OpenShift Container Platform 4.14

断开连接的安装镜像

同步 (mirror) 安装容器镜像

Last Updated: 2025-10-26

OpenShift Container Platform 4.14 断开连接的安装镜像

同步 (mirror) 安装容器镜像

Legal Notice

Abstract

本文档论述了如何为断开连接的 OpenShift Container Platform 安装同步（mirror）安装容器镜像。

Table of Contents

第 1 章 关于断开连接的安装镜像

作为一个集群管理员，您可以使用镜像 registry 确保集群只使用满足机构对外部内容控制的容器镜像。

1.1. 用于在断开连接的环境中安装集群的镜像 REGISTRY

您必须将所需的容器镜像镜像(mirror)到断开连接的环境中，然后才能安装并置备集群。要 mirror 这些容器镜像，您必须有一个 mirror registry。考虑以下用于创建和使用 mirror registry 的选项：

- 如果您已有容器镜像 registry，如 Red Hat Quay，您可以使用它作为您的 mirror registry。如果您还没有 registry，您必须创建一个。
- 建立 registry 后，您需要镜像工具。要在断开连接的环境中将 OpenShift Container Platform 镜像存储库 mirror 到 mirror registry，您可以使用 oc-mirror OpenShift CLI (**oc**) 插件。oc-mirror 插件是一个单一工具，它会将所有所需的 OpenShift Container Platform 内容和其他镜像 mirror 到您的 mirror registry。oc-mirror 插件是镜像的首选方法。
- 另外，您可以使用 **oc adm** 命令 mirror OpenShift Container Platform 的发行版本和目录镜像。

1.2. 后续步骤

- 使用 *mirror registry for Red Hat OpenShift* 创建镜像 registry
- 为断开连接的安装 mirror 镜像
- 使用 oc-mirror 插件为断开连接的安装镜像镜像
- 在断开连接的环境中使用 Operator Lifecycle Manager。

第 2 章 使用 MIRROR REGISTRY FOR RED HAT OPENSHIFT 创建 MIRROR REGISTRY

mirror registry for Red Hat OpenShift 是一个小型灵活的容器 registry，作为目标，用于为断开连接的安装镜像(mirror)的 OpenShift Container Platform 所需的容器镜像。

如果您已有容器镜像 registry，如 Red Hat Quay，您可以跳过本节并直接 mirror OpenShift Container Platform 镜像仓库。

重要

mirror registry for Red Hat OpenShift 的目的并不是替代 Red Hat Quay 的生产环境部署。

2.1. 先决条件

- OpenShift Container Platform 订阅。
- 安装了 Podman 3.4.2 或更高版本以及 OpenSSL 的 Red Hat Enterprise Linux (RHEL) 8 和 9。
- Red Hat Quay 服务的完全限定域名，它必须通过 DNS 服务器解析。
- 目标主机上的基于密钥的 SSH 连接。为本地安装自动生成 SSH 密钥。对于远程主机，您必须生成自己的 SSH 密钥。
- 2 个或更多 vCPU。
- 8 GB RAM。
- OpenShift Container Platform 4.14 发行镜像大约需要 12 GB；OpenShift Container Platform 4.14 发行镜像和 OpenShift Container Platform 4.14 Red Hat Operator 镜像大约需要 358 GB。

重要

- 每个流推荐具有 1 TB 或更多空间。
- 这些要求基于本地测试结果，且只测试了发行镜像和 Operator 镜像。存储要求可能会因您的组织的需求而有所不同。例如，当镜像了多个 z-streams 时，则可能需要更多空间。您可以使用标准 Red Hat Quay 功能 或适当的 API 调用来 删除不必要的镜像并释放空间。

2.2. MIRROR REGISTRY FOR RED HAT OPENSHIFT 介绍

对于断开连接的 OpenShift Container Platform 部署，需要一个容器 registry 来安装集群。要在这样的集群中运行 production-grade registry 服务，您必须创建一个单独的 registry 部署来安装第一个集群。*mirror registry for Red Hat OpenShift* 可以解决这个问题，并包含在每个 OpenShift 订阅中。它可用于从 OpenShift 控制台 **Downloads** 页面下载。

mirror registry for Red Hat OpenShift 允许用户使用 **mirror-registry** 命令行界面(CLI)工具安装一个较小的 Red Hat Quay 版本及其所需的组件。*mirror registry for Red Hat OpenShift* 会自动部署，带有预先配置的本地存储和本地数据库。它还包括自动生成的用户凭证和访问权限，其中只有一个输入集，且不需要额外配置选项。

mirror registry for Red Hat OpenShift 提供了一个预先确定的网络配置，并在成功时报告部署的组件凭证并访问 URL。另外还提供了一组有限的可选配置输入，如完全限定域名(FQDN)服务、超级用户名称和密

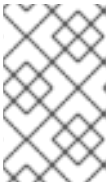
码，以及自定义 TLS 证书。这为用户提供了一个容器 registry，以便在受限网络环境中运行 OpenShift Container Platform 时，轻松创建所有 OpenShift Container Platform 发行版本内容的离线镜像。

如果在安装环境中已有另一个容器 registry，则使用 *mirror registry for Red Hat OpenShift* 是可选的。

2.2.1. Mirror registry for Red Hat OpenShift 限制

以下限制适用于 *mirror registry for Red Hat OpenShift* ：

- *mirror registry for Red Hat OpenShift* 并不是一个高度可用的 registry，且只支持本地文件系统存储。它并不适用于 OpenShift Container Platform 替换 Red Hat Quay 或内部镜像 registry。
- *mirror registry for Red Hat OpenShift* 的目的并不是替代 Red Hat Quay 的生产环境部署。
- *mirror registry for Red Hat OpenShift* 只支持托管安装断开连接的 OpenShift Container Platform 集群（如发行镜像或 Red Hat Operator 镜像）所需的镜像。它使用 Red Hat Enterprise Linux (RHEL)机器上的本地存储，RHEL 支持的存储则由 *mirror registry for Red Hat OpenShift* 支持。



注意

因为 *mirror registry for Red Hat OpenShift* 使用本地存储，所以您应该了解 mirror 镜像时消耗的存储使用量，并使用 Red Hat Quay 的垃圾回收功能来缓解潜在的问题。有关此功能的更多信息，请参阅"Red Hat Quay 垃圾回收"。

- 对于推送到 *Red Hat OpenShift for bootstrap 目的的红帽产品镜像* 的支持，每个相应的产品订阅会涵盖有效的订阅。进一步启用 bootstrap 体验的例外列表可在 自助管理的 Red Hat OpenShift 大小和订阅指南 中找到。
- 由客户创建的内容不应由 *mirror registry for Red Hat OpenShift* 托管。
- 不建议将 *mirror registry for Red Hat OpenShift* 与多个集群一起使用，因为多个集群可以在更新集群时造成单点故障。反之，使用 *mirror registry for Red Hat OpenShift* 安装一个集群，该集群可以托管一个生产环境级别的、高度可用的 registry，如 Red Hat Quay，可将 OpenShift Container Platform 内容提供给其他集群。

2.3. 使用 MIRROR REGISTRY FOR RED HAT OPENSHIFT 镜像一个本地主机

此流程解释了如何使用 **mirror-registry** 安装程序工具在本地主机上安装 *mirror registry for Red Hat OpenShift*。这样，用户可以创建在端口 443 上运行的本地主机 registry，以存储 OpenShift Container Platform 镜像的镜像。

注意

使用 **mirror-registry** CLI 工具安装 *mirror registry for Red Hat OpenShift* 会对机器进行几个更改。安装后，会创建一个 **$HOME/quay-install** 目录，其中包含安装文件、本地存储和配置捆绑包。如果部署目标是本地主机，则生成可信 SSH 密钥，并且设置主机计算机上的 systemd 文件，以确保容器运行时持久。另外，会创建一个名为 **init** 的初始用户，并自动生成的密码。所有访问凭证都会在安装例程的末尾打印。

流程

1. 从 OpenShift console **Downloads** 页下载最新版本的 *mirror registry for Red Hat OpenShift* 的 **mirror-registry.tar.gz** 软件包。

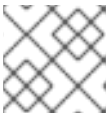
2. 使用 **mirror-registry** 工具，在本地主机上安装 *mirror registry for Red Hat OpenShift*。有关可用标志的完整列表，请参阅 "mirror registry for Red Hat OpenShift flags"。

```
$ ./mirror-registry install \
  --quayHostname <host_example_com> \
  --quayRoot <example_directory_name>
```

3. 运行以下命令，使用安装期间生成的用户名和密码登录 registry：

```
$ podman login -u init \
  -p <password> \
  <host_example_com>:8443> \
  --tls-verify=false 1
```

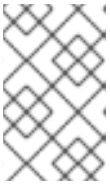
1 您可以通过将您的系统配置为信任生成的 rootCA 证书来避免运行 **--tls-verify=false**。如需更多信息，请参阅"保护 Red Hat Quay"和"将系统配置为信任证书颁发机构"。



注意

您还可以在安装后通过 **https://<host.example.com>:8443** 访问 UI 登录。

4. 您可以在登录后镜像 OpenShift Container Platform 镜像。根据您的需要，请参阅本文档的"镜像 OpenShift Container Platform 镜像存储库"或"镜像 Operator 目录"部分以用于此文档。

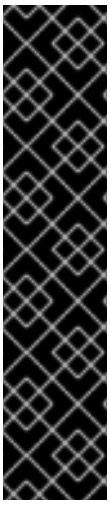


注意

如果因为存储层问题导致 *Red Hat OpenShift 镜像存储了镜像 registry* 存在问题，您可以在更稳定的存储上重新镜像 OpenShift Container Platform 镜像或重新安装 registry。

2.4. 从一个本地主机更新 MIRROR REGISTRY FOR RED HAT OPENSHIFT

此流程解释了如何使用 **upgrade** 命令从本地主机更新 *mirror registry for Red Hat OpenShift*。更新至最新版本可确保新的功能、错误修复和安全漏洞修复。



重要

当从版本 1 升级到版本 2 时，请注意以下限制：

- worker 数量被设置为 **1**，因为 SQLite 中不允许多个写入。
- 您不能使用 *mirror registry for Red Hat OpenShift* 用户接口 (UP)。
- 不要在升级过程中访问 **sqlite-storage** Podman 卷。
- 镜像 registry 会出现间歇性停机时间，因为它会在升级过程中重启。
- PostgreSQL 数据在 **/$HOME/quay-install/quay-postgres-backup/** 目录下备份，以进行恢复。

先决条件

- 您已在本地主机上安装了 *mirror registry for Red Hat OpenShift*。

流程

- 如果您要将 *mirror registry for Red Hat OpenShift* 从 1.3 升级到 2.y，且您的安装目录默认为 **/etc/quay-install**，您可以输入以下命令：

```
$ sudo ./mirror-registry upgrade -v
```

注意

- *mirror registry for Red Hat OpenShift* 将 Red Hat Quay 的 Podman 卷、Postgres 数据和 **/etc/quay-install** 数据迁移到新的 **$HOME/quay-install** 位置。这可让您在以后的升级过程中使用没有 **--quayRoot** 标记的 *mirror registry for Red Hat OpenShift*。
- 在使用 **./mirror-registry upgrade -v** 标记升级 *mirror registry for Red Hat OpenShift* 时需要包括在创建 mirror registry 时使用的相同的凭证。例如，如果使用 **--quayHostname <host_example_com>** 和 **--quayRoot <example_directory_name>** 安装了 *mirror registry for Red Hat OpenShift*，则必须包括该字符串来正确地升级 mirror registry。

- 如果您要将 *mirror registry for Red Hat OpenShift* 从 1.3 升级到 2.y，且您在 1.y 部署中使用自定义 quay 配置和存储目录，则必须传递-- **quayRoot** 和- **quayStorage** 标志。例如：

```
$ sudo ./mirror-registry upgrade --quayHostname <host_example_com> --quayRoot
<example_directory_name>  --quayStorage <example_directory_name>/quay-storage -v
```

- 如果您要从 1.3 升级到 2.y *mirror registry for Red Hat OpenShift*，并希望指定自定义 SQLite 存储路径，您必须传递 **--sqliteStorage** 标志，例如：

```
$ sudo ./mirror-registry upgrade --sqliteStorage <example_directory_name>/sqlite-storage -v
```

验证

1. 运行以下命令，确保 *mirror registry for Red Hat OpenShift* 已更新：

```
$ podman ps
```

输出示例

```
registry.redhat.io/quay/quay-rhel8:v3.12.10
```

2.5. 使用 MIRROR REGISTRY FOR RED HAT OPENSHIFT 在远程主机上 MIRROR

此流程解释了如何使用 **mirror-registry** 工具在远程主机上安装 *mirror registry for Red Hat OpenShift*。这样，用户可以创建 registry 来保存 OpenShift Container Platform 镜像的镜像。

注意

使用 **mirror-registry** CLI 工具安装 *mirror registry for Red Hat OpenShift* 会对机器进行几个更改。安装后，会创建一个 **$HOME/quay-install** 目录，其中包含安装文件、本地存储和配置捆绑包。如果部署目标是本地主机，则生成可信 SSH 密钥，并且设置主机计算机上的 systemd 文件，以确保容器运行时持久。另外，会创建一个名为 **init** 的初始用户，并自动生成的密码。所有访问凭证都会在安装例程的末尾打印。

流程

1. 从 OpenShift console **Downloads** 页下载最新版本的 *mirror registry for Red Hat OpenShift* 的 **mirror-registry.tar.gz** 软件包。

