示例输出：

```
openshift-console console console-openshift-console.apps.new-coral.purple-chesterfield.com
console https reencrypt/Redirect None
```

3. 在浏览器中打开 URL 并检查结果。如果控制台 URL 显示 **console-openshift-console.router.default.svc.cluster.local**，当安装 OpenShift Container Platform 时把 **openshift_master_default_subdomain** 设置为这个值。请参阅以下 URL 示例：
<https://console-openshift-console.apps.new-coral.purple-chesterfield.com>。

您可以使用控制台或 CLI 继续运行 Red Hat Advanced Cluster Management。这两个流程都已被记录在文档中。

1.2.3. 从 OperatorHub Web 控制台界面安装

最佳实践：从 OpenShift Container Platform 导航中的 *Administrator* 视图，安装 OpenShift Container Platform 提供的 OperatorHub Web 控制台界面。

1. 选择 **Operators > OperatorHub** 以访问可用操作器列表，然后选择 *Advanced Cluster Management for Kubernetes operator*。
2. 在 *Operator subscription* 页中，选择安装选项：
 - 命名空间信息：
 - Red Hat Advanced Cluster Management hub 集群必须安装在自己的命名空间或项目中。

- 默认情况下，OperatorHub 控制台安装过程会创建一个名为 **open-cluster-management** 的命名空间。**最佳实践**：继续使用 **open-cluster-management** 命名空间（如果可用）。
- 如果已有一个名为 **open-cluster-management** 的命名空间，请选择不同的命名空间。
- Channel：选择与要安装的发行版本相对应的频道。当您选择频道时，它会安装指定的发行版本，并确定以后获得该发行版本中的勘误更新。
- 更新的批准策略：批准策略指定了用户需要如何处理应用到您的频道或发行版本的更新。
 - 选择 **Automatic** 以确保该版本内的任何更新被自动应用。
 - 选择 **Manual** 在有更新可用时接收通知。如果您对更新的应用有疑问，这可能是您的最佳实践。

重要信息：要升级到下一个次版本，您必须返回到 *OperatorHub* 页面，并为更当前的发行版本选择一个新频道。

3. 选择 **Install** 以应用您的更改并创建 Operator。
4. 创建 *MultiClusterHub* 自定义资源。
 - a. 在 OpenShift Container Platform 控制台导航中，选择 **Installed Operators > Advanced Cluster Management for Kubernetes**。
 - b. 选择 **MultiClusterHub** 标签页。
 - c. 选择 **Create MultiClusterHub**。
 - d. 更新 YAML 文件中的默认值。请参阅文档中的 *MultiClusterHub* **高级配置** 部分中的选项。
 - 下例显示了默认模板。确认 **namespace** 是项目的命名空间。请参阅以下示例：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
```

5. 选择 **Create** 来初始化自定义资源。Red Hat Advanced Cluster Management hub 集群最多可能需要 10 分钟才能构建和启动。创建 Red Hat Advanced Cluster Management hub 集群后，**MultiClusterHub** 资源状态会在 Red Hat Advanced Cluster Management operator 的 *MultiClusterHub* 标签页中显示 *Running*。要获取控制台的访问权限，请参阅[访问控制台](#)主题。

1.2.4. 通过 OpenShift Container Platform CLI 安装

1. 创建一个包含 Operator 要求的 Red Hat Advanced Cluster Management hub 集群命名空间。运行以下命令，其中 **namespace** 是 Red Hat Advanced Cluster Management hub 集群命名空间的名称。在 OpenShift Container Platform 环境中，**namespace** 的值可能被称为 *Project*（项目）。

```
oc create namespace <namespace>
```

- 将项目命名空间切换到您创建的命名空间。使用在第 1 步中创建的 Red Hat Advanced Cluster Management hub 集群命名空间的名称替换 **namespace**。

```
oc project <namespace>
```

- 创建 YAML 文件来配置 **OperatorGroup** 资源。每个命名空间只能有一个 operator 组。将 **default** 替换为 operator 组的名称。将 **namespace** 替换为项目命名空间的名称。请参见以下示例：

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <default>
  namespace: <namespace>
spec:
  targetNamespaces:
  - <namespace>
```

- 运行以下命令来创建 **OperatorGroup** 资源。将 **operator-group** 替换为您创建的 operator 组 YAML 文件的名称：

```
oc apply -f <path-to-file>/<operator-group>.yaml
```

- 创建 YAML 文件来配置 OpenShift Container Platform 订阅。文件内容类似以下示例，将 **release-2.x** 替换为当前版本：

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: acm-operator-subscription
spec:
  sourceNamespace: openshift-marketplace
  source: redhat-operators
  channel: release-2.x
  installPlanApproval: Automatic
  name: advanced-cluster-management
```

注：要在基础架构节点上安装 Red Hat Advanced Cluster Management hub 集群，请参阅 [Operator Lifecycle Manager 订阅附加配置](#) 部分。

- 运行以下命令来创建 OpenShift Container Platform 订阅。使用您创建的订阅文件的名称替换 **subscription**：

```
oc apply -f <path-to-file>/<subscription>.yaml
```

- 创建一个 YAML 文件来配置 **MultiClusterHub** 自定义资源。您的默认模板应当类似于以下示例。将 **namespace** 替换为项目命名空间的名称：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec: {}
```

注：要在基础架构节点上安装 Red Hat Advanced Cluster Management hub 集群，请参阅 [MultiClusterHub 自定义资源附加配置](#) 部分：

- 运行以下命令来创建 **MultiClusterHub** 自定义资源。将 **custom-resource** 替换为自定义资源文件的名称：

```
oc apply -f <path-to-file>/<custom-resource>.yaml
```

如果此步骤失败并显示以下错误，则仍然会创建并应用这些资源。创建资源后几分钟内再次运行命令：

```
error: unable to recognize "/.mch.yaml": no matches for kind "MultiClusterHub" in version "operator.open-cluster-management.io/v1"
```

- 运行以下命令来获取自定义资源。在运行命令后，**MultiClusterHub** 自定义资源状态可能需要最多 10 分钟才能在 **status.phase** 字段中显示为 **Running**：

```
oc get mch -o=jsonpath='{.items[0].status.phase}'
```

