# OpenShift Container Platform 4.3

## Updating clusters

Updating OpenShift Container Platform clusters

Last Updated: 2020-10-22

# OpenShift Container Platform 4.3 Updating clusters

Updating OpenShift Container Platform clusters

## Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## Abstract

This document provides instructions for updating, or upgrading, OpenShift Container Platform clusters. In version , updating your cluster is a simple process that does not require you to take your cluster offline.

# Table of Contents

CHAPTER 1. UPDATING A CLUSTER BETWEEN MINOR VERSIONS ..... 4
1.1. PREREQUISITES 4
1.2. ABOUT THE OPENSHIFT CONTAINER PLATFORM UPDATE SERVICE 4
1.3. OPENSHIFT CONTAINER PLATFORM UPGRADE CHANNELS AND RELEASES 5
candidate-4.3 channel 6
fast-4.3 channel 6
stable-4.3 channel 6
Upgrade version paths 6
Fast and stable channel use and strategies 7
Restricted network clusters 7
Switching between channels 7
1.4. UPDATING A CLUSTER BY USING THE WEB CONSOLE 7

CHAPTER 2. UPDATING A CLUSTER WITHIN A MINOR VERSION FROM THE WEB CONSOLE ..... 9
2.1. PREREQUISITES 9
2.2. ABOUT THE OPENSHIFT CONTAINER PLATFORM UPDATE SERVICE 9
2.3. OPENSHIFT CONTAINER PLATFORM UPGRADE CHANNELS AND RELEASES 10
candidate-4.3 channel 10
fast-4.3 channel 11
stable-4.3 channel 11
Upgrade version paths 11
Fast and stable channel use and strategies 12
Restricted network clusters 12
Switching between channels 12
2.4. UPDATING A CLUSTER BY USING THE WEB CONSOLE 12

CHAPTER 3. UPDATING A CLUSTER WITHIN A MINOR VERSION BY USING THE CLI ..... 14
3.1. PREREQUISITES 14
3.2. ABOUT THE OPENSHIFT CONTAINER PLATFORM UPDATE SERVICE 14
3.3. OPENSHIFT CONTAINER PLATFORM UPGRADE CHANNELS AND RELEASES 15
candidate-4.3 channel 15
fast-4.3 channel 16
stable-4.3 channel 16
Upgrade version paths 16
Fast and stable channel use and strategies 17
Restricted network clusters 17
Switching between channels 17
3.4. UPDATING A CLUSTER BY USING THE CLI 17

CHAPTER 4. UPDATING A CLUSTER THAT INCLUDES RHEL COMPUTE MACHINES ..... 21
4.1. PREREQUISITES 21
4.2. ABOUT THE OPENSHIFT CONTAINER PLATFORM UPDATE SERVICE 21
4.3. OPENSHIFT CONTAINER PLATFORM UPGRADE CHANNELS AND RELEASES 22
candidate-4.3 channel 22
fast-4.3 channel 23
stable-4.3 channel 23
Upgrade version paths 23
Fast and stable channel use and strategies 24
Restricted network clusters 24
Switching between channels 24
4.4. UPDATING A CLUSTER BY USING THE WEB CONSOLE 24
4.5. (OPTIONAL) ADDING HOOKS TO PERFORM ANSIBLE TASKS ON RHEL MACHINES 25

4.5.1. About Ansible hooks for upgrades 26
4.5.2. Configuring the Ansible inventory file to use hooks 26
4.5.3. Available hooks for RHEL compute machines 27
4.6. UPDATING RHEL COMPUTE MACHINES IN YOUR CLUSTER 27

**CHAPTER 5. UPDATING A RESTRICTED NETWORK CLUSTER 30**
5.1. PREPARING YOUR MIRROR HOST 30
5.1.1. Installing the CLI by downloading the binary 30
5.1.1.1. Installing the CLI on Linux 30
5.1.1.2. Installing the CLI on Windows 31
5.1.1.3. Installing the CLI on macOS 31
5.2. CONFIGURING CREDENTIALS THAT ALLOW IMAGES TO BE MIRRORED 32
5.3. MIRRORING THE OPENSHIFT CONTAINER PLATFORM IMAGE REPOSITORY 35
5.4. CREATING THE IMAGE CONFIGMAP 37
5.4.1. Creating the ConfigMap for image signature verification by using the oc CLI 37
5.4.2. Creating an image signature ConfigMap manually 37
5.5. UPGRADING THE RESTRICTED NETWORK CLUSTER 38
5.6. CONFIGURING IMAGE REGISTRY REPOSITORY MIRRORING 39

# CHAPTER 1. UPDATING A CLUSTER BETWEEN MINOR VERSIONS

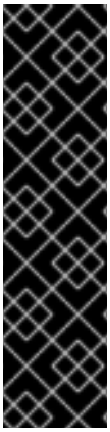
You can update, or upgrade, an OpenShift Container Platform cluster between minor versions.

**NOTE**

Because of the difficulty of changing update channels by using **oc**, use the web console to change the update channel. It is recommended to complete the update process within the web console. You can follow the steps in Updating a cluster within a minor version by using the CLI to complete the update after you change to a 4.3 channel.

## 1.1. PREREQUISITES

- Have access to the cluster as a user with **admin** privileges. See Using RBAC to define and apply permissions.
- Have a recent etcd backup in case your upgrade fails and you must restore your cluster to a previous state.

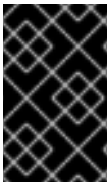


**IMPORTANT**

If you are upgrading to this release from OpenShift Container Platform 4.2, you must restart all Pods after the upgrade is complete.

This is because the service CA is automatically rotated as of OpenShift Container Platform 4.3.5. The service CA is rotated during the upgrade and a restart is required afterward to ensure that all services are using the new service CA before the previous service CA expires.

After this one-time manual restart, subsequent upgrades and rotations will ensure restart before the service CA expires without requiring manual intervention.



**IMPORTANT**

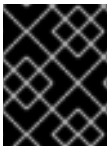
Using the **unsupportedConfigOverrides** section to modify an Operator's configuration is unsupported and might block cluster upgrades. You must remove this setting before you can upgrade your cluster.

## 1.2. ABOUT THE OPENSHIFT CONTAINER PLATFORM UPDATE SERVICE

The OpenShift Container Platform update service is the hosted service that provides over-the-air updates to both OpenShift Container Platform and Red Hat Enterprise Linux CoreOS (RHCOS). It provides a graph, or diagram that contain *vertices* and the *edges* that connect them, of component Operators. The edges in the graph show which versions you can safely update to, and the vertices are update payloads that specify the intended state of the managed cluster components.

The Cluster Version Operator (CVO) in your cluster checks with the OpenShift Container Platform update service to see the valid updates and update paths based on current component versions and information in the graph. When you request an update, the OpenShift Container Platform CVO uses the release image for that update to upgrade your cluster. The release artifacts are hosted in Quay as container images.

To allow the OpenShift Container Platform update service to provide only compatible updates, a release verification pipeline exists to drive automation. Each release artifact is verified for compatibility with supported cloud platforms and system architectures as well as other component packages. After the pipeline confirms the suitability of a release, the OpenShift Container Platform update service notifies you that it is available.



**IMPORTANT**

Because the update service displays all valid updates, you must not force an update to a version that the update service does not display.

During continuous update mode, two controllers run. One continuously updates the payload manifests, applies them to the cluster, and outputs the status of the controlled rollout of the Operators, whether they are available, upgrading, or failed. The second controller polls the OpenShift Container Platform update service to determine if updates are available.

**IMPORTANT**

Reverting your cluster to a previous version, or a rollback, is not supported. Only upgrading to a newer version is supported.

