



Red Hat OpenShift

Mirroring images for a disconnected installation by using the oc adm command

- [Prerequisites](#)
- [About the mirror registry](#)
- [Preparing your mirror host](#)
 - [Installing the OpenShift CLI](#)
- [Configuring credentials that allow images to be mirrored](#)
- [Mirroring the OKD image repository](#)
- [The Cluster Samples Operator in a disconnected environment](#)
 - [Cluster Samples Operator assistance for mirroring](#)
- [Mirroring Operator catalogs for use with disconnected clusters](#)
 - [Prerequisites](#)
 - [Extracting and mirroring catalog contents](#)
 - [Generated manifests](#)
 - [Postinstallation requirements](#)
- [Next steps](#)
- [Additional resources](#)

You can ensure your clusters only use container images that satisfy your organizational controls on external content. Before you install a cluster on infrastructure that you provision in a restricted network, you must mirror the required container images into that environment. By using the `oc adm` command, you can mirror release and catalog images in OpenShift. To mirror container images, you must have a registry for mirroring.



You must have access to the internet to obtain the necessary container images. In this procedure, you place your mirror registry on a mirror host that has access to both your network and the internet. If you do not have access to a mirror host, use the [Mirroring Operator catalogs for use with disconnected clusters](#) procedure to copy images to a device you can move across network boundaries with.

Prerequisites

- You must have a container image registry that supports [Docker v2-2](#) in the location that will host the OKD cluster, such as one of the following registries:
 - [Red Hat Quay](#)
 - [JFrog Artifactory](#)
 - [Sonatype Nexus Repository](#)
 - [Harbor](#)

If you have an entitlement to Red Hat Quay, see the documentation on deploying Red Hat Quay [for proof-of-concept purposes](#) or [by using the Red Hat Quay Operator](#). If you need additional assistance selecting and installing a registry, contact your sales representative or Red Hat Support.

- If you do not already have an existing solution for a container image registry, subscribers of OKD are provided a [mirror registry for Red Hat OpenShift](#). The *mirror registry for Red Hat OpenShift* is included with your subscription and is a small-scale container registry that can be used to mirror the required container images of OKD in disconnected installations.

About the mirror registry

You can mirror the images that are required for OKD installation and subsequent product updates to a container mirror registry such as Red Hat Quay, JFrog Artifactory, Sonatype Nexus Repository, or Harbor. If you do not have access to a large-scale container registry, you can use the *mirror registry for Red Hat OpenShift*, a small-scale container registry included with OKD subscriptions.

You can use any container registry that supports Docker v2-2, such as Red Hat Quay, the *mirror registry for Red Hat OpenShift*, Artifactory, Sonatype Nexus Repository, or Harbor. Regardless of your chosen registry, the procedure to mirror content from Red Hat hosted sites on the internet to an isolated image registry is the same. After you mirror the content, you configure each cluster to retrieve this content from your mirror registry.



The OpenShift image registry cannot be used as the target registry because it does not support pushing without a tag, which is required during the mirroring process.

If choosing a container registry that is not the *mirror registry for Red Hat OpenShift*, it must be reachable by every machine in the clusters that you provision. If the registry is unreachable, installation, updating, or normal operations such as workload relocation might fail. For that reason, you must run mirror registries in a highly available way, and the mirror registries must at least match the production availability of your OKD clusters.

When you populate your mirror registry with OKD images, you can follow two scenarios. If you have a host that can access both the internet and your mirror registry, but not your cluster nodes, you can directly mirror the content from that machine. This process is referred to as *connected mirroring*. If you have no such host, you must mirror the images to a file system and then bring that host or removable media into your restricted environment. This process is referred to as *disconnected mirroring*.

For mirrored registries, to view the source of pulled images, you must review the `Trying to access log` entry in the CRI-O logs. Other methods to view the image pull source, such as using the `crictl images` command on a node, show the non-mirrored image name, even though the image is pulled from the mirrored location.



Red Hat does not test third party registries with OKD.

Additional information

For information about viewing the CRI-O logs to view the image source, see [Viewing the image pull source](#).

Preparing your mirror host

Before you perform the mirror procedure, you must prepare the host to retrieve content and push it to the remote location.

Installing the OpenShift CLI

You can install the OpenShift CLI (`oc`) to interact with OKD from a command-line interface. You can install `oc` on Linux, Windows, or macOS.



If you installed an earlier version of `oc` , you cannot use it to complete all of the commands in OKD.

Download and install the new version of `oc` .

Installing the OpenShift CLI on Linux

You can install the OpenShift CLI (`oc`) binary on Linux by using the following procedure.