如果您重新安装 Red Hat Advanced Cluster Management，且 pod 没有启动，请参阅 [对安装失败进行故障排除](#) 以了解解决这个问题的步骤。

备注：

- 具有 **ClusterRoleBinding** 的 **ServiceAccount** 会自动向 Red Hat Advanced Cluster Management 以及有权访问安装 Red Hat Advanced Cluster Management 的命名空间的用户凭证授予集群管理员特权。
- 安装还会创建一个名为 **local-cluster** 的命名空间，该命名空间在由自身管理时为 Red Hat Advanced Cluster Management hub 集群保留。因此，不能已存在名为 **local-cluster** 的命名空间。为安全起见，请不要将 **local-cluster** 命名空间的访问权限授予任何尚未具有 **cluster-administrator** 访问权限的用户。

1.2.5. 在基础架构节点上安装 Red Hat Advanced Cluster Management hub 集群

OpenShift Container Platform 集群可以配置为包含用于运行批准的管理组件的基础架构节点。在基础架构节点上运行组件可避免为运行这些管理组件的节点分配 OpenShift Container Platform 订阅配额。

将基础架构节点添加到 OpenShift Container Platform 集群后，请遵循 [OpenShift Container Platform CLI 指令的安装](#)，并将配置添加到 Operator Lifecycle Manager 订阅和 **MultiClusterHub** 自定义资源中。

1.2.5.1. 将基础架构节点添加到 OpenShift Container Platform 集群

按照 OpenShift Container Platform 文档中的 [创建基础架构机器集](#) 中所述的步骤进行操作。基础架构节点配置有 Kubernetes 污点 (**taint**) 和标签 (**label**)，以便防止非管理工作负载在它们上运行。

要与 Red Hat Advanced Cluster Management 提供的基础架构节点启用兼容，请确保您的基础架构节点应用了以下污点和标签：

```
metadata:
  labels:
    node-role.kubernetes.io/infra: ""
spec:
```

```
taints:
- effect: NoSchedule
  key: node-role.kubernetes.io/infra
```

1.2.5.2. Operator Lifecycle Manager Subscription 额外配置

在应用 Operator Lifecycle Manager 订阅前，添加以下配置：

```
spec:
  config:
    nodeSelector:
      node-role.kubernetes.io/infra: ""
  tolerations:
  - key: node-role.kubernetes.io/infra
    effect: NoSchedule
    operator: Exists
```

1.2.5.3. MultiClusterHub 自定义资源附加配置

在应用 **MultiClusterHub** 自定义资源前添加以下附加配置：

```
spec:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
```

1.3. 在断开连接的网络环境中安装

您可能需要在断开连接的环境中安装 Red Hat Advanced Cluster Management for Kubernetes。要在断开连接的 hub 集群上安装，除了用于连接网络环境的常规安装或升级步骤外，还执行下列步骤。

需要的访问权限： 需要所有安装和升级任务的 **集群管理** 访问权限。

请参见以下部分：

- [先决条件](#)
- [确认 OpenShift Container Platform 安装](#)
- [配置 Operator Lifecycle Manager](#)
- [配置镜像内容源策略](#)
- [安装 Red Hat Advanced Cluster Management for Kubernetes operator 和 hub](#)

1.3.1. 先决条件

在安装 Red Hat Advanced Cluster Management for Kubernetes 前，需要满足以下要求：

- 由于您在断开连接的网络环境中安装，您需要访问本地镜像 registry 以存储镜像 Operator Lifecycle Manager 目录和 Operator 镜像。在此环境中安装 OpenShift Container Platform 集群时，您可能已经设置了一个本地镜像 registry，因此应该可以使用相同的本地镜像 registry。
- 您必须有一个可以访问互联网和您的本地镜像 registry 的工作站。

- 支持的 Red Hat OpenShift Container Platform 版本必须部署到您的环境中，且必须使用 CLI 登录。有关安装 Red Hat [OpenShift Container Platform](#) 的信息，请参阅 [OpenShift Container Platform 版本 4.11 安装文档](#)。有关使用 Red Hat OpenShift [CLI 安装和配置 oc 命令](#) 的信息，请参阅 [CLI 入门](#)。
- 请参阅 [调整集群大小](#) 以了解如何为您的 hub 集群设置容量。

1.3.2. 确认 OpenShift Container Platform 安装

- 连接后，运行 `oc -n openshift-console get route` 命令来访问 OpenShift Container Platform Web 控制台。请参见以下示例输出：

```
openshift-console      console      console-openshift-console.apps.new-coral.purple-
chesterfield.com      console      https reencrypt/Redirect  None
```

在浏览器中打开 URL 并检查结果。如果控制台 URL 显示 `console-openshift-console.router.default.svc.cluster.local`，当安装 OpenShift Container Platform 时把 `openshift_master_default_subdomain` 设置为这个值。

1.3.3. 确认本地镜像 registry 的可用性

最佳实践： 将现有的镜像 registry 用于 Operator Lifecycle Manager Operator 相关内容。

在断开连接的环境中安装 Red Hat Advanced Cluster Management for Kubernetes 涉及使用本地镜像 registry。由于您已在断开连接的环境中完成 OpenShift Container Platform 集群的安装，所以您已设置了一个镜像 registry，以便在 Red Hat OpenShift Container Platform 集群安装过程中使用。

如果您还没有本地镜像 registry，请完成以下步骤，为 Red Hat OpenShift Container Platform 文档的 [断开连接的安装 mirror 镜像中所述的步骤](#) 创建一个。

1.3.4. 配置 Operator Lifecycle Manager

因为 Red Hat Advanced Cluster Management for Kubernetes 打包为 Operator，所以使用 Operator Lifecycle Manager 进行安装。

在断开连接的环境中，Operator Lifecycle Manager 无法访问红帽提供的 Operator 的标准 Operator 源，因为它们托管在无法从断开连接的集群访问的镜像 registry 上。集群管理员可以使用镜像 registry 和 Operator 目录在断开连接的环境中启用安装和升级 Operator。

要准备断开连接的集群以安装 Red Hat Advanced Cluster Management for Kubernetes，请按照 OpenShift Container Platform 文档中 [在受限网络中使用 Operator Lifecycle Manager](#) 中所述的步骤进行。