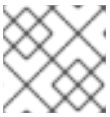
2. 使用 **mirror-registry** 工具，在本地主机上安装 *mirror registry for Red Hat OpenShift*。有关可用标志的完整列表，请参阅 "mirror registry for Red Hat OpenShift flags"。

```
$ ./mirror-registry install -v \
  --targetHostname <host_example_com> \
  --targetUsername <example_user> \
  -k ~/.ssh/my_ssh_key \
  --quayHostname <host_example_com> \
  --quayRoot <example_directory_name>
```

3. 运行以下命令，使用安装期间生成的用户名和密码登录 到镜像的 registry：

```
$ podman login -u init \
  -p <password> \
  <host_example_com>:8443> \
  --tls-verify=false 1
```

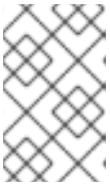
1 您可以通过将您的系统配置为信任生成的 rootCA 证书来避免运行 **--tls-verify=false**。如需更多信息，请参阅"使用 SSL 保护到 Red Hat Quay 的连接"和"将系统配置为信任证书颁发机构"。



注意

您还可以在安装后通过 **https://<host.example.com>:8443** 访问 UI 登录。

4. 您可以在登录后镜像 OpenShift Container Platform 镜像。根据您的需要，请参阅本文档的"镜像 OpenShift Container Platform 镜像存储库"或"镜像 Operator 目录"部分以用于此文档。

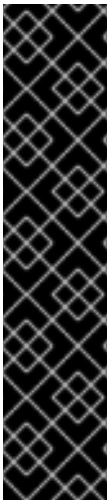


注意

如果因为存储层问题导致 *Red Hat OpenShift 镜像存储了镜像 registry* 存在问题，您可以在更稳定的存储上重新镜像 OpenShift Container Platform 镜像或重新安装 registry。

2.6. 从一个远程主机为 RED HAT OPENSHIFT 更新 MIRROR REGISTRY

此流程解释了如何使用 **upgrade** 命令从远程主机更新 *mirror registry for Red Hat OpenShift*。更新至最新版本可确保程序错误修复和安全漏洞。



重要

当从版本 1 升级到版本 2 时，请注意以下限制：

- worker 数量被设置为 **1**，因为 SQLite 中不允许多个写入。
- 您不能使用 *mirror registry for Red Hat OpenShift* 用户接口 (UP)。
- 不要在升级过程中访问 **sqlite-storage** Podman 卷。
- 镜像 registry 会出现间歇性停机时间，因为它会在升级过程中重启。
- PostgreSQL 数据在 **/$HOME/quay-install/quay-postgres-backup/** 目录下备份，以进行恢复。

先决条件

- 您已在远程主机上安装了 *mirror registry for Red Hat OpenShift*。

流程

- 要从远程主机升级 *mirror registry for Red Hat OpenShift*，请输入以下命令：

```
$ ./mirror-registry upgrade -v --targetHostname <remote_host_url> --targetUsername
<user_name> -k ~/.ssh/my_ssh_key
```

注意

在使用 **./mirror-registry upgrade -v** 标记升级 *mirror registry for Red Hat OpenShift* 时需要包括在创建 mirror registry 时使用的相同的凭证。例如，如果使用 **--quayHostname <host_example_com>** 和 **--quayRoot <example_directory_name>** 安装了 *mirror registry for Red Hat OpenShift*，则必须包括该字符串来正确地升级 mirror registry。

- 如果您要从 1.3 升级到 2.y *mirror registry for Red Hat OpenShift*，并希望指定自定义 SQLite 存储路径，您必须传递 **--sqliteStorage** 标志，例如：

```
$ ./mirror-registry upgrade -v --targetHostname <remote_host_url> --targetUsername
<user_name> -k ~/.ssh/my_ssh_key --sqliteStorage <example_directory_name>/quay-
storage
```

验证

1. 运行以下命令，确保 *mirror registry for Red Hat OpenShift* 已更新：

```
$ podman ps
```

输出示例

```
registry.redhat.io/quay/quay-rhel8:v3.12.10
```

2.7. 替换 MIRROR REGISTRY FOR RED HAT OPENSHIFT SSL/TLS 证书

在某些情况下，您可能想要为 *mirror registry for Red Hat OpenShift* 更新 SSL/TLS 证书。这在以下情况中很有用：

- 如果您需要替换当前的 *mirror registry for Red Hat OpenShift* 证书。
- 如果您使用与之前 *mirror registry for Red Hat OpenShift* 安装相同的证书。
- 如果您希望定期更新 *mirror registry for Red Hat OpenShift* 证书。

使用以下步骤替换 *mirror registry for Red Hat OpenShift* SSL/TLS 证书。

先决条件

- 您已从 OpenShift 控制台的 **Downloads** 页中下载并安装了 **./mirror-registry** 二进制文件。

流程

1. 输入以下命令安装 *mirror registry for Red Hat OpenShift*：

```
$ ./mirror-registry install \
--quayHostname <host_example_com> \
--quayRoot <example_directory_name>
```

   这会将 *mirror registry for Red Hat OpenShift* 安装到 **$HOME/quay-install** 目录中。

2. 准备一个新的证书颁发机构(CA)捆绑包，并生成新的 **ssl.key** 和 **ssl.crt** 密钥文件。如需更多信息，请参阅为 Red Hat Quay 配置 SSL 和 TLS 。

3. 输入以下命令为 **/$HOME/quay-install** 分配一个环境变量，如 **QUAY** ：

```
$ export QUAY=/$HOME/quay-install
```

4. 输入以下命令将新的 **ssl.crt** 文件复制到 **/$HOME/quay-install** 目录中：

```
$ cp ~/ssl.crt $QUAY/quay-config
```

5. 输入以下命令将新的 **ssl.key** 文件复制到 **/$HOME/quay-install** 目录中：

```
$ cp ~/ssl.key $QUAY/quay-config
```

6. 输入以下命令重启 **quay-app** 应用程序 pod：

```
$ systemctl --user restart quay-app
```

2.8. 卸载 MIRROR REGISTRY FOR RED HAT OPENSHIFT

使用以下步骤从本地主机卸载 *Red Hat OpenShift 的 mirror registry*。

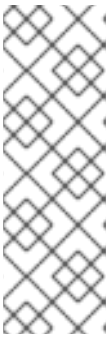
先决条件

- 您已在本地主机上安装了 *mirror registry for Red Hat OpenShift*。

流程

- 运行以下命令，从本地主机中卸载 *mirror registry for Red Hat OpenShift* ：

```
$ ./mirror-registry uninstall -v \
  --quayRoot <example_directory_name>
```



注意

- 删除 *mirror registry for Red Hat OpenShift* 会在删除前提示用户。您可以使用 **--autoApprove** 来跳过此提示。
- 如果使用 **--quayRoot** 标志安装了 *mirror registry for Red Hat OpenShift*，则卸载时也需要使用 **--quayRoot** 标志。例如，如果您安装了带有 **--quayRoot example_directory_name** 的 *Red Hat OpenShift 的 mirror registry*，则必须包含该字符串才能正确卸载 mirror registry。

2.9. MIRROR REGISTRY FOR RED HAT OPENSHIFT 标记

以下标记可用于 *mirror registry for Red Hat OpenShift* ：

标记	描述
--autoApprove	禁用交互式提示的布尔值。如果设置为 **true**，则在卸载镜像 registry 时自动删除 **quayRoot** 目录。如果未指定，则默认为 **false**。
--initPassword	在 Quay 安装过程中创建的 init 用户的密码。必须至少包含八个字符，且不包含空格。
--initUser string	显示初始用户的用户名。若未指定，则默认为 **init**。
--no-color, **-c**	允许用户禁用颜色序列，在运行安装、卸载和升级命令时将其传播到 Ansible。
--quayHostname	客户端用来联系 registry 的镜像 registry 的完全限定域名。等同于 Quay **config.yaml** 中的 **SERVER_HOSTNAME**。必须可以被 DNS 解析。如果未指定，则默认为 **<targetHostname>:8443**。[1]
--quayStorage	保存 Quay 持久性存储数据的文件夹。默认为 **quay-storage** Podman 卷。卸载需要 root 权限。
--quayRoot, **-r**	保存容器镜像层和配置数据的目录，包括 **rootCA.key**、rootCA**.pem** 和 **rootCA.srl** 证书。如果未指定，则默认为 **$HOME/quay-install**。
--sqliteStorage	保存 SQLite 数据库数据的文件夹。如果没有指定，则默认为 **sqlite-storage** Podman 卷。卸载需要 root。
--ssh-key,**-k**	SSH 身份密钥的路径。如果未指定，则默认为 **~/.ssh/quay_installer**。

标记	描述
--sslCert	SSL/TLS 公钥/证书的路径。默认为 **{quayRoot}/quay-config**，并在未指定时自动生成。
--sslCheckSkip	跳过对 **config.yaml** 文件中的 **SERVER_HOSTNAME** 的检查证书主机名。[2]
--sslKey	用于 HTTPS 通信的 SSL/TLS 私钥路径。默认为 **{quayRoot}/quay-config**，并在未指定时自动生成。
--targetHostname, -H	要安装 Quay 的目标的主机名。默认为 **$HOST**，如本地主机（如果未指定）。
--targetUsername, -u	目标主机上的用户，将用于 SSH。默认为 **$USER**，例如，如果未指定，则默认为当前用户。
--verbose,-v	显示调试日志和 Ansible playbook 输出。
--version	显示 *mirror registry for Red Hat OpenShift*的版本。

1. 如果您的系统的公共 DNS 名称与本地主机名不同，则必须修改 **--quayHostname**。另外，**--quayHostname** 标志不支持使用 IP 地址的安装。需要使用主机名进行安装。
2. 当镜像 registry 在代理后面设置时，会使用 **--sslCheckSkip**，并且公开的主机名与内部 Quay 主机名不同。当用户不希望在安装过程中对提供的 Quay 主机名验证证书时，也可以使用它。

2.10. MIRROR REGISTRY FOR RED HAT OPENSHIFT 发行注记

mirror registry for Red Hat OpenShift 是一个小型灵活的容器 registry，作为目标，用于为断开连接的安装镜像(mirror)的 OpenShift Container Platform 所需的容器镜像。

本发行注记介绍了 OpenShift Container Platform 的 *mirror registry for Red Hat OpenShift* 。

2.10.1. Mirror registry for Red Hat OpenShift 2.0 发行注记

以下小节详细介绍了 mirror registry for Red Hat OpenShift 的每个 2.0 发行版本。

2.10.1.1. Mirror registry for Red Hat OpenShift 2.0.8

发布日期： 2025 年 10 月 16 日

Mirror registry for Red Hat OpenShift 现在包括在 Red Hat Quay 3.12.12 中。

以下公告适用于 *mirror registry for Red Hat OpenShift* ：

- RHBA-2025:17062 - mirror registry for Red Hat OpenShift 2.0.8

2.10.1.2. Mirror registry for Red Hat OpenShift 2.0.7

发布日期： 2025 年 7 月 14 日

Mirror registry for Red Hat OpenShift 现在包括在 Red Hat Quay 3.12.10 中。

以下公告适用于 *mirror registry for Red Hat OpenShift* ：

- RHBA-2025:9645 - mirror registry for Red Hat OpenShift 2.0.7

2.10.1.3. Mirror registry for Red Hat OpenShift 2.0.6

发布日期：2025 年 4 月 28 日

Mirror registry for Red Hat OpenShift 现在包括在 Red Hat Quay 3.12.8 中。

以下公告适用于 *mirror registry for Red Hat OpenShift* ：

- RHBA-2025:4251 - mirror registry for Red Hat OpenShift 2.0.6

2.10.1.4. Mirror registry for Red Hat OpenShift 2.0.5

发布日期：2025 年 1 月 13 日

Mirror registry for Red Hat OpenShift 现在包括在 Red Hat Quay 3.12.5 中。

以下公告适用于 *mirror registry for Red Hat OpenShift* ：

- RHBA-2025:0298 - mirror registry for Red Hat OpenShift 2.0.5

2.10.1.5. Mirror registry for Red Hat OpenShift 2.0.4

发布日期： 2025 年 1 月 6 日

Mirror registry for Red Hat OpenShift 现在包括在 Red Hat Quay 3.12.4 中。

以下公告适用于 *mirror registry for Red Hat OpenShift* ：

- RHBA-2025:0033 - mirror registry for Red Hat OpenShift 2.0.4

2.10.1.6. Mirror registry for Red Hat OpenShift 2.0.3

发布日期： 2024 年 11 月 25 日

Mirror registry for Red Hat OpenShift 现在包括在 Red Hat Quay 3.12.3 中。

以下公告适用于 *mirror registry for Red Hat OpenShift* ：

- RHBA-2024:10181 - mirror registry for Red Hat OpenShift 2.0.3

2.10.1.7. Mirror registry for Red Hat OpenShift 2.0.2

发布日期：2024 年 10 月 31 日

Mirror registry for Red Hat OpenShift 现在包括在 Red Hat Quay 3.12.2 中。

以下公告适用于 *mirror registry for Red Hat OpenShift* ：

- RHBA-2024:8370 - mirror registry for Red Hat OpenShift 2.0.2

2.10.1.8. Mirror registry for Red Hat OpenShift 2.0.1

发布日期：2024 年 9 月 26 日

Mirror registry for Red Hat OpenShift 现在包括在 Red Hat Quay 3.12.1 中。

以下公告适用于 *mirror registry for Red Hat OpenShift* ：

- RHBA-2024:7070 - mirror registry for Red Hat OpenShift 2.0.1

2.10.1.9. Mirror registry for Red Hat OpenShift 2.0.0

发布日期： 2024 年 9 月 3 日

Mirror registry for Red Hat OpenShift 现在包括在 Red Hat Quay 3.12.0 中。

以下公告适用于 *mirror registry for Red Hat OpenShift* ：

- RHBA-2024:5277 - mirror registry for Red Hat OpenShift 2.0.0

以下新功能包括在 *mirror registry for Red Hat OpenShift* 2.0.0 中：

- 随着 *mirror registry for Red Hat OpenShift* 的发布，内部数据库已从 PostgreSQL 升级到 SQLite。因此，数据现在默认存储在 **sqlite-storage** Podman 卷中，整个 tarball 大小会减少 300 MB。
  新的安装默认使用 SQLite。在升级到 2.0 之前，请参阅"根据您的环境，"从本地主机更新 mirror registry for Red Hat OpenShift"或"从远程主机更新 mirror registry for Red Hat OpenShift"。
- 添加了新功能标志 **--sqliteStorage**。使用这个标志，您可以手动设置保存 SQLite 数据库数据的位置。
- *Mirror registry for Red Hat OpenShift* 现在可用于 IBM Power 和 IBM Z 架构（ **s390x** 和 **ppc64le**）。