During the upgrade process, the Machine Config Operator (MCO) applies the new configuration to your cluster machines. It cordons the number of nodes that is specified by the **maxUnavailable** field on the machine configuration pool and marks them as unavailable. By default, this value is set to **1**. It then applies the new configuration and reboots the machine. If you use Red Hat Enterprise Linux (RHEL) machines as workers, the MCO does not update the kubelet on these machines because you must update the OpenShift API on them first. Because the specification for the new version is applied to the old kubelet, the RHEL machine cannot return to the **Ready** state. You cannot complete the update until the machines are available. However, the maximum number of nodes that are unavailable is set to ensure that normal cluster operations are likely to continue with that number of machines out of service.

**Additional resources**

- Support policy for unmanaged Operators

## 1.3. OPENSHIFT CONTAINER PLATFORM UPGRADE CHANNELS AND RELEASES

In OpenShift Container Platform 4.1, Red Hat introduced the concept of channels for recommending the appropriate release versions for cluster upgrade. By controlling the pace of upgrades, these upgrade channels allow you to choose an upgrade strategy. Upgrade channels are tied to a minor version of OpenShift Container Platform. For instance, OpenShift Container Platform 4.3 upgrade channels will never include an upgrade to a 4.4 release. This strategy ensures that administrators explicitly decide to upgrade to the next minor version of OpenShift Container Platform. Upgrade channels control only release selection and do not impact the version of the cluster that you install; the **openshift-install** binary file for a specific version of OpenShift Container Platform always installs that version.

OpenShift Container Platform 4.3 offers the following upgrade channels:

- **candidate-4.3**
- **fast-4.3**
- **stable-4.3**

### candidate-4.3 channel

The **candidate-4.3** channel contains candidate builds for a z-stream (4.3.z) release. Release candidates contain all the features of the product but are not supported. Use release candidate versions to test feature acceptance and assist in qualifying the next version of OpenShift Container Platform. A release candidate is any build that is available in the candidate channel, including ones that do not contain **-rc** in their names. After a version is available in the candidate channel, it goes through more quality checks. If it meets the quality standard, it is promoted to the **fast-4.3** or **stable-4.3** channels. Because of this strategy, if a specific release is available in both the **candidate-4.3** channel and in the **fast-4.3** or **stable-4.3** channels, it is a Red Hat supported version. The **candidate-4.3** channel can include release versions from which there are no recommended updates in any channel.

You can use the **candidate-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

NOTE

Release candidates differ from the nightly builds. Nightly builds are available for early access to features, but updating to or from nightly builds is neither recommended nor supported. Nightly builds are not available in any upgrade channel. You can reference the OpenShift Container Platform release statuses for more build information.

### fast-4.3 channel

The **fast-4.3** channel is updated with new 4.3 versions as soon as Red Hat declares the given version as a general availability release. As such, these releases are fully supported, are production quality, and have performed well while available as a release candidate in the **candidate-4.3** channel from where they were promoted. Some time after a release appears in the **fast-4.3** channel, it is added to the **stable-4.3** channel. Releases never appear in the **stable-4.3** channel before they appear in the **fast-4.3** channel.

You can use the **fast-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

### stable-4.3 channel

While the **fast-4.3** channel contains releases as soon as their errata are published, releases are added to the **stable-4.3** channel after a delay. During this delay, data is collected from Red Hat SRE teams, Red Hat support services, and pre-production and production environments that participate in connected customer program about the stability of the release.

You can use the **stable-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

### Upgrade version paths

OpenShift Container Platform maintains an upgrade recommendation service that understands the version of OpenShift Container Platform you have installed as well as the path to take within the channel you choose to get you to the next release. You can imagine seeing the following in the **fast-4.3** channel:

- 4.3.0
- 4.3.1
- 4.3.3
- 4.3.4

The service recommends only upgrades that have been tested and have no serious issues. If your cluster is on 4.3.1 and OpenShift Container Platform suggests 4.3.4, then it is safe for you to update from 4.3.1 to 4.3.4. Do not rely on consecutive patch numbers. In this example, 4.3.2 is not, and never was, available

in the channel. The update service will not suggest updating to a version of OpenShift Container Platform that contains known vulnerabilities.

Update stability depends on your channel. The presence of an update recommendation in the **candidate-4.3** channel does not imply that the update is supported. It means that no serious issues have been found with the update yet, but there might not be significant traffic through the update to suggest stability. The presence of an update recommendation in the **fast-4.3** or **stable-4.3** channels is a declaration that the update is fully supported while it is in the channel. While releases will never be removed from a channel, update recommendations that exhibit serious issues will be removed from all channels. Updates initiated after the update recommendation has been removed might not be supported.

Red Hat will eventually provide supported update paths from any supported release in the **fast-4.3** or **stable-4.3** channels to the latest release in 4.3.z, although there can be delays while safe paths away from troubled releases are constructed and verified.

### Fast and stable channel use and strategies

The **fast-4.3** and **stable-4.3** channels present a choice between receiving general availability releases as soon as they are available or allowing Red Hat to control the rollout of those updates. If issues are detected during rollout or at a later time, upgrades to that version might be blocked in both the **fast-4.3** and **stable-4.3** channels, and a new version might be introduced that becomes the new preferred upgrade target.

Customers can improve this process by configuring pre-production systems on the **fast-4.3** channel, configuring production systems on the **stable-4.3** channel, and participating in Red Hat's connected customer program. Red Hat uses this program to observe the impact of updates on your specific hardware and software configurations. Future releases might improve or alter the pace at which updates move from the **fast-4.3** to the **stable-4.3** channel.

### Restricted network clusters

If you manage the container images for your OpenShift Container Platform clusters yourself, you must consult the Red Hat errata that is associated with product releases and note any comments that impact upgrades. During upgrade, the user interface might warn you about switching between these versions, so you must ensure that you selected an appropriate version before you bypass those warnings.

### Switching between channels

Your cluster is still supported if you change from the **stable-4.3** channel to the **fast-4.3** channel. Although you can switch to the **candidate-4.3** channel at any time, some releases in that channel might be unsupported release candidates. You can switch from the **candidate-4.3** channel to the **fast-4.3** channel if your current release is a general availability release. You can always switch from the **fast-4.3** channel to the **stable-4.3** channel, although if the current release was recently promoted to **fast-4.3** there can be a delay of up to a day for the release to be promoted to **stable-4.3**. If you change to a channel that does not include your current release, an alert displays and no updates can be recommended, but you can safely change back to your original channel at any point.

## 1.4. UPDATING A CLUSTER BY USING THE WEB CONSOLE

If updates are available, you can update your cluster from the web console.

You can find information about available OpenShift Container Platform advisories and updates in the errata section of the Customer Portal.

### Prerequisites

- Have access to the web console as a user with **admin** privileges.

**Procedure**

1. From the web console, click **Administration** > **Cluster Settings** and review the contents of the **Overview** tab.

2. For production clusters, ensure that the **CHANNEL** is set to the correct channel for your current minor version, such as **stable-4.3**.

IMPORTANT

For production clusters, you must subscribe to a stable-* or fast-* channel.

   - If the **UPDATE STATUS** is not **Updates Available**, you cannot upgrade your cluster.
   - The **DESIRED VERSION** indicates the cluster version that your cluster is running or is updating to.

3. Click **Updates Available**, select the highest available version and click **Update**. The **UPDATE STATUS** changes to **Updating**, and you can review the progress of the Operator upgrades on the **Cluster Operators** tab.

4. If you are upgrading to this release from OpenShift Container Platform 4.2, you must restart all Pods after the upgrade is complete. You can do this using the following command, which requires the OpenShift CLI (**oc**):

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

NOTE

Restarting all Pods is required because the service CA is automatically rotated as of OpenShift Container Platform 4.3.5. The service CA is rotated during the upgrade and a restart is required afterward to ensure that all services are using the new service CA before the previous service CA expires.