1.3.4.1. 额外要求

完成前面的步骤后，请注意以下特定于 Red Hat Advanced Cluster Management for Kubernetes 的要求：

1.3.4.1.1. 在镜像目录中包含 operator 软件包

- 在镜像目录中包括所需的 Operator 软件包。红帽在 Red Hat operator 目录中提供了 Red Hat Advanced Cluster Management for Kubernetes operator，它由 `registry.redhat.io/redhat/redhat-operator-index` 索引镜像提供。当您准备此目录索引镜像的镜像时，您可以选择将镜像整个目录作为红帽提供的镜像，或者镜像只包含您要使用的 Operator 软件包的子集。

如果您要创建完整镜像目录，则不需要特别考虑安装 Red Hat Advanced Cluster Management for Kubernetes 所需的所有软件包。但是，如果您要创建部分或过滤了镜像目录，您要为其识别要包含的特定软件包，则需要在您的列表中包含以下软件包名称：

- **advanced-cluster-management**
- **multicluster-engine**
- 使用两个镜像程序中的一个。
- 如果您使用 OPM 实用程序、**opm index prune** 创建镜像的目录或 registry，请在 **-p** 选项的值中包含以下软件包名称，如以下示例所示，使用当前版本替换 **4.x**：

```
opm index prune \
  -f registry.redhat.io/redhat/redhat-operator-index:v4.x \
  -p advanced-cluster-management,multicluster-engine \
  -t myregistry.example.com:5000/mirror/my-operator-index:v4.x
```

- 如果您使用 **oc-mirror** 插件填充镜像的目录或 registry，请在 **ImageSetConfiguration** 的 **packages** 列表中包含以下软件包名称，如下例所示，使用当前版本替换 **4.x**：

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: myregistry.example.com:5000/mirror/oc-mirror-metadata
  mirror:
    platform:
      channels:
        - name: stable-4.x
      type: ocp
    operators:
      - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.11
      packages:
        - name: advanced-cluster-management
        - name: multicluster-engine
    additionalImages: []
  helm: {}
```

1.3.4.1.2. 配置以使用您的镜像 registry

当您填充了安装 Red Hat Advanced Cluster Management for Kubernetes 所需的较早软件包的本地镜像 registry 时，请完成 [在受限网络中使用 Operator Lifecycle Manager](#) 中描述的步骤，以便在断开连接的环境中提供您的镜像 registry 和目录，其中包括以下步骤：

1. [禁用默认的 OperatorHub 源](#)
2. [对 Operator 目录进行镜像\(mirror\)](#)
3. [为镜像目录添加目录源](#)

1.3.4.1.3. 查找目录源名称

如 Red Hat OpenShift Container Platform 文档中的步骤所述，您需要在断开连接的集群中添加 **CatalogSource** 资源。**重要信息**：请注意 **metadata.name** 字段的值，稍后您将需要。

使用类似以下示例的 YAML 文件将 **CatalogSource** 资源添加到 **openshift-marketplace** 命名空间中，将 **4.x** 替换为当前版本：

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: my-mirror-catalog-source
  namespace: openshift-marketplace
spec:
  image: myregistry.example.com:5000/mirror/my-operator-index:v4.x
  sourceType: grpc
```

在稍后创建的 **MulticloudHub** 资源中需要注解的 **metadata.name** 字段值。

1.3.5. 验证所需的软件包可用

Operator Lifecycle Manager 定期轮询可用软件包的目录源。Operator Lifecycle Manager 轮询镜像目录的目录源后，您可以通过查询可用的 **PackageManifest** 资源来验证在断开连接的集群中是否有所需软件包可用。

运行以下命令，定向至断开连接的集群：

```
oc -n openshift-marketplace get packagemanifests
```

此时会显示的列表应该包括以下软件包由您的镜像目录的目录源提供：

- **advanced-cluster-management**
- **multicloud-engine**

1.3.6. 配置镜像内容源策略

为了让集群从您的镜像 registry 获取 Red Hat Advanced Cluster Management for Kubernetes Operator 的容器镜像，而不是从互联网托管的 registry 中配置 **ImageContentSourcePolicy**，以便在断开连接的环境中配置 **ImageContentSourcePolicy**，以将镜像引用重定向到您的镜像 registry。

如果使用 **oc adm catalog mirror** 命令对目录进行镜像，所需的镜像内容源策略配置位于由该命令创建的 **manifests-*** 目录的 **imageContentSourcePolicy.yaml** 文件中。

如果您使用 **oc-mirror** 插件替代您的目录，则 **imageContentSourcePolicy.yaml** 文件位于 **oc-mirror-workspace/results-*** 目录中。

在这两种情况下，您可以使用 **oc apply** 或 **oc replace** 命令将策略应用到断开连接的命令中，例如：

```
oc replace -f ./<path>/imageContentSourcePolicy.yaml
```

所需的镜像内容源策略语句会根据您创建镜像 registry 的方式而有所不同，但与以下示例类似：

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  labels:
    operators.openshift.org/catalog: "true"
  name: operator-0
```

```
spec:
  repositoryDigestMirrors:
  - mirrors:
    - myregistry.example.com:5000/rhacm2
    source: registry.redhat.io/rhacm2
  - mirrors:
    - myregistry.example.com:5000/multicluster-engine
    source: registry.redhat.io/multicluster-engine
  - mirrors:
    - myregistry.example.com:5000/openshift4
    source: registry.redhat.io/openshift4
  - mirrors:
    - myregistry.example.com:5000/redhat
    source: registry.redhat.io/redhat
```

1.3.7. 安装 Red Hat Advanced Cluster Management for Kubernetes operator 和 hub 集群

按照前所述配置了 Operator Lifecycle Manager 和 Red Hat OpenShift Container Platform，您可以使用 OperatorHub 控制台或 CLI 安装 Red Hat Advanced Cluster Management for Kubernetes。遵循[在连接的环境中安装](#)中介绍的相同指导。

重要： 创建 **MulticlusterHub** 资源是 hub 集群的安装过程开始。

因为集群上的 Operator 安装需要为镜像目录使用非默认目录源，所以 **MulticlusterHub** 资源中需要一个特殊的注解来向 Operator 提供镜像目录源的名称。以下示例显示了所需的 **mce-subscription-spec** 注解：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  namespace: open-cluster-management
  name: hub
  annotations:
    installer.open-cluster-management.io/mce-subscription-spec: '{"source": "my-mirror-catalog-source"}'
spec: {}
```