2.10.2. Mirror registry for Red Hat OpenShift 1.3 发行注记

要查看 *mirror registry for Red Hat OpenShift* 1.3 发行注记，请参阅 Mirror registry for Red Hat OpenShift 1.3 发行注记。

2.10.3. Mirror registry for Red Hat OpenShift 1.2 发行注记

要查看 *mirror registry for Red Hat OpenShift* 1.2 发行注记，请参阅 Mirror registry for Red Hat OpenShift 1.2 发行注记。

2.10.4. Mirror registry for Red Hat OpenShift 1.1 发行注记

要查看 *mirror registry for Red Hat OpenShift* 1.1 发行注记，请参阅 Mirror registry for Red Hat OpenShift 1.1 发行注记。

2.11. MIRROR REGISTRY FOR RED HAT OPENSHIFT 故障排除

为了帮助对 *mirror registry for Red Hat OpenShift* 进行故障排除，您可以收集由 mirror registry 安装的 systemd 服务的日志。安装以下服务：

- quay-app.service
- quay-postgres.service
- quay-redis.service
- quay-pod.service

先决条件

- 您已安装了 *mirror registry for Red Hat OpenShift*。

流程

- 如果使用 root 权限安装 *mirror registry for Red Hat OpenShift*，您可以输入以下命令获取其 systemd 服务的状态信息：

```
$ sudo systemctl status <service>
```

- 如果作为标准用户安装 *mirror registry for Red Hat OpenShift*，您可以输入以下命令获取其 systemd 服务的状态信息：

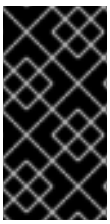
```
$ systemctl --user status <service>
```

2.12. 其他资源

- Red Hat Quay 垃圾回收
- 保护 Red Hat Quay
- 将系统配置为信任证书颁发机构
- 镜像 OpenShift Container Platform 镜像存储库
- 镜像用于断开连接的集群的 Operator 目录

第 3 章 为断开连接的安装 MIRROR 镜像

您可以确保集群只使用满足您机构对外部内容控制的容器镜像。在受限网络中置备的基础架构上安装集群前，您必须将所需的容器镜像镜像(mirror)到那个环境中。要镜像容器镜像，您必须有一个 registry 才能进行镜像(mirror)。



重要

您必须可以访问互联网来获取所需的容器镜像。在这一流程中，您要将镜像 registry 放在可访问您的网络以及互联网的镜像（mirror）主机上。如果您没有镜像主机的访问权限，请使用镜像 Operator 目录与断开连接的集群流程一起使用，将镜像复制到可跨网络界限的设备。

3.1. 先决条件

- 您必须在托管 OpenShift Container Platform 集群的位置（如以下 registry 之一）中有一个支持 Docker v2-2 的容器镜像 registry：
  - Red Hat Quay
  - JFrog Artifactory
  - Sonatype Nexus 仓库
  - Harbor

  如果您有 Red Hat Quay 权利，请参阅有关部署 Red Hat Quay 以了解概念验证的文档，或使用 Red Hat Quay Operator。如果您需要额外的帮助来选择并安装 registry，请联络您的销售代表或红帽支持。
- 如果您还没有容器镜像 registry，OpenShift Container Platform 可以为订阅者提供一个 mirror registry for Red Hat OpenShift。*mirror registry for Red Hat OpenShift* 包含在您的订阅中，它是一个小型容器 registry，可用于在断开连接的安装中镜像 OpenShift Container Platform 所需的容器镜像。

3.2. 关于镜像 REGISTRY

您可以镜像 OpenShift Container Platform 安装所需的镜像，以及容器镜像 registry 的后续产品更新，如 Red Hat Quay、JFrog Artifactory、Sonatype Nexus Repository 或 Harbor。如果您无法访问大型容器 registry，可以使用 *mirror registry for Red Hat OpenShift*，它是包括在 OpenShift Container Platform 订阅中的一个小型容器 registry。

您可以使用支持 Docker v2-2 的任何容器 registry，如 Red Hat Quay，*mirror registry for Red Hat OpenShift*, Artifactory, Sonatype Nexus Repository, 或 Harbor。无论您所选 registry 是什么，都会将互联网上红帽托管站点的内容镜像到隔离的镜像 registry 相同。镜像内容后，您要将每个集群配置为从镜像 registry 中检索此内容。

重要

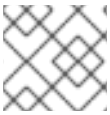
OpenShift 镜像 registry 不能用作目标 registry，因为它不支持没有标签的推送，在镜像过程中需要这个推送。

如果选择的容器 registry 不是 *mirror registry for Red Hat OpenShift*，则需要集群中置备的每台机器都可以访问它。如果 registry 无法访问，安装、更新或常规操作（如工作负载重新定位）可能会失败。因此，

您必须以高度可用的方式运行镜像 registry，镜像 registry 至少必须与 OpenShift Container Platform 集群的生产环境可用性相匹配。

使用 OpenShift Container Platform 镜像填充镜像 registry 时，可以遵循以下两种情况。如果您的主机可以同时访问互联网和您的镜像 registry，而不能访问您的集群节点，您可以直接从该机器中镜像该内容。这个过程被称为 *连接的镜像(mirror)*。如果没有这样的主机，则必须将该镜像文件镜像到文件系统中，然后将该主机或者可移动介质放入受限环境中。这个过程被称为 *断开连接的镜像*。

对于已镜像的 registry，若要查看拉取镜像的来源，您必须查看 **Trying 以访问** CRI-O 日志中的日志条目。查看镜像拉取源的其他方法（如在节点上使用 **crictl images** 命令）显示非镜像镜像名称，即使镜像是从镜像位置拉取的。



注意

红帽没有针对 OpenShift Container Platform 测试第三方 registry。

附加信息

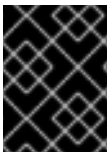
有关查看 CRI-O 日志以查看镜像源的详情，请参阅查看镜像拉取源。

3.3. 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。

3.3.1. 通过下载二进制文件安装 OpenShift CLI

您可以安装 OpenShift CLI(**oc**)来使用命令行界面与 OpenShift Container Platform 进行交互。您可以在 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 4.14 中的所有命令。下载并安装新版本的 **oc**。

3.3.1.1. 在 Linux 上安装 OpenShift 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. 导航到红帽客户门户网站上的 OpenShift Container Platform 下载页面。
2. 从 **产品变体** 下拉列表中选择架构。
3. 从 **版本** 下拉列表中选择适当的版本。
4. 点 **OpenShift v4.14 Linux Client** 条目旁的 **Download Now** 来保存文件。
5. 解包存档：

```
$ tar xvf <file>
```

6. 将 **oc** 二进制文件放到 **PATH 中的目录中**。

要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

3.3.1.2. 在 Windows 上安装 OpenShift CLI

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

流程

1. 导航到红帽客户门户网站上的 OpenShift Container Platform 下载页面。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 Windows Client** 条目旁的 **Download Now** 来保存文件。
4. 使用 ZIP 程序解压存档。
5. 将 **oc** 二进制文件移到 **PATH 中的目录中**。
   要查看您的 **PATH**，请打开命令提示并执行以下命令：

```
C:\> path
```

验证

- 安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

3.3.1.3. 在 macOS 上安装 OpenShift CLI

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

流程

1. 导航到红帽客户门户网站上的 OpenShift Container Platform 下载页面。
2. 从 **版本** 下拉列表中选择适当的版本。
3. 点 **OpenShift v4.14 macOS Client** 条目旁的 **Download Now** 来保存文件。

注意

对于 macOS arm64，请选择 **OpenShift v4.14 macOS arm64 Client** 条目。

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**，请打开终端并执行以下命令：

```
$ echo $PATH
```

验证

- 使用 **oc** 命令验证安装：

```
$ oc <command>
```

3.4. 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

- 您已将镜像 registry 配置为在断开连接的环境中使用。
- 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 下载 **registry.redhat.io** pull secret。
2. 以 JSON 格式创建您的 pull secret 副本：

```
$ cat ./pull-secret | jq . > <path>/<pull_secret_file_in_json> 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"
    }
  }
}
```

3. 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。

4. 编辑 JSON 文件并添加描述 registry 的部分：

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

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:8443**

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": {
    "registry.example.com": {
      "auth": "BGVtbYk3ZHAtqXs=",
```

```
    "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"
  }
 }
}
```

3.5. 镜像 OPENSHIFT CONTAINER PLATFORM 镜像存储库

镜像要在集群安装或升级过程中使用的 OpenShift Container Platform 镜像仓库。

先决条件

- 您的镜像主机可访问互联网。
- 您已将镜像 registry 配置为在受限网络中使用，并可访问您配置的证书和凭证。
- 您已从 Red Hat OpenShift Cluster Manager 下载了 pull secret，并已修改为包含镜像存储库的身份验证。
- 如果您使用自签名证书，已在证书中指定 Subject Alternative Name。

流程

在镜像主机上完成以下步骤：

1. 查看 OpenShift Container Platform 下载页面，以确定您要安装的 OpenShift Container Platform 版本，并决定 Repository Tags 页中的相应标签（tag）。
2. 设置所需的环境变量：
   a. 导出发行版本信息：

   ```
   $ OCP_RELEASE=<release_version>
   ```

   对于 **<release_version>**，请指定与 OpenShift Container Platform 版本对应的标签，用于您的架构，如 **4.5.4**。

   b. 导出本地 registry 名称和主机端口：

   ```
   $ LOCAL_REGISTRY='<local_registry_host_name>:<local_registry_host_port>'
   ```

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。

c. 导出本地存储库名称：

```
$ LOCAL_REPOSITORY='<local_repository_name>'
```

对于 **<local_repository_name>**，请指定要在 registry 中创建的仓库名称，如 **ocp4/openshift4**。

d. 导出要进行镜像的存储库名称：

```
$ PRODUCT_REPO='openshift-release-dev'
```

对于生产环境版本，必须指定 **openshift-release-dev**。

e. 导出 registry pull secret 的路径：

```
$ LOCAL_SECRET_JSON='<path_to_pull_secret>'
```

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。

f. 导出发行版本镜像：

```
$ RELEASE_NAME="ocp-release"
```

对于生产环境版本，您必须指定 **ocp-release**。

g. 为您的集群导出构架类型：

```
$ ARCHITECTURE=<cluster_architecture> 1
```

1 指定集群的构架，如 **x86_64**, **aarch64**, **s390x**, 获 **ppc64le**。

h. 导出托管镜像的目录的路径：

```
$ REMOVABLE_MEDIA_PATH=<path> 1
```

1 指定完整路径，包括开始的前斜杠(/)字符。

3. 将版本镜像(mirror)到镜像 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. 查看要镜像的镜像和配置清单：

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

```
    --to-release-
image=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}-
${ARCHITECTURE} --dry-run
```

iii. 记录上一命令输出中的 **imageContentSources** 部分。您的镜像信息与您的镜像存储库相对应，您必须在安装过程中将 **imageContentSources** 部分添加到 **install-config.yaml** 文件中。

iv. 将镜像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}
```

v. 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。

重要

运行 **oc image mirror** 可能会导致以下错误： **error: unable to retrieve source image**。当镜像索引包括对镜像 registry 中不再存在的镜像的引用时，会发生此错误。镜像索引可能会保留旧的引用，以便为运行这些镜像的用户在升级图表中显示新的升级路径。作为临时解决方案，您可以使用 **--skip-missing** 选项绕过错误并继续下载镜像索引。如需更多信息，请参阅 Service Mesh Operator 镜像失败。

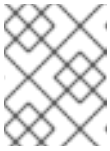
- 如果本地容器 registry 连接到镜像主机，请执行以下操作：

i. 使用以下命令直接将发行版镜像推送到本地 registry:

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

该命令将发行信息提取为摘要，其输出包括安装集群时所需的 **imageContentSources** 数据。

ii. 记录上一命令输出中的 **imageContentSources** 部分。您的镜像信息与您的镜像存储库相对应，您必须在安装过程中将 **imageContentSources** 部分添加到 **install-config.yaml** 文件中。



注意

镜像名称在镜像过程中被修补到 Quay.io， podman 镜像将在 bootstrap 虚拟机的 registry 中显示 Quay.io。

4. 要创建基于您镜像内容的安装程序，请提取内容并将其固定到发行版中：

   - 如果您的镜像主机无法访问互联网，请运行以下命令：

     ```
     $ oc adm release extract -a ${LOCAL_SECRET_JSON} --icsp-file=<file> --
     command=openshift-install
     "${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}-
     ${ARCHITECTURE}"
     ```

   - 如果本地容器 registry 连接到镜像主机，请运行以下命令：

     ```
     $ oc adm release extract -a ${LOCAL_SECRET_JSON} --command=openshift-install
     "${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}-
     ${ARCHITECTURE}"
     ```