Procedure

- Navigate to <https://mirror.openshift.com/pub/openshift-v4/clients/oc/latest/> and choose the folder for your operating system and architecture.

- Download `oc.tar.gz`.
- Unpack the archive:

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

- Place the `oc` binary in a directory that is on your `PATH`.

To check your `PATH`, execute the following command:

```
$ echo $PATH
```

Verification

- After you install the OpenShift CLI, it is available using the `oc` command:

```
$ oc <command>
```

Installing the OpenShift CLI on Windows

You can install the OpenShift CLI (`oc`) binary on Windows by using the following procedure.

Procedure

- Navigate to <https://mirror.openshift.com/pub/openshift-v4/clients/oc/latest/> and choose the folder for your operating system and architecture.
- Download `oc.zip`.
- Unzip the archive with a ZIP program.
- Move the `oc` binary to a directory that is on your `PATH`.

To check your `PATH`, open the command prompt and execute the following command:

```
C:\> path
```

Verification

- After you install the OpenShift CLI, it is available using the `oc` command:

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

Installing the OpenShift CLI on macOS

You can install the OpenShift CLI (`oc`) binary on macOS by using the following procedure.

Procedure

- Navigate to <https://mirror.openshift.com/pub/openshift-v4/clients/oc/latest/> and choose the folder for your operating system and architecture.
- Download `oc.tar.gz`.
- Unpack and unzip the archive.
- Move the `oc` binary to a directory on your `PATH`.

To check your `PATH`, open a terminal and execute the following command:

```
$ echo $PATH
```

Verification

- Verify your installation by using an `oc` command:

```
$ oc <command>
```

Configuring credentials that allow images to be mirrored

Create a container image registry credentials file that enables you to mirror images from Red Hat to your mirror.



Do not use this image registry credentials file as the pull secret when you install a cluster. If you provide this file when you install cluster, all of the machines in the cluster will have write access to your mirror registry.

Prerequisites

- You configured a mirror registry to use in your disconnected environment.
- You identified an image repository location on your mirror registry to mirror images into.
- You provisioned a mirror registry account that allows images to be uploaded to that image repository.
- You have write access to the mirror registry.

Procedure

Complete the following steps on the installation host:

- Generate the base64-encoded user name and password or token for your mirror registry by running the following command:

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

- 1 For <user_name> and <password>, specify the user name and password that you configured for your registry.

Example output

```
BGVtbYk3ZHAtqXs=
```

- Create a .json file and add a section that describes your registry to it:

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

Specify the registry domain name, and optionally the port, that your mirror

- 1 registry uses to serve content. For example, registry.example.com or registry.example.com:8443
- 2 Specify the base64-encoded user name and password for the mirror registry.

Mirroring the OKD image repository

Mirror the OKD image repository to your registry to use during cluster installation or upgrade.

Prerequisites

- Your mirror host has access to the internet.
- You configured a mirror registry to use in your restricted network and can access the certificate and credentials that you configured.
- You have created a pull secret for your mirror repository.
- If you use self-signed certificates, you have specified a Subject Alternative Name in the certificates.

Procedure

Complete the following steps on the mirror host:

- Review the [OKD releases page](#) to determine the version and tag of OKD that you want to install.
- Set the required environment variables:

- Export the release version:

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

For <release_version>, specify the tag that corresponds to the version of OKD to install, such as 4.5.4.

- Export the local registry name and host port:

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

For <local_registry_host_name>, specify the registry domain name for your mirror repository, and for <local_registry_host_port>, specify the port that it serves content on.

- Export the local repository name:


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

For `<local_repository_name>`, specify the name of the repository to create in your registry, such as `ocp4/openshift4`.

- Export the name of the repository to mirror:

```
$ PRODUCT_REPO='okd'
```

- Export the path to your registry pull secret:

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

For `<path_to_pull_secret>`, specify the absolute path to and file name of the pull secret for your mirror registry that you created.

- Export the release mirror:

```
$ RELEASE_NAME="scos-release"
```

- Export the path to the directory to host the mirrored images:

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

1 Specify the full path, including the initial forward slash (/) character.

- Mirror the version images to the mirror registry:
 - If your mirror host does not have internet access, take the following actions:
 - Connect the removable media to a system that is connected to the internet.
 - Review the images and configuration manifests to mirror:

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

- Record the entire `imageContentSources` section from the output of the previous command. The information about your mirrors is unique to your mirrored repository, and you must add the `imageContentSources` section to the `install-config.yaml` file during installation.
- Mirror the images to a directory on the removable media:

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

- Take the media to the restricted network environment and upload the images to the local container 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 For `REMOVABLE_MEDIA_PATH`, you must use the same path that you specified when you mirrored the images.