需要 **mce-subscription-spec** 注解，因为 multicluster engine operator 在 Red Hat Advanced Cluster Management 安装过程中被自动安装。如果您要使用 CLI 创建资源，请在使用 **oc apply** 命令应用的 YAML 中包含 **mce-subscription-spec** 注解来创建 **MulticlusterHub** 资源。

如果使用 OperatorHub 控制台创建资源，切换到 *YAML 视图* 并插入前面显示的注解。**重要：** OperatorHub 控制台中的字段在 *Field 视图* 面板中没有注解的字段来创建 **MulticlusterHub**。

1.4. MULTICLUSTERHUB 高级配置

Red Hat Advanced Cluster Management for Kubernetes 会使用一个会部署所有需要的组件的 operator 进行安装。一些列出的组件会被默认启用。如果一个组件 *被禁用*，则该资源不会在启用前部署到集群中。Operator 的工作原理来部署以下组件：

表 1.7. 部署的组件表列表

Name	描述	Enabled

app-lifecycle	统一和简化用于构建和部署应用程序及应用程序更新的选项。	True
cluster-backup	为所有 hub 集群资源（如受管集群、应用程序和策略）提供备份和恢复支持。	False
cluster-lifecycle	为 OpenShift Container Platform 和 Red Hat Advanced Cluster Management hub 集群提供集群管理功能。	True
cluster-permission	自动将 RBAC 资源分发到受管集群并管理这些资源的生命周期。	True
控制台	启用 Red Hat Advanced Cluster Management Web 控制台插件。	True
grc	为集群定义策略启用安全增强。	True
insights	识别集群中的现有或潜在问题。	True
multicluster-observability	启用监控以深入了解受管集群的健康状态。	True
search	为所有集群中的 Kubernetes 资源提供可见性。	True
submariner-addon	启用环境中两个或多个受管集群（内部环境或云中）之间的直接网络和服务发现。	True
volsync	支持异步复制集群中的持久性卷，或使用存储类型不兼容进行复制的集群中。	True

当您将在 Red Hat Advanced Cluster Management 安装到集群中时，不会默认启用所有列出的组件。

您可以通过在 **MultiClusterHub** 自定义资源中添加一个或多个属性来进一步配置 Red Hat Advanced Cluster Management。继续阅读以了解有关您可以添加属性的信息。

1.4.1. 控制台和组件配置

以下示例显示了可用于启用或禁用组件的 **spec.overrides** 默认模板：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace> 1
```

```
spec:
  overrides:
    components:
      - name: <name> 2
        enabled: true
```

1. 使用项目的名称替换 **namespace**。
2. 使用组件的名称替换 **name**。

或者，您可以运行以下命令。将 **namespace** 替换为项目的名称，将 **name** 替换为组件的名称：

```
oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p='[{"op": "add", "path": "/spec/overrides/components/-", "value": {"name": "<name>", "enabled": true}}]'
```

注：当禁用**控制台**组件时，Red Hat OpenShift Container Platform 控制台被禁用。

1.4.2. 自定义镜像 Pull Secret

如果您计划导入不是由 OpenShift Container Platform 或 Red Hat Advanced Cluster Management 创建的 Kubernetes 集群，生成一个具有 OpenShift Container Platform pull secret 信息的 secret，以便从发布 registry 中访问授权内容。

OpenShift Container Platform 集群的 secret 要求由 OpenShift Container Platform 和 Red Hat Advanced Cluster Management 自动解决，因此如果您没有导入其他类型的 Kubernetes 集群，则不必创建 secret。您的 OpenShift Container Platform pull secret 与您的 Red Hat Customer Portal ID 相关联，在所有 Kubernetes 供应商中都是相同的。

重要：这些 secret 是特定于命名空间的，因此请确保处于用于 hub 集群的命名空间中。

1. 进入 cloud.redhat.com/openshift/install/pull-secret 以下载 OpenShift Container Platform pull secret 文件。
2. 点 **Download pull secret**。
3. 运行以下命令来创建 secret:

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

- 将 **secret** 替换为您要创建的 secret 的名称。
- 将 **namespace** 替换为项目命名空间，因为 secret 是特定于命名空间的。
- 将 **path-to-pull-secret** 替换为您下载的 OpenShift Container Platform pull secret 的路径。

以下示例显示，如果使用自定义 pull secret，要使用的 **spec.imagePullSecret** 模板。将 secret 替换为 pull secret 的名称：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
```

```
namespace: <namespace>
spec:
  imagePullSecret: <secret>
```

1.4.3. availabilityConfig

Red Hat Advanced Cluster Management hub 集群有两个可用功能：**High** 和 **Basic**。默认情况下，hub 集群的可用性为 **High**，hub 集群组件副本数为 **2**。它提供了对故障转移功能的支持，但消耗的资源数量比可用性为 **Basic**（副本数为**1**）的集群多。

重要： 如果您在单节点 OpenShift 集群中使用 multicluster engine operator，请将 **spec.availabilityConfig** 设置为 **Basic**。

以下示例显示了具有 **Basic** 可用性的 **spec.availabilityConfig** 模板：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  availabilityConfig: "Basic"
```

1.4.4. nodeSelector

您可以在 Red Hat Advanced Cluster Management hub 集群中定义一组节点选择器，以安装到集群中的特定节点。以下示例显示了将 Red Hat Advanced Cluster Management pod 分配给带有标签 **node-role.kubernetes.io/infra** 的节点的 **spec.nodeSelector**：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
```

1.4.5. 容限（tolerations）

您可以定义容限列表，以便 Red Hat Advanced Cluster Management hub 集群可以容忍在集群中定义的特定污点。

以下示例显示了与 **node-role.kubernetes.io/infra** 污点匹配的 **spec.tolerations**：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  tolerations:
```

