OpenShift Container Platform 4.4

更新集群

更新 OpenShift Container Platform 集群

Last Updated: 2021-03-11

OpenShift Container Platform 4.4 更新集群

更新 OpenShift Container Platform 集群

法律通告

摘要

本文档提供了有关更新和升级 OpenShift Container Platform 集群的信息。更新集群的过程较简单，可以在不需要使集群离线的情况下进行。

目录

第 1 章 在次版本间更新集群

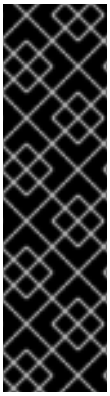
您可以在次版本（minor version）间更新或升级 OpenShift Container Platform 集群。

注意

由于使用 **oc**更改更新频道会比较困难，所以请使用 web 控制台来更改更新频道。建议您在 web 控制台中完成更新过程。在改到一个 4.4 频道后，按照使用 CLI 在一个次版本中更新集群中介绍的步骤进行操作。

1.1. 先决条件

- 使用具有 **admin** 权限的用户访问集群。请参阅使用 RBAC 定义并应用权限 。
- 请保存一个最新的 etcd backup。如果升级失败，则需要把集群恢复到一个以前的状态。



重要

如果您要从 OpenShift Container Platform 4.3.3 或更早版本升级到这个版本，则必须在升级完成后重启所有 Pod。

这是因为服务 CA 会在 OpenShift Container Platform 4.3.5 中自动轮转。升级过程中会轮转服务 CA，之后需要重启服务以确保所有服务在上一个服务 CA 过期前都使用新的服务 CA。

这个手动重启操作只需要执行一次，后续的升级和轮转将在服务 CA 过期前确保重启，而无需人工干预。

重要

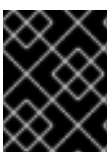
使用 **unsupportedConfigOverrides** 部分修改 Operator 配置不受支持，并可能会阻止集群升级。您必须先删除此设置才能升级集群。

1.2. 关于 OPENSHIFT CONTAINER PLATFORM 更新服务

OpenShift Container Platform 更新服务是一种托管服务，为 OpenShift Container Platform 和 Red Hat Enterprise Linux CoreOS (RHCOS) 提供无线更新 (over-the air update)。它提供了一个组件 Operator 图，其中包含各个*顶点*及连接它们的*边*。图中的边显示可以安全更新的版本，顶点是更新有效负载，用于指定托管集群组件的预期状态。

集群中的 Cluster Version Operator (CVO) 会检查 OpenShift Container Platform 更新服务，并根据当前组件版本和图中的信息决定有效的更新和更新路径。当您请求更新时，OpenShift Container Platform CVO 使用该更新的发行镜像来升级您的集群。发行工件 (artifact) 作为容器镜像托管在 Quay 中。

为了使 OpenShift Container Platform 更新服务仅提供兼容的更新，提供了一个版本验证管道来驱动自动化。每个发行工件都会被验证是否与支持的云平台和系统架构以及其他组件包兼容。在管道确认有适用的版本后，OpenShift Container Platform 更新服务会通知您可以进行更新。



重要

因为更新服务会显示所有有效的更新，所以不能强制更新到一个更新服务没有显示的版本。

对于连续更新模式，会运行两个控制器。一个控制器不断更新有效负载清单，将它们应用于集群，并输出受控 Operator 部署的状态（可用、正在进行升级或失败）。第二个控制器轮询 OpenShift Container Platform 更新服务以确定更新是否可用。

重要

不支持将集群还原到以前的版本或执行回滚。仅支持升级到较新版本。

在升级过程中，Machine Config Operator (MCO) 会将新配置应用到集群机器。它将机器配置池中由 **maxUnavailable** 字段指定数量的节点保护起来，并将其标记为不可用。在默认情况下，这个值被设置为 **1**。然后，它会应用新配置并重启机器。如果您将 Red Hat Enterprise Linux (RHEL) 机器用作 worker，MCO 不会在这些机器上更新 kubelet，因为您必须首先在这些机器上更新 OpenShift API。因为新版本的规格被应用到旧的 kubelet，所以 RHEL 机器无法返回 **Ready** 状态。在机器可用前，您无法完成更新。但是，通过设置不可用节点的最大数量可以确保当不可用机器的数量没有超过这个值时，正常的集群操作仍然可以继续进行。

其他资源

- 非受管 Operator 的支持策略

1.3. OPENSHIFT CONTAINER PLATFORM 升级频道和发行版本

在 OpenShift Container Platform 4.1 中，红帽引进了升级频道的概念，用于为集群升级推荐适当的版本。通过控制升级的速度，这些升级频道允许您选择升级策略。升级频道与 OpenShift Container Platform 的次要版本关联。例如，OpenShift Container Platform 4.4 升级频道永远不会包括到版本 4.5 的升级。这可确保管理员明确决定升级到下一个 OpenShift Container Platform 次要版本。升级频道仅控制版本选择，它不会影响您安装的集群版本; 特定版本的 OpenShift Container Platform 的 **openshift-install** 二进制文件始终会安装这个特定版本。

OpenShift Container Platform 4.4 提供了以下升级频道：

- **candidate-4.4**
- **fast-4.4**
- **stable-4.4**

candidate-4.4 频道

Candidate-4.4 频道包含 z-stream (4.4.z) 发行版本的候选构建。发行候选版本包含该产品的所有功能但不被正式支持。发行候选版本可以用来测试新版本的功能以决定下一个 OpenShift Container Platform 版本是否适用于您的系统。发行候选是指候选频道中的一个构建，包括那些名称中没有 **-rc** 的构建。当一个版本出现在候选频道中后，它仍然会进行更多的质量测试。如果达到质量标准，则会将其推广至 **fast-4.4** 或 **stable-4.4** 频道。因此，如果一个特定的版本同时存在于 **candidate-4.4**频道以及 **fast-4.4** 或 **stable-4.4** 频道中，则代表红帽会支持这个版本。**candidate-4.4** 频道可能会包括任何频道都不推荐更新的发行版本。

您可以使用 **candidate-4.4** 频道以前的 OpenShift Container Platform 次版本进行升级。

注意

发行候选版本与每天构建的版本不同。用户可以使用每日构建的版本试用新功能，但升级到每日构建的版本或从每日构建的版本升级不被支持。所有升级频道都没有包括每日构建的版本。您可以引用 OpenShift Container Platform 发行版本状态来获取更多构建信息。

fast-4.4 频道

当红帽声明某个特定版本成为正式发行版本时，**fast-4.4** 频道被更新来包括这个新的 4.4 版本。这意味着，这些版本被完全支持，且具有符合生产环境的质量，当它们作为发行候选版本出现在 **candidate-4.4** 频道期间，被证明可以正常工作。当一个发行版本出现在 **fast-4.4** 频道中的一段时间后，会被添加到 **stable-4.4** 频道。如果版本没有出现在 **fast-4.4** 频道中，则这个版本一定不会出现在 **stable-4.4** 频道中。

您可以使用 **fast-4.4** 频道来从以前的 OpenShift Container Platform 次版本进行升级。

stable-4.4 频道

虽然当它们的勘误被发布后马上就会出现在 **fast-4.4** 频道中，但这些内容可能需要一段延迟时间会被添加到 **stable-4.4** 频道中。在此延迟期间，红帽 SRE 团队、红帽支持服务以及参与连接的客户程序的生产前和产品环境中收集有关此发行版本的稳定性数据。

您可以使用 **stable-4.4** 频道来从以前的 OpenShift Container Platform 次要版本升级。

升级版本路径

OpenShift Container Platform 维护一个升级建议服务，它了解已安装的 OpenShift Container Platform 版本以及您选择用来获取下一版本的频道中的路径。您可在 **fast-4.4** 频道中看到以下内容：

- 4.4.0
- 4.4.1
- 4.4.3
- 4.4.4

该服务只建议经过测试且不存在严重问题的升级。如果您的集群为 4.4.1，OpenShift Container Platform 建议为 4.4.4，那么可以安全地从 .4.4.1 升级到 4.4.4。您不需要一定在连续的补丁号间进行升级。在这个示例中，该频道并没有（且重来没有）包括 4.4.2。更新服务不会建议把系统更新到一个包含具有已知漏洞的 OpenShift Container Platform 版本。

更新的稳定性取决于您的频道。在 **candidate-4.4** 频道中存在一个更新建议并不意味着这个更新会被支持。它代表，在更新中还没有发现任何严重问题，这可能是因为此更新还没有足够的使用情况来证明它的稳定性。如果在 **fast-4.4** 或 **stable-4.4** 频道中出现了一个更新建议，则代表这个更新被完全支持。虽然发行版本永远不会从一个频道中删除，但存在严重问题的更新建议会从所有频道中删除。在更新建议被删除后才初始的更新可能不被支持。

红帽最终会为 **fast-4.4** 或 **stable-4.4** 频道中支持的发行版本提供到最新的 4.4.z 版本的更新路径，但可能会因为创建并验证解决已知问题的更新路径而有一定的延迟。

fast 和 stable 频道的使用和策略

通过 **fast-4.4** 和 **stable-4.4** 频道，您可以选择在一个发行版本正式发行后马上接收到这个版本，或选择由红帽控制向用户推出更新的过程。如果在推出部署的过程或之后发现问题，到这个版本的升级会在**fast-4.4** 和 **stable-4.4** 频道中被禁止。一个新版本可能会出现，做为新的首选升级目标。

通过在 **fast-4.4** 频道中配置预生产环境的系统、在 **stable-4.4** 频道中配置生产环境的系统，并参与红帽连接的客户项目，用户可以改进更新的过程。红帽使用这个程序观察更新对您特定的硬件和软件配置的影响。将来的版本可能会改进或修改更新从 **fast-4.4** 频道进入 **stable-4.4** 频道的速度。

受限网络集群

如果您自己为 OpenShift Container Platform 集群管理容器镜像，您必须考虑与产品关联的红帽勘误中的升级信息。在升级过程中，用户界面可能会提醒您在这些版本间进行切换，因此您必须在跳过这些警告前确定选择了正确的版本。