Running `oc image mirror` might result in the following error: `error: unable to retrieve source image`. This error occurs when image indexes include references to images that no longer exist on the image registry. Image indexes might retain older references to allow users running those images an upgrade path to newer points on the upgrade graph. As a temporary workaround, you can use the `--skip-missing` option to bypass the error and continue downloading the image index. For more information, see [Service Mesh Operator mirroring failed](#).

- If the local container registry is connected to the mirror host, take the following actions:
 - Directly push the release images to the local registry by using following command:

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

This command pulls the release information as a digest, and its output includes the `imageContentSources` data that you require when you install your cluster.

- Record the entire `imageContentSources` section from the output of the previous command. The information about your mirrors is unique to your mirrored repository, and you must add the `imageContentSources` section to the `install-config.yaml` file during installation.



The image name gets patched to Quay.io during the mirroring process, and the podman images will show Quay.io in the registry on the bootstrap virtual machine.

- To create the installation program that is based on the content that you mirrored, extract it and pin it to the release:
 - If your mirror host does not have internet access, run the following command:

```
$ oc adm release extract -a ${LOCAL_SECRET_JSON} --icsp-file=
<file> --command=openshift-install
"${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}-${ARCHI
TECTURE}" \
--insecure=true (1)
```

1 Optional: If you do not want to configure trust for the target registry, add the `--insecure=true` flag.

- If the local container registry is connected to the mirror host, run the following command:

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



To ensure that you use the correct images for the version of OKD that you selected, you must extract the installation program from the mirrored content.

You must perform this step on a machine with an active internet connection.

- For clusters using installer-provisioned infrastructure, run the following command:

```
$ openshift-install
```

The Cluster Samples Operator in a disconnected environment

In a disconnected environment, you must take additional steps after you install a cluster to configure the Cluster Samples Operator. Review the following information in preparation.

Cluster Samples Operator assistance for mirroring

During installation, OKD creates a config map named `imagestreamtag-to-image` in the `openshift-cluster-samples-operator` namespace. The `imagestreamtag-to-image` config map contains an entry, the populating image, for each image stream tag.

The format of the key for each entry in the data field in the config map is `<image_stream_name>_<image_stream_tag_name>`.

During a disconnected installation of OKD, the status of the Cluster Samples Operator is set to `Removed`. If you choose to change it to `Managed`, it installs samples.



The use of samples in a network-restricted or discontinued environment may require access to services external to your network. Some example services include: Github, Maven Central, npm, RubyGems, PyPi and others. There might be additional steps to take that allow the cluster samples operators's objects to reach the services they require.

You can use this config map as a reference for which images need to be mirrored for your image streams to import.

- While the Cluster Samples Operator is set to `Removed`, you can create your mirrored registry, or determine which existing mirrored registry you want to use.
- Mirror the samples you want to the mirrored registry using the new config map as your guide.
- Add any of the image streams you did not mirror to the `skippedImagestreams` list of the Cluster Samples Operator configuration object.
- Set `samplesRegistry` of the Cluster Samples Operator configuration object to the mirrored registry.
- Then set the Cluster Samples Operator to `Managed` to install the image streams you have mirrored.

Mirroring Operator catalogs for use with disconnected clusters

You can mirror the Operator contents of a Red Hat-provided catalog, or a custom catalog, into a container image registry using the `oc adm catalog mirror` command. The target registry must support [Docker v2-2](#). For a cluster on a restricted network, this registry can be one that the cluster has network access to, such as a mirror registry created during a restricted network cluster installation.



- The OpenShift image registry cannot be used as the target registry because it does not support pushing without a tag, which is required during the mirroring process.
- Running `oc adm catalog mirror` might result in the following error: `error: unable to retrieve source image`. This error occurs when image indexes include references to images that no longer exist on the image registry. Image indexes might retain older references to allow users running those images an upgrade path to newer points on the upgrade graph. As a temporary workaround, you can use the `--skip-missing` option to bypass the error and continue downloading the image index. For more information, see [Service Mesh Operator mirroring failed](#).

The `oc adm catalog mirror` command also automatically mirrors the index image that is specified during the mirroring process, whether it be a Red Hat-provided index image or your own custom-built index image, to the target registry. You can then use the mirrored index image to create a catalog source that allows Operator Lifecycle Manager (OLM) to load the mirrored catalog onto your OKD cluster.