```
- key: node-role.kubernetes.io/infra
  effect: NoSchedule
  operator: Exists
```

默认情况下，以上 infra-node 容限在 pod 上设置，而不在配置中指定任何容限。在配置中自定义容限会替换这个默认值。

1.4.6. disableHubSelfManagement

默认情况下，Red Hat Advanced Cluster Management hub 集群会自动导入和管理。此受管 hub 集群名为 **local-cluster**。指定 hub 集群是否管理本身在 **multiclusterengine** 自定义资源中的设置。在 Red Hat Advanced Cluster Management 中更改此设置会自动更改 **multiclusterengine** 自定义资源中的设置。

注：在管理多集群引擎 operator 集群的 Red Hat Advanced Cluster Management hub 集群中，任何早期的手动配置都会被此操作替代。

如果您不希望 Red Hat Advanced Cluster Management hub 集群管理自己，请将 **spec.disableHubSelfManagement** 的设置从 **false** 改为 **true**。如果该设置没有包括在定义自定义资源的 YAML 文件中，请添加它。hub 集群只能通过这个选项进行管理。

将这个选项设置为 **true** 并尝试手动管理 hub，会导致意外行为。

以下示例显示，如果要禁用 hub 集群自助管理功能，要使用的默认模板。将 **namespace** 替换为项目的名称：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  disableHubSelfManagement: true
```

要启用默认的 **local-cluster**，请将设置返回为 **false**，或删除此设置。

1.4.7. disableUpdateClusterImageSets

如果要确保所有集群都使用相同的发行镜像，您可以创建自己的自定义列表，以便在创建集群时可用的发行镜像。

在 [连接到管理可用发行镜像时](#)，请参阅[维护自定义发行镜像列表](#)，并设置 **spec.disableUpdateClusterImageSets** 属性，这将停止自定义镜像列表被覆盖。

以下示例显示了禁用集群镜像集更新的默认模板。将 **namespace** 替换为项目的名称：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  disableUpdateClusterImageSets: true
```

1.4.8. customCAConfigmap（已弃用）

默认情况下，Red Hat OpenShift Container Platform 使用 Ingress Operator 创建内部 CA。

以下示例显示了用于为 Red Hat Advanced Cluster Management 提供自定义 OpenShift Container Platform 默认入口 CA 证书的默认模板。使用项目的名称替换 **namespace**。将 **spec.customCAConfigmap** 值替换为 **ConfigMap** 的名称：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  customCAConfigmap: <configmap>
```

1.4.9. sslCiphers（已弃用）

默认情况下，Red Hat Advanced Cluster Management hub 集群包含所支持的 SSL 密码的完整列表。

以下示例显示了用于列出管理入口的 **sslCiphers** 模板的默认 **spec.ingress.sslCiphers** 模板。将 **namespace** 替换为项目的名称：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  ingress:
    sslCiphers:
      - "ECDHE-ECDSA-AES128-GCM-SHA256"
      - "ECDHE-RSA-AES128-GCM-SHA256"
```

1.4.10. ClusterBackup

enableClusterBackup 字段不再被支持，并被这个组件替代。

以下示例显示了用来启用 **ClusterBackup** 的 **spec.overrides** 默认模板。将 **namespace** 替换为项目的名称：

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  overrides:
    components:
      - name: cluster-backup
        enabled: true
```

或者，您可以运行以下命令。使用项目的名称替换 **namespace**。

```
oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p='[{"op": "add", "path": "/spec/overrides/components/-", "value": {"name": "cluster-backup", "enabled": true}}]'
```

1.5. 升级

您可以使用 Red Hat OpenShift Container Platform 控制台中的 operator 订阅设置来控制 Red Hat Advanced Cluster Management for Kubernetes 的升级。Operator Lifecycle Manager **operatorcondition** 可帮助控制如何升级版本。当使用 Operator 部署 Red Hat Advanced Cluster Management 的最初阶段，您可以进行以下选择：

重要： 升级只在前一个版本中被支持。您可以升级到下一个可用功能发行版本，但无法在升级过程中跳过发行版本。

- **Channel**：与您要安装的产品版本相对应的频道。初始频道设置通常是安装时可用的最当前的频道。
- **Approval**：指定是否需要在频道中批准更新，或者是否自动进行更新。
 - 如果设置为 **Automatic**，那么所选频道中的次要发行本(Errata)更新会在没有管理员干预的情况下部署。
 - 如果设置为 **Manual**，则每个更新到频道中的次发行版本(Errata)都需要管理员批准更新。

需要的访问权限： OpenShift Container Platform 管理员

您还可以使用 Operator 升级到 Red Hat Advanced Cluster Management 的最新版本时使用这些设置。完成以下步骤以升级 Operator:

重要： 您在频道选择中升级到更新的版本后无法恢复到更早的版本。您必须卸载 Operator，并使用更早的版本重新安装它才能使用以前的版本。登录您的 OpenShift Container Platform operator hub。

1. 在 OpenShift Container Platform 导航中，选择 **Operators > Installed operator**。
2. 选择 **Red Hat Advanced Cluster Management for Kubernetesoperator**。
3. 选择 *Subscription* 选项卡来编辑订阅设置。
4. 确保 *Upgrade Status* 被标记为 *Up to date*。此状态表示 Operator 处于所选频道中可用的最新版本。如果 *Upgrade Status* 表示升级处于待处理的状态，请完成以下步骤，将其更新至频道中可用的最新次版本：
 - a. 点 *Approval* 字段中的 **Manual** 设置来编辑值。
 - b. 选择 **Automatic** 来启用自动更新。
 - c. 选择 **Save** 提交您的更改。
 - d. 等待自动更新应用到 Operator。更新会自动将所需更新添加到所选频道的最新版本。当完成所有更新后，*Upgrade Status* 字段将显示 **Up to date**。
备注：**MultiClusterHub** 自定义资源最多需要 10 分钟才能完成升级。您可以输入以下命令来检查升级是否仍然在进行中：

```
oc get mch
```

在进行升级时，**Status** 字段会显示 **Updating**。升级完成后，**Status** 字段会显示 **Running**。

5. 现在，*Upgrade Status* 是 **Up to date**，点 *Channel* 字段中的值来编辑它。

6. 选择下一个可用功能发行版本的频道，但不会尝试跳过频道。
重要： Operator Lifecycle Manager `operatorcondition` 资源会在当前升级过程中检查以前的升级，并防止跳过版本。您可以检查相同的资源状态，以查看 `upgradable` 状态是否为 `true` 或 `false`。
7. 选择 **Save** 保存您的更改。
8. 等待自动升级完成。当升级到下一个功能版本后，会部署对频道中最新补丁版本的更新。
9. 如果需要升级到之后的版本，请重复步骤 7-9，直到 Operator 处于所需频道的最新级别。请确定为您的最终频道部署了所有补丁版本。
10. 可选：如果您希望以后在频道中的更新需要手动批准，将 `Approval` 设置为 **Manual**。