在频道间切换

如果您从 **stable-4.4** 频道改到 **fast-4.4** 频道，您的集群仍然被支持。虽然您可以在任何时候切换到

candidate-4.4 频道，但该频道中的一些发行版本可能不被支持。如果您当前的发行本是正式发布版本，则可以从 **candidate-4.4** 频道切换到 **fast-4.4** 频道。从 **fast-4.4** 频道切换到 **stable-4.4** 频道一直被允许，但如果当前的发行版本最近被升级到 **fast-4.4**，则可能会有最多一天的延迟该发行版本才会出现在 **stable-4.4** 中。如果您改到的频道没有包括您当前的发行版本，则会出现一个警告信息且不会有建议的更新，但您可以随时安全地切换回您原来地频道。

1.4. 使用WEB控制台更新集群

如果有可用更新，您可以从Web控制台更新集群。

您可以在客户门户网站的勘误部分找到有关可用的OpenShift Container Platform公告和更新的信息。

先决条件

- 使用具有 **admin** 权限的用户登陆到 web 控制台。

流程

1. 在 web 控制台中点 **Administration** > **Cluster Settings**，查看 **Overview** 标签页中的内容。
2. 对于生产环境中的集群，请确保将 **CHANNEL** 设置为您当前使用的次版本的正确频道，如 **stable-4.3**。

重要

对于生产环境中的集群，需要订阅到 stable-* 或 fast-* 频道。

- 如果 **UPDATE STATUS** 的值不是 **Updates Available**，则不能升级您的集群。
- **DESIRED VERSION**显示正在运行的集群版本，或正在更新到的集群版本。

3. 点 **Updates Available**，选择最高可用版本并点 **Update**。**UPDATE STATUS**会变为**Updating**，您可以在**Cluster Operators**页中查看Operator升级的进度。
4. 如果您要从 OpenShift Container Platform 4.3.3 或更早版本升级到这个版本，则必须在升级完成后重启所有 Pod。您可以使用以下命令进行此操作，该命令需要 OpenShift CLI（**oc**）：

```
$ for I in $(oc get ns -o jsonpath='{range .items[*]} {.metadata.name}{"\n"} {end}'); \
    do oc delete pods --all -n $I; \
    sleep 1; \
    done
```

注意

需要重启所有 Pod，因为服务 CA 会在 OpenShift Container Platform 4.3.5 中自动轮转。升级过程中会轮转服务 CA，之后需要重启服务以确保所有服务在上一个服务 CA 过期前都使用新的服务 CA。

这个手动重启操作只需要执行一次，后续的升级和轮转将在服务 CA 过期前确保重启，而无需人工干预。

5. 更新完成后，Cluster Version Operator 会刷新可用更新，检查当前频道中是否有更多可用更新。
   - 如果有可用更新，请继续在当前频道中执行更新，直到您无法再更新为止。

- 如果没有可用的更新，将 **CHANNEL** 改为下一个次版本的 stable-* 或者 fast-* 频道，并更新至您在该频道中想要的版本。

您可能需要执行一些过渡的更新，直到您到达您想要的版本。

第 2 章 通过 WEB 控制台将集群更新为一个新的次版本

您可以使用 web 控制台对 OpenShift Container Platform 集群进行更新或升级。

2.1. 先决条件

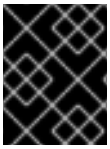
- 使用具有 **admin** 权限的用户访问集群。请参阅使用 RBAC 定义并应用权限 。
- 请保存一个最新的 etcd backup。如果升级失败，则需要把集群恢复到一个以前的状态。

2.2. 关于 OPENSHIFT CONTAINER PLATFORM 更新服务

OpenShift Container Platform 更新服务是一种托管服务，为 OpenShift Container Platform 和 Red Hat Enterprise Linux CoreOS (RHCOS) 提供无线更新 (over-the air update)。它提供了一个组件 Operator 图，其中包含各个*顶点*及连接它们的*边*。图中的边显示可以安全更新的版本，顶点是更新有效负载，用于指定托管集群组件的预期状态。

集群中的 Cluster Version Operator (CVO) 会检查 OpenShift Container Platform 更新服务，并根据当前组件版本和图中的信息决定有效的更新和更新路径。当您请求更新时，OpenShift Container Platform CVO 使用该更新的发行镜像来升级您的集群。发行工件 (artifact) 作为容器镜像托管在 Quay 中。

为了使 OpenShift Container Platform 更新服务仅提供兼容的更新，提供了一个版本验证管道来驱动自动化。每个发行工件都会被验证是否与支持的云平台和系统架构以及其他组件包兼容。在管道确认有适用的版本后，OpenShift Container Platform 更新服务会通知您可以进行更新。



重要

因为更新服务会显示所有有效的更新，所以不能强制更新到一个更新服务没有显示的版本。

对于连续更新模式，会运行两个控制器。一个控制器不断更新有效负载清单，将它们应用于集群，并输出受控 Operator 部署的状态（可用、正在进行升级或失败）。第二个控制器轮询 OpenShift Container Platform 更新服务以确定更新是否可用。

重要

不支持将集群还原到以前的版本或执行回滚。仅支持升级到较新版本。

在升级过程中，Machine Config Operator (MCO) 会将新配置应用到集群机器。它将机器配置池中由 **maxUnavailable** 字段指定数量的节点保护起来，并将其标记为不可用。在默认情况下，这个值被设置为 **1**。然后，它会应用新配置并重启机器。如果您将 Red Hat Enterprise Linux (RHEL) 机器用作 worker，MCO 不会在这些机器上更新 kubelet，因为您必须首先在这些机器上更新 OpenShift API。因为新版本的规格被应用到旧的 kubelet，所以 RHEL 机器无法返回 **Ready** 状态。在机器可用前，您无法完成更新。但是，通过设置不可用节点的最大数量可以确保当不可用机器的数量没有超过这个值时，正常的集群操作仍然可以继续进行。

其他资源

- 非受管 Operator 的支持策略

2.3. OPENSHIFT CONTAINER PLATFORM 升级频道和发行版本

在 OpenShift Container Platform 4.1 中，红帽引进了升级频道的概念，用于为集群升级推荐适当的版本。通过控制升级的速度，这些升级频道允许您选择升级策略。升级频道与 OpenShift Container Platform 的次要版本关联。例如，OpenShift Container Platform 4.4 升级频道永远不会包括到版本 4.5 的升级。这可确保管理员明确决定升级到下一个 OpenShift Container Platform 次要版本。升级频道仅控制版本选择，它不会影响您安装的集群版本; 特定版本的 OpenShift Container Platform 的 **openshift-install** 二进制文件始终会安装这个特定版本。

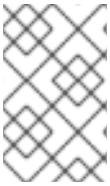
OpenShift Container Platform 4.4 提供了以下升级频道：

- **candidate-4.4**
- **fast-4.4**
- **stable-4.4**

candidate-4.4 频道

Candidate-4.4 频道包含 z-stream (4.4.z) 发行版本的候选构建。发行候选版本包含该产品的所有功能但不被正式支持。发行候选版本可以用来测试新版本的功能以决定下一个 OpenShift Container Platform 版本是否适用于您的系统。发行候选是指候选频道中的一个构建，包括那些名称中没有 **-rc** 的构建。当一个版本出现在候选频道中后，它仍然会进行更多的质量测试。如果达到质量标准，则会将其推广至 **fast-4.4** 或 **stable-4.4** 频道。因此，如果一个特定的版本同时存在于 **candidate-4.4**频道以及 **fast-4.4** 或 **stable-4.4** 频道中，则代表红帽会支持这个版本。**candidate-4.4** 频道可能会包括任何频道都不推荐更新的发行版本。

您可以使用 **candidate-4.4** 频道以前的 OpenShift Container Platform 次版本进行升级。



注意

发行候选版本与每天构建的版本不同。用户可以使用每日构建的版本试用新功能，但升级到每日构建的版本或从每日构建的版本升级不被支持。所有升级频道都没有包括每日构建的版本。您可以引用 OpenShift Container Platform 发行版本状态来获取更多构建信息。

fast-4.4 频道

当红帽声明某个特定版本成为正式发行版本时，**fast-4.4** 频道被更新来包括这个新的 4.4 版本。这意味着，这些版本被完全支持，且具有符合生产环境的质量，当它们作为发行候选版本出现在 **candidate-4.4** 频道期间，被证明可以正常工作。当一个发行版本出现在 **fast-4.4** 频道中的一段时间后，会被添加到 **stable-4.4** 频道。如果版本没有出现在 **fast-4.4** 频道中，则这个版本一定不会出现在 **stable-4.4** 频道中。

您可以使用 **fast-4.4** 频道来从以前的 OpenShift Container Platform 次版本进行升级。

stable-4.4 频道

虽然当它们的勘误被发布后马上就会出现在 **fast-4.4** 频道中，但这些内容可能需要一段延迟时间会被添加到 **stable-4.4** 频道中。在此延迟期间，红帽 SRE 团队、红帽支持服务以及参与连接的客户程序的生产前和产品环境中收集有关此发行版本的稳定性数据。

您可以使用 **stable-4.4** 频道来从以前的 OpenShift Container Platform 次要版本升级。

升级版本路径

OpenShift Container Platform 维护一个升级建议服务，它了解已安装的 OpenShift Container Platform 版本以及您选择用来获取下一版本的频道中的路径。您可在 **fast-4.4** 频道中看到以下内容：

- 4.4.0
- 4.4.1
- 4.4.3

- 4.4.4

该服务只建议经过测试且不存在严重问题的升级。如果您的集群为 4.4.1，OpenShift Container Platform 建议为 4.4.4，那么可以安全地从 .4.4.1 升级到 4.4.4。您不需要一定在连续的补丁号间进行升级。在这个示例中，该频道并没有（且重来没有）包括 4.4.2。更新服务不会建议把系统更新到一个包含具有已知漏洞的 OpenShift Container Platform 版本。

更新的稳定性取决于您的频道。在 **candidate-4.4** 频道中存在一个更新建议并不意味着这个更新会被支持。它代表，在更新中还没有发现任何严重问题，这可能是因为此更新还没有足够的使用情况来证明它的稳定性。如果在 **fast-4.4** 或 **stable-4.4** 频道中出现了一个更新建议，则代表这个更新被完全支持。虽然发行版本永远不会从一个频道中删除，但存在严重问题的更新建议会从所有频道中删除。在更新建议被删除后才初始的更新可能不被支持。