After this one-time manual restart, subsequent upgrades and rotations will ensure restart before the service CA expires without requiring manual intervention.

5. After the update completes and the Cluster Version Operator refreshes the available updates, check if more updates are available in your current channel.

   - If updates are available, continue to perform updates in the current channel until you can no longer update.
   - If no updates are available, change the **CHANNEL** to the stable-* or fast-* channel for the next minor version, and update to the version that you want in that channel.

   You might need to perform several intermediate updates until you reach the version that you want.

# CHAPTER 2. UPDATING A CLUSTER WITHIN A MINOR VERSION FROM THE WEB CONSOLE

You can update, or upgrade, an OpenShift Container Platform cluster by using the web console.

## 2.1. PREREQUISITES

- Have access to the cluster as a user with **admin** privileges. See Using RBAC to define and apply permissions.
- Have a recent etcd backup in case your upgrade fails and you must restore your cluster to a previous state.

**IMPORTANT**

If you are upgrading to this release from OpenShift Container Platform 4.3.3 or earlier, you must restart all Pods after the upgrade is complete.

This is because the service CA is automatically rotated as of OpenShift Container Platform 4.3.5. The service CA is rotated during the upgrade and a restart is required afterward to ensure that all services are using the new service CA before the previous service CA expires.

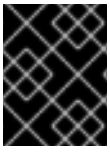
After this one-time manual restart, subsequent upgrades and rotations will ensure restart before the service CA expires without requiring manual intervention.

## 2.2. ABOUT THE OPENSHIFT CONTAINER PLATFORM UPDATE SERVICE

The OpenShift Container Platform update service is the hosted service that provides over-the-air updates to both OpenShift Container Platform and Red Hat Enterprise Linux CoreOS (RHCOS). It provides a graph, or diagram that contain *vertices* and the *edges* that connect them, of component Operators. The edges in the graph show which versions you can safely update to, and the vertices are update payloads that specify the intended state of the managed cluster components.

The Cluster Version Operator (CVO) in your cluster checks with the OpenShift Container Platform update service to see the valid updates and update paths based on current component versions and information in the graph. When you request an update, the OpenShift Container Platform CVO uses the release image for that update to upgrade your cluster. The release artifacts are hosted in Quay as container images.

To allow the OpenShift Container Platform update service to provide only compatible updates, a release verification pipeline exists to drive automation. Each release artifact is verified for compatibility with supported cloud platforms and system architectures as well as other component packages. After the pipeline confirms the suitability of a release, the OpenShift Container Platform update service notifies you that it is available.

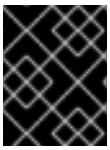


**IMPORTANT**

Because the update service displays all valid updates, you must not force an update to a version that the update service does not display.

During continuous update mode, two controllers run. One continuously updates the payload manifests,

applies them to the cluster, and outputs the status of the controlled rollout of the Operators, whether they are available, upgrading, or failed. The second controller polls the OpenShift Container Platform update service to determine if updates are available.



IMPORTANT

Reverting your cluster to a previous version, or a rollback, is not supported. Only upgrading to a newer version is supported.

During the upgrade process, the Machine Config Operator (MCO) applies the new configuration to your cluster machines. It cordons the number of nodes that is specified by the **maxUnavailable** field on the machine configuration pool and marks them as unavailable. By default, this value is set to **1**. It then applies the new configuration and reboots the machine. If you use Red Hat Enterprise Linux (RHEL) machines as workers, the MCO does not update the kubelet on these machines because you must update the OpenShift API on them first. Because the specification for the new version is applied to the old kubelet, the RHEL machine cannot return to the **Ready** state. You cannot complete the update until the machines are available. However, the maximum number of nodes that are unavailable is set to ensure that normal cluster operations are likely to continue with that number of machines out of service.

**Additional resources**

- Support policy for unmanaged Operators

## 2.3. OPENSHIFT CONTAINER PLATFORM UPGRADE CHANNELS AND RELEASES

In OpenShift Container Platform 4.1, Red Hat introduced the concept of channels for recommending the appropriate release versions for cluster upgrade. By controlling the pace of upgrades, these upgrade channels allow you to choose an upgrade strategy. Upgrade channels are tied to a minor version of OpenShift Container Platform. For instance, OpenShift Container Platform 4.3 upgrade channels will never include an upgrade to a 4.4 release. This strategy ensures that administrators explicitly decide to upgrade to the next minor version of OpenShift Container Platform. Upgrade channels control only release selection and do not impact the version of the cluster that you install; the **openshift-install** binary file for a specific version of OpenShift Container Platform always installs that version.

OpenShift Container Platform 4.3 offers the following upgrade channels:

- **candidate-4.3**
- **fast-4.3**
- **stable-4.3**

### candidate-4.3 channel

The **candidate-4.3** channel contains candidate builds for a z-stream (4.3.z) release. Release candidates contain all the features of the product but are not supported. Use release candidate versions to test feature acceptance and assist in qualifying the next version of OpenShift Container Platform. A release candidate is any build that is available in the candidate channel, including ones that do not contain **-rc** in their names. After a version is available in the candidate channel, it goes through more quality checks. If it meets the quality standard, it is promoted to the **fast-4.3** or **stable-4.3** channels. Because of this strategy, if a specific release is available in both the **candidate-4.3** channel and in the **fast-4.3** or **stable-4.3** channels, it is a Red Hat supported version. The **candidate-4.3** channel can include release versions from which there are no recommended updates in any channel.

You can use the **candidate-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

NOTE

Release candidates differ from the nightly builds. Nightly builds are available for early access to features, but updating to or from nightly builds is neither recommended nor supported. Nightly builds are not available in any upgrade channel. You can reference the OpenShift Container Platform release statuses for more build information.

### fast-4.3 channel

The **fast-4.3** channel is updated with new 4.3 versions as soon as Red Hat declares the given version as a general availability release. As such, these releases are fully supported, are production quality, and have performed well while available as a release candidate in the **candidate-4.3** channel from where they were promoted. Some time after a release appears in the **fast-4.3** channel, it is added to the **stable-4.3** channel. Releases never appear in the **stable-4.3** channel before they appear in the **fast-4.3** channel.

You can use the **fast-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

### stable-4.3 channel

While the **fast-4.3** channel contains releases as soon as their errata are published, releases are added to the **stable-4.3** channel after a delay. During this delay, data is collected from Red Hat SRE teams, Red Hat support services, and pre-production and production environments that participate in connected customer program about the stability of the release.

You can use the **stable-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

### Upgrade version paths

OpenShift Container Platform maintains an upgrade recommendation service that understands the version of OpenShift Container Platform you have installed as well as the path to take within the channel you choose to get you to the next release. You can imagine seeing the following in the **fast-4.3** channel:

- 4.3.0
- 4.3.1
- 4.3.3
- 4.3.4

The service recommends only upgrades that have been tested and have no serious issues. If your cluster is on 4.3.1 and OpenShift Container Platform suggests 4.3.4, then it is safe for you to update from 4.3.1 to 4.3.4. Do not rely on consecutive patch numbers. In this example, 4.3.2 is not, and never was, available in the channel. The update service will not suggest updating to a version of OpenShift Container Platform that contains known vulnerabilities.

Update stability depends on your channel. The presence of an update recommendation in the **candidate-4.3** channel does not imply that the update is supported. It means that no serious issues have been found with the update yet, but there might not be significant traffic through the update to suggest stability. The presence of an update recommendation in the **fast-4.3** or **stable-4.3** channels is a declaration that the update is fully supported while it is in the channel. While releases will never be removed from a channel, update recommendations that exhibit serious issues will be removed from all channels. Updates initiated after the update recommendation has been removed might not be supported.