有关升级 Operator 的更多信息，请参阅 [OpenShift Container Platform 文档中的 Operator](#)。

1.5.1. 使用升级管理集群池

如果您要管理 [集群池（技术预览）](#)，则需要进一步配置，以便在升级后停止对这些集群池的自动管理。

在 `ClusterClaim metadata.annotations` 中设置 `cluster.open-cluster-management.io/createmanagercluster: "false"`。

除非更改此设置，否则所有现有集群声明会在升级时自动导入。

1.6. 在断开连接的环境中升级

在断开连接的环境中，请参阅升级 Red Hat Advanced Cluster Management for Kubernetes 的步骤和信息。

注： 此信息遵循 [升级](#) 中的步骤。查看该流程，然后查看以下信息：

在安装过程中，或升级可能会遇到与 Red Hat Advanced Cluster Management 和 multicluster engine operator 之间的独立关系相关的重要信息。如需了解在安装或升级过程中考虑的，请参阅 [在断开连接的网络环境中安装](#)。

与在连接的网络环境中升级一样，升级过程通过更改 Red Hat Advanced Cluster Management for Kubernetes 的 Operator Lifecycle Manager 订阅中的升级频道来启动到新版本升级频道。

但是，由于断开连接的环境特殊特性，您需要在更改更新频道以启动升级过程前解决以下镜像要求：

1. 确保在镜像目录中更新所需的软件包。
在安装过程中，或在以前的更新过程中，您创建了一个镜像目录和一个 registry，其中包含在断开连接的环境中安装 Red Hat Advanced Cluster Management for Kubernetes 的 operator 软件包和镜像。要升级，您需要更新镜像目录和 registry 来获取 Operator 软件包的更新版本。

与安装操作类似，您需要确保您的镜像目录和 registry 在要包含或更新的 Operator 列表中包含以下 Operator 软件包：

- **advanced-cluster-manager**
- **multicluster-engine**

2. 验证 `MutliclusterHub` 资源实例。
在安装或升级过程中，您创建了 `MulticlusterHub` 资源的实例，并因为断开连接的环境，您要为该资源添加了一个 `mce-subscription-spec` 注解。

如果您更新您的镜像目录和 registry 的步骤会导致 OpenShift Container Platform 集群上可用的名为 **CatalogSource** 的 CatalogSource 与之前使用的名称相同，则不需要更新 **MulticusterHub** 资源来更新 **mce-subscriptino-spec** 注解。

但是，如果更新您的镜像目录和 registry 的步骤会导致新创建的名为 **CatalogSource**，请更新 **MulticusterHub** 资源中的 **mce-subscription-spec** 注解以反映新的目录源名称。

1.6.1. 使用目录镜像升级

Red Hat Advanced Cluster Management 使用相关的多集群引擎 operator 功能来提供作为产品一部分交付的基础服务。Red Hat Advanced Cluster Management 会自动安装和管理所需的多集群引擎 operator 和 **MulticusterEngine** 资源实例，作为 hub 集群安装和升级的一部分。

在连接的网络环境中，集群管理员可以在没有特殊镜像目录和目录源的情况下安装或升级 Red Hat Advanced Cluster Management。但是，因为在断开连接的环境中安装任何 Operator Lifecycle Manager Operator 涉及使用特殊镜像目录和目录源，所以在安装后需要一些额外的步骤。

1. 更新用于填充镜像目录的步骤。

如果安装 Red Hat Advanced Cluster Management 镜像过程会创建 Red Hat Operator 目录的完整副本，则不需要特殊的镜像更新。刷新目录以为新 Operator 版本获取更新的内容。

但是，如果您的步骤填充了 *filtered* 的目录，您需要更新您的镜像流程，以确保在镜像目录中包括 **multicuster-engine** Operator 软件包（除了 **advanced-cluster-management** 软件包外）。

请参阅 [镜像目录部分中的 Include 所需的 Operator 软件包](#)，它提供了填充镜像目录时要使用的选项示例。更新您流程中使用的 operator-package 列表，以匹配这些新要求。

2. 更新 **MulticusterHub** 资源实例。

如在 [断开连接的网络环境](#) 主题中所述，当在断开连接的环境中安装或升级 hub 集群时，您需要一个 **MulticusterHub** 资源上的新注解。

最佳实践：更新 **MulticusterHub** 资源实例，在将 Operator Lifecycle Manager 订阅中的 Operator Lifecycle Manager 更新频道改为 **advanced-cluster-management** operator 软件包前包括所需的注解，以启动升级。在这个版本中，升级可以在没有延迟的情况下进行。

使用 **oc edit** 命令更新 **Multicusterub** 资源，以添加 **mce-subscription-spec** 注解，如下例所示：

```
metadata:
  annotations:
    installer.open-cluster-management.io/mce-subscription-spec: '{"source": "<my-mirror-catalog-source>"}
```

将示例中的 **<my-mirror-catalog-source>** 替换为位于 mirror 目录的 **openshift-marketplace** 命名空间中的 **CatalogSource** 资源的名称。

重要：如果您在添加注解前开始升级，升级会开始，但当 Operator 尝试在后台将订阅安装到 **multicuster-engine** 时，升级会停滞。**MulticusterHub** 资源的状态在此期间继续显示 **upgrading**。

要解决这个问题，请运行 **oc edit** 来添加 **mce-subscription-spec** 注解，如前面所示。

1.7. 使用策略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 策略搭配使用，以便在断开连接的环境中升级多个集群。

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. 使用控制台，通知您的集群有可用的升级。

- [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](#)

1.7.1. 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：

- 在 Red Hat OpenShift Container Platform 版本 4.13 或更高版本上运行的 hub 集群，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 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存储库的更多信息，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。

1.7.2. 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=4.13.2-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。

1.7.3. 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。

1.7.4. 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**。

1.7.5. 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[为镜像 registry 访问配置额外的信任存储](#)。
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。