红帽最终会为 **fast-4.4** 或 **stable-4.4** 频道中支持的发行版本提供到最新的 4.4.z 版本的更新路径，但可能会因为创建并验证解决已知问题的更新路径而有一定的延迟。

fast 和 stable 频道的使用和策略
通过 **fast-4.4** 和 **stable-4.4** 频道，您可以选择在一个发行版本正式发行后马上接收到这个版本，或选择由红帽控制向用户推出更新的过程。如果在推出部署的过程或之后发现问题，到这个版本的升级会在**fast-4.4** 和 **stable-4.4** 频道中被禁止。一个新版本可能会出现，做为新的首选升级目标。

通过在 **fast-4.4** 频道中配置预生产环境的系统、在 **stable-4.4** 频道中配置生产环境的系统，并参与红帽连接的客户项目，用户可以改进更新的过程。红帽使用这个程序观察更新对您特定的硬件和软件配置的影响。将来的版本可能会改进或修改更新从 **fast-4.4** 频道进入 **stable-4.4** 频道的速度。

受限网络集群
如果您自己为 OpenShift Container Platform 集群管理容器镜像，您必须考虑与产品关联的红帽勘误中的升级信息。在升级过程中，用户界面可能会提醒您在这些版本间进行切换，因此您必须在跳过这些警告前确定选择了正确的版本。

在频道间切换
如果您从 **stable-4.4** 频道改到 **fast-4.4** 频道，您的集群仍然被支持。虽然您可以在任何时候切换到 **candidate-4.4** 频道，但该频道中的一些发行版本可能不被支持。如果您当前的发行本是正式发布版本，则可以从 **candidate-4.4** 频道切换到 **fast-4.4** 频道。从 **fast-4.4** 频道切换到 **stable-4.4** 频道一直被允许，但如果当前的发行版本最近被升级到 **fast-4.4**，则可能会有最多一天的延迟该发行版本才会出现在 **stable-4.4** 中。如果您改到的频道没有包括您当前的发行版本，则会出现一个警告信息且不会有建议的更新，但您可以随时安全地切换回您原来地频道。

2.4. 使用WEB控制台更新集群

如果有可用更新，您可以从Web控制台更新集群。

您可以在客户门户网站的勘误部分找到有关可用的OpenShift Container Platform公告和更新的信息。

先决条件

- 使用具有 **admin** 权限的用户登陆到 web 控制台。

流程

1. 在 web 控制台中点 **Administration** > **Cluster Settings**，查看 **Overview** 标签页中的内容。
2. 对于生产环境中的集群，请确保将 **CHANNEL** 设置为您要升级到的版本的正确频道，如 **stable-4.4**。

重要

对于生产环境中的集群，需要订阅到 stable-* 或 fast-* 频道。

- 如果 **UPDATE STATUS** 的值不是 **Updates Available**，则不能升级您的集群。
- **DESIRED VERSION**显示正在运行的集群版本，或正在更新到的集群版本。

3. 点 **Updates Available**，选择要更新到的版本，最新可用版本并点 **Update**。**UPDATE STATUS**会变为**Updating** ，您可以在**Cluster Operators**页中查看Operator升级的进度。

4. 如果您要从 OpenShift Container Platform 4.3.3 或更早版本升级到这个版本，则必须在升级完成后重启所有 Pod。您可以使用以下命令进行此操作，该命令需要 OpenShift CLI（**oc**）：

```
$ for I in $(oc get ns -o jsonpath='{range .items[*]} {.metadata.name}{"\n"} {end}'); \
    do oc delete pods --all -n $I; \
    sleep 1; \
    done
```

注意

需要重启所有 Pod，因为服务 CA 会在 OpenShift Container Platform 4.3.5 中自动轮转。升级过程中会轮转服务 CA，之后需要重启服务以确保所有服务在上一个服务 CA 过期前都使用新的服务 CA。

这个手动重启操作只需要执行一次，后续的升级和轮转将在服务 CA 过期前确保重启，而无需人工干预。

5. 更新完成后，Cluster Version Operator 会刷新可用更新，检查当前频道中是否有更多可用更新。

- 如果有可用更新，请继续在当前频道中执行更新，直到您无法再更新为止。
- 如果没有可用的更新，将 **CHANNEL** 改为下一个次版本的 stable-* 或者 fast-* 频道，并更新至您在该频道中想要的版本。

您可能需要执行一些过渡的更新，直到您到达您想要的版本。

第 3 章 使用 CLI 将集群更新为一个新的次版本

您可以使用 OpenShift CLI (**oc**) 将 OpenShift Container Platform 集群更新或升级到一个次版本。

3.1. 先决条件

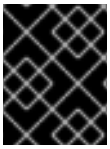
- 使用具有 **admin** 权限的用户访问集群。请参阅使用 RBAC 定义并应用权限 。
- 请保存一个最新的 etcd backup。如果升级失败，则需要把集群恢复到一个以前的状态。

3.2. 关于 OPENSHIFT CONTAINER PLATFORM 更新服务

OpenShift Container Platform 更新服务是一种托管服务，为 OpenShift Container Platform 和 Red Hat Enterprise Linux CoreOS (RHCOS) 提供无线更新 (over-the air update)。它提供了一个组件 Operator 图，其中包含各个*顶点*及连接它们的*边*。图中的边显示可以安全更新的版本，顶点是更新有效负载，用于指定托管集群组件的预期状态。

集群中的 Cluster Version Operator (CVO) 会检查 OpenShift Container Platform 更新服务，并根据当前组件版本和图中的信息决定有效的更新和更新路径。当您请求更新时，OpenShift Container Platform CVO 使用该更新的发行镜像来升级您的集群。发行工件 (artifact) 作为容器镜像托管在 Quay 中。

为了使 OpenShift Container Platform 更新服务仅提供兼容的更新，提供了一个版本验证管道来驱动自动化。每个发行工件都会被验证是否与支持的云平台和系统架构以及其他组件包兼容。在管道确认有适用的版本后，OpenShift Container Platform 更新服务会通知您可以进行更新。



重要

因为更新服务会显示所有有效的更新，所以不能强制更新到一个更新服务没有显示的版本。

对于连续更新模式，会运行两个控制器。一个控制器不断更新有效负载清单，将它们应用于集群，并输出受控 Operator 部署的状态（可用、正在进行升级或失败）。第二个控制器轮询 OpenShift Container Platform 更新服务以确定更新是否可用。

重要

不支持将集群还原到以前的版本或执行回滚。仅支持升级到较新版本。

在升级过程中，Machine Config Operator (MCO) 会将新配置应用到集群机器。它将机器配置池中由 **maxUnavailable** 字段指定数量的节点保护起来，并将其标记为不可用。在默认情况下，这个值被设置为 **1**。然后，它会应用新配置并重启机器。如果您将 Red Hat Enterprise Linux (RHEL) 机器用作 worker，MCO 不会在这些机器上更新 kubelet，因为您必须首先在这些机器上更新 OpenShift API。因为新版本的规格被应用到旧的 kubelet，所以 RHEL 机器无法返回 **Ready** 状态。在机器可用前，您无法完成更新。但是，通过设置不可用节点的最大数量可以确保当不可用机器的数量没有超过这个值时，正常的集群操作仍然可以继续进行。

其他资源

- 非受管 Operator 的支持策略

3.3. OPENSHIFT CONTAINER PLATFORM 升级频道和发行版本

在 OpenShift Container Platform 4.1 中，红帽引进了升级频道的概念，用于为集群升级推荐适当的版本。通过控制升级的速度，这些升级频道允许您选择升级策略。升级频道与 OpenShift Container Platform 的次要版本关联。例如，OpenShift Container Platform 4.4 升级频道永远不会包括到版本 4.5 的升级。这可确保管理员明确决定升级到下一个 OpenShift Container Platform 次要版本。升级频道仅控制版本选择，它不会影响您安装的集群版本; 特定版本的 OpenShift Container Platform 的 **openshift-install** 二进制文件始终会安装这个特定版本。

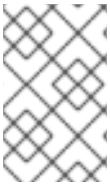
OpenShift Container Platform 4.4 提供了以下升级频道：

- **candidate-4.4**
- **fast-4.4**
- **stable-4.4**

candidate-4.4 频道

Candidate-4.4 频道包含 z-stream (4.4.z) 发行版本的候选构建。发行候选版本包含该产品的所有功能但不被正式支持。发行候选版本可以用来测试新版本的功能以决定下一个 OpenShift Container Platform 版本是否适用于您的系统。发行候选是指候选频道中的一个构建，包括那些名称中没有 **-rc** 的构建。当一个版本出现在候选频道中后，它仍然会进行更多的质量测试。如果达到质量标准，则会将其推广至 **fast-4.4** 或 **stable-4.4** 频道。因此，如果一个特定的版本同时存在于 **candidate-4.4**频道以及 **fast-4.4** 或 **stable-4.4** 频道中，则代表红帽会支持这个版本。**candidate-4.4** 频道可能会包括任何频道都不推荐更新的发行版本。

您可以使用 **candidate-4.4** 频道以前的 OpenShift Container Platform 次版本进行升级。



注意

发行候选版本与每天构建的版本不同。用户可以使用每日构建的版本试用新功能，但升级到每日构建的版本或从每日构建的版本升级不被支持。所有升级频道都没有包括每日构建的版本。您可以引用 OpenShift Container Platform 发行版本状态来获取更多构建信息。

fast-4.4 频道

当红帽声明某个特定版本成为正式发行版本时，**fast-4.4** 频道被更新来包括这个新的 4.4 版本。这意味着，这些版本被完全支持，且具有符合生产环境的质量，当它们作为发行候选版本出现在 **candidate-4.4** 频道期间，被证明可以正常工作。当一个发行版本出现在 **fast-4.4** 频道中的一段时间后，会被添加到 **stable-4.4** 频道。如果版本没有出现在 **fast-4.4** 频道中，则这个版本一定不会出现在 **stable-4.4** 频道中。