Additional resources

- [Using Operator Lifecycle Manager in disconnected environments](#)

Prerequisites

Mirroring Operator catalogs for use with disconnected clusters has the following prerequisites:

- Workstation with unrestricted network access.
- `podman` version 1.9.3 or later.
- If you want to filter, or *prune*, an existing catalog and selectively mirror only a subset of Operators, see the following sections:

- Installing the opm CLI
- Updating or filtering a file-based catalog image
- Access to a mirror registry that supports Docker v2-2.
- On your mirror registry, decide which repository, or namespace, to use for storing mirrored Operator content. For example, you might create an `olm-mirror` repository.
- If your mirror registry does not have internet access, connect removable media to your workstation with unrestricted network access.
- If you are working with private registries, including `registry.redhat.io`, set the `REG_CREDS` environment variable to the file path of your registry credentials for use in later steps. For example, for the `podman` CLI:

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

Extracting and mirroring catalog contents

The `oc adm catalog mirror` command extracts the contents of an index image to generate the manifests required for mirroring. The default behavior of the command generates manifests, then automatically mirrors all of the image content from the index image, as well as the index image itself, to your mirror registry.

Alternatively, if your mirror registry is on a completely disconnected, or *airgapped*, host, you can first mirror the content to removable media, move the media to the disconnected environment, then mirror the content from the media to the registry.

Mirroring catalog contents to registries on the same network

If your mirror registry is co-located on the same network as your workstation with unrestricted network access, take the following actions on your workstation.

Procedure

- If your mirror registry requires authentication, run the following command to log in to the registry:


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

- Run the following command to extract and mirror the content to the mirror 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)
```

- Specify the index image for the catalog that you want to mirror.
Specify the fully qualified domain name (FQDN) for the target registry to mirror the Operator contents to. The mirror registry `<repository>` can be any existing repository, or namespace, on the registry, for example `olm-mirror` as outlined in the prerequisites. If there is an existing repository found during mirroring, the repository name is added to the resulting image name. If you do not want the image name to include the repository name, omit the `<repository>` value from this line, for example `<mirror_registry>:<port>`.
- Optional: If required, specify the location of your registry credentials file. `{REG_CREDS}` is required for `registry.redhat.io`.
- Optional: If you do not want to configure trust for the target registry, add the `--insecure` flag.
Optional: Specify which platform and architecture of the index image to select when multiple variants are available. Images are passed as `'<platform>/<arch>[/<variant>]'`. This does not apply to images referenced by the index. Valid values are `linux/amd64`, `linux/ppc64le`, `linux/s390x`, `linux/arm64`.
- Optional: Generate only the manifests required for mirroring without actually mirroring the image content to a registry. This option can be useful for reviewing what will be mirrored, and lets you make any changes to the mapping list, if you require only a subset of packages. You can then use the `mapping.txt` file with the `oc image mirror` command to mirror the modified list of images in a later step. This flag is intended for only advanced selective mirroring of content from the catalog.

Example output

```
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-catalog-1614211642 (2)
```

- 1 Directory for the temporary `index.db` database generated by the command.

Record the manifests directory name that is generated. This directory is referenced in subsequent procedures.

2



Red Hat Quay does not support nested repositories. As a result, running the `oc adm catalog mirror` command will fail with a 401 unauthorized error. As a workaround, you can use the `--max-components=2` option when running the `oc adm catalog mirror` command to disable the creation of nested repositories. For more information on this workaround, see the [Unauthorized error thrown while using catalog mirror command with Quay registry](#) Knowledgebase Solution.

Mirroring catalog contents to airgapped registries

If your mirror registry is on a completely disconnected, or airgapped, host, take the following actions.

Procedure

- Run the following command on your workstation with unrestricted network access to mirror the content to local files:

```
$ 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 Specify the index image for the catalog that you want to mirror.
- 2 Specify the content to mirror to local files in your current directory.
- 3 Optional: If required, specify the location of your registry credentials file.
- 4 Optional: If you do not want to configure trust for the target registry, add the `--insecure` flag.
Optional: Specify which platform and architecture of the index image to select when multiple variants are available. Images are specified as
- 5 '`<platform>/<arch>[/<variant>]`'. This does not apply to images referenced by the index. Valid values are `linux/amd64`, `linux/ppc64le`, `linux/s390x`, `linux/arm64`, and `.*`