Red Hat will eventually provide supported update paths from any supported release in the **fast-4.3** or **stable-4.3** channels to the latest release in 4.3.z, although there can be delays while safe paths away from troubled releases are constructed and verified.

### Fast and stable channel use and strategies

The **fast-4.3** and **stable-4.3** channels present a choice between receiving general availability releases as soon as they are available or allowing Red Hat to control the rollout of those updates. If issues are detected during rollout or at a later time, upgrades to that version might be blocked in both the **fast-4.3** and **stable-4.3** channels, and a new version might be introduced that becomes the new preferred upgrade target.

Customers can improve this process by configuring pre-production systems on the **fast-4.3** channel, configuring production systems on the **stable-4.3** channel, and participating in Red Hat's connected customer program. Red Hat uses this program to observe the impact of updates on your specific hardware and software configurations. Future releases might improve or alter the pace at which updates move from the **fast-4.3** to the **stable-4.3** channel.

### Restricted network clusters

If you manage the container images for your OpenShift Container Platform clusters yourself, you must consult the Red Hat errata that is associated with product releases and note any comments that impact upgrades. During upgrade, the user interface might warn you about switching between these versions, so you must ensure that you selected an appropriate version before you bypass those warnings.

### Switching between channels

Your cluster is still supported if you change from the **stable-4.3** channel to the **fast-4.3** channel. Although you can switch to the **candidate-4.3** channel at any time, some releases in that channel might be unsupported release candidates. You can switch from the **candidate-4.3** channel to the **fast-4.3** channel if your current release is a general availability release. You can always switch from the **fast-4.3** channel to the **stable-4.3** channel, although if the current release was recently promoted to **fast-4.3** there can be a delay of up to a day for the release to be promoted to **stable-4.3**. If you change to a channel that does not include your current release, an alert displays and no updates can be recommended, but you can safely change back to your original channel at any point.

## 2.4. UPDATING A CLUSTER BY USING THE WEB CONSOLE

If updates are available, you can update your cluster from the web console.

You can find information about available OpenShift Container Platform advisories and updates in the errata section of the Customer Portal.

### Prerequisites

- Have access to the web console as a user with **admin** privileges.

### Procedure

1. From the web console, click **Administration** > **Cluster Settings** and review the contents of the **Overview** tab.

2. For production clusters, ensure that the **CHANNEL** is set to the correct channel for the version that you want to update to, your current minor version, such as **stable-4.3**.

**IMPORTANT**

For production clusters, you must subscribe to a stable-* or fast-* channel.

- If the **UPDATE STATUS** is not **Updates Available**, you cannot upgrade your cluster.
- The **DESIRED VERSION** indicates the cluster version that your cluster is running or is updating to.

3. Click **Updates Available**, select a version to update to, the highest available version and click **Update**. The **UPDATE STATUS** changes to **Updating**, and you can review the progress of the Operator upgrades on the **Cluster Operators** tab.

4. If you are upgrading to this release from OpenShift Container Platform 4.3.3 or earlier, 4.2, you must restart all Pods after the upgrade is complete. You can do this using the following command, which requires the OpenShift CLI (**oc**):

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

NOTE

Restarting all Pods is required because the service CA is automatically rotated as of OpenShift Container Platform 4.3.5. 4.3.5. The service CA is rotated during the upgrade and a restart is required afterward to ensure that all services are using the new service CA before the previous service CA expires.

After this one-time manual restart, subsequent upgrades and rotations will ensure restart before the service CA expires without requiring manual intervention.

5. After the update completes and the Cluster Version Operator refreshes the available updates, check if more updates are available in your current channel.
   - If updates are available, continue to perform updates in the current channel until you can no longer update.
   - If no updates are available, change the **CHANNEL** to the stable-* or fast-* channel for the next minor version, and update to the version that you want in that channel.

   You might need to perform several intermediate updates until you reach the version that you want.

# CHAPTER 3. UPDATING A CLUSTER WITHIN A MINOR VERSION BY USING THE CLI

You can update, or upgrade, an OpenShift Container Platform cluster within a minor version by using the OpenShift CLI (**oc**).

## 3.1. PREREQUISITES

- Have access to the cluster as a user with **admin** privileges. See Using RBAC to define and apply permissions.
- Have a recent etcd backup in case your upgrade fails and you must restore your cluster to a previous state.

**IMPORTANT**

If you are upgrading to this release from OpenShift Container Platform 4.3.3 or earlier, you must restart all Pods after the upgrade is complete.

This is because the service CA is automatically rotated as of OpenShift Container Platform 4.3.5. The service CA is rotated during the upgrade and a restart is required afterward to ensure that all services are using the new service CA before the previous service CA expires.

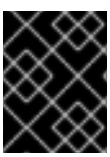
After this one-time manual restart, subsequent upgrades and rotations will ensure restart before the service CA expires without requiring manual intervention.

## 3.2. ABOUT THE OPENSHIFT CONTAINER PLATFORM UPDATE SERVICE

The OpenShift Container Platform update service is the hosted service that provides over-the-air updates to both OpenShift Container Platform and Red Hat Enterprise Linux CoreOS (RHCOS). It provides a graph, or diagram that contain *vertices* and the *edges* that connect them, of component Operators. The edges in the graph show which versions you can safely update to, and the vertices are update payloads that specify the intended state of the managed cluster components.

The Cluster Version Operator (CVO) in your cluster checks with the OpenShift Container Platform update service to see the valid updates and update paths based on current component versions and information in the graph. When you request an update, the OpenShift Container Platform CVO uses the release image for that update to upgrade your cluster. The release artifacts are hosted in Quay as container images.

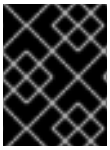
To allow the OpenShift Container Platform update service to provide only compatible updates, a release verification pipeline exists to drive automation. Each release artifact is verified for compatibility with supported cloud platforms and system architectures as well as other component packages. After the pipeline confirms the suitability of a release, the OpenShift Container Platform update service notifies you that it is available.



**IMPORTANT**

Because the update service displays all valid updates, you must not force an update to a version that the update service does not display.

During continuous update mode, two controllers run. One continuously updates the payload manifests, applies them to the cluster, and outputs the status of the controlled rollout of the Operators, whether they are available, upgrading, or failed. The second controller polls the OpenShift Container Platform update service to determine if updates are available.



IMPORTANT

Reverting your cluster to a previous version, or a rollback, is not supported. Only upgrading to a newer version is supported.

During the upgrade process, the Machine Config Operator (MCO) applies the new configuration to your cluster machines. It cordons the number of nodes that is specified by the **maxUnavailable** field on the machine configuration pool and marks them as unavailable. By default, this value is set to **1**. It then applies the new configuration and reboots the machine. If you use Red Hat Enterprise Linux (RHEL) machines as workers, the MCO does not update the kubelet on these machines because you must update the OpenShift API on them first. Because the specification for the new version is applied to the old kubelet, the RHEL machine cannot return to the **Ready** state. You cannot complete the update until the machines are available. However, the maximum number of nodes that are unavailable is set to ensure that normal cluster operations are likely to continue with that number of machines out of service.