您可以使用 **fast-4.4** 频道来从以前的 OpenShift Container Platform 次版本进行升级。

stable-4.4 频道

虽然当它们的勘误被发布后马上就会出现在 **fast-4.4** 频道中，但这些内容可能需要一段延迟时间会被添加到 **stable-4.4** 频道中。在此延迟期间，红帽 SRE 团队、红帽支持服务以及参与连接的客户程序的生产前和产品环境中收集有关此发行版本的稳定性数据。

您可以使用 **stable-4.4** 频道来从以前的 OpenShift Container Platform 次要版本升级。

升级版本路径

OpenShift Container Platform 维护一个升级建议服务，它了解已安装的 OpenShift Container Platform 版本以及您选择用来获取下一版本的频道中的路径。您可在 **fast-4.4** 频道中看到以下内容：

- 4.4.0
- 4.4.1
- 4.4.3

- 4.4.4

该服务只建议经过测试且不存在严重问题的升级。如果您的集群为 4.4.1，OpenShift Container Platform 建议为 4.4.4，那么可以安全地从 .4.4.1 升级到 4.4.4。您不需要一定在连续的补丁号间进行升级。在这个示例中，该频道并没有（且重来没有）包括 4.4.2。更新服务不会建议把系统更新到一个包含具有已知漏洞的 OpenShift Container Platform 版本。

更新的稳定性取决于您的频道。在 **candidate-4.4** 频道中存在一个更新建议并不意味着这个更新会被支持。它代表，在更新中还没有发现任何严重问题，这可能是因为此更新还没有足够的使用情况来证明它的稳定性。如果在 **fast-4.4** 或 **stable-4.4** 频道中出现了一个更新建议，则代表这个更新被完全支持。虽然发行版本永远不会从一个频道中删除，但存在严重问题的更新建议会从所有频道中删除。在更新建议被删除后才初始的更新可能不被支持。

红帽最终会为 **fast-4.4** 或 **stable-4.4** 频道中支持的发行版本提供到最新的 4.4.z 版本的更新路径，但可能会因为创建并验证解决已知问题的更新路径而有一定的延迟。

fast 和 stable 频道的使用和策略
通过 **fast-4.4** 和 **stable-4.4** 频道，您可以选择在一个发行版本正式发行后马上接收到这个版本，或选择由红帽控制向用户推出更新的过程。如果在推出部署的过程或之后发现问题，到这个版本的升级会在**fast-4.4** 和 **stable-4.4** 频道中被禁止。一个新版本可能会出现，做为新的首选升级目标。

通过在 **fast-4.4** 频道中配置预生产环境的系统、在 **stable-4.4** 频道中配置生产环境的系统，并参与红帽连接的客户项目，用户可以改进更新的过程。红帽使用这个程序观察更新对您特定的硬件和软件配置的影响。将来的版本可能会改进或修改更新从 **fast-4.4** 频道进入 **stable-4.4** 频道的速度。

受限网络集群
如果您自己为 OpenShift Container Platform 集群管理容器镜像，您必须考虑与产品关联的红帽勘误中的升级信息。在升级过程中，用户界面可能会提醒您在这些版本间进行切换，因此您必须在跳过这些警告前确定选择了正确的版本。

在频道间切换
如果您从 **stable-4.4** 频道改到 **fast-4.4** 频道，您的集群仍然被支持。虽然您可以在任何时候切换到 **candidate-4.4** 频道，但该频道中的一些发行版本可能不被支持。如果您当前的发行本是正式发布版本，则可以从 **candidate-4.4** 频道切换到 **fast-4.4** 频道。从 **fast-4.4** 频道切换到 **stable-4.4** 频道一直被允许，但如果当前的发行版本最近被升级到 **fast-4.4**，则可能会有最多一天的延迟该发行版本才会出现在 **stable-4.4** 中。如果您改到的频道没有包括您当前的发行版本，则会出现一个警告信息且不会有建议的更新，但您可以随时安全地切换回您原来地频道。

3.4. 使用 CLI 更新集群

如果有可用更新，您可以使用OpenShift CLI (**oc**)更新集群。

您可以在客户门户网站的勘误部分找到有关可用的OpenShift Container Platform公告和更新的信息。

先决条件

- 安装与更新版本的版本匹配的 OpenShift CLI(**oc**)。
- 使用具有 **cluster-admin** 权限的用户登陆到集群。
- 安装**jq**软件包。

流程

1. 确认集群可用
   -

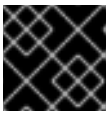
```
$ oc get clusterversion

NAME     VERSION   AVAILABLE   PROGRESSING   SINCE   STATUS
version  4.4.0     True        False         158m    Cluster version is 4.4.0
```

2. 检查当前的更新频道信息，并确认您的频道已设置为**stable-4.4** ：

```
$ oc get clusterversion -o json|jq ".items[0].spec"

{
  "channel": "stable-4.4",
  "clusterID": "990f7ab8-109b-4c95-8480-2bd1deec55ff",
  "upstream": "https://api.openshift.com/api/upgrades_info/v1/graph"
}
```



重要

对于生产环境中的集群，需要订阅到 **stable-*** 或 **fast-*** 频道。

3. 查看可用更新，记录下要应用的更新的版本号：

```
$ oc adm upgrade

Cluster version is 4.1.0

Updates:

VERSION IMAGE
4.1.2   quay.io/openshift-release-dev/ocp-
release@sha256:9c5f0df8b192a0d7b46cd5f6a4da2289c155fd5302dec7954f8f06c878160b8b
```

4. 应用更新：

- 要更新到最新版本：

```
$ oc adm upgrade --to-latest=true (1)
```

- 要更新到一个特定版本：

```
$ oc adm upgrade --to=<version> (1)
```

(1) (1) **<version>**是从上一个命令输出中获取的更新版本。

5. 查看 Cluster Version Operator 的状态：

```
$ oc get clusterversion -o json|jq ".items[0].spec"

{
  "channel": "stable-4.4",
  "clusterID": "990f7ab8-109b-4c95-8480-2bd1deec55ff",
  "desiredUpdate": {
    "force": false,
```

```
    "image": "quay.io/openshift-release-dev/ocp-
release@sha256:9c5f0df8b192a0d7b46cd5f6a4da2289c155fd5302dec7954f8f06c878160b8b",

    "version": "4.4.0" 1
  },
  "upstream": "https://api.openshift.com/api/upgrades_info/v1/graph"
}
```

1 如果**desiredUpdate**中的**version**值与您指定的值匹配，则更新正在进行中。

6. 查看集群版本状态历史记录以监控更新的状态。这可能需要一些时间才能完成对所有对象的更新。

```
$ oc get clusterversion -o json|jq ".items[0].status.history"

[
  {
    "completionTime": null,
    "image": "quay.io/openshift-release-dev/ocp-
release@sha256:9c5f0df8b192a0d7b46cd5f6a4da2289c155fd5302dec7954f8f06c878160b8b",

    "startedTime": "2019-06-19T20:30:50Z",
    "state": "Partial",
    "verified": true,
    "version": "4.1.2"
  },
  {
    "completionTime": "2019-06-19T20:30:50Z",
    "image": "quay.io/openshift-release-dev/ocp-
release@sha256:b8307ac0f3ec4ac86c3f3b52846425205022da52c16f56ec31cbe428501001d6
",
    "startedTime": "2019-06-19T17:38:10Z",
    "state": "Completed",
    "verified": false,
    "version": "4.1.0"
  }
]
```

历史记录包含了应用于集群的最新版本的列表。当CVO应用更新时，此值将会被相应更新。该列表按日期排序，最新的更新会在列表中第一个显示。如果历史信息中的更新状态为 **Completed**，则表示部署已完成；如果状态为 **Partial**，则表示更新失败或还未完成。

重要

如果升级失败，Operator 将停止操作并报告故障组件的状态。当前还不支持将集群还原到以前的版本。如果升级失败，请联系红帽支持。

7. 更新完成后，可以通过以下方法确认集群已更新为新版本：

```
$ oc get clusterversion

NAME      VERSION   AVAILABLE   PROGRESSING   SINCE     STATUS
version   4.4.0     True        False         2m        Cluster version is 4.4.0
```

第 4 章 更新包含使用 RHEL 的计算（COMPUTE）系统的集群

您可以更新或升级OpenShift Container Platform集群。如果您的集群包含Red Hat Enterprise Linux（RHEL）系统，则必须执行额外的步骤来更新这些系统。

4.1. 先决条件

- 使用具有 **admin** 权限的用户访问集群。请参阅使用 RBAC 定义并应用权限 。
- 请保存一个最新的 etcd backup。如果升级失败，则需要把集群恢复到一个以前的状态。

重要

如果您要从 OpenShift Container Platform 4.3.3 或更早版本升级到这个版本，则必须在升级完成后重启所有 Pod。

这是因为服务 CA 会在 OpenShift Container Platform 4.3.5 中自动轮转。升级过程中会轮转服务 CA，之后需要重启服务以确保所有服务在上一个服务 CA 过期前都使用新的服务 CA。

这个手动重启操作只需要执行一次，后续的升级和轮转将在服务 CA 过期前确保重启，而无需人工干预。

4.2. 关于 OPENSHIFT CONTAINER PLATFORM 更新服务

OpenShift Container Platform 更新服务是一种托管服务，为 OpenShift Container Platform 和 Red Hat Enterprise Linux CoreOS (RHCOS) 提供无线更新 (over-the air update)。它提供了一个组件 Operator 图，其中包含各个*顶点*及连接它们的*边*。图中的边显示可以安全更新的版本，顶点是更新有效负载，用于指定托管集群组件的预期状态。

集群中的 Cluster Version Operator (CVO) 会检查 OpenShift Container Platform 更新服务，并根据当前组件版本和图中的信息决定有效的更新和更新路径。当您请求更新时，OpenShift Container Platform CVO 使用该更新的发行镜像来升级您的集群。发行工件 (artifact) 作为容器镜像托管在 Quay 中。

为了使 OpenShift Container Platform 更新服务仅提供兼容的更新，提供了一个版本验证管道来驱动自动化。每个发行工件都会被验证是否与支持的云平台和系统架构以及其他组件包兼容。在管道确认有适用的版本后，OpenShift Container Platform 更新服务会通知您可以进行更新。