Example output

```
...
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 Record the manifests directory name that is generated. This directory is referenced in subsequent procedures.
- 2 Record the expanded `file:///` path that is based on your provided index image. This path is referenced in a subsequent step.

This command creates a `v2/` directory in your current directory.

- Copy the `v2/` directory to removable media.
- Physically remove the media and attach it to a host in the disconnected environment that has access to the mirror registry.
- If your mirror registry requires authentication, run the following command on your host in the disconnected environment to log in to the registry:

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

- Run the following command from the parent directory containing the `v2/` directory to upload the images from local files to the mirror 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 Specify the `file://` path from the previous command output.
Specify the fully qualified domain name (FQDN) for the target registry to mirror the Operator contents to. The mirror registry `<repository>` can be any existing repository, or namespace, on the registry, for example `olm-mirror` as outlined in the prerequisites. If there is an existing repository found during mirroring, the repository name is added to the resulting image name. If you do not want the image name to include the repository name, omit the `<repository>` value from this line, for example `<mirror_registry>:<port>`.
- 2
- 3 Optional: If required, specify the location of your registry credentials file.
- 4 Optional: If you do not want to configure trust for the target registry, add the `--insecure` flag.
Optional: Specify which platform and architecture of the index image to select when multiple variants are available. Images are specified as `'<platform>/<arch>[/<variant>]'`. This does not apply to images referenced by the index. Valid values are `linux/amd64`, `linux/ppc64le`, `linux/s390x`, `linux/arm64`, and `.*`.
- 5



Red Hat Quay does not support nested repositories. As a result, running the `oc adm catalog mirror` command will fail with a 401 unauthorized error. As a workaround, you can use the `--max-components=2` option when running the `oc adm catalog mirror` command to disable the creation of nested repositories. For more information on this workaround, see the [Unauthorized error thrown while using catalog mirror command with Quay registry](#) Knowledgebase Solution.

- Run the `oc adm catalog mirror` command again. Use the newly mirrored index image as the source and the same mirror registry target used in the previous step:

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

- 1 The --manifests-only flag is required for this step so that the command does not copy all of the mirrored content again.



This step is required because the image mappings in the `imageContentSourcePolicy.yaml` file generated during the previous step must be updated from local paths to valid mirror locations. Failure to do so will cause errors when you create the `ImageContentSourcePolicy` object in a later step.

After you mirror the catalog, you can continue with the remainder of your cluster installation. After your cluster installation has finished successfully, you must specify the manifests directory from this procedure to create the `ImageContentSourcePolicy` and `CatalogSource` objects. These objects are required to enable installation of Operators from OperatorHub.

Generated manifests

After mirroring Operator catalog content to your mirror registry, a manifests directory is generated in your current directory.

If you mirrored content to a registry on the same network, the directory name takes the following pattern:

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

If you mirrored content to a registry on a disconnected host in the previous section, the directory name takes the following pattern:

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



The manifests directory name is referenced in subsequent procedures.

The manifests directory contains the following files, some of which might require further modification:

- The `catalogSource.yaml` file is a basic definition for a `CatalogSource` object that is pre-populated with your index image tag and other relevant metadata. This file can be used as is or modified to add the catalog source to your cluster.



If you mirrored the content to local files, you must modify your `catalogSource.yaml` file to remove any backslash (/) characters from the `metadata.name` field. Otherwise, when you attempt to create the object, it fails with an "invalid resource name" error.

- The `imageContentSourcePolicy.yaml` file defines an `ImageContentSourcePolicy` object that can configure nodes to translate between the image references stored in Operator manifests and the mirrored registry.



If your cluster uses an `ImageContentSourcePolicy` object to configure repository mirroring, you can use only global pull secrets for mirrored registries. You cannot add a pull secret to a project.

- The `mapping.txt` file contains all of the source images and where to map them in the target registry. This file is compatible with the `oc image mirror` command and can be used to further customize the mirroring configuration.



If you used the `--manifests-only` flag during the mirroring process and want to further trim the subset of packages to mirror, see the steps in the [Mirroring a package manifest format catalog image](#) procedure of the OKD 4.7 documentation about modifying your `mapping.txt` file and using the file with the `oc image mirror` command.

Postinstallation requirements

After you mirror the catalog, you can continue with the remainder of your cluster installation. After your cluster installation has finished successfully, you must specify the manifests directory from this procedure to create the

`ImageContentSourcePolicy` and `CatalogSource` objects. These objects are required to populate and enable installation of Operators from OperatorHub.

Additional resources

- [Populating OperatorHub from mirrored Operator catalogs](#)
- [Updating or filtering a file-based catalog image](#)

Next steps

- Install a cluster on infrastructure that you provision in your restricted network, such as on [VMware vSphere](#), [bare metal](#), or [Amazon Web Services](#).

Additional resources

- See [Gathering data about specific features](#) for more information about using must-gather.