重要

要确保将正确的镜像用于您选择的 OpenShift Container Platform 版本，您必须从镜像内容中提取安装程序。

您必须在有活跃互联网连接的机器上执行这个步骤。

5. 对于使用安装程序置备的基础架构的集群，运行以下命令：

   ```
   $ openshift-install
   ```

3.6. 在断开连接的环境中的 CLUSTER SAMPLES OPERATOR

在断开连接的环境中，在安装集群后执行额外的步骤来配置 Cluster Samples Operator。在准备过程中查阅以下信息：

3.6.1. 协助镜像的 Cluster Samples Operator

在安装过程中，OpenShift Container Platform 在 **openshift-cluster-samples-operator** 命名空间中创建一个名为 **imagestreamtag-to-image** 的配置映射。**imagestreamtag-to-image** 配置映射包含每个镜像流标签的条目（填充镜像）。

配置映射中 data 字段中每个条目的键格式为 **<image_stream_name>_<image_stream_tag_name>**。

在断开连接的 OpenShift Container Platform 安装过程中，Cluster Samples Operator 的状态被设置为 **Removed**。如果您将其改为 **Managed**，它会安装示例。

注意

在网络限制或断开连接的环境中使用示例可能需要通过网络访问服务。某些示例服务包括：Github、Maven Central、npm、RubyGems、PyPi 等。这可能需要执行额外的步骤，让集群 samples operator 对象能够访问它们所需的服务。

您可以使用此配置映射作为导入镜像流所需的镜像的引用。

- 在 Cluster Samples Operator 被设置为 **Removed** 时，您可以创建镜像的 registry，或决定您要使用哪些现有镜像 registry。
- 使用新的配置映射作为指南来镜像您要镜像的 registry 的示例。
- 将没有镜像的任何镜像流添加到 Cluster Samples Operator 配置对象的 **skippedImagestreams** 列表中。
- 将 Cluster Samples Operator 配置对象的 **samplesRegistry** 设置为已镜像的 registry。
- 然后，将 Cluster Samples Operator 设置为 **Managed** 来安装您已镜像的镜像流。

3.7. 镜像用于断开连接的集群的 OPERATOR 目录

您可以使用 **oc adm catalog mirror** 命令将红帽提供的目录或自定义目录的 Operator 内容镜像到容器镜像 registry 中。目标 registry 必须支持 Docker v2-2。对于受限网络中的集群，此 registry 可以是集群有网络访问权限的 registry，如在受限网络集群安装过程中创建的镜像 registry。

重要

- OpenShift 镜像 registry 不能用作目标 registry，因为它不支持没有标签的推送，在镜像过程中需要这个推送。
- 运行 **oc adm catalog mirror** 可能会导致以下错误： **error: unable to retrieve source image**。当镜像索引包括对镜像 registry 中不再存在的镜像的引用时，会发生此错误。镜像索引可能会保留旧的引用，以便为运行这些镜像的用户在升级图表中显示新的升级路径。作为临时解决方案，您可以使用 **--skip-missing** 选项绕过错误并继续下载镜像索引。如需更多信息，请参阅 Service Mesh Operator 镜像失败。

oc adm catalog mirror 命令还会自动将在镜像过程中指定的索引镜像（无论是红帽提供的索引镜像还是您自己的自定义构建索引镜像）镜像到目标 registry。然后，您可以使用镜像的索引镜像创建一个目录源，允许 Operator Lifecycle Manager（OLM）将镜像目录加载到 OpenShift Container Platform 集群。

其他资源

- 在受限网络中使用 Operator Lifecycle Manager

3.7.1. 先决条件

与断开连接的集群一起使用的 Operator 目录具有以下先决条件：

- 没有网络访问限制的工作站
- **podman** 1.9.3 或更高版本。
- 如果要过滤或 *prune* 一个现存的目录，且仅选择性地镜像部分 Operator，请参阅以下部分：
  - 安装 opm CLI
  - 更新或过滤基于文件的目录镜像

- 如果要镜像红帽提供的目录，请在具有无网络访问限制的工作站中运行以下命令，以便与 **registry.redhat.io** 进行身份验证：

```
$ podman login registry.redhat.io
```

- 访问支持 Docker v22 的镜像 registry。
- 在镜像 registry 上，决定使用哪个存储库或命名空间来存储已镜像的 Operator 内容。例如，您可以创建一个 **olm-mirror** 存储库。
- 如果您的镜像 registry 无法访问互联网，请将可移动介质连接到您的没有网络访问限制的工作站。
- 如果您正在使用私有 registry，包括 **registry.redhat.io**，请将 **REG_CREDS** 环境变量设置为 registry 凭证的文件路径，以便在后续步骤中使用。例如，对于 **podman** CLI:

```
$ REG_CREDS=${XDG_RUNTIME_DIR}/containers/auth.json
```

3.7.2. 提取和镜像目录内容

oc adm catalog mirror 命令提取索引镜像的内容，以生成镜像所需的清单。命令的默认行为会生成清单，然后会自动将索引镜像以及索引镜像本身中的所有镜像内容镜像（mirror）到您的镜像 registry。

另外，如果您的镜像 registry 位于完全断开连接的主机上，或者断开连接的或 *airgapped* 主机上，您可以首先将内容镜像到可移动介质，将介质移到断开连接的环境中，然后将内容从介质镜像到 registry。

3.7.2.1. 将目录内容镜像到同一网络上的 registry

如果您的镜像 registry 与您的没有网络访问限制的工作站位于同一个网络中，请在您的工作站上执行以下操作：

流程

1. 如果您的镜像 registry 需要身份验证，请运行以下命令登录到 registry:

```
$ podman login <mirror_registry>
```

2. 运行以下命令，将内容提取并镜像到镜像 registry:

```
$ oc adm catalog mirror \
    <index_image> \ 1
    <mirror_registry>:<port>[/<repository>] \ 2
    [-a ${REG_CREDS}] \ 3
    [--insecure] \ 4
    [--index-filter-by-os='<platform>/<arch>'] \ 5
    [--manifests-only] 6
```

1. 指定您要镜像的目录的索引镜像。
2. 指定要将 Operator 内容镜像到的目标 registry 的完全限定域名(FQDN)。镜像 registry **<repository>** 可以是 registry 上的任何现有存储库或命名空间，如先决条件中所述，如 **olm-mirror**。如果在镜像过程中找到现有的存储库，存储库名称将添加到生成的镜像名称中。如果您不希望镜像名称包含存储库名称，请省略此行中的 **<repository>** 值，例如

<mirror_registry>:<port>。

3 可选：如果需要，指定 registry 凭证文件的位置。**registry.redhat.io** 需要 **{REG_CREDS}**。

4 可选：如果您不想为目标 registry 配置信任，请添加 **--insecure** 标志。

5 可选：在有多个变体可用时，指定索引镜像的平台和架构。镜像被传递为 **'<platform>/<arch>[/<variant>]'**。这不适用于索引引用的镜像。有效值为 **linux/amd64**, **linux/ppc64le**, **linux/s390x**, **linux/arm64**.

6 可选：只生成镜像所需的清单，而不实际将镜像内容镜像到 registry。这个选项对检查哪些将被镜像（mirror）非常有用，如果您只需要一小部分软件包，可以对映射列表进行修改。然后，您可以使用带有 **oc image mirror** 命令的 **mapping.txt** 文件来在以后的步骤中镜像修改的镜像列表。此标志用于从目录中对内容进行高级选择性镜像。

输出示例

```
src image has index label for database path: /database/index.db
using database path mapping: /database/index.db:/tmp/153048078
wrote database to /tmp/153048078 1
...
wrote mirroring manifests to manifests-redhat-operator-index-1614211642 2
```

1 命令生成的临时 **index.db** 数据库的目录。

2 记录生成的 manifests 目录名称。该目录在后续过程中被引用。

注意

Red Hat Quay 不支持嵌套存储库。因此，运行 **oc adm catalog mirror** 命令会失败，并显示 **401** 未授权错误。作为临时解决方案，您可以在运行 **oc adm catalog mirror** 命令时使用 **--max-components=2** 选项来禁用嵌套存储库的创建。有关此临时解决方案的更多信息，请参阅 Unauthorized error thrown while using catalog mirror command with Quay registry。

其他资源

- Operator 的架构和操作系统支持

3.7.2.2. 将目录内容镜像到 airgapped registry

如果您的镜像 registry 位于完全断开连接的主机上，或 airgapped 主机上，请执行以下操作。

流程

1. 在您的工作站中运行以下命令，且没有网络访问权限将内容镜像到本地文件中：

```
$ oc adm catalog mirror \
    <index_image> \ 1
    file:///local/index \ 2
```

```
-a ${REG_CREDS} \ 3
--insecure \ 4
--index-filter-by-os='<platform>/<arch>' 5
```

1. 指定您要镜像的目录的索引镜像。
2. 指定要镜像到当前目录中的本地文件的内容。
3. 可选：如果需要，指定 registry 凭证文件的位置。
4. 可选：如果您不想为目标 registry 配置信任，请添加 **--insecure** 标志。
5. 可选：在有多个变体可用时，指定索引镜像的平台和架构。镜像被指定为 **'<platform>/<arch>[/<variant>]'**。这不适用于索引引用的镜像。有效值为 **linux/amd64**, **linux/ppc64le**, **linux/s390x**, **linux/arm64**, 和 **.***

输出示例

```
...
info: Mirroring completed in 5.93s (5.915MB/s)
wrote mirroring manifests to manifests-my-index-1614985528 1

To upload local images to a registry, run:

 oc adm catalog mirror file://local/index/myrepo/my-index:v1 REGISTRY/REPOSITORY 2
```

1. 记录生成的 manifests 目录名称。该目录在后续过程中被引用。
2. 记录根据您提供的索引镜像扩展的 **file://** 路径。这个路径在后续步骤中被引用。

此命令会在当前目录中创建 **v2/** 目录中。

2. 将 **v2/** 目录复制到可移动介质。
3. 物理删除该介质并将其附加到断开连接的环境中可访问镜像 registry 的主机。
4. 如果您的镜像 registry 需要身份验证，请在断开连接的环境中的主机上运行以下命令以登录到 registry：

```
$ podman login <mirror_registry>
```

5. 从包含 **v2/** 目录的父目录运行以下命令，将镜像从本地文件上传到镜像 registry：

```
$ oc adm catalog mirror \
  file://local/index/<repository>/<index_image>:<tag> \ 1
  <mirror_registry>:<port>[/<repository>] \ 2
  -a ${REG_CREDS} \ 3
  --insecure \ 4
  --index-filter-by-os='<platform>/<arch>' 5
```

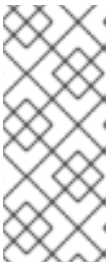
1. 指定上一命令输出中的 **file://** 路径。

2 指定要将 Operator 内容镜像到的目标 registry 的完全限定域名(FQDN)。镜像 registry **<repository>** 可以是 registry 上的任何现有存储库或命名空间，如先决条件中所述，如

3 可选：如果需要，指定 registry 凭证文件的位置。

4 可选：如果您不想为目标 registry 配置信任，请添加 **--insecure** 标志。

5 可选：在有多个变体可用时，指定索引镜像的平台和架构。镜像被指定为 **'<platform>/<arch>[/<variant>]'**。这不适用于索引引用的镜像。有效值为 **linux/amd64**, **linux/ppc64le**, **linux/s390x**, **linux/arm64**, 和 **.***



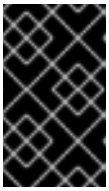
注意

Red Hat Quay 不支持嵌套存储库。因此，运行 **oc adm catalog mirror** 命令会失败，并显示 **401** 未授权错误。作为临时解决方案，您可以在运行 **oc adm catalog mirror** 命令时使用 **--max-components=2** 选项来禁用嵌套存储库的创建。有关此临时解决方案的更多信息，请参阅 Unauthorized error thrown while using catalog mirror command with Quay registry。

6. 再次运行 **oc adm catalog mirror** 命令。使用新镜像的索引镜像作为源，以及上一步中使用的同一镜像 registry 目标：

```
$ oc adm catalog mirror \
    <mirror_registry>:<port>/<index_image> \
    <mirror_registry>:<port>[/<repository>] \
    --manifests-only \ 1
    [-a ${REG_CREDS}] \
    [--insecure]
```

1 此步骤需要 **--manifests-only** 标志，以便该命令不会再次复制所有镜像的内容。



重要

这一步是必需的，因为上一步中生成的 **imageContentSourcePolicy.yaml** 文件中的镜像映射必须从本地路径更新为有效的镜像位置。如果不这样做，会在稍后的步骤中创建 **ImageContentSourcePolicy** 对象时会导致错误。

在镜像目录后，您可以继续执行集群的其余部分。在集群安装成功完成后，您必须指定此流程中的 manifests 目录来创建 **ImageContentSourcePolicy** 和 **CatalogSource** 对象。需要这些对象才能从 OperatorHub 安装 Operator。

其他资源

- Operator 的架构和操作系统支持

3.7.3. 生成的清单

将 Operator 目录内容镜像到镜像 registry 后，会在当前目录中生成清单目录。

如果您将内容镜像到同一网络上的 registry，则目录名称采用以下模式：

```
manifests-<index_image_name>-<random_number>
```

如果您在上一节中将内容镜像到断开连接的主机上的 registry，则目录名称采用以下模式：

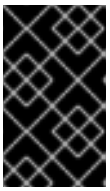
```
manifests-index/<repository>/<index_image_name>-<random_number>
```

注意

清单目录名称在后续过程中被引用。

manifests 目录包含以下文件，其中的一些文件可能需要进一步修改：

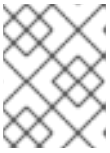
- **catalogSource.yaml** 文件是 **CatalogSource** 对象的基本定义，它预先填充索引镜像标签及其他相关元数据。此文件可原样使用，或进行相应修改来在集群中添加目录源。