重要

因为更新服务会显示所有有效的更新，所以不能强制更新到一个更新服务没有显示的版本。

对于连续更新模式，会运行两个控制器。一个控制器不断更新有效负载清单，将它们应用于集群，并输出受控 Operator 部署的状态（可用、正在进行升级或失败）。第二个控制器轮询 OpenShift Container Platform 更新服务以确定更新是否可用。

重要

不支持将集群还原到以前的版本或执行回滚。仅支持升级到较新版本。

在升级过程中，Machine Config Operator (MCO) 会将新配置应用到集群机器。它将机器配置池中由 **maxUnavailable** 字段指定数量的节点保护起来，并将其标记为不可用。在默认情况下，这个值被设置为

1。然后，它会应用新配置并重启机器。如果您将 Red Hat Enterprise Linux (RHEL) 机器用作 worker，MCO 不会在这些机器上更新 kubelet，因为您必须首先在这些机器上更新 OpenShift API。因为新版本的规格被应用到旧的 kubelet，所以 RHEL 机器无法返回 **Ready** 状态。在机器可用前，您无法完成更新。但是，通过设置不可用节点的最大数量可以确保当不可用机器的数量没有超过这个值时，正常的集群操作仍然可以继续进行。

其他资源

- 非受管 Operator 的支持策略

4.3. OPENSHIFT CONTAINER PLATFORM 升级频道和发行版本

在 OpenShift Container Platform 4.1 中，红帽引进了升级频道的概念，用于为集群升级推荐适当的版本。通过控制升级的速度，这些升级频道允许您选择升级策略。升级频道与 OpenShift Container Platform 的次要版本关联。例如，OpenShift Container Platform 4.4 升级频道永远不会包括到版本 4.5 的升级。这可确保管理员明确决定升级到下一个 OpenShift Container Platform 次要版本。升级频道仅控制版本选择，它不会影响您安装的集群版本; 特定版本的 OpenShift Container Platform 的 **openshift-install** 二进制文件始终会安装这个特定版本。

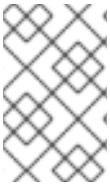
OpenShift Container Platform 4.4 提供了以下升级频道：

- **candidate-4.4**
- **fast-4.4**
- **stable-4.4**

candidate-4.4 频道

Candidate-4.4 频道包含 z-stream (4.4.z) 发行版本的候选构建。发行候选版本包含该产品的所有功能但不被正式支持。发行候选版本可以用来测试新版本的功能以决定下一个 OpenShift Container Platform 版本是否适用于您的系统。发行候选是指候选频道中的一个构建，包括那些名称中没有 **-rc** 的构建。当一个版本出现在候选频道中后，它仍然会进行更多的质量测试。如果达到质量标准，则会将其推广至 **fast-4.4** 或 **stable-4.4** 频道。因此，如果一个特定的版本同时存在于 **candidate-4.4**频道以及 **fast-4.4** 或 **stable-4.4** 频道中，则代表红帽会支持这个版本。**candidate-4.4** 频道可能会包括任何频道都不推荐更新的发行版本。

您可以使用 **candidate-4.4** 频道以前的 OpenShift Container Platform 次版本进行升级。



注意

发行候选版本与每天构建的版本不同。用户可以使用每日构建的版本试用新功能，但升级到每日构建的版本或从每日构建的版本升级不被支持。所有升级频道都没有包括每日构建的版本。您可以引用 OpenShift Container Platform 发行版本状态来获取更多构建信息。

fast-4.4 频道

当红帽声明某个特定版本成为正式发行版本时，**fast-4.4** 频道被更新来包括这个新的 4.4 版本。这意味着，这些版本被完全支持，且具有符合生产环境的质量，当它们作为发行候选版本出现在 **candidate-4.4** 频道期间，被证明可以正常工作。当一个发行版本出现在 **fast-4.4** 频道中的一段时间后，会被添加到 **stable-4.4** 频道。如果版本没有出现在 **fast-4.4** 频道中，则这个版本一定不会出现在 **stable-4.4** 频道中。

您可以使用 **fast-4.4** 频道来从以前的 OpenShift Container Platform 次版本进行升级。

stable-4.4 频道

虽然当它们的勘误被发布后马上就会出现在 **fast-4.4** 频道中，但这些内容可能需要一段延迟时间会被添加到 **stable-4.4** 频道中。在此延迟期间，红帽 SRE 团队、红帽支持服务以及参与连接的客户程序的生产前和产品环境中收集有关此发行版本的稳定性数据。

您可以使用 **stable-4.4** 频道来从以前的 OpenShift Container Platform 次要版本升级。

升级版本路径
OpenShift Container Platform 维护一个升级建议服务，它了解已安装的 OpenShift Container Platform 版本以及您选择用来获取下一版本的频道中的路径。您可在 **fast-4.4** 频道中看到以下内容：

- 4.4.0
- 4.4.1
- 4.4.3
- 4.4.4

该服务只建议经过测试且不存在严重问题的升级。如果您的集群为 4.4.1，OpenShift Container Platform 建议为 4.4.4，那么可以安全地从 .4.4.1 升级到 4.4.4。您不需要一定在连续的补丁号间进行升级。在这个示例中，该频道并没有（且重来没有）包括 4.4.2。更新服务不会建议把系统更新到一个包含具有已知漏洞的 OpenShift Container Platform 版本。

更新的稳定性取决于您的频道。在 **candidate-4.4** 频道中存在一个更新建议并不意味着这个更新会被支持。它代表，在更新中还没有发现任何严重问题，这可能是因为此更新还没有足够的使用情况来证明它的稳定性。如果在 **fast-4.4** 或 **stable-4.4** 频道中出现了一个更新建议，则代表这个更新被完全支持。虽然发行版本永远不会从一个频道中删除，但存在严重问题的更新建议会从所有频道中删除。在更新建议被删除后才初始的更新可能不被支持。

红帽最终会为 **fast-4.4** 或 **stable-4.4** 频道中支持的发行版本提供到最新的 4.4.z 版本的更新路径，但可能会因为创建并验证解决已知问题的更新路径而有一定的延迟。

fast 和 stable 频道的使用和策略
通过 **fast-4.4** 和 **stable-4.4** 频道，您可以选择在一个发行版本正式发行后马上接收到这个版本，或选择由红帽控制向用户推出更新的过程。如果在推出部署的过程或之后发现问题，到这个版本的升级会在**fast-4.4** 和 **stable-4.4** 频道中被禁止。一个新版本可能会出现，做为新的首选升级目标。

通过在 **fast-4.4** 频道中配置预生产环境的系统、在 **stable-4.4** 频道中配置生产环境的系统，并参与红帽连接的客户项目，用户可以改进更新的过程。红帽使用这个程序观察更新对您特定的硬件和软件配置的影响。将来的版本可能会改进或修改更新从 **fast-4.4** 频道进入 **stable-4.4** 频道的速度。

受限网络集群
如果您自己为 OpenShift Container Platform 集群管理容器镜像，您必须考虑与产品关联的红帽勘误中的升级信息。在升级过程中，用户界面可能会提醒您在这些版本间进行切换，因此您必须在跳过这些警告前确定选择了正确的版本。

在频道间切换
如果您从 **stable-4.4** 频道改到 **fast-4.4** 频道，您的集群仍然被支持。虽然您可以在任何时候切换到 **candidate-4.4** 频道，但该频道中的一些发行版本可能不被支持。如果您当前的发行本是正式发布版本，则可以从 **candidate-4.4** 频道切换到 **fast-4.4** 频道。从 **fast-4.4** 频道切换到 **stable-4.4** 频道一直被允许，但如果当前的发行版本最近被升级到 **fast-4.4**，则可能会有最多一天的延迟该发行版本才会出现在 **stable-4.4** 中。如果您改到的频道没有包括您当前的发行版本，则会出现一个警告信息且不会有建议的更新，但您可以随时安全地切换回您原来地频道。

4.4. 使用WEB控制台更新集群

如果有可用更新，您可以从Web控制台更新集群。

您可以在客户门户网站的勘误部分找到有关可用的OpenShift Container Platform公告和更新的信息。

先决条件

- 使用具有 **admin** 权限的用户登陆到 web 控制台。

流程

1. 在 web 控制台中点 **Administration** > **Cluster Settings**，查看 **Overview** 标签页中的内容。
2. 对于生产环境中的集群，请确保将 **CHANNEL** 设置为您要升级到的版本的正确频道，如 **stable-4.4**。

重要

对于生产环境中的集群，需要订阅到 stable-* 或 fast-* 频道。

- 如果 **UPDATE STATUS** 的值不是 **Updates Available**，则不能升级您的集群。
- **DESIRED VERSION**显示正在运行的集群版本，或正在更新到的集群版本。

3. 点 **Updates Available**，选择要更新到的版本，最新可用版本并点 **Update**。**UPDATE STATUS**会变为**Updating**，您可以在**Cluster Operators**页中查看Operator升级的进度。
4. 如果您要从 OpenShift Container Platform 4.3.3 或更早版本升级到这个版本，则必须在升级完成后重启所有 Pod。您可以使用以下命令进行此操作，该命令需要 OpenShift CLI（**oc**）：

```
$ for I in $(oc get ns -o jsonpath='{range .items[*]} {.metadata.name}{"\n"} {end}'); \
    do oc delete pods --all -n $I; \
    sleep 1; \
    done
```

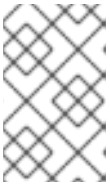
注意

需要重启所有 Pod，因为服务 CA 会在 OpenShift Container Platform 4.3.5 中自动轮转。升级过程中会轮转服务 CA，之后需要重启服务以确保所有服务在上一个服务 CA 过期前都使用新的服务 CA。

这个手动重启操作只需要执行一次，后续的升级和轮转将在服务 CA 过期前确保重启，而无需人工干预。

5. 更新完成后，Cluster Version Operator 会刷新可用更新，检查当前频道中是否有更多可用更新。
   - 如果有可用更新，请继续在当前频道中执行更新，直到您无法再更新为止。
   - 如果没有可用的更新，将 **CHANNEL** 改为下一个次版本的 stable-* 或者 fast-* 频道，并更新至您在该频道中想要的版本。