**Additional resources**

- Support policy for unmanaged Operators

## 3.3. OPENSHIFT CONTAINER PLATFORM UPGRADE CHANNELS AND RELEASES

In OpenShift Container Platform 4.1, Red Hat introduced the concept of channels for recommending the appropriate release versions for cluster upgrade. By controlling the pace of upgrades, these upgrade channels allow you to choose an upgrade strategy. Upgrade channels are tied to a minor version of OpenShift Container Platform. For instance, OpenShift Container Platform 4.3 upgrade channels will never include an upgrade to a 4.4 release. This strategy ensures that administrators explicitly decide to upgrade to the next minor version of OpenShift Container Platform. Upgrade channels control only release selection and do not impact the version of the cluster that you install; the **openshift-install** binary file for a specific version of OpenShift Container Platform always installs that version.

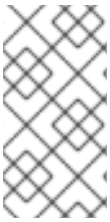
OpenShift Container Platform 4.3 offers the following upgrade channels:

- **candidate-4.3**
- **fast-4.3**
- **stable-4.3**

### candidate-4.3 channel

The **candidate-4.3** channel contains candidate builds for a z-stream (4.3.z) release. Release candidates contain all the features of the product but are not supported. Use release candidate versions to test feature acceptance and assist in qualifying the next version of OpenShift Container Platform. A release candidate is any build that is available in the candidate channel, including ones that do not contain **-rc** in their names. After a version is available in the candidate channel, it goes through more quality checks. If it meets the quality standard, it is promoted to the **fast-4.3** or **stable-4.3** channels. Because of this strategy, if a specific release is available in both the **candidate-4.3** channel and in the **fast-4.3** or **stable-4.3** channels, it is a Red Hat supported version. The **candidate-4.3** channel can include release versions from which there are no recommended updates in any channel.

You can use the **candidate-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.



NOTE

Release candidates differ from the nightly builds. Nightly builds are available for early access to features, but updating to or from nightly builds is neither recommended nor supported. Nightly builds are not available in any upgrade channel. You can reference the OpenShift Container Platform release statuses for more build information.

### fast-4.3 channel

The **fast-4.3** channel is updated with new 4.3 versions as soon as Red Hat declares the given version as a general availability release. As such, these releases are fully supported, are production quality, and have performed well while available as a release candidate in the **candidate-4.3** channel from where they were promoted. Some time after a release appears in the **fast-4.3** channel, it is added to the **stable-4.3** channel. Releases never appear in the **stable-4.3** channel before they appear in the **fast-4.3** channel.

You can use the **fast-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

### stable-4.3 channel

While the **fast-4.3** channel contains releases as soon as their errata are published, releases are added to the **stable-4.3** channel after a delay. During this delay, data is collected from Red Hat SRE teams, Red Hat support services, and pre-production and production environments that participate in connected customer program about the stability of the release.

You can use the **stable-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

### Upgrade version paths

OpenShift Container Platform maintains an upgrade recommendation service that understands the version of OpenShift Container Platform you have installed as well as the path to take within the channel you choose to get you to the next release. You can imagine seeing the following in the **fast-4.3** channel:

- 4.3.0
- 4.3.1
- 4.3.3
- 4.3.4

The service recommends only upgrades that have been tested and have no serious issues. If your cluster is on 4.3.1 and OpenShift Container Platform suggests 4.3.4, then it is safe for you to update from 4.3.1 to 4.3.4. Do not rely on consecutive patch numbers. In this example, 4.3.2 is not, and never was, available in the channel. The update service will not suggest updating to a version of OpenShift Container Platform that contains known vulnerabilities.

Update stability depends on your channel. The presence of an update recommendation in the **candidate-4.3** channel does not imply that the update is supported. It means that no serious issues have been found with the update yet, but there might not be significant traffic through the update to suggest stability. The presence of an update recommendation in the **fast-4.3** or **stable-4.3** channels is a declaration that the update is fully supported while it is in the channel. While releases will never be removed from a channel, update recommendations that exhibit serious issues will be removed from all channels. Updates initiated after the update recommendation has been removed might not be supported.

Red Hat will eventually provide supported update paths from any supported release in the **fast-4.3** or **stable-4.3** channels to the latest release in 4.3.z, although there can be delays while safe paths away from troubled releases are constructed and verified.

### Fast and stable channel use and strategies

The **fast-4.3** and **stable-4.3** channels present a choice between receiving general availability releases as soon as they are available or allowing Red Hat to control the rollout of those updates. If issues are detected during rollout or at a later time, upgrades to that version might be blocked in both the **fast-4.3** and **stable-4.3** channels, and a new version might be introduced that becomes the new preferred upgrade target.

Customers can improve this process by configuring pre-production systems on the **fast-4.3** channel, configuring production systems on the **stable-4.3** channel, and participating in Red Hat's connected customer program. Red Hat uses this program to observe the impact of updates on your specific hardware and software configurations. Future releases might improve or alter the pace at which updates move from the **fast-4.3** to the **stable-4.3** channel.

### Restricted network clusters

If you manage the container images for your OpenShift Container Platform clusters yourself, you must consult the Red Hat errata that is associated with product releases and note any comments that impact upgrades. During upgrade, the user interface might warn you about switching between these versions, so you must ensure that you selected an appropriate version before you bypass those warnings.

### Switching between channels

Your cluster is still supported if you change from the **stable-4.3** channel to the **fast-4.3** channel. Although you can switch to the **candidate-4.3** channel at any time, some releases in that channel might be unsupported release candidates. You can switch from the **candidate-4.3** channel to the **fast-4.3** channel if your current release is a general availability release. You can always switch from the **fast-4.3** channel to the **stable-4.3** channel, although if the current release was recently promoted to **fast-4.3** there can be a delay of up to a day for the release to be promoted to **stable-4.3**. If you change to a channel that does not include your current release, an alert displays and no updates can be recommended, but you can safely change back to your original channel at any point.

## 3.4. UPDATING A CLUSTER BY USING THE CLI

If updates are available, you can update your cluster by using the OpenShift CLI (**oc**).

You can find information about available OpenShift Container Platform advisories and updates in the errata section of the Customer Portal.

### Prerequisites

- Install the OpenShift CLI (**oc**) that matches the version for your updated version.
- Log in to the cluster as user with **cluster-admin** privileges.
- Install the **jq** package.

### Procedure

1. Ensure that your cluster is available:

```
$ oc get clusterversion

NAME     VERSION   AVAILABLE   PROGRESSING   SINCE   STATUS
```

```
version   4.3.0     True        False         158m    Cluster version is 4.3.0
```

2. Review the current update channel information and confirm that your channel is set to **stable-4.3**:

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

{
  "channel": "stable-4.3",
  "clusterID": "990f7ab8-109b-4c95-8480-2bd1deec55ff",
  "upstream": "https://api.openshift.com/api/upgrades_info/v1/graph"
}
```

**IMPORTANT**

For production clusters, you must subscribe to a **stable-*** channel.

3. View the available updates and note the version number of the update that you want to apply:

```
$ oc adm upgrade

Cluster version is 4.1.0

Updates:

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

4. Apply an update:

- To update to the latest version:

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

- To update to a specific version:

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

(1) **<version>** is the update version that you obtained from the output of the previous command.