重要

如果将内容镜像到本地文件，您必须修改 catalogSource **.yaml** 文件，从 **metadata.name** 字段中删除任何反斜杠(/)字符。否则，当您试图创建对象时，会失败并显示 "invalid resource name" 错误。

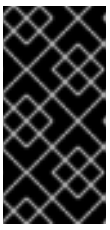
- 用来定义 **ImageContentSourcePolicy** 对象的 **imageContentSourcePolicy.yaml**，它可以将节点配置为在 Operator 清单中存储的镜像（image）引用和镜像 (mirror) 的 registry 间进行转换。



注意

如果您的集群使用 **ImageContentSourcePolicy** 对象来配置存储库镜像，则只能将全局 pull secret 用于镜像 registry。您不能在项目中添加 pull secret。

- **mapping.txt** 文件，在其中包含所有源镜像，并将它们映射到目标 registry。此文件与 **oc image mirror** 命令兼容，可用于进一步自定义镜像（mirror）配置。



重要

如果您在镜像过程中使用 **--manifests-only** 标志，并希望进一步调整要镜像的软件包子集，请参阅 OpenShift Container Platform 4.7 文档中的镜像软件包清单格式目录镜像流程中有关修改 **mapping.txt** 文件并使用 **oc image mirror** 命令的步骤。

3.7.4. 安装后的要求

在镜像目录后，您可以继续执行集群的其余部分。在集群安装成功完成后，您必须指定此流程中的 manifests 目录来创建 **ImageContentSourcePolicy** 和 **CatalogSource** 对象。这些对象需要填充和启用从 OperatorHub 安装 Operator。

其他资源

- 从镜像的 Operator 目录填充 OperatorHub
- 更新或过滤基于文件的目录镜像

3.8. 后续步骤

- 在您在受限网络中置备的基础架构上安装集群，如 VMware vSphere、裸机或 Amazon Web Services。

3.9. 其他资源

- 有关使用 must-gather 的更多信息，请参阅收集有关特定功能的数据。

第 4 章 使用 OC-MIRROR 插件为断开连接的安装镜像镜像

可以在没有直接的互联网连接的受限网络中运行集群，方法是使用在一个私有 registry 中的 mirror OpenShift Container Platform 容器镜像安装集群。集群运行时必须始终运行此 registry。如需更多信息，请参阅先决条件部分。

您可以使用 oc-mirror OpenShift CLI (**oc**)插件在完全或部分断开连接的环境中将镜像镜像到镜像 registry。您必须从具有互联网连接的系统运行 oc-mirror，以便从官方红帽 registry 中下载所需的镜像。

下列步骤概述了如何使用 oc-mirror 插件将镜像镜像到镜像 registry 的高级别工作流：

1. 创建镜像设置配置文件。
2. 使用以下方法之一将镜像设置为镜像 registry：
   - 将镜像直接设置为镜像 registry。
   - 镜像集合镜像到磁盘，将镜像设置为目标环境，然后将镜像上传到目标镜像 registry。
3. 配置集群以使用 oc-mirror 插件生成的资源。
4. 根据需要重复这些步骤以更新您的镜像 registry。

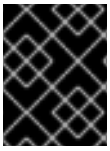
4.1. 关于 OC-MIRROR 插件

您可以使用 oc-mirror OpenShift CLI(**oc**)插件，使用单个工具将所有所需的 OpenShift Container Platform 内容和其他镜像(mirror)镜像到您的镜像 registry。它提供以下功能：

- 提供镜像 OpenShift Container Platform 发行版本、Operator、helm chart 和其他镜像的集中方法。
- 维护 OpenShift Container Platform 和 Operator 的更新路径。
- 使用声明的镜像设置配置文件来仅包含集群所需的 OpenShift Container Platform 发行版本、Operator 和镜像。
- 执行增量镜像，从而减少将来镜像集的大小。
- 从上一执行以来，从镜像集配置中排除的目标镜像 registry 中修剪镜像的镜像。
- （可选）为 OpenShift Update Service (OSUS) 使用生成支持工件。

使用 oc-mirror 插件时，您可以在镜像设置配置文件中指定要镜像的内容。在这个 YAML 文件中，您可以将配置微调为仅包含集群需要的 OpenShift Container Platform 发行版本和 Operator。这可减少您下载和传输所需的数据量。oc-mirror 插件也可以镜像任意 helm chart 和附加容器镜像，以帮助用户将其工作负载无缝同步到镜像 registry 中。

第一次运行 oc-mirror 插件时，它会使用所需内容填充您的镜像 registry，以执行断开连接的集群安装或更新。要让断开连接的集群继续接受更新，您必须更新镜像 registry。要更新您的镜像 registry，请使用与第一次运行相同的配置运行 oc-mirror 插件。oc-mirror 插件引用存储后端的元数据，并只下载上次运行该工具后所发布的元数据。这为 OpenShift Container Platform 和 Operator 提供了更新路径，并根据需要执行依赖项解析。

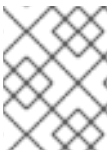


重要

当使用 oc-mirror CLI 插件填充镜像 registry 时，必须使用 oc-mirror 工具对镜像 registry 进行进一步的更新。

4.2. OC-MIRROR 兼容性和支持

oc-mirror 插件支持为 OpenShift Container Platform 版本 4.10 及之后的版本的镜像 OpenShift Container Platform 有效负载镜像和 Operator 目录。



注意

在 **aarch64**, **ppc64le**, 和 **s390x** 架构中，oc-mirror 插件只支持 OpenShift Container Platform 版本 4.14 及更新的版本。

使用 oc-mirror 插件的最新版本，无论您需要镜像的 OpenShift Container Platform 版本是什么。

重要

如果您为 OpenShift Container Platform 4.12 的 oc-mirror 插件使用了预览预览的 OCI 本地目录功能，则无法再使用 oc-mirror 插件的 OCI 本地目录功能在本地复制目录，并将其转换为 OCI 格式作为 mirror 到一个完全断开连接的集群中的第一步。

4.3. 关于镜像 REGISTRY

您可以镜像 OpenShift Container Platform 安装所需的镜像，以及支持 Docker v2-2 （如 Red Hat Quay）的容器镜像 registry 所需的镜像。如果您无法访问大型容器 registry，可以使用 *mirror registry for Red Hat OpenShift*，它是一个包括在 OpenShift Container Platform 订阅中的小型容器 registry。

无论您所选 registry 是什么，都会将互联网上红帽托管站点的内容镜像到隔离的镜像 registry 相同。镜像内容后，您要将每个集群配置为从镜像 registry 中检索此内容。

重要

OpenShift 镜像 registry 不能用作目标 registry，因为它不支持没有标签的推送，在镜像过程中需要这个推送。

如果选择不是 *mirror registry for Red Hat OpenShift* 的容器 registry，您必须可以被您置备的集群中的每个机器访问。如果 registry 无法访问，安装、更新或常规操作（如工作负载重新定位）可能会失败。因此，您必须以高度可用的方式运行镜像 registry，镜像 registry 至少必须与 OpenShift Container Platform 集群的生产环境可用性相匹配。

使用 OpenShift Container Platform 镜像填充镜像 registry 时，可以遵循以下两种情况。如果您的主机可以同时访问互联网和您的镜像 registry，而不能访问您的集群节点，您可以直接从该机器中镜像该内容。这个过程被称为 *连接的镜像(mirror)*。如果没有这样的主机，则必须将该镜像文件镜像到文件系统中，然后将该主机或者可移动介质放入受限环境中。这个过程被称为 *断开连接的镜像*。

对于已镜像的 registry，若要查看拉取镜像的来源，您必须查看 **Trying 以访问** CRI-O 日志中的日志条目。查看镜像拉取源的其他方法（如在节点上使用 **crictl images** 命令）显示非镜像镜像名称，即使镜像是从镜像位置拉取的。

注意

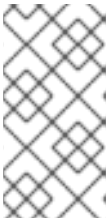
红帽没有针对 OpenShift Container Platform 测试第三方 registry。

其他资源

- 有关查看 CRI-O 日志以查看镜像源的详情，请参阅查看镜像拉取源。

4.4. 先决条件

- 您必须在托管 OpenShift Container Platform 集群的位置（如 Red Hat Quay）中有一个支持 Docker v2-2 的容器镜像 registry。



注意

如果使用 Red Hat Quay，则必须在 oc-mirror 插件中使用 3.6 或更高版本。如果您有 Red Hat Quay 权利，请参阅有关部署 Red Hat Quay for 概念验证 的文档，或使用 Red Hat Quay Operator。如果您需要额外的帮助来选择并安装 registry，请联络您的销售代表或红帽支持。

如果您还没有容器镜像 registry，OpenShift Container Platform 可以为订阅者提供一个 mirror registry for Red Hat OpenShift。您的订阅中包含 *mirror registry for Red Hat OpenShift*，它是一个小型容器 registry，可用于在断开连接的安装中镜像 OpenShift Container Platform 所需的容器镜像。

4.5. 准备您的镜像主机

在使用 oc-mirror 插件镜像(mirror)前，您必须安装插件并创建容器镜像 registry 凭据文件，以允许从红帽镜像到您的镜像。

4.5.1. 安装 oc-mirror OpenShift CLI 插件

要使用 oc-mirror OpenShift CLI 插件来镜像 registry 镜像，您必须安装插件。如果您在一个完全断开连接的环境中镜像镜像集，请确保在具有互联网访问的主机上的 oc-mirror 插件以及可访问镜像 registry 的断开连接的环境中安装 oc-mirror 插件。

先决条件

- 已安装 OpenShift CLI（**oc**）。
- 您已在使用 oc-mirror 的操作系统中，将 **umask** 参数设置为 **0022**。
- 您已为您要使用的 RHEL 版本安装了正确的二进制文件。

流程

1. 下载 oc-mirror CLI 插件。
   a. 导航到 OpenShift Cluster Manager 的 Downloads 页面。
   b. 在 **OpenShift disconnected 安装工具**部分下，点 **Download** for **OpenShift Client(oc)mirror 插件** 并保存该文件。
2. 解压归档：

```
$ tar xvzf oc-mirror.tar.gz
```

3. 如有必要，将插件文件更新为可执行。

```
$ chmod +x oc-mirror
```

注意

不要重命名 **oc-mirror** 文件。

4. 通过将文件放在 **PATH** 中，例如 **/usr/local/bin**，安装 oc-mirror CLI 插件：

```
$ sudo mv oc-mirror /usr/local/bin/.
```

验证

- 运行 **oc mirror help** 来验证插件是否已成功安装：

```
$ oc mirror help
```

其他资源

- 安装和使用 CLI 插件

4.5.2. 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

- 您已将镜像 registry 配置为在断开连接的环境中使用。

- 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 下载 **registry.redhat.io** pull secret。
2. 以 JSON 格式创建您的 pull secret 副本：

```
$ cat ./pull-secret | jq . > <path>/<pull_secret_file_in_json> 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"
    }
  }
}
```

3. 将文件保存为 **~/.docker/config.json** 或 **$XDG_RUNTIME_DIR/containers/auth.json**：
   a. 如果 **.docker** 或 **$XDG_RUNTIME_DIR/containers** 目录不存在，请输入以下命令来创建：

```
$ mkdir -p <directory_name>
```

   其中 **<directory_name>** 是 **~/.docker** 或 **$XDG_RUNTIME_DIR/containers**。

   b. 输入以下命令将 pull secret 复制到适当的目录中：

```
$ cp <path>/<pull_secret_file_in_json> <directory_name>/<auth_file>
```

   其中 **<directory_name>** 是 **~/.docker** 或 **$XDG_RUNTIME_DIR/containers**，**<auth_file>** 是 **config.json** 或 **auth.json**。

4. 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。

5. 编辑 JSON 文件并添加描述 registry 的部分：

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

(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:8443**

(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": {
    "registry.example.com": {
      "auth": "BGVtbYk3ZHAtqXs=",
      "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"
    }
  }
}
```

4.6. 创建镜像设置配置

在使用 oc-mirror 插件镜像集之前，必须先创建镜像设置配置文件。此镜像设置配置文件定义哪些 OpenShift Container Platform 发行版本、Operator 和其他镜像要镜像，以及 oc-mirror 插件的其他配置设置。

您必须在镜像设置配置文件中指定存储后端。此存储后端可以是本地目录或支持 Docker v2-2 的 registry。oc-mirror 插件在创建镜像的过程中将元数据存储在这个存储后端中。

重要

不要删除或修改 oc-mirror 插件生成的元数据。每次针对同一镜像 registry 运行 oc-mirror 插件时，都必须使用相同的存储后端。

先决条件

- 您已创建了容器镜像 registry 凭证文件。具体步骤，请参阅*配置允许镜像镜像的凭证*。

流程

1. 使用 **oc mirror init** 命令为镜像设置配置创建模板，并将其保存到名为 **imageset-config.yaml** 的文件中：

```
$ oc mirror init --registry example.com/mirror/oc-mirror-metadata > imageset-config.yaml (1)
```

(1) 将 **example.com/mirror/oc-mirror-metadata** 替换为存储后端的 registry 的位置。

2. 编辑该文件并根据需要调整设置：

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
archiveSize: 4                                          (1)
storageConfig:                                          (2)
  registry:
    imageURL: example.com/mirror/oc-mirror-metadata     (3)
    skipTLS: false
mirror:
  platform:
    channels:
    - name: stable-4.14                                 (4)
      type: ocp
    graph: true                                         (5)
  operators:
  - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.14   (6)
    packages:
    - name: serverless-operator                         (7)
      channels:
      - name: stable                                    (8)
  additionalImages:
  - name: registry.redhat.io/ubi9/ubi:latest            (9)
  helm: {}
```