   您可能需要执行一些过渡的更新，直到您到达您想要的版本。



注意

当您更新包含有 Red Hat Enterprise Linux (RHEL) worker 机器的集群时，这些 worker 会在更新过程中暂时不可用。当集群进入 **NotReady** 状态时，您需要针对每个 RHEL 机器运行升级 playbook 以完成更新。

4.5. （可选）添加 HOOK 以在RHEL系统上执行ANSIBLE任务

在OpenShift Container Platform更新期间，您可以使用*hook*在RHEL计算系统上运行Ansible任务。

4.5.1. 在升级过程中使用 Ansible hook

更新OpenShift Container Platform时，可以使用*hook*在执行特定操作时在Red Hat Enterprise Linux（RHEL）节点上运行自定义的任务。您可以使用 hook 提供定义了在执行特定任务之前或之后要运行的任务的文件。在OpenShift Container Platform集群中更新RHEL计算节点时，可以使用 hook 来验证或修改自定义的基础架构。

因为当 hook 失败时，这个操作将会失败，所以您必须把 hook 设计为可以多次运行，并且获得相同的结果。

hook 有以下限制： - hook 没有已定义或版本化的界面。它们可以使用内部的**openshift-ansible**变量，但这些变量可能会在将来的OpenShift Container Platform版本被修改或删除。 - hook 本身没有错误处理机制，因此 hook 中的错误会暂停更新过程。如果出现错误，则需要解决相关的问题，然后再次进行升级。

4.5.2. 配置Ansible inventory文件以使用 hook

您可以在 **hosts** inventory 文件的**all:vars** 部分中定义 Red Hat Enterprise Linux（RHEL）compute 机器（也称为 worker 机器）更新时使用的 hook 。

先决条件

- 您可以访问用于添加RHEL compute 系统集群的计算机。您必须有访问定义RHEL系统的**hosts** Ansible 清单文件的权限。

流程

1. 在设计了 hook 后，创建一个YAML文件，为其定义Ansible任务。此文件必须是一组任务，不能是一个 playbook，如以下示例所示：

```
---
# Trivial example forcing an operator to acknowledge the start of an upgrade
# file=/home/user/openshift-ansible/hooks/pre_compute.yml

- name: note the start of a compute machine update
  debug:
    msg: "Compute machine upgrade of {{ inventory_hostname }} is about to start"

- name: require the user agree to start an upgrade
  pause:
    prompt: "Press Enter to start the compute machine update"
```

2. 修改 **hosts** Ansible inventory 文件来指定 hook 文件。hook 文件作为参数值在 **[all:vars]** 部分指定。如下所示：

清单文件中的 hook 定义示例

```
[all:vars]
openshift_node_pre_upgrade_hook=/home/user/openshift-ansible/hooks/pre_node.yml
openshift_node_post_upgrade_hook=/home/user/openshift-ansible/hooks/post_node.yml
```

为了避免歧义，请在其定义中使用 hook 文件的绝对路径而不要使用相对路径。

4.5.3. RHEL计算系统可用的 hook

在更新OpenShift Container Platform集群中的Red Hat Enterprise Linux（RHEL）compute 系统时，可以使用以下 hook。

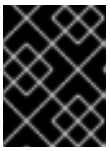
Hook 名	描述
openshift_node_pre_cordon_hook	• 在每个节点被封锁（cordon）**之前**运行。 • 此 hook 以串行方式针对**每个节点**运行。 • 如果某个任务必须针对其他主机运行，则该任务必须使用**delegate_to**或**local_action**。
openshift_node_pre_upgrade_hook	• 在每个节点被封锁**后**，且被更新**前**运行。 • 此 hook 以串行方式针对**每个节点**运行。 • 如果某个任务必须针对其他主机运行，则该任务必须使用**delegate_to**或**local_action**。
openshift_node_pre_uncordon_hook	• 在每个节点被更新**后**，且被取消封锁（uncordon）**前**运行。 • 此 hook 以串行方式针对**每个节点**运行。 • 如果某个任务必须针对其他主机运行，则该任务必须使用**delegate_to**或**local_action**。
openshift_node_post_upgrade_hook	• 每个节点未被取消封锁**后**运行。这是**最后一个**节点更新操作。 • 此 hook 以串行方式针对**每个节点**运行。 • 如果某个任务必须针对其他主机运行，则该任务必须使用**delegate_to**或**local_action**。

4.6. 更新集群中的RHEL COMPUTE 系统

在对集群进行更新后，必须更新集群中的Red Hat Enterprise Linux（RHEL）compute 系统。

先决条件

- 已更新了集群



重要

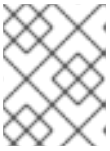
由于RHEL系统需要集群生成的资产才能完成更新过程，因此必须在更新其中的 RHEL compute 系统前更新集群。

- 您可以访问用于添加RHEL compute 系统集群的计算机。您必须有权访问定义了 RHEL 系统及 **upgrade** playbook 的 **hosts** Ansible 清单文件。

流程

1. 停止并禁用主机上的防火墙：

```
# systemctl disable --now firewalld.service
```



注意

请不要在以后启用防火墙。如果这样做，则无法访问 worker 上的 OpenShift Container Platform 日志。

2. 启用 OpenShift Container Platform 4.4 所需的存储库：
   a. 在运行 Ansible playbook 的机器上，更新所需的存储库：

```
# subscription-manager repos --disable=rhel-7-server-ansible-2.8-rpms  \
                             --disable=rhel-7-server-ose-4.3-rpms \
                             --enable=rhel-7-server-ansible-2.9-rpms \
                             --enable=rhel-7-server-ose-4.4-rpms
```

   b. 在运行 Ansible playbook 的机器上，更新所需的软件包，包括 **openshift-ansible**：

```
# yum update openshift-ansible openshift-clients
```

   c. 在每个 RHEL 计算节点上，更新所需的软件仓库：

```
# subscription-manager repos --disable=rhel-7-server-ose-4.3-rpms \
                             --enable=rhel-7-server-ose-4.4-rpms
```

3. 更新 RHEL worker 机器：
   a. 查看当前节点状态，以确定要更新哪个 RHEL worker：

```
# oc get node
NAME                  STATUS                     ROLES    AGE    VERSION
mycluster-control-plane-0   Ready                      master   145m   v1.17.1
mycluster-control-plane-1   Ready                      master   145m   v1.17.1
mycluster-control-plane-2   Ready                      master   145m   v1.17.1
mycluster-rhel7-0           NotReady,SchedulingDisabled   worker   98m
```

```
v1.14.6+97c81d00e
mycluster-rhel7-1       Ready                  worker   98m   v1.14.6+97c81d00e
mycluster-rhel7-2       Ready                  worker   98m   v1.14.6+97c81d00e
mycluster-rhel7-3       Ready                  worker   98m   v1.14.6+97c81d00e
```

记录下哪个机器具有 **NotReady，SchedulingDisabled** 状态。

b. 查看位于 **/<path>/inventory/hosts** 中的 Ansible 清单文件，并更新其内容，以便只有具有 **NotReady,SchedulingDisabled** 状态的机器才列在 **[workers]** 部分中，如下例所示：

```
[all:vars]
ansible_user=root
#ansible_become=True

openshift_kubeconfig_path="~/.kube/config"

[workers]
mycluster-rhel7-0.example.com
```

c. 切换到**openshift-ansible**目录并运行**升级** playbook：

```
$ cd /usr/share/ansible/openshift-ansible
$ ansible-playbook -i /<path>/inventory/hosts playbooks/upgrade.yml 1
```

1 对于**<path>**，指定您创建的Ansible库存文件的路径。

4. 按照上一步中的流程更新集群中的每个 RHEL worker 机器。

5. 更新完所有 worker 后，确认所有集群节点已更新至新版本：

```
# oc get node
NAME                 STATUS                     ROLES    AGE    VERSION
mycluster-control-plane-0   Ready                      master   145m   v1.17.1
mycluster-control-plane-1   Ready                      master   145m   v1.17.1
mycluster-control-plane-2   Ready                      master   145m   v1.17.1
mycluster-rhel7-0        NotReady,SchedulingDisabled   worker   98m    v1.17.1
mycluster-rhel7-1        Ready                      worker   98m    v1.17.1
mycluster-rhel7-2        Ready                      worker   98m    v1.17.1
mycluster-rhel7-3        Ready                      worker   98m    v1.17.1
```

第 5 章 更新受限网络集群

您可以使用 **oc** 命令行界面（CLI）升级受限网络 OpenShift Container Platform 集群。

受限网络环境是集群节点无法访问互联网的环境。因此，您必须在 registry 中填充安装镜像。如果您的 registry 主机无法同时访问互联网和集群，您可以将镜像镜像到与这个环境断开连接的文件系统中，然后使用主机或可移动介质填补该空白。如果本地容器 registry 和集群连接到镜像 registry 的主机，您可以直接将发行镜像推送到本地 registry。

如果受限网络中有多个集群，请将所需的发行镜像镜像到单个容器镜像 registry，并使用该 registry 更新所有集群。

5.1. 先决条件

- 可以访问互联网来获取所需的容器镜像。
- 具有对 restricted-network 环境中的容器 registry 的写入权限，以便推送和拉取镜像。容器 registry 必须与 Docker registry API v2 兼容。
- 您必须安装了 **oc** 命令行界面（CLI）工具。
- 使用具有 **admin** 权限的用户访问集群。请参阅使用 RBAC 定义并应用权限 。
- 请保存一个最新的 etcd backup。如果升级失败，则需要把集群恢复到一个以前的状态。

5.2. 准备您的镜像主机

执行镜像步骤前，必须准备主机以检索内容并将其推送到远程位置。

5.2.1. 通过下载二进制文件安装 CLI

您需要安装 CLI（**oc**） 来使用命令行界面与 OpenShift Container Platform 进行交互。您可在 Linux、Windows 或 macOS 上安装 **oc**。

重要

如果安装了旧版本的 **oc**，则无法使用 OpenShift Container Platform 4.4 中的所有命令。下载并安装新版本的 **oc**。如果您要在受限网络中升级集群，安装计划升级到的 **oc** 版本。