1.7.6. 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> 1
  replicas: 1
  repository: ${LOCAL_REGISTRY}/ocp4/release
  graphDataImage: '<host_name>:<port>/cincinnati-graph-data-container' 2
```

1 1 将 `spec.registry` 值替换为镜像的本地断开连接 registry 的路径。

2 2 将 `spec.graphDataImage` 值替换为图形数据 init 容器的路径。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。

1.7.7. 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。**已弃用：PlacementRule**

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 *policy-mirror*。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. 在控制台中，选择 **Governance > Create policy**。
3. 将 **YAML** 开关设置为 *On* 以查看策略的 YAML 版本。
4. 删除 YAML 代码中的所有内容。
5. 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-mirror
  namespace: default
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
      apiVersion: policy.open-cluster-management.io/v1
      kind: ConfigurationPolicy
      metadata:
        name: policy-image-content-source-policy
      spec:
        object-templates:
        - complianceType: musthave
          objectDefinition:
            apiVersion: operator.openshift.io/v1alpha1
            kind: ImageContentSourcePolicy
            metadata:
              name: <your-local-mirror-name>
            spec:
              repositoryDigestMirrors:
                - mirrors:
                    - <your-registry> 1
                  source: registry.redhat.io
          ---
            apiVersion: policy.open-cluster-management.io/v1
            kind: PlacementBinding
            metadata:
              name: binding-policy-mirror
              namespace: default
            placementRef:
              name: placement-policy-mirror
              kind: PlacementRule
```

```

  apiGroup: apps.open-cluster-management.io
  subjects:
  - name: policy-mirror
    kind: Policy
    apiGroup: policy.open-cluster-management.io
  ---
  apiVersion: apps.open-cluster-management.io/v1
  kind: PlacementRule
  metadata:
    name: placement-policy-mirror
    namespace: default
  spec:
    clusterConditions:
    - status: "True"
      type: ManagedClusterConditionAvailable
    clusterSelector:
      matchExpressions:
      [] # selects all clusters if not specified

```

- 1 将 **your-registry** 替换为本地镜像存储库的路径。您可以通过输入 **oc adm release mirror** 命令来查找到本地镜像的路径。

6. 选择 **Enforce if supported**。

7. 选择 **Create** 来创建策略。

1.7.8. 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. 在控制台菜单中选择 **Governance > Create policy**。
2. 将 **YAML** 切换设置为 *On* 以查看策略的 YAML 版本。
3. 删除 **YAML** 代码中的所有内容。
4. 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-catalog
    namespace: default
  spec:
    disabled: false
    remediationAction: enforce
    policy-templates:
    - objectDefinition:
      apiVersion: policy.open-cluster-management.io/v1
      kind: ConfigurationPolicy
      metadata:
        name: policy-catalog
      spec:
        object-templates:
        - complianceType: musthave

```



```

    objectDefinition:
      apiVersion: config.openshift.io/v1
      kind: OperatorHub
      metadata:
        name: cluster
      spec:
        disableAllDefaultSources: true
- complianceType: musthave
    objectDefinition:
      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' ❶
        displayName: My Operator Catalog
        publisher: grpc
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-catalog
  namespace: default
placementRef:
  name: placement-policy-catalog
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-catalog
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-catalog
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
    [] # selects all clusters if not specified

```

❶ 将 **spec.image** 值替换为本地受限目录源镜像的路径。

5. 选择 **Enforce if supported**。

6. 选择 **Create** 来创建策略。

1.7.9. 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`。记录下这个值以便在以后的步骤中使用。
3. 在 hub 集群控制台菜单中，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-cluster-version
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards: null
    policy.open-cluster-management.io/categories: null
    policy.open-cluster-management.io/controls: null
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
      apiVersion: policy.open-cluster-management.io/v1
      kind: ConfigurationPolicy
      metadata:
        name: policy-cluster-version
      spec:
        object-templates:
        - complianceType: musthave
          objectDefinition:
            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 1
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-cluster-version
  namespace: default
placementRef:
  name: placement-policy-cluster-version
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-cluster-version
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-cluster-version
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
    [] # selects all clusters if not specified

```

- 1** 将 **objectDefinition.spec.upstream** 的值替换为 hub 集群 OpenShift Update Service 操作对象的路径。

您可以完成以下步骤来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。
7. 选择 **Enforce if supported**。
 8. 选择 **Create** 来创建策略。
 9. 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
```

1.7.10. 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. 登录到 Kubernetes operator 控制台的多集群引擎。
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。

1.7.11. 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。

1.7.12. 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。

1.8. 卸载

在卸载 Red Hat Advanced Cluster Management for Kubernetes 时，您会看到两个不同的卸载过程级别：*删除自定义资源* 和 *完整的 Operator 卸载*。卸载过程最多可能需要 20 分钟。

- 第一个级别是自定义资源移除，这是最基本的卸载类型，用于删除 **MultiClusterHub** 实例的自定义资源，但会保留其他所需的 Operator 资源。如果您计划使用相同的设置和组件重新安装，这个卸载级别很有用。
- 第二个级别是更完整的卸载，可删除大多数 Operator 组件，不包括自定义资源定义等组件。当您继续执行此步骤时，它会删除所有没有通过删除自定义资源而被删除的组件和订阅。在卸载后，您必须在重新安装自定义资源前重新安装 Operator。

1.8.1. 先决条件

在卸载 Red Hat Advanced Cluster Management hub 集群前，您必须分离所有由该 hub 集群管理的集群。分离仍由 hub 集群管理的所有集群，然后尝试再次卸载。

- 如果使用发现功能，在尝试卸载时可能会看到以下错误：

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

要禁用发现功能，请完成以下步骤：

- 从控制台导航到 **Discovered Clusters** 表，再单击 **Disable cluster discovery**。确认您要删除该服务。
- 您还可以使用终端。运行以下命令以禁用发现：

```
oc delete discoveryconfigs --all --all-namespaces
```

- 如果您附加了代理服务配置，您可能会看到以下信息：

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

- 要使用命令行界面禁用和删除 **AgentServiceConfig** 资源，请完成以下步骤：
 - a. 登录到您的 hub 集群。
 - b. 输入以下命令删除 **AgentServiceConfig** 自定义资源：