(1) 添加 **archiveSize** 以设置镜像集合中的每个文件的最大大小（以 GiB 为单位）。

2 设置后端位置，以将镜像设置元数据保存到。此位置可以是 registry 或本地目录。必须指定 **storageConfig** 值。

3 设置存储后端的 registry URL。

4 将频道设置为从中检索 OpenShift Container Platform 镜像。

5 添加 **graph: true** 以构建并推送 graph-data 镜像推送到镜像 registry。创建 OpenShift Update Service (OSUS) 需要 graph-data 镜像。**graph: true** 字段还会生成 **UpdateService** 自定义资源清单。**oc** 命令行界面 (CLI) 可以使用 **UpdateService** 自定义资源清单来创建 OSUS。如需更多信息，请参阅*关于 OpenShift Update Service*。

6 将 Operator 目录设置为从中检索 OpenShift Container Platform 镜像。

7 仅指定要包含在镜像集中的某些 Operator 软件包。删除此字段以检索目录中的所有软件包。

8 仅指定要包含在镜像集中的 Operator 软件包的某些频道。即使您没有使用该频道中的捆绑包，还必须始终包含 Operator 软件包的默认频道。您可以运行以下命令来找到默认频道：**oc mirror list operators --catalog=<catalog_name> --package=<package_name>**。

9 指定要在镜像集中包含的任何其他镜像。

注意

graph: true 字段还会镜像 **ubi-micro** 镜像，以及其他镜像的镜像。

当升级 OpenShift Container Platform 延长更新支持 (EUS) 版本时，在当前和目标版本之间可能需要一个中间版本。例如，如果当前版本是 **4.14**，目标版本为 **4.16**，您可能需要在使用 oc-mirror 插件 v1 时在 **ImageSetConfiguration** 中包含版本，如 **4.15.8**。

oc-mirror 插件 v1 可能并不总是自动检测，因此请检查 Cincinnati 图形网页以确认任何所需的中间版本并手动添加到您的配置中。

如需完整的参数列表，请参阅 *Image set configuration parameters*；对于不同的镜像用例，请参阅 *Image set configuration examples*。

3. 保存更新的文件。
在镜像内容时，**oc mirror** 命令需要此镜像设置配置文件。

其他资源

- 镜像设置配置参数
- 镜像设置配置示例
- 在断开连接的环境中使用 OpenShift Update Service

4.7. 将镜像集镜像(MIRROR)到镜像 REGISTRY

您可以使用 oc-mirror CLI 插件在 部分断开连接的环境中或完全断开连接的环境中将镜像镜像到镜像 registry。

这些步骤假定您已设置了镜像 registry。

4.7.1. 在部分断开连接的环境中镜像设置的镜像

在部分断开连接的环境中，您可以直接镜像到目标镜像 registry 的镜像。

4.7.1.1. 镜像(mirror)到镜像(mirror)的镜像

您可以使用 oc-mirror 插件将镜像直接设置为在镜像设置过程中可访问的目标镜像 registry。

您必须在镜像设置配置文件中指定存储后端。这个存储后端可以是本地目录或 Docker v2 registry。oc-mirror 插件在创建镜像的过程中将元数据存储在这个存储后端中。

重要

不要删除或修改 oc-mirror 插件生成的元数据。每次针对同一镜像 registry 运行 oc-mirror 插件时，都必须使用相同的存储后端。

先决条件

- 您可以访问互联网来获取所需的容器镜像。
- 已安装 OpenShift CLI(**oc**)。
- 已安装 **oc-mirror** CLI 插件。
- 您已创建了镜像设置配置文件。

流程

- 运行 **oc mirror** 命令将指定镜像集配置中的镜像镜像到指定的 registry:

```
$ oc mirror --config=./imageset-config.yaml \ 1
  docker://registry.example:5000 2
```

1. 传递创建的镜像设置配置文件。此流程假设它名为 **imageset-config.yaml**。
2. 指定要镜像设置文件的 registry。registry 必须以 **docker://** 开头。如果为镜像 registry 指定顶层命名空间，则必须在后续执行时使用此命名空间。

验证

1. 进入生成的 **oc-mirror-workspace/** 目录。
2. 导航到结果目录，例如，**results-1639608409/**。
3. 验证 **ImageContentSourcePolicy** 和 **CatalogSource** 资源是否存在 YAML 文件。

注意

ImageContentSourcePolicy YAML 文件的 **repositoryDigestMirrors** 部分在安装过程中用于 **install-config.yaml** 文件。

+

后续步骤

- 配置集群以使用 oc-mirror 生成的资源。

故障排除

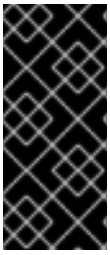
- 无法检索源镜像。

4.7.2. 镜像在完全断开连接的环境中设置的镜像

要镜像在完全断开连接的环境中设置的镜像，您必须首先将镜像集镜像到磁盘， 然后将磁盘上的镜像集文件镜像到一个镜像。

4.7.2.1. 从镜像镜像到磁盘

您可以使用 oc-mirror 插件生成镜像集，并将内容保存到磁盘。然后，生成的镜像集可以转移到断开连接的环境中，并镜像到目标 registry。



重要

根据镜像设置配置文件中指定的配置，使用 oc-mirror 的镜像可能会将几百 GB 数据下载到磁盘。

您填充镜像 registry 时初始镜像集下载通常是最大镜像。因为您只下载自上次运行命令以来更改的镜像，所以再次运行 oc-mirror 插件时，所生成的镜像集通常比较小。

您必须在镜像设置配置文件中指定存储后端。这个存储后端可以是本地目录或 docker v2 registry。oc-mirror 插件在创建镜像的过程中将元数据存储在这个存储后端中。

重要

不要删除或修改 oc-mirror 插件生成的元数据。每次针对同一镜像 registry 运行 oc-mirror 插件时，都必须使用相同的存储后端。

先决条件

- 您可以访问互联网来获取所需的容器镜像。
- 已安装 OpenShift CLI(**oc**)。
- 已安装 **oc-mirror** CLI 插件。
- 您已创建了镜像设置配置文件。

流程

- 运行 **oc mirror** 命令将指定镜像集配置镜像到磁盘：

```
$ oc mirror --config=./imageset-config.yaml \ 1
  file://<path_to_output_directory>  2
```

1. 传递创建的镜像设置配置文件。此流程假设它名为 **imageset-config.yaml**。
2. 指定要输出镜像集文件的目标目录。目标目录路径必须以 **file://** 开头。

验证

1. 进入您的输出目录：

```
$ cd <path_to_output_directory>
```

2. 验证是否创建了镜像设置 **.tar** 文件：

```
$ ls
```

输出示例

```
mirror_seq1_000000.tar
```

后续步骤

- 将镜像集 .tar 文件移动到断开连接的环境中。

故障排除

- 无法检索源镜像。

4.7.2.2. 从磁盘镜像到镜像

您可以使用 oc-mirror 插件将生成的镜像集的内容镜像到目标镜像 registry。

先决条件

- 您已在断开连接的环境中安装了 OpenShift CLI(**oc**)。
- 您已在断开连接的环境中安装了 **oc-mirror** CLI 插件。
- 已使用 **oc mirror** 命令生成镜像集文件。
- 您已将镜像集文件传送到断开连接的环境中。

流程

- 运行 **oc mirror** 命令，以处理磁盘上镜像集文件，并将内容镜像到目标镜像 registry：

```
$ oc mirror --from=./mirror_seq1_000000.tar \ 1
  docker://registry.example:5000              2
```

1. 传递镜像集 .tar 文件以进行镜像，在本例中名为 **mirror_seq1_000000.tar**。如果在镜像设置配置文件中指定了 **archiveSize** 值，则镜像集可能会划分为多个 .tar 文件。在这种情况下，您可以传递一个包含镜像设置 .tar 文件的目录。

2 指定要镜像设置文件的 registry。registry 必须以 **docker://** 开头。如果为镜像 registry 指定顶层命名空间，则必须在后续执行时使用此命名空间。

此命令使用镜像集更新镜像 registry，并生成 **ImageContentSourcePolicy** 和 **CatalogSource** 资源。

验证

1. 进入生成的 **oc-mirror-workspace/** 目录。
2. 导航到结果目录，例如，**results-1639608409/**。
3. 验证 **ImageContentSourcePolicy** 和 **CatalogSource** 资源是否存在 YAML 文件。

后续步骤

- 配置集群以使用 oc-mirror 生成的资源。

故障排除

- 无法检索源镜像。

4.8. 配置集群以使用 OC-MIRROR 生成的资源

将镜像设置为镜像 registry 后，您必须将生成的 **ImageContentSourcePolicy**、**CatalogSource** 和发行版本镜像签名资源应用到集群。

ImageContentSourcePolicy 资源将镜像 registry 与源 registry 关联，并将在线 registry 中的镜像拉取请求重定向到镜像 registry。Operator Lifecycle Manager(OLM)使用 **CatalogSource** 资源检索有关镜像 registry 中可用 Operator 的信息。发行镜像签名用于验证镜像的发行镜像。

先决条件

- 您已将镜像设置为断开连接的环境中的 registry 镜像。
- 您可以使用具有 **cluster-admin** 角色的用户访问集群。

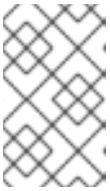
流程

1. 以具有 **cluster-admin** 角色的用户身份登录 OpenShift CLI。
2. 运行以下命令，将结果目录中的 YAML 文件应用到集群：

```
$ oc apply -f ./oc-mirror-workspace/results-1639608409/
```

3. 如果镜像(mirror)镜像，请运行以下命令将发行版本镜像签名应用到集群：

```
$ oc apply -f ./oc-mirror-workspace/results-1639608409/release-signatures/
```



注意

如果要镜像 Operator 而不是集群，则不需要运行 **$ oc apply -f ./oc-mirror-workspace/results-1639608409/release-signatures/**。运行该命令将返回错误，因为没有要应用的发行版本镜像签名。

验证

1. 运行以下命令验证 **ImageContentSourcePolicy** 资源是否已成功安装：

```
$ oc get imagecontentsourcepolicy
```

2. 运行以下命令验证 **CatalogSource** 资源是否已成功安装：

```
$ oc get catalogsource -n openshift-marketplace
```

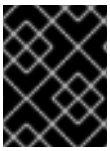
4.9. 保持镜像 REGISTRY 内容更新

在目标镜像 registry 填充了初始镜像集后，请务必定期更新它，使其具有最新的内容。您可以选择设置 cron 任务（如果可能），以便定期更新镜像 registry。

请确定您更新镜像设置的配置，以根据需要添加或删除 OpenShift Container Platform 和 Operator 版本。任何移除的镜像都会从镜像 registry 中修剪。

4.9.1. 关于更新您的镜像 registry 内容

当您再次运行 oc-mirror 插件时，它会生成一个镜像集，该集合仅包含与之前执行后的全新和更新镜像。因为它只拉取自以前的镜像集的差异，所以所生成的镜像集通常比初始镜像集更小且更快。



重要

生成的镜像集是有序的，且必须推送到目标镜像 registry。您可以从生成的镜像设置归档文件的文件名中获取序列号。

4.9.1.1. 添加新和更新的镜像

根据镜像设置配置中的设置，oc-mirror 的将来执行可能会镜像新的和更新镜像。查看镜像设置配置中的设置，以确保根据需要检索新版本。例如，如果要限制特定版本，可以将 Operator 的最小和最大版本设置为 mirror。另外，您可以将最小版本设置为镜像的起点，但保持对版本范围保持打开状态，以便在以后的 oc-mirror 上执行新的 Operator 版本。省略任何最小或最大版本会为您提供频道中的 Operator 的完整版本历史记录。省略明确指定的频道会为您提供指定 Operator 的所有频道的所有发行版本。省略任何命名 Operator 都会为您提供所有 Operator 的整个目录及其所有版本。

所有这些约束和条件均针对红帽每次调用 oc-mirror 时根据公开发布的内容进行评估。这样，它会自动获取新版本和全新的 Operator。约束只能通过列出所需的 Operator 集合来指定，它不会自动将其他新发布的 Operator 添加到镜像集中。您还可以指定特定的发行版本频道，将镜像限制为只为此频道，而不是任何已添加的新频道。这对于 Operator 产品（如 Red Hat Quay）来说，在其次发行版本中使用不同的发行频道。最后，您可以指定一个特定 Operator 的最大版本，这会导致工具只镜像指定的版本范围，这样您不会自动获得任何更新版本镜像 (mirror) 版本。在所有用例中，您必须更新镜像设置配置文件以扩大 Operator 镜像范围，以获取其他 Operator、新频道和较新版本的 Operator，以便在目标 registry 中提供。

建议将诸如频道规格或版本范围等约束与所选 Operator 的发行策略保持一致。例如，当 Operator 使用

stable 频道时，您应该将镜像限制到该频道，并有可能最小的版本来查找下载卷之间的正确平衡并定期获取稳定更新。如果 Operator 选择了发行版本频道方案，如 **stable-3.7**，您应该镜像该频道中的所有发行版本。这可让您继续使用 Operator 的补丁版本，如 **3.7.1**。您还可以定期调整镜像设置配置，为新产品版本添加频道，如 **stable-3.8**。

4.9.1.2. 修剪镜像

如果镜像不再包含在生成和镜像的最新镜像集中，则会自动从目标镜像 registry 中修剪镜像。这可让您轻松管理和清理不需要的内容并回收存储资源。

如果不再需要 OpenShift Container Platform 发行版本或 Operator 版本，您可以修改镜像设置配置以排除它们，并在镜像 (mirror) 后从镜像 registry 中修剪它们。这可以通过调整镜像设置配置文件中的每个 Operator 的最小或最大版本范围设置，或者从目录中镜像 (mirror) 的 Operator 中移除。您还可以从配置文件中删除整个 Operator 目录或整个 OpenShift Container Platform 版本。