5.2.1.1. 在 Linux 上安装 CLI

您可以按照以下流程在 Linux 上安装 OpenShift CLI（**oc**）二进制文件。

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Linux**，并点 **Download command-line tools**。
4. 解包存档：

```
$ tar xvzf <file>
```

5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
   执行以下命令可以查看当前的 **PATH** 设置：

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

5.2.1.2. 在 Windows 上安装 CLI

您可以按照以下流程在 Windows 上安装 OpenShift CLI（**oc**）二进制代码。

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **Windows**，点 **Download command-line tools**。
4. 使用 ZIP 程序解压存档。
5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
   要查看您的 **PATH**，请打开命令提示窗口并执行以下命令：

```
C:\> path
```

安装 CLI 后，就可以使用**oc**命令：

```
C:\> oc <command>
```

5.2.1.3. 在 macOS 上安装 CLI

您可以按照以下流程在 macOS 上安装 OpenShift CLI（**oc**）二进制代码。

流程

1. 访问 Red Hat OpenShift Cluster Manager 网站的 Infrastructure Provider 页面。
2. 选择您的基础架构供应商及安装类型。
3. 在 **Command-line interface** 部分，从下拉菜单中选择 **MacOS**，并点 **Download command-line tools**。
4. 解包和解压存档。
5. 将 **oc** 二进制文件移到 PATH 的目录中。
   要查看您的 **PATH**，打开一个终端窗口并执行以下命令：

```
-
```

```
$ echo $PATH
```

安装 CLI 后，就可以使用**oc**命令：

```
$ oc <command>
```

5.3. 配置允许对容器镜像进行镜像的凭证

创建容器镜像 registry 凭证文件，允许将红帽的镜像镜像到您的镜像环境中。

警告

安装集群时不要使用此镜像 registry 凭据文件作为 pull secret。如果在安装集群时提供此文件，集群中的所有机器都将具有镜像 registry 的写入权限。

警告

此过程需要您可以对镜像 registry 上的容器镜像 registry 进行写操作，并将凭证添加到 registry pull secret。

先决条件

- 配置了一个镜像（mirror） registry 在受限网络中使用。
- 您在镜像 registry 中标识了镜像仓库的位置，以将容器镜像镜像(mirror)到这个位置。
- 您置备了一个镜像 registry 帐户，允许将镜像上传到该镜像仓库。

流程

在安装主机上完成以下步骤：

1. 从 Red Hat OpenShift Cluster Manager 站点的 Pull Secret 页面下载 **registry.redhat.io** 的 pull secret，将它保存为一个 **.json** 文件。
2. 为您的镜像 registry 生成 base64 编码的用户名和密码或令牌：

```
$ echo -n '<user_name>:<password>' | base64 -w0 (1)
BGVtbYk3ZHAtqXs=
```

(1) 通过 **<user_name>** 和 **<password>** 指定 registry 的用户名和密码。

3. 以 JSON 格式创建您的 pull secret 副本：

```
$ cat ./pull-secret.text | jq .  > <path>/<pull-secret-file>
```
(1)

(1) 指定到存储 pull secret 的文件夹的路径，以及您创建的 JSON 文件的名称。

该文件类似于以下示例：

```
{
  "auths": {
    "cloud.openshift.com": {
      "auth": "b3BlbnNo...",
      "email": "you@example.com"
    },
    "quay.io": {
      "auth": "b3BlbnNo...",
      "email": "you@example.com"
    },
    "registry.connect.redhat.com": {
      "auth": "NTE3Njg5Nj...",
      "email": "you@example.com"
    },
    "registry.redhat.io": {
      "auth": "NTE3Njg5Nj...",
      "email": "you@example.com"
    }
  }
}
```

4. 编辑新文件并添加描述 registry 的部分：

```
  "auths": {
    "<mirror_registry>": {
      "auth": "<credentials>",
      "email": "you@example.com"
    },
```
(1) (2)

(1) 对于 **<mirror_registry>**，指定 registry 域名，以及您的镜像 registry 用来提供内容的可选端口。例如： **registry.example.com** 或 **registry.example.com:5000**

(2) 使用 **<credentials>** 为您的镜像 registry 指定 base64 编码的用户名和密码。

该文件类似于以下示例：

```
{
  "auths": {
    "<mirror_registry>": {
      "auth": "<credentials>",
      "email": "you@example.com"
    },
    "cloud.openshift.com": {
      "auth": "b3BlbnNo...",
      "email": "you@example.com"
    },
    "quay.io": {
```

```
    "auth": "b3BlbnNo...",
    "email": "you@example.com"
  },
  "registry.connect.redhat.com": {
    "auth": "NTE3Njg5Nj...",
    "email": "you@example.com"
  },
  "registry.redhat.io": {
    "auth": "NTE3Njg5Nj...",
    "email": "you@example.com"
  }
 }
}
```

5.4. 镜像 OPENSHIFT CONTAINER PLATFORM 镜像存储库

在受限网络中置备的基础架构上升级集群前，您必须将所需的容器镜像镜像(mirror)到那个环境中。您也可以在不受限制的网络中使用此流程来确保集群只使用满足您机构对外部内容控制的容器镜像。

流程

1. 查看 OpenShift Container Platform 升级路径，以确认您的当前和预定集群版本之间有升级路径。
2. 设置所需的环境变量：

```
$ OCP_RELEASE=<release_version> 1
$ LOCAL_REGISTRY='<local_registry_host_name>:<local_registry_host_port>' 2
$ LOCAL_REPOSITORY='<local_repository_name>' 3
$ PRODUCT_REPO='openshift-release-dev' 4
$ LOCAL_SECRET_JSON='<path_to_pull_secret>' 5
$ RELEASE_NAME='ocp-release' 6
$ ARCHITECTURE=<server_architecture> 7
$ REMOVABLE_MEDIA_PATH=<path> 8
```

1. 对于 **<release_version>**，请指定与升级到的 OpenShift Container Platform 版本对应的标签，如 **4.5.0**。
2. 对于 **<local_registry_host_name>**，请指定镜像存储库的 registry 域名；对于 **<local_registry_host_port>**，请指定用于提供内容的端口。
3. 对于 **<repository_name>**，请指定要在 registry 中创建的存储库名称，如 **ocp4/openshift4**。
4. 要镜像的存储库。对于生产环境版本，必须指定 **openshift-release-dev**。
5. 对于 **<path_to_pull_secret>**，请指定您创建的镜像 registry 的 pull secret 的绝对路径和文件名。
6. 对于生产环境版本，您必须指定 **ocp-release**。
7. 对于 **<server_architecture>**，指定服务器的构架，如 **x86_64**。
8. 对于 **<path>**，指定完整路径，包括开始的正斜杠(/)字符。

3. 查看要镜像的镜像和配置清单：

```
$ oc adm release mirror -a ${LOCAL_SECRET_JSON} --to-
dir=${REMOVABLE_MEDIA_PATH}/mirror
quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-${ARCHITECTURE}
--dry-run
```

4. 将版本镜像镜像(mirror)到内部容器 registry。

- 如果您的镜像主机无法访问互联网，请执行以下操作：

  i. 将可移动介质连接到连接到互联网的系统。

  ii. 将镜像和配置清单镜像到可移动介质的目录中：

```
$ oc adm release mirror -a ${LOCAL_SECRET_JSON} --to-
dir=${REMOVABLE_MEDIA_PATH}/mirror
quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-
${ARCHITECTURE}
```

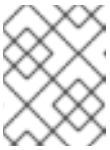
  iii. 将介质上传到受限网络环境中，并将镜像上传到本地容器 registry。

```
$ oc image mirror  -a ${LOCAL_SECRET_JSON} --from-
dir=${REMOVABLE_MEDIA_PATH}/mirror
"file://openshift/release:${OCP_RELEASE}*"
${LOCAL_REGISTRY}/${LOCAL_REPOSITORY} 1
```

1 对于 **REMOVABLE_MEDIA_PATH**，您必须使用与镜像镜像时指定的同一路径。

- 如果本地容器 registry 和集群连接到镜像主机，将发行镜像直接推送到本地 registry，并使用以下命令将配置映射应用到集群：

```
$ oc adm release mirror -a ${LOCAL_SECRET_JSON} --
from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-
${ARCHITECTURE} \
  --to=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY} --apply-release-image-signature
```



注意

如果包含 **--apply-release-image-signature** 选项，不要为镜像签名验证创建配置映射。

5.5. 创建镜像签名配置映射

在更新集群前，需要手动创建包含您使用的发行版本镜像签名的配置映射。此签名允许 Cluster Version Operator（CVO）通过比较预期的及实际镜像签名来验证发行的镜像没有被修改。

如果要从 4.4.8 或更高版本升级,您可以使用 **oc** CLI 创建配置映射。如果您是从更早的版本升级，则必须使用手动方法。

5.5.1. 使用 oc CLI 创建用于镜像签名验证的配置映射

在更新集群前，需要手动创建包含您使用的发行版本镜像签名的配置映射。此签名允许 Cluster Version Operator（CVO）通过比较预期的及实际镜像签名来验证发行的镜像没有被修改。

注意

如果从 4.4.8 之前的发行版本升级，则必须使用手动方法创建配置映射，而不是使用此流程。此流程使用的命令不在早期版本的 **oc** 命令行界面（CLI）中。

先决条件

- 安装 OpenShift CLI(**oc**)版本 4.4.8 或更高版本。

流程

1. 从 mirror.openshift.com 或 Google Cloud Storage(GCS) 获取您要升级到的版本的镜像签名。
2. 使用 **oc** 命令行界面(CLI)登录您要升级的集群。
3. 将镜像发行镜像签名配置映射应用到连接的集群：

```
$ oc apply -f <image_signature_file> 1
```

1 对于 **<image_signature_file>**，指定文件的路径和名称，例如 **mirror/config/signature-sha256-81154f5c03294534.yaml**。