```
oc delete agentserviceconfig --all
```

- 如果附加了受管集群，您可能会看到以下信息。**注**：这包括 **local-cluster**，它是您自助管理的 hub 集群：

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

有关分离集群的更多信息，请参阅[从管理部分删除集群](#)，方法是在[集群创建](#)中选择您的供应商的信息。

- 如果您使用可观察性功能，可能会看到以下内容：

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

- 要使用终端禁用和删除 **MultiClusterObservability** 自定义资源，请参阅以下步骤：

- a. 登录到您的 hub 集群。
- b. 输入以下命令删除 **MultiClusterObservability** 自定义资源：

```
oc delete mco observability
```

- o 要使用控制台删除 **MultiClusterObservability** 自定义资源，请参阅以下流程：
 - a. 如果安装了 **MultiClusterObservability** 自定义资源，请选择 *MultiClusterObservability* 选项卡。
 - b. 选择 **MultiClusterObservability** 自定义资源的 **Options** 菜单。
 - c. 选择 **Delete MultiClusterObservability**。
当您删除资源时，Red Hat Advanced Cluster Management hub 集群上的 **open-cluster-management-observability** 命名空间中的 pod 以及所有受管集群上的 **open-cluster-management-addon-observability** 命名空间中的 pod 都会被删除。

注：删除 observability 服务后，您的对象存储不会受到影响。

1.8.2. 使用命令删除资源

1. 如果还没有运行 oc 命令，请确保 OpenShift Container Platform CLI 配置为运行 **oc** 命令。如需有关如何配置 **oc** 命令的更多信息，请参阅 OpenShift Container Platform 文档中的 [OpenShift CLI 入门](#)。
2. 输入以下命令来更改到您的项目命名空间。将 *namespace* 替换为项目命名空间的名称：

```
oc project <namespace>
```

3. 输入以下命令删除 **MultiClusterHub** 自定义资源：

```
oc delete multiclusterhub --all
```

4. 要查看进度，请输入以下命令：

```
oc get mch -o yaml
```

5. 运行清理脚本删除所有潜在的剩余工件。如果您计划在同一集群中使用旧版本的 Red Hat Advanced Cluster Management 重新安装，则运行此清理脚本。
 - a. 按照安装 Helm 中的说明，安装 Helm CLI 二进制版本 3.2.0 或更高版本。<https://helm.sh/docs/intro/install/>
 - b. 将以下脚本复制到一个文件中：

```
#!/bin/bash
ACM_NAMESPACE=<namespace>
oc delete mch --all -n $ACM_NAMESPACE
oc delete apiservice v1.admission.cluster.open-cluster-management.io
v1.admission.work.open-cluster-management.io
oc delete clusterimageset --all
oc delete clusterrole multiclusterengines.multicluster.openshift.io-v1-admin
multiclusterengines.multicluster.openshift.io-v1-crdview
multiclusterengines.multicluster.openshift.io-v1-edit
```

```

multiclusterengines.multicluster.openshift.io-v1-view open-cluster-
management:addons:application-manager open-cluster-management:admin-aggregate
open-cluster-management:cert-policy-controller-hub open-cluster-management:cluster-
manager-admin-aggregate open-cluster-management:config-policy-controller-hub open-
cluster-management:edit-aggregate open-cluster-management:iam-policy-controller-hub
open-cluster-management:policy-framework-hub open-cluster-management:view-
aggregate
oc delete crd klusterletaddonconfigs.agent.open-cluster-management.io
placementbindings.policy.open-cluster-management.io policies.policy.open-cluster-
management.io userpreferences.console.open-cluster-management.io
discoveredclusters.discovery.open-cluster-management.io
discoveryconfigs.discovery.open-cluster-management.io
oc delete mutatingwebhookconfiguration ocm-mutating-webhook
managedclustermutators.admission.cluster.open-cluster-management.io multicluster-
observability-operator
oc delete validatingwebhookconfiguration
channels.apps.open.cluster.management.webhook.validator application-webhook-
validator multiclusterhub-operator-validating-webhook ocm-validating-webhook
multicluster-observability-operator multiclusterengines.multicluster.openshift.io

```

将脚本中的 **<namespace>** 替换为安装 Red Hat Advanced Cluster Management 的命名空间的名称。

重要： 确保指定正确的命名空间，因为命名空间会被清理和删除。

- a. 运行该脚本删除所有从之前安装中保留的工件。如果没有剩余的工件，则会返回一个没有找到资源的信息。

注： 如果您计划重新安装相同的 Red Hat Advanced Cluster Management 版本，您可以跳过这个过程中的下一步并重新安装自定义资源。继续以完整 Operator 的卸载过程。

6. 输入以下命令在安装它的命名空间中删除 Red Hat Advanced Cluster Management **ClusterServiceVersion** 和 **Subscription**：将 **2.x.0** 值替换为当前的主版本或次版本：

```

oc get csv
NAME                                DISPLAY                                VERSION  REPLACES
PHASE
advanced-cluster-management.v2.x.0  Advanced Cluster Management for Kubernetes
2.x.0                                Succeeded

oc delete clusterserviceversion advanced-cluster-management.v2.x.0

oc get sub
NAME                                PACKAGE                                SOURCE                                CHANNEL
acm-operator-subscription          advanced-cluster-management          acm-custom-registry          release-
2.x

oc delete sub acm-operator-subscription

```

注： CSV 的订阅名称和版本可能有所不同。

1.8.3. 使用控制台删除组件

当使用 Red Hat OpenShift Container Platform 控制台卸载时，需要删除 operator。使用控制台完成以下步骤进行卸载：

1. 在 OpenShift Container Platform 控制台导航中，选择 **Operators > Installed Operators > Advanced Cluster Manager for Kubernetes。**
2. 删除 **MultiClusterHub** 自定义资源。
 - a. 选择 *Multiclusterhub* 标签页。
 - b. 选择 MultiClusterHub 自定义资源的 **Options** 菜单。
 - c. 选择 **Delete MultiClusterHub。**
3. 使用以下命令，根据[删除 MultiClusterHub 实例中的步骤](#)运行清理脚本。
备注：如果您计划重新安装相同的 Red Hat Advanced Cluster Management 版本，您可以跳过这个过程中的其余步骤并重新安装自定义资源。
4. 进入 **Installed Operators。**
5. 选择 *Options* 菜单并选择 **Uninstall operator** 来删除 *Red Hat Advanced Cluster Management。*