重要

如果没有要镜像的新的或更新的镜像，则不会从目标镜像 registry 中修剪排除的镜像。另外，如果 Operator publisher 从频道中删除 Operator 版本，则从目标镜像 registry 中删除了删除的版本。

要禁用从目标镜像 registry 中自动修剪镜像，请将 **--skip-pruning** 标志传递给 **oc mirror** 命令。

4.9.2. 更新您的镜像 registry 内容

将初始镜像设置为镜像 registry 后，您可以使用 oc-mirror 插件来保持断开连接的集群更新。

根据您的镜像设置配置，oc-mirror 会自动检测 OpenShift Container Platform 的较新版本，以及在完成 inital mirror 后发布的所选 Operator。建议您定期运行 oc-mirror，例如在每日 cron 作业中运行 oc-mirror，以及时接收产品和安全更新。

先决条件

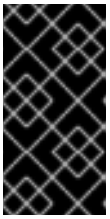
- 您已使用 oc-mirror 插件将初始镜像设置为您的镜像 registry。
- 您可以访问用于进行 oc-mirror 插件的初始执行的存储后端。

注意

您必须使用与 oc-mirror 的初始执行相同镜像 registry 的存储后端。不要删除或修改 oc-mirror 插件生成的元数据镜像。

流程

1. 如有必要，更新您的镜像设置配置文件，以获取新的 OpenShift Container Platform 和 Operator 版本。对于示例镜像使用情况，请参阅 *Image set configuration examples* 。
2. 按照您用来将初始镜像设置为镜像 registry 的步骤操作。具体步骤，请参阅*在部分断开连接的环境中镜像镜像集*，或*在完全断开连接的环境中镜像镜像集*。



重要

- 您必须提供相同的存储后端，以便只创建并镜像不同的镜像集。
- 如果您在初始镜像集创建过程中为镜像 registry 指定顶层命名空间，则每次针对同一镜像 registry 运行 oc-mirror 插件时都必须使用此命名空间。

3. 配置集群以使用 oc-mirror 生成的资源。

其他资源

- 镜像设置配置示例
- 在部分断开连接的环境中镜像设置的镜像
- 镜像在完全断开连接的环境中设置的镜像
- 配置集群以使用 oc-mirror 生成的资源

4.10. 执行空运行

您可以使用 oc-mirror 来执行空运行，而无需实际镜像(mirror)。这可让您查看要镜像的镜像列表，以及从镜像 registry 修剪的所有镜像。它还允许您在早期版本中捕获与镜像集配置相关的任何错误，或使用生成的镜像列表以及其他工具来执行镜像操作。

先决条件

- 您可以访问互联网来获取所需的容器镜像。
- 已安装 OpenShift CLI(**oc**)。
- 已安装 **oc-mirror** CLI 插件。
- 您已创建了镜像设置配置文件。

流程

1. 使用 **--dry-run** 标志运行 **oc mirror** 命令来执行空运行：

```
$ oc mirror --config=./imageset-config.yaml \ 1
  docker://registry.example:5000           \ 2
  --dry-run                                  3
```

1. 传递创建的镜像设置配置文件。此流程假设它名为 **imageset-config.yaml**。
2. 指定镜像 registry。在使用 **--dry-run** 标志时，不会镜像这个 registry。
3. 使用 **--dry-run** 标志来生成空运行工件，而不是实际的镜像设置文件。

输出示例

```
Checking push permissions for registry.example:5000
Creating directory: oc-mirror-workspace/src/publish
```

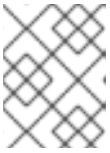
```
Creating directory: oc-mirror-workspace/src/v2
Creating directory: oc-mirror-workspace/src/charts
Creating directory: oc-mirror-workspace/src/release-signatures
No metadata detected, creating new workspace
wrote mirroring manifests to oc-mirror-workspace/operators.1658342351/manifests-redhat-operator-index

...

info: Planning completed in 31.48s
info: Dry run complete
Writing image mapping to oc-mirror-workspace/mapping.txt
```

2. 进入生成的工作区目录：

```
$ cd oc-mirror-workspace/
```

3. 查看生成的 **mapping.txt** 文件。
此文件包含将要镜像的所有镜像的列表。

4. 查看生成的 **prune-plan.json** 文件。
此文件包含在发布镜像集时从镜像 registry 中修剪的所有镜像的列表。



注意

只有在 oc-mirror 命令指向您的镜像 registry 且需要修剪的镜像时，才会生成 **prune-plan.json** 文件。

4.11. 包括本地 OCI OPERATOR 目录

虽然将 OpenShift Container Platform 发行版本、Operator 目录和其他额外的镜像从 registry mirror 到一个部分断开连接的集群中，但您还可以在一个本地磁盘中的基于文件的目录中包含 Operator 目录镜像。本地目录必须采用开放容器项目 (OCI) 格式。

本地目录及其内容会根据镜像设置配置文件中的过滤信息，mirror 到您的目标 mirror registry。

重要

在镜像本地 OCI 目录时，所有您要 mirror 的 OpenShift Container Platform 发行版本或其他需要和本地 OCI 格式目录一起 mirror 的镜像都必须从 registry 中拉取。

您无法在磁盘中一起 mirror OCI 目录和一个 oc-mirror 镜像集文件镜像。

使用 OCI 功能的一个用例是，您有一个 CI/CD 系统将 OCI 目录构建到磁盘上的位置，并需要将 OCI 目录以及 OpenShift Container Platform 发行版本一起 mirror 到您的 mirror 镜像 registry。

注意

如果您为 OpenShift Container Platform 4.12 的 oc-mirror 插件使用了预览预览的 OCI 本地目录功能，则无法再使用 oc-mirror 插件的 OCI 本地目录功能在本地复制目录，并将其转换为 OCI 格式作为 mirror 到一个完全断开连接的集群中的第一步。

先决条件

- 您可以访问互联网来获取所需的容器镜像。
- 已安装 OpenShift CLI(**oc**)。
- 已安装 **oc-mirror** CLI 插件。

流程

1. 创建镜像设置配置文件，并根据需要调整设置。
   以下示例镜像设置配置在磁盘上 mirror 一个 OCI 目录，以及来自 **registry.redhat.io** 的 OpenShift Container Platform 发行版本和 UBI 镜像。

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  local:
    path: /home/user/metadata                                    (1)
mirror:
  platform:
    channels:
    - name: stable-4.14                                          (2)
      type: ocp
    graph: false
  operators:
  - catalog: oci:///home/user/oc-mirror/my-oci-catalog           (3)
    targetCatalog: my-namespace/redhat-operator-index            (4)
    packages:
    - name: aws-load-balancer-operator
  - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.14   (5)
    packages:
    - name: rhacs-operator
  additionalImages:
  - name: registry.redhat.io/ubi9/ubi:latest                     (6)
```

(1) 设置后端位置，以将镜像设置元数据保存到。此位置可以是 registry 或本地目录。必须指定 **storageConfig** 值。

(2) （可选）包括一个 OpenShift Container Platform 发行版本，以便从 **registry.redhat.io** mirror。

(3) 指定磁盘上 OCI 目录位置的绝对路径。使用 OCI 功能时，路径必须以 **oci:**// 开头。

(4) 另外，还可指定替代命名空间和名称来镜像目录。

(5) （可选）指定要从 registry 中拉取的额外 Operator 目录。

(6) （可选）指定要从 registry 中拉取的额外镜像。

2. 运行 **oc mirror** 命令将 OCI 目录 mirror 到目标 mirror registry：

```
$ oc mirror --config=./imageset-config.yaml \   (1)
  docker://registry.example:5000                 (2)
```

1. 传递镜像设置配置文件。此流程假设它名为 **imageset-config.yaml**。
2. 指定要将内容 mirror 到的 registry。registry 必须以 **docker://** 开头。如果为镜像 registry 指定顶层命名空间，则必须在后续执行时使用此命名空间。

另外，您可以指定其他标记来调整 OCI 功能的行为：

--oci-insecure-signature-policy

不要将签名推送到目标 mirror registry。

--oci-registries-config

指定 TOML 格式的 **registry.conf** 文件的路径。您可以使用它来从不同的 registry 中镜像，如用于测试的预生产位置，而无需更改镜像设置配置文件。这个标志只会影响本地 OCI 目录，而不会影响任何其他被镜像的内容。

registry.conf 文件示例

```
[[registry]]
 location = "registry.redhat.io:5000"
 insecure = false
 blocked = false
 mirror-by-digest-only = true
 prefix = ""
 [[registry.mirror]]
    location = "preprod-registry.example.com"
    insecure = false
```

后续步骤

- 配置集群以使用 oc-mirror 生成的资源。

其他资源

- 配置集群以使用 oc-mirror 生成的资源

4.12. 镜像设置配置参数

oc-mirror 插件需要一个镜像设置配置文件，该文件定义哪些镜像要镜像(mirror)。下表列出了 **ImageSetConfiguration** 资源的可用参数。

表 4.1. ImageSetConfiguration 参数

参数	描述	值
apiVersion	**ImageSetConfiguration** 内容的 API 版本。	字符串.例如：**mirror.openshift.io/v1alpha2**。
archiveSize	镜像集中的每个存档文件的最大大小（以 GiB 为单位）。	整数.例如： **4**

参数	描述	值
mirror	镜像集的配置。	对象

mirror.additionalImages	镜像集的额外镜像配置。	对象数组。例如：
		`additionalImages:`
		`  - name: registry.redhat.io/ubi8/ubi:latest`

mirror.additionalImages.name	要 mirror 的镜像的标签或摘要。	字符串.例如：**registry.redhat.io/ubi8/ubi:latest**
mirror.blockedImages	用于阻止镜像(tag)或摘要(SHA)的镜像列表。	字符串数组。例如：**docker.io/library/alpine**
mirror.helm	镜像集的 helm 配置。请注意，oc-mirror 插件只支持 helm chart，在呈现时不需要用户输入。	对象

mirror.helm.local	要镜像的本地 helm chart。	对象数组。例如：
		`local:`
		`  - name: podinfo`
		`  path: /test/podinfo-5.0.0.tar.gz`

mirror.helm.local.name	要镜像的本地 helm chart 的名称。	字符串.例如：**podinfo**。
mirror.helm.local.path	到镜像的本地 helm chart 的路径。	字符串.例如：**/test/podinfo-5.0.0.tar.gz**。

参数	描述	值
mirror.helm.repositories	从其中镜像的的远程 helm 软件仓库。	对象数组。例如： repositories: - name: podinfo url: https://example.github.io/podinfo charts: - name: podinfo version: 5.0.0
mirror.helm.repositories.name	从其中镜像(mirror)的 helm 存储库的名称。	字符串.例如：**podinfo**。
mirror.helm.repositories.url	从其中镜像(mirror)的 helm 存储库的 URL。	字符串.例如：**https://example.github.io/podinfo**。
mirror.helm.repositories.charts	要镜像的远程 helm chart。	对象数组。
mirror.helm.repositories.charts.name	要镜像的 helm chart 的名称。	字符串.例如：**podinfo**。
mirror.helm.repositories.charts.version	要镜像命名 helm chart 的版本。	字符串.例如：**5.0.0**。
mirror.operators	镜像集的 Operator 配置。	对象数组。例如： operators: - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.14 packages: - name: elasticsearch-operator minVersion: '2.4.0'

参数	描述	值
mirror.operators.catalog	包括在镜像集中的 Operator 目录。	字符串.例如：**registry.redhat.io/redhat/redhat-operator-index:v4.14**。
mirror.operators.full	为 **true** 时，下载完整的目录、Operator 软件包或 Operator 频道。	布尔值.默认值为 **false**。
mirror.operators.packages	Operator 软件包配置。	对象数组。例如： operators: - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.14 packages: - name: elasticsearch-operator minVersion: '5.2.3-31'
mirror.operators.packages.name	镜像集中要包含的 Operator 软件包名称	字符串.例如：**elasticsearch-operator**。
mirror.operators.packages.channels	Operator 软件包频道配置。	对象
mirror.operators.packages.channels.name	Operator 频道名称（软件包中唯一）要包括在镜像集中。	字符串.例如： **fast** 或 **stable-v4.14**。
mirror.operators.packages.channels.maxVersion	Operator 镜像的最高版本，在其中存在所有频道。详情请查看以下备注。	字符串.例如：**5.2.3-31**
mirror.operators.packages.channels.minBundle	要包含的最小捆绑包的名称，以及频道头更新图中的所有捆绑包。仅在命名捆绑包没有语义版本元数据时设置此字段。	字符串.例如：**bundleName**
mirror.operators.packages.channels.minVersion	Operator 的最低版本，用于镜像存在的所有频道。详情请查看以下备注。	字符串.例如：**5.2.3-31**
mirror.operators.packages.maxVersion	Operator 最高版本，可跨所有存在的频道进行镜像。详情请查看以下备注。	字符串.例如：**5.2.3-31**。

参数	描述	值
mirror.operators.packages.minVersion	Operator 的最低版本，用于镜像存在的所有频道。详情请查看以下备注。	字符串.例如：**5.2.3-31**。
mirror.operators.skipDependencies	如果为 **true**，则不会包含捆绑包的依赖项。	布尔值.默认值为 **false**。
mirror.operators.targetCatalog	要镜像引用的目录的替代名称和可选命名空间层次结构。	字符串.例如：**my-namespace/my-operator-catalog**
mirror.operators.targetName	将引用的目录镜像为。 **targetName** 参数已弃用。改为使用 **targetCatalog** 参数。	字符串.例如： **my-operator-catalog**
mirror.operators.targetTag	附加到 **targetName** 或 **targetCatalog** 的替代标签。	字符串.例如： **v1**
mirror.platform	镜像集的平台配置。	对象
mirror.platform.architectures	要镜像的平台发行版本有效负载的架构。	字符串数组。例如： architectures: - amd64 - arm64 - multi - ppc64le - s390x 默认值为 **amd64**。值 **multi** 确保镜像支持所有可用架构，无需指定单个架构。
mirror.platform.channels	镜像集的平台频道配置。	对象数组。例如： channels: - name: stable-4.10 - name: stable-4.14

参数	描述	值
mirror.platform.channels.full	为 **true** 时，将 **minVersion** 设置为频道中的第一个发行版本，将 **maxVersion** 设置为该频道的最后一个发行版本。	布尔值.默认值为 **false**。
mirror.platform.channels.name	发行频道的名称。	字符串.例如：**stable-4.14**
mirror.platform.channels.minVersion	要镜像引用的平台的最低版本。	字符串.例如：**4.12.6**
mirror.platform.channels.maxVersion	要镜像引用的平台的最高版本。	字符串.例如：**4.14.1**
mirror.platform.channels.shortestPath	切换最短的路径镜像或完整范围镜像。	布尔值.默认值为 **false**。
mirror.platform.channels.type	要镜像的平台的类型。	字符串.例如： **ocp** 或 **okd**。默认为 **ocp**。
mirror.platform.graph	指明是否将 OSUS 图表添加到镜像集中，然后发布到镜像。	布尔值.默认值为 **false**。
storageConfig	镜像集的后端配置。	对象
storageConfig.local	镜像集的本地后端配置。	对象
storageConfig.local.path	包含镜像设置元数据的目录路径。	字符串.例如：**./path/to/dir/**。
storageConfig.registry	镜像集的 registry 后端配置。	对象
storageConfig.registry.imageURL	后端 registry URI。可以选择在 URI 中包含命名空间引用。	字符串.例如：**quay.io/myuser/imageset:metadata**。
storageConfig.registry.skipTLS	（可选）跳过引用的后端 registry 的 TLS 验证。	布尔值.默认值为 **false**。

注意

使用 **minVersion** 和 **maxVersion** 属性过滤特定 Operator 版本范围可能会导致多个频道头错误。错误信息将显示有**多个频道**头。这是因为在应用过滤器时，Operator 的更新图会被截断。

Operator Lifecycle Manager 要求每个 operator 频道都包含一个端点组成更新图表的版本，即 Operator 的最新版本。在应用图形的过滤器范围时，可以进入两个或多个独立图形或具有多个端点的图形。

要避免这个错误，请不要过滤 Operator 的最新版本。如果您仍然遇到错误，根据具体的 Operator，增加 **maxVersion** 属性，或者减少 **minVersion** 属性。因为每个 Operator 图都可以不同，所以您可能需要根据流程调整这些值，直到错误丢失为止。

4.13. 镜像设置配置示例

以下 **ImageSetConfiguration** 文件示例演示了各种镜像用例的配置。

4.13.1. 使用案例：包含最短的 OpenShift Container Platform 更新路径

以下 **ImageSetConfiguration** 文件使用本地存储后端，并包括所有 OpenShift Container Platform 版本，以及从最低 **4.11.37** 版本到最大 **4.12.15** 版本的更新路径。

ImageSetConfiguration 文件示例