5.5.2. 手动创建镜像签名配置映射

创建并应用镜像签名配置映射到您要更新的集群。

注意

每次更新集群时都必须执行以下步骤。

流程

1. 请参阅 OpenShift Container Platform 升级路径 知识库文章，以确定集群的有效升级路径。
2. 将版本添加到 **OCP_RELEASE_NUMBER** 环境变量中：

```
$ OCP_RELEASE_NUMBER=<release_version> 1
```

1 对于 **<release_version>**，请指定与集群升级到的 OpenShift Container Platform 版本对应的标签，如 **4.4.0**。

3. 将集群的系统构架添加到 **ARCHITECTURE** 环境变量中：

```
$ ARCHITECTURE=<server_architecture> 1
```

1 对于 **server_architecture**，指定服务器的构架，如 **x86_64**。

4. 从 Quay 获取发行版本镜像摘要：

```
$ DIGEST="$(oc adm release info quay.io/openshift-release-dev/ocp-release:${OCP_RELEASE_NUMBER}-${ARCHITECTURE} | sed -n 's/Pull From: .*@//p')"
```

5. 设置摘要算法：

```
$ DIGEST_ALGO="${DIGEST%%:*}"
```

6. 设置摘要签名：

```
$ DIGEST_ENCODED="${DIGEST#*:}"
```

7. 从 mirror.openshift.com 网站获取镜像签名。

```
$ SIGNATURE_BASE64=$(curl -s "https://mirror.openshift.com/pub/openshift-v4/signatures/openshift/release/${DIGEST_ALGO}=${DIGEST_ENCODED}/signature-1" | base64 -w0 && echo)
```

8. 创建配置映射：

```
$ cat >checksum-${OCP_RELEASE_NUMBER}.yaml <<EOF
apiVersion: v1
kind: ConfigMap
metadata:
  name: release-image-${OCP_RELEASE_NUMBER}
  namespace: openshift-config-managed
  labels:
    release.openshift.io/verification-signatures: ""
binaryData:
  ${DIGEST_ALGO}-${DIGEST_ENCODED}: ${SIGNATURE_BASE64}
EOF
```

9. 将配置映射应用到集群以更新：

```
$ oc apply -f checksum-${OCP_RELEASE_NUMBER}.yaml
```

5.6. 升级受限网络集群

将受限网络集群更新至您下载的发行镜像的 OpenShift Container Platform 版本。

先决条件

- 您已将新发行版本的镜像镜像（mirror）到 registry。
- 您已将发行镜像签名 ConfigMap 在新发行版本中应用到集群。
- 从镜像签名 ConfigMap 中获取了发行版本的 sha256 sum 值。
- 安装 OpenShift CLI(**oc**)版本 4.4.8 或更高版本。

流程

- 更新集群：

```
$ oc adm upgrade --allow-explicit-upgrade --to-image
${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}<sha256_sum_value> 1
```

1 **<sha256_sum_value>** 值是镜像签名 ConfigMap 的 sha256 sum 值，例如 **@sha256:81154f5c03294534e1eaf0319BEF7a601134f891689ccede5d705ef659aa8c92**

如果镜像 registry 使用 **ImageContentSourcePolicy**，可以使用 Canonical registry 名称而非 **LOCAL_REGISTRY**。

5.7. 配置镜像 REGISTRY 存储库镜像

通过设置容器 registry 存储库镜像，您可以：

- 配置 OpenShift Container Platform 集群，以便重定向从源镜像 registry 上的存储库拉取（pull）镜像的请求，并通过已镜像 (mirror) 的镜像 registry 上的存储库来解决该请求。
- 为每个目标存储库识别多个已镜像 (mirror)的存储库，以确保如果一个镜像停止运作，仍可使用其他镜像。

OpenShift Container Platform 中存储库镜像的属性包括：

- 镜像拉取（pull）可应对 registry 停机的问题
- 受限网络中的集群可以从关键位置（如 quay.io）拉取镜像，并让位于公司防火墙后的 registry 提供请求的镜像。
- 发出镜像拉取（pull）请求时尝试特定 registry 顺序，通常最后才会尝试持久性 registry。
- 您所输入的镜像信息会添加到 OpenShift Container Platform 集群中每个节点上的 **/etc/containers/registries.conf** 文件中。
- 当节点从源存储库中请求镜像时，它会依次尝试每个已镜像的存储库，直到找到所请求的内容。如果所有镜像均失败，集群则会尝试源存储库。如果成功，镜像则会被拉取（pull）至节点中。

可通过以下方式设置存储库镜像：

- 安装 OpenShift Container Platform 时：通过拉取（pull） OpenShift Container Platform 所需的容器镜像，然后将这些镜像放至公司防火墙后，即可将 OpenShift Container Platform 安装到受限网络中的数据中心。详情请参阅镜像 OpenShift Container Platform 的镜像存储库。
- 安装 OpenShift Container Platform 后：即使没有在 OpenShift Container Platform 安装期间配置镜像 (mirror)，之后您仍可使用 **ImageContentSourcePolicy** 对象进行配置。

以下流程提供安装后镜像配置，您可在此处创建 **ImageContentSourcePolicy** 对象来识别：

- 您希望镜像 (mirror) 的容器镜像存储库的源
- 您希望为其提供从源存储库请求的内容的每个镜像存储库的单独条目。

先决条件

- 使用具有 **cluster-admin** 角色的用户访问集群。

流程

1. 通过以下方法配置已镜像的存储库：

   - 按照 Red Hat Quay 存储库镜像 中所述，使用 Red Hat Quay 来设置已镜像的存储库。使用 Red Hat Quay 有助于您将镜像从一个存储库复制到另一存储库，并可随着时间的推移重复自动同步这些存储库。

   - 使用 **skopeo** 等工具手动将镜像从源目录复制到已镜像的存储库。
     例如：在 Red Hat Enterprise Linux（RHEL 7 或 RHEL 8）系统上安装 skopeo RPM 软件包后，使用 **skopeo** 命令，如下例所示：

     ```
     $ skopeo copy \
     docker://registry.access.redhat.com/ubi8/ubi-minimal@sha256:5cfbaf45ca96806917830c183e9f37df2e913b187adb32e89fd83fa455ebaa6 \
     docker://example.io/example/ubi-minimal
     ```

     在本例中，您有一个名为 **example.io** 的容器镜像 registry，其中包含一个名为 **example** 的镜像存储库，您希望将 **ubi8/ubi-minimal** 镜像从 **registry.access.redhat.com** 复制到此镜像存储库。创建该 registry 后，您可将 OpenShift Container Platform 集群配置为将源存储库的请求重定向到已镜像的存储库。

2. 登录您的 OpenShift Container Platform 集群。

3. 创建 **ImageContentSourcePolicy** 文件（如：**registryrepomirror.yaml**），将源和镜像 (mirror) 替换为您自己的 registry、存储库对和镜像中的源和镜像：

   ```
   apiVersion: operator.openshift.io/v1alpha1
   kind: ImageContentSourcePolicy
   metadata:
     name: ubi8repo
   spec:
     repositoryDigestMirrors:
     - mirrors:
       - example.io/example/ubi-minimal (1)
       source: registry.access.redhat.com/ubi8/ubi-minimal (2)
     - mirrors:
       - example.com/example/ubi-minimal
       source: registry.access.redhat.com/ubi8/ubi-minimal
   ```

   (1) 指明镜像 registry 和存储库的名称

   (2) 指明包含所镜像内容的 registry 和存储库

4. 创建新的 **ImageContentSourcePolicy** 对象：

   ```
   $ oc create -f registryrepomirror.yaml
   ```

   创建 **ImageContentSourcePolicy** 对象后，新的设置将部署到每个节点，集群开始使用已镜像的存储库来响应源存储库请求。

5. 要检查是否应用了已镜像的配置设置，在其中一个节点上执行以下内容。

   a. 列出您的节点：

      ```
      $ oc get node
      ```

输出示例

```
NAME                          STATUS                     ROLES    AGE  VERSION
ip-10-0-137-44.ec2.internal   Ready                      worker   7m   v1.17.1
ip-10-0-138-148.ec2.internal  Ready                      master   11m  v1.17.1
ip-10-0-139-122.ec2.internal  Ready                      master   11m  v1.17.1
ip-10-0-147-35.ec2.internal   Ready,SchedulingDisabled   worker   7m   v1.17.1
ip-10-0-153-12.ec2.internal   Ready                      worker   7m   v1.17.1
ip-10-0-154-10.ec2.internal   Ready                      master   11m  v1.17.1
```

您可以发现，在应用更改时每个 worker 节点上的调度都会被禁用。

b. 启动调试过程以访问节点：

```
$ oc debug node/ip-10-0-147-35.ec2.internal
```

输出示例

```
Starting pod/ip-10-0-147-35ec2internal-debug ...
To use host binaries, run `chroot /host`
```

c. 访问节点的文件：

```
sh-4.2# chroot /host
```

d. 检查 **/etc/containers/registries.conf** 文件，确保已完成更改：

```
sh-4.2# cat /etc/containers/registries.conf
```

输出示例

```
unqualified-search-registries = ["registry.access.redhat.com", "docker.io"]
[[registry]]
  location = "registry.access.redhat.com/ubi8/"
  insecure = false
  blocked = false
  mirror-by-digest-only = true
  prefix = ""

  [[registry.mirror]]
    location = "example.io/example/ubi8-minimal"
    insecure = false

  [[registry.mirror]]
    location = "example.com/example/ubi8-minimal"
    insecure = false
```

e. 将镜像摘要从源拉取到节点，并检查是否通过镜像解析。**ImageContentSourcePolicy** 对象仅支持镜像摘要，不支持镜像标签。

```
sh-4.2# podman pull --log-level=debug registry.access.redhat.com/ubi8/ubi-
minimal@sha256:5cfbaf45ca96806917830c183e9f37df2e913b187adb32e89fd83fa455ebaa6
```

存储库镜像故障排除

如果存储库镜像流程未按规定工作，请使用以下有关存储库镜像如何工作的信息协助排查问题。

- 首个工作镜像用于提供拉取（pull）的镜像。
- 只有在无其他镜像工作时，才会使用主 registry。
- 从系统上下文，**Insecure** 标志用作回退。
- 最近更改了 **/etc/containers/registries** 文件的格式。现在它是第 2 版，采用 TOML 格式。