```
apiVersion: mirror.openshift.io/v1alpha2
kind: ImageSetConfiguration
storageConfig:
  local:
    path: /home/user/metadata
mirror:
  platform:
    channels:
      - name: stable-4.12
        minVersion: 4.11.37
        maxVersion: 4.12.15
        shortestPath: true
```

4.13.2. 使用案例：包含对于多架构的版本的从最低到最新版本的所有 OpenShift Container Platform 版本

以下 **ImageSetConfiguration** 文件使用一个 registry 存储后端，并包括从最小 **4.13.4** 迁移到频道中最新版本的所有 OpenShift Container Platform 版本。对于每个使用此镜像集合配置的 oc-mirror，评估 **stable-4.13** 频道的最新发行版本，因此定期运行 oc-mirror 可确保您自动收到最新版本的 OpenShift Container Platform 镜像。

通过将 **platform.architectures** 的值设置为 **multi**，您可以确保支持多架构版本的镜像。

ImageSetConfiguration 文件示例

```
apiVersion: mirror.openshift.io/v1alpha2
kind: ImageSetConfiguration
storageConfig:
```

```
  registry:
    imageURL: example.com/mirror/oc-mirror-metadata
    skipTLS: false
mirror:
  platform:
    architectures:
      - "multi"
    channels:
      - name: stable-4.13
        minVersion: 4.13.4
        maxVersion: 4.13.6
```

4.13.3. 使用案例：包含从最低到最新的 Operator 版本

以下 **ImageSetConfiguration** 文件使用本地存储后端，仅包含 **stable** 频道中从 4.0.1 及之后的版本开始的 Red Hat Advanced Cluster Security for Kubernetes Operator。

注意

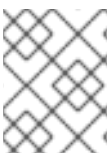
当您指定了一个最小或最大版本范围时，可能不会接收该范围内的所有 Operator 版本。

默认情况下，oc-mirror 排除了 Operator Lifecycle Manager (OLM)规格中跳过或被较新的版本替换的任何版本。跳过的 Operator 版本可能会受到 CVE 或包含错误的影响。改为使用较新版本。有关跳过和替换版本的更多信息，请参阅使用 OLM 创建更新图表。

要接收指定范围内的所有 Operator 版本，您可以将 **mirror.operators.full** 字段设置为 **true**。

ImageSetConfiguration 文件示例

```
apiVersion: mirror.openshift.io/v1alpha2
kind: ImageSetConfiguration
storageConfig:
  local:
    path: /home/user/metadata
mirror:
  operators:
    - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.14
      packages:
        - name: rhacs-operator
          channels:
          - name: stable
            minVersion: 4.0.1
```



注意

要指定最大版本而不是最新的版本，请设置 **mirror.operators.packages.channels.maxVersion** 字段。

4.13.4. 使用案例：包含 Nutanix CSI Operator

以下 **ImageSetConfiguration** 文件使用本地存储后端，并包括 Nutanix CSI Operator、OpenShift Update Service (OSUS)图形镜像以及额外的 Red Hat Universal Base Image (UBI)。

ImageSetConfiguration 文件示例

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: mylocalregistry/ocp-mirror/openshift4
    skipTLS: false
mirror:
  platform:
    channels:
    - name: stable-4.11
      type: ocp
    graph: true
  operators:
  - catalog: registry.redhat.io/redhat/certified-operator-index:v4.14
    packages:
    - name: nutanixcsioperator
      channels:
      - name: stable
  additionalImages:
  - name: registry.redhat.io/ubi9/ubi:latest
```

4.13.5. 使用案例：包含默认 Operator 频道

以下 **ImageSetConfiguration** 文件包括 OpenShift Elasticsearch Operator 的 **stable-5.7** 和 **stable** 频道。即使只需要 **stable-5.7** 频道中的软件包，**stable** 频道也必须包含在 **ImageSetConfiguration** 文件中，因为它是 Operator 的默认频道。即使您没有使用该频道中的捆绑包，还必须始终包含 Operator 软件包的默认频道。

提示

您可以运行以下命令来找到默认频道：**oc mirror list operators --catalog=<catalog_name> --package=<package_name>**。

ImageSetConfiguration 文件示例

```
apiVersion: mirror.openshift.io/v1alpha2
kind: ImageSetConfiguration
storageConfig:
  registry:
    imageURL: example.com/mirror/oc-mirror-metadata
    skipTLS: false
mirror:
  operators:
  - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.14
    packages:
    - name: elasticsearch-operator
      channels:
      - name: stable-5.7
      - name: stable
```

4.13.6. 使用案例：包含整个目录（所有版本）

以下 **ImageSetConfiguration** 文件将 **mirror.operators.full** 字段设置为 **true**，使其包含整个 Operator 目录的所有版本。

ImageSetConfiguration 文件示例

```
apiVersion: mirror.openshift.io/v1alpha2
kind: ImageSetConfiguration
storageConfig:
  registry:
    imageURL: example.com/mirror/oc-mirror-metadata
    skipTLS: false
mirror:
  operators:
    - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.14
      full: true
```

4.13.7. 使用案例：包含整个目录（仅限频道头）

以下 **ImageSetConfiguration** 文件包含整个 Operator 目录的频道头。

默认情况下，对于目录中的每个 Operator，oc-mirror 都包含来自默认频道的最新 Operator 版本（频道头）。如果要镜像所有 Operator 版本，而不仅仅是频道头，您必须将 **mirror.operators.full** 字段设置为 **true**。

本例还使用 **targetCatalog** 字段指定替代命名空间和名称来镜像目录。

ImageSetConfiguration 文件示例

```
apiVersion: mirror.openshift.io/v1alpha2
kind: ImageSetConfiguration
storageConfig:
  registry:
    imageURL: example.com/mirror/oc-mirror-metadata
    skipTLS: false
mirror:
  operators:
  - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.14
    targetCatalog: my-namespace/my-operator-catalog
```

4.13.8. 用例：包含任意镜像和 helm chart

以下 **ImageSetConfiguration** 文件使用 registry 存储后端，并包含 helm chart 和额外的 Red Hat Universal Base Image(UBI)。

ImageSetConfiguration 文件示例

```
apiVersion: mirror.openshift.io/v1alpha2
kind: ImageSetConfiguration
archiveSize: 4
storageConfig:
  registry:
```

```
    imageURL: example.com/mirror/oc-mirror-metadata
    skipTLS: false
mirror:
 platform:
   architectures:
     - "s390x"
   channels:
     - name: stable-4.14
 operators:
   - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.14
 helm:
   repositories:
     - name: redhat-helm-charts
       url: https://raw.githubusercontent.com/redhat-developer/redhat-helm-charts/master
       charts:
         - name: ibm-mongodb-enterprise-helm
           version: 0.2.0
 additionalImages:
   - name: registry.redhat.io/ubi9/ubi:latest
```

4.13.9. 用例：包含 EUS 版本的升级路径

以下 **ImageSetConfiguration** 文件包含 **eus-<version>** 频道，其中 **maxVersion** 值至少要比 **minVersion** 的值高两个次版本。

例如，在这个 **ImageSetConfiguration** 文件中，**minVersion** 设置为 **4.12.28**，而 **eus-4.14** 频道的 **maxVersion** 为 **4.14.16**。

ImageSetConfiguration 文件示例

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v2alpha1
mirror:
  platform:
    graph: true # Required for the OSUS Operator
    architectures:
    - amd64
    channels:
    - name: stable-4.12
      minVersion: '4.12.28'
      maxVersion: '4.12.28'
      shortestPath: true
      type: ocp
    - name: eus-4.14
      minVersion: '4.12.28'
      maxVersion: '4.14.16'
      shortestPath: true
      type: ocp
```

4.14. OC-MIRROR 的命令参考

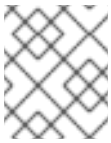
下表描述了 **oc mirror** 子命令和标志：

表 4.2. oc mirror 子命令

子命令	描述
completion	为指定的 shell 生成自动完成脚本。
describe	输出镜像集合的内容。
帮助	显示有关任何子命令的帮助。
init	输出初始镜像设置配置模板。
list	列出可用平台和 Operator 内容及其版本。
version	输出 oc-mirror 版本。

表 4.3. oc mirror 标记

标记	描述
-c, **--config <string>**	指定镜像设置配置文件的路径。
--continue-on-error	如果发生任何非镜像拉取相关的错误，请继续并尝试进行镜像(mirror)。
--dest-skip-tls	禁用目标 registry 的 TLS 验证。
--dest-use-http	使用 HTTP 用于目标 registry。
--dry-run	仅输出操作情况，不实际 mirror 镜像。生成 **mapping.txt** 和 **pruning-plan.json** 文件。
--from <string>	指定由执行 oc-mirror 生成的镜像设置归档的路径，以加载到目标 registry 中。
-h,**--help**	显示帮助。
--ignore-history	下载镜像和打包层时，忽略过去的镜像。禁用增量镜像，并可能会下载更多数据。
--manifests-only	为 **ImageContentSourcePolicy** 对象生成清单，将集群配置为使用镜像 registry，但不实际镜像任何镜像。要使用此标志，您必须使用 **--from** 标志传递镜像集存档。
--max-nested-paths <int>	指定限制嵌套路径的目标 registry 的最大嵌套路径数。默认值为 **0**。
--max-per-registry <int>	指定每个 registry 允许的并发请求数。默认值为 **6**。

标记	描述
--oci-insecure-signature-policy	在镜像本地 OCI 目录时不要推送签名（使用 **--include-local-oci-catalogs**）。
--oci-registries-config	提供 registry 配置文件，以指定在镜像本地 OCI 目录（使用 **--include-local-oci-catalogs**）时复制的替代 registry 位置。
--skip-cleanup	跳过删除工件目录。
--skip-image-pin	不要将镜像标签替换为 Operator 目录中的摘要。
--skip-metadata-check	发布镜像集时跳过元数据。  **注意** 只有在使用 **--ignore-history** 标志创建镜像集时才建议使用。
--skip-missing	如果没有找到镜像，则跳过它而不是报告错误并中止执行。不适用于在镜像设置配置中明确指定的自定义镜像。
--skip-pruning	从目标镜像 registry 禁用自动修剪镜像。
--skip-verification	跳过摘要验证。
--source-skip-tls	为源 registry 禁用 TLS 验证。
--source-use-http	将普通 HTTP 用于源 registry。
-v, --verbose <int>	指定日志级别详细程度的数量。有效值为 **0** - **9**。默认值为 **0**。

4.15. 其他资源

- 关于在断开连接的环境中的集群更新