5. Review the status of the Cluster Version Operator:

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

{
  "channel": "stable-4.3",
  "clusterID": "990f7ab8-109b-4c95-8480-2bd1deec55ff",
  "desiredUpdate": {
    "force": false,
    "image": "quay.io/openshift-release-dev/ocp-
```

```
release@sha256:9c5f0df8b192a0d7b46cd5f6a4da2289c155fd5302dec7954f8f06c878160b8b",

    "version": "4.3.1" (1)
  },
  "upstream": "https://api.openshift.com/api/upgrades_info/v1/graph"
}
```

(1) If the **version** number in the **desiredUpdate** stanza matches the value that you specified, the update is in progress.

6. Review the cluster version status history to monitor the status of the update. It might take some time for all the objects to finish updating.

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

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

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

The history contains a list of the most recent versions applied to the cluster. This value is updated when the CVO applies an update. The list is ordered by date, where the newest update is first in the list. Updates in the history have state **Completed** if the rollout completed and **Partial** if the update failed or did not complete.

**IMPORTANT**

If an upgrade fails, the Operator stops and reports the status of the failing component. Rolling your cluster back to a previous version is not supported. If your upgrade fails, contact Red Hat support.

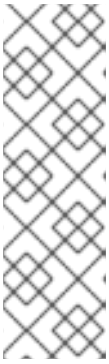
7. After the update completes, you can confirm that the cluster version has updated to the new version:

```
$ oc get clusterversion
```

```
NAME      VERSION     AVAILABLE   PROGRESSING   SINCE     STATUS
version   4.1.2       True        False         2m        Cluster version is 4.1.2
```

8. If you are upgrading to this release from OpenShift Container Platform 4.3.3 or earlier, you must restart all Pods after the upgrade is complete. You can do this using the following command:

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



NOTE

Restarting all Pods is required because the service CA is automatically rotated as of OpenShift Container Platform 4.3.5. The service CA is rotated during the upgrade and a restart is required afterward to ensure that all services are using the new service CA before the previous service CA expires.

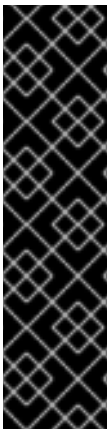
After this one-time manual restart, subsequent upgrades and rotations will ensure restart before the service CA expires without requiring manual intervention.

# CHAPTER 4. UPDATING A CLUSTER THAT INCLUDES RHEL COMPUTE MACHINES

You can update, or upgrade, an OpenShift Container Platform cluster. If your cluster contains Red Hat Enterprise Linux (RHEL) machines, you must perform more steps to update those machines.

## 4.1. PREREQUISITES

- Have access to the cluster as a user with **admin** privileges. See Using RBAC to define and apply permissions.
- Have a recent etcd backup in case your upgrade fails and you must restore your cluster to a previous state.



**IMPORTANT**

If you are upgrading to this release from OpenShift Container Platform 4.2, you must restart all Pods after the upgrade is complete.

This is because the service CA is automatically rotated as of OpenShift Container Platform 4.3.5. The service CA is rotated during the upgrade and a restart is required afterward to ensure that all services are using the new service CA before the previous service CA expires.

After this one-time manual restart, subsequent upgrades and rotations will ensure restart before the service CA expires without requiring manual intervention.

## 4.2. ABOUT THE OPENSHIFT CONTAINER PLATFORM UPDATE SERVICE

The OpenShift Container Platform update service is the hosted service that provides over-the-air updates to both OpenShift Container Platform and Red Hat Enterprise Linux CoreOS (RHCOS). It provides a graph, or diagram that contain *vertices* and the *edges* that connect them, of component Operators. The edges in the graph show which versions you can safely update to, and the vertices are update payloads that specify the intended state of the managed cluster components.

The Cluster Version Operator (CVO) in your cluster checks with the OpenShift Container Platform update service to see the valid updates and update paths based on current component versions and information in the graph. When you request an update, the OpenShift Container Platform CVO uses the release image for that update to upgrade your cluster. The release artifacts are hosted in Quay as container images.

To allow the OpenShift Container Platform update service to provide only compatible updates, a release verification pipeline exists to drive automation. Each release artifact is verified for compatibility with supported cloud platforms and system architectures as well as other component packages. After the pipeline confirms the suitability of a release, the OpenShift Container Platform update service notifies you that it is available.

**IMPORTANT**

Because the update service displays all valid updates, you must not force an update to a version that the update service does not display.

During continuous update mode, two controllers run. One continuously updates the payload manifests, applies them to the cluster, and outputs the status of the controlled rollout of the Operators, whether they are available, upgrading, or failed. The second controller polls the OpenShift Container Platform update service to determine if updates are available.

IMPORTANT

Reverting your cluster to a previous version, or a rollback, is not supported. Only upgrading to a newer version is supported.

During the upgrade process, the Machine Config Operator (MCO) applies the new configuration to your cluster machines. It cordons the number of nodes that is specified by the **maxUnavailable** field on the machine configuration pool and marks them as unavailable. By default, this value is set to **1**. It then applies the new configuration and reboots the machine. If you use Red Hat Enterprise Linux (RHEL) machines as workers, the MCO does not update the kubelet on these machines because you must update the OpenShift API on them first. Because the specification for the new version is applied to the old kubelet, the RHEL machine cannot return to the **Ready** state. You cannot complete the update until the machines are available. However, the maximum number of nodes that are unavailable is set to ensure that normal cluster operations are likely to continue with that number of machines out of service.

**Additional resources**

- Support policy for unmanaged Operators

## 4.3. OPENSHIFT CONTAINER PLATFORM UPGRADE CHANNELS AND RELEASES

In OpenShift Container Platform 4.1, Red Hat introduced the concept of channels for recommending the appropriate release versions for cluster upgrade. By controlling the pace of upgrades, these upgrade channels allow you to choose an upgrade strategy. Upgrade channels are tied to a minor version of OpenShift Container Platform. For instance, OpenShift Container Platform 4.3 upgrade channels will never include an upgrade to a 4.4 release. This strategy ensures that administrators explicitly decide to upgrade to the next minor version of OpenShift Container Platform. Upgrade channels control only release selection and do not impact the version of the cluster that you install; the **openshift-install** binary file for a specific version of OpenShift Container Platform always installs that version.

OpenShift Container Platform 4.3 offers the following upgrade channels:

- **candidate-4.3**
- **fast-4.3**
- **stable-4.3**

### candidate-4.3 channel

The **candidate-4.3** channel contains candidate builds for a z-stream (4.3.z) release. Release candidates contain all the features of the product but are not supported. Use release candidate versions to test feature acceptance and assist in qualifying the next version of OpenShift Container Platform. A release candidate is any build that is available in the candidate channel, including ones that do not contain **-rc** in their names. After a version is available in the candidate channel, it goes through more quality checks. If it meets the quality standard, it is promoted to the **fast-4.3** or **stable-4.3** channels. Because of this strategy, if a specific release is available in both the **candidate-4.3** channel and in the **fast-4.3** or **stable-4.3** channels, it is a Red Hat supported version. The **candidate-4.3** channel can include release versions from which there are no recommended updates in any channel.

You can use the **candidate-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

NOTE

Release candidates differ from the nightly builds. Nightly builds are available for early access to features, but updating to or from nightly builds is neither recommended nor supported. Nightly builds are not available in any upgrade channel. You can reference the OpenShift Container Platform release statuses for more build information.

### fast-4.3 channel

The **fast-4.3** channel is updated with new 4.3 versions as soon as Red Hat declares the given version as a general availability release. As such, these releases are fully supported, are production quality, and have performed well while available as a release candidate in the **candidate-4.3** channel from where they were promoted. Some time after a release appears in the **fast-4.3** channel, it is added to the **stable-4.3** channel. Releases never appear in the **stable-4.3** channel before they appear in the **fast-4.3** channel.

You can use the **fast-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

### stable-4.3 channel

While the **fast-4.3** channel contains releases as soon as their errata are published, releases are added to the **stable-4.3** channel after a delay. During this delay, data is collected from Red Hat SRE teams, Red Hat support services, and pre-production and production environments that participate in connected customer program about the stability of the release.

You can use the **stable-4.3** channel to upgrade from a previous minor version of OpenShift Container Platform.

### Upgrade version paths

OpenShift Container Platform maintains an upgrade recommendation service that understands the version of OpenShift Container Platform you have installed as well as the path to take within the channel you choose to get you to the next release. You can imagine seeing the following in the **fast-4.3** channel:

- 4.3.0
- 4.3.1
- 4.3.3
- 4.3.4

The service recommends only upgrades that have been tested and have no serious issues. If your cluster is on 4.3.1 and OpenShift Container Platform suggests 4.3.4, then it is safe for you to update from 4.3.1 to 4.3.4. Do not rely on consecutive patch numbers. In this example, 4.3.2 is not, and never was, available in the channel. The update service will not suggest updating to a version of OpenShift Container Platform that contains known vulnerabilities.

Update stability depends on your channel. The presence of an update recommendation in the **candidate-4.3** channel does not imply that the update is supported. It means that no serious issues have been found with the update yet, but there might not be significant traffic through the update to suggest stability. The presence of an update recommendation in the **fast-4.3** or **stable-4.3** channels is a declaration that the update is fully supported while it is in the channel. While releases will never be removed from a channel, update recommendations that exhibit serious issues will be removed from all channels. Updates initiated after the update recommendation has been removed might not be supported.

Red Hat will eventually provide supported update paths from any supported release in the **fast-4.3** or **stable-4.3** channels to the latest release in 4.3.z, although there can be delays while safe paths away from troubled releases are constructed and verified.

### Fast and stable channel use and strategies

The **fast-4.3** and **stable-4.3** channels present a choice between receiving general availability releases as soon as they are available or allowing Red Hat to control the rollout of those updates. If issues are detected during rollout or at a later time, upgrades to that version might be blocked in both the **fast-4.3** and **stable-4.3** channels, and a new version might be introduced that becomes the new preferred upgrade target.

Customers can improve this process by configuring pre-production systems on the **fast-4.3** channel, configuring production systems on the **stable-4.3** channel, and participating in Red Hat's connected customer program. Red Hat uses this program to observe the impact of updates on your specific hardware and software configurations. Future releases might improve or alter the pace at which updates move from the **fast-4.3** to the **stable-4.3** channel.

### Restricted network clusters

If you manage the container images for your OpenShift Container Platform clusters yourself, you must consult the Red Hat errata that is associated with product releases and note any comments that impact upgrades. During upgrade, the user interface might warn you about switching between these versions, so you must ensure that you selected an appropriate version before you bypass those warnings.

### Switching between channels

Your cluster is still supported if you change from the **stable-4.3** channel to the **fast-4.3** channel. Although you can switch to the **candidate-4.3** channel at any time, some releases in that channel might be unsupported release candidates. You can switch from the **candidate-4.3** channel to the **fast-4.3** channel if your current release is a general availability release. You can always switch from the **fast-4.3** channel to the **stable-4.3** channel, although if the current release was recently promoted to **fast-4.3** there can be a delay of up to a day for the release to be promoted to **stable-4.3**. If you change to a channel that does not include your current release, an alert displays and no updates can be recommended, but you can safely change back to your original channel at any point.

## 4.4. UPDATING A CLUSTER BY USING THE WEB CONSOLE

If updates are available, you can update your cluster from the web console.

You can find information about available OpenShift Container Platform advisories and updates in the errata section of the Customer Portal.

### Prerequisites

- Have access to the web console as a user with **admin** privileges.

### Procedure

1. From the web console, click **Administration** > **Cluster Settings** and review the contents of the **Overview** tab.

2. For production clusters, ensure that the **CHANNEL** is set to the correct channel for the version that you want to update to, your current minor version, such as **stable-4.3**.

**IMPORTANT**

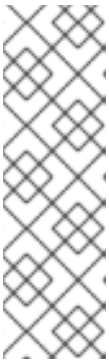
For production clusters, you must subscribe to a stable-* or fast-* channel.

- If the **UPDATE STATUS** is not **Updates Available**, you cannot upgrade your cluster.
- The **DESIRED VERSION** indicates the cluster version that your cluster is running or is updating to.

3. Click **Updates Available**, select a version to update to, the highest available version and click **Update**. The **UPDATE STATUS** changes to **Updating**, and you can review the progress of the Operator upgrades on the **Cluster Operators** tab.

4. If you are upgrading to this release from OpenShift Container Platform 4.3.3 or earlier, 4.2, you must restart all Pods after the upgrade is complete. You can do this using the following command, which requires the OpenShift CLI (**oc**):

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



NOTE

Restarting all Pods is required because the service CA is automatically rotated as of OpenShift Container Platform 4.3.5. 4.3.5. The service CA is rotated during the upgrade and a restart is required afterward to ensure that all services are using the new service CA before the previous service CA expires.

After this one-time manual restart, subsequent upgrades and rotations will ensure restart before the service CA expires without requiring manual intervention.

5. After the update completes and the Cluster Version Operator refreshes the available updates, check if more updates are available in your current channel.
   - If updates are available, continue to perform updates in the current channel until you can no longer update.
   - If no updates are available, change the **CHANNEL** to the stable-* or fast-* channel for the next minor version, and update to the version that you want in that channel.

   You might need to perform several intermediate updates until you reach the version that you want.

NOTE

When you update a cluster that contains Red Hat Enterprise Linux (RHEL) worker machines, those workers temporarily become unavailable during the update process. You must run the upgrade playbook against each RHEL machine as it enters the **NotReady** state for the cluster to finish updating.

## 4.5. (OPTIONAL) ADDING HOOKS TO PERFORM ANSIBLE TASKS ON RHEL MACHINES

You can use *hooks* to run Ansible tasks on the RHEL compute machines during the OpenShift Container Platform update.

### 4.5.1. About Ansible hooks for upgrades

When you update OpenShift Container Platform, you can run custom tasks on your Red Hat Enterprise Linux (RHEL) nodes during specific operations by using *hooks*. Hooks allow you to provide files that define tasks to run before or after specific update tasks. You can use hooks to validate or modify custom infrastructure when you update the RHEL compute nodes in you OpenShift Container Platform cluster.

Because when a hook fails, the operation fails, you must design hooks that are idempotent, or can run multiple times and provide the same results.

Hooks have the following important limitations: - Hooks do not have a defined or versioned interface. They can use internal **openshift-ansible** variables, but it is possible that the variables will be modified or removed in future OpenShift Container Platform releases. - Hooks do not have error handling, so an error in a hook halts the update process. If you get an error, you must address the problem and then start the upgrade again.

### 4.5.2. Configuring the Ansible inventory file to use hooks

You define the hooks to use when you update the Red Hat Enterprise Linux (RHEL) compute machines, which are also known as worker machines, in the **hosts** inventory file under the **all:vars** section.

#### Prerequisites

- You have access to the machine that you used to add the RHEL compute machines cluster. You must have access to the **hosts** Ansible inventory file that defines your RHEL machines.

#### Procedure

1. After you design the hook, create a YAML file that defines the Ansible tasks for it. This file must be a set of tasks and cannot be a playbook, as shown in the following example:

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

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

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

2. Modify the **hosts** Ansible inventory file to specify the hook files. The hook files are specified as parameter values in the **[all:vars]** section, as shown:

   **Example hook definitions in an inventory file**

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

   To avoid ambiguity in the paths to the hook, use absolute paths instead of a relative paths in their definitions.

### 4.5.3. Available hooks for RHEL compute machines

You can use the following hooks when you update the Red Hat Enterprise Linux (RHEL) compute machines in your OpenShift Container Platform cluster.

| Hook name | Description |
|---|---|
| **openshift_node_pre_cordon_hook** | • Runs **before** each node is cordoned.<br>• This hook runs against **each node** in serial.<br>• If a task must run against a different host, the task must use **delegate_to** or **local_action**. |
| **openshift_node_pre_upgrade_hook** | • Runs **after** each node is cordoned but **before** it is updated.<br>• This hook runs against **each node** in serial.<br>• If a task must run against a different host, the task must use **delegate_to** or **local_action**. |
| **openshift_node_pre_uncordon_hook** | • Runs **after** each node is updated but **before** it is uncordoned.<br>• This hook runs against **each node** in serial.<br>• If a task must run against a different host, they task must use **delegate_to** or **local_action**. |
| **openshift_node_post_upgrade_hook** | • Runs **after** each node uncordoned. It is the **last** node update action.<br>• This hook runs against **each node** in serial.<br>• If a task must run against a different host, the task must use **delegate_to** or **local_action**. |

## 4.6. UPDATING RHEL COMPUTE MACHINES IN YOUR CLUSTER

After you update your cluster, you must update the Red Hat Enterprise Linux (RHEL) compute machines in your cluster.

**Prerequisites**

- You updated your cluster.

IMPORTANT

Because the RHEL machines require assets that are generated by the cluster to complete the update process, you must update the cluster before you update the RHEL compute machines in it.

- You have access to the machine that you used to add the RHEL compute machines cluster. You must have access to the **hosts** Ansible inventory file that defines your RHEL machines and the **upgrade** playbook.

**Procedure**

1. Stop and disable firewalld on the host:

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

NOTE

You must not enable firewalld later. If you do, you cannot access OpenShift Container Platform logs on the worker.

2. Enable the repositories that are required for OpenShift Container Platform 4.3:
   a. On the machine that you run the Ansible playbooks, update the required repositories:

```
# subscription-manager repos --disable=rhel-7-server-ansible-2.7-rpms  \
                           --disable=rhel-7-server-ose-4.2-rpms \
                           --enable=rhel-7-server-ansible-2.8-rpms \
                           --enable=rhel-7-server-ose-4.3-rpms
```

   b. On the machine that you run the Ansible playbooks, update the required packages, including **openshift-ansible**:

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

   c. On each RHEL compute node, update the required repositories:

```
# subscription-manager repos --disable=rhel-7-server-ose-4.2-rpms \
                           --enable=rhel-7-server-ose-4.3-rpms
```

3. Update a RHEL worker machine:
   a. Review the current node status to determine which RHEL worker to update:

```
# oc get node
NAME                    STATUS                     ROLES    AGE    VERSION
mycluster-control-plane-0   Ready                      master   145m   v1.16.2
mycluster-control-plane-1   Ready                      master   145m   v1.16.2
mycluster-control-plane-2   Ready                      master   145m   v1.16.2
mycluster-rhel7-0           NotReady,SchedulingDisabled   worker   98m
v1.14.6+97c81d00e
```

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

Note which machine has the **NotReady,SchedulingDisabled** status.

b. Review your Ansible inventory file at **/<path>/inventory/hosts** and update its contents so that only the machine with the **NotReady,SchedulingDisabled** status is listed in the **[workers]** section, as shown in the following example:

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

openshift_kubeconfig_path="~/.kube/config"

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

c. Change to the **openshift-ansible** directory and run the **upgrade** playbook:

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

1 For **<path>**, specify the path to the Ansible inventory file that you created.

4. Follow the process in the previous step to update each RHEL worker machine in your cluster.

5. After you update all of the workers, confirm that all of your cluster nodes have updated to the new version:

```
# oc get node
NAME                      STATUS                     ROLES    AGE    VERSION
mycluster-control-plane-0   Ready                      master   145m   v1.16.2
mycluster-control-plane-1   Ready                      master   145m   v1.16.2
mycluster-control-plane-2   Ready                      master   145m   v1.16.2
mycluster-rhel7-0         NotReady,SchedulingDisabled   worker   98m    v1.16.2
mycluster-rhel7-1         Ready                      worker   98m    v1.16.2
mycluster-rhel7-2         Ready                      worker   98m    v1.16.2
mycluster-rhel7-3         Ready                      worker   98m    v1.16.2
```

# CHAPTER 5. UPDATING A RESTRICTED NETWORK CLUSTER

You can upgrade a restricted network OpenShift Container Platform cluster by using the **oc** command-line interface (CLI).

A restricted network environment is the one in which your cluster nodes cannot access the internet. For this reason, you must populate a registry with the installation images. If your registry host cannot access both the internet and the cluster, you can mirror the images to a file system that disconnected from that environment and then bring that host or removable media across that gap. If the local container registry is connected to the mirror registry's host, you can directly push the release images to the local registry.

If multiple clusters are present within the restricted network, mirror the required release images to a single container image registry and use that registry to update all the clusters.

**Prerequisites**

- Have access to the internet to obtain the necessary container images.
- Have write access to a container registry in the restricted-network environment to push and pull images. The container registry must be compatible with Docker registry API v2.
- You must have the **oc** command-line interface (CLI) tool installed.
- Have access to the cluster as a user with **admin** privileges. See Using RBAC to define and apply permissions.
- Have a recent etcd backup in case your upgrade fails and you must restore your cluster to a previous state.

## 5.1. 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.

### 5.1.1. Installing the CLI by downloading the binary

You can install the OpenShift CLI (**oc**) in order to interact with OpenShift Container Platform from a command-line interface. You can install **oc** on Linux, Windows, or macOS.

**IMPORTANT**

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.3. Download and install the new version of **oc**. If you are upgrading a cluster in a restricted network, install the **oc** version that you plan to upgrade to.

#### 5.1.1.1. Installing the CLI on Linux

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

**Procedure**

1. Navigate to the Infrastructure Provider page on the Red Hat OpenShift Cluster Manager site.

2. Select your infrastructure provider, and, if applicable, your installation type.
3. In the **Command-line interface** section, select **Linux** from the drop-down menu and click **Download command-line tools**.
4. Unpack the archive:

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

5. Place the **oc** binary in a directory that is on your **PATH**.
   To check your **PATH**, execute the following command:

```
$ echo $PATH
```

After you install the CLI, it is available using the **oc** command:

```
$ oc <command>
```

#### 5.1.1.2. Installing the CLI on Windows

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

**Procedure**

1. Navigate to the Infrastructure Provider page on the Red Hat OpenShift Cluster Manager site.
2. Select your infrastructure provider, and, if applicable, your installation type.
3. In the **Command-line interface** section, select **Windows** from the drop-down menu and click **Download command-line tools**.
4. Unzip the archive with a ZIP program.
5. 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
```

After you install the CLI, it is available using the **oc** command:

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

#### 5.1.1.3. Installing the CLI on macOS

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

**Procedure**

1. Navigate to the Infrastructure Provider page on the Red Hat OpenShift Cluster Manager site.
2. Select your infrastructure provider, and, if applicable, your installation type.

3. In the **Command-line interface** section, select **MacOS** from the drop-down menu and click **Download command-line tools**.
4. Unpack and unzip the archive.
5. Move the **oc** binary to a directory on your PATH.
   To check your **PATH**, open a terminal and execute the following command:

```
$ echo $PATH
```

After you install the CLI, it is available using the **oc** command:

```
$ oc <command>
```

## 5.2. CONFIGURING CREDENTIALS THAT ALLOW IMAGES TO BE MIRRORED

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

**WARNING**

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.

**WARNING**

This process requires that you have write access to a container image registry on the mirror registry and adds the credentials to a registry pull secret.

**IMPORTANT**

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 restricted network.
- 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.

### Procedure

Complete the following steps on the installation host:

1. Download your **registry.redhat.io** pull secret from the Pull Secret page on the Red Hat OpenShift Cluster Manager site and save it to a **.json** file.

2. Generate the base64-encoded user name and password or token for your mirror registry:

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

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

3. Make a copy of your pull secret in JSON format:

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

1 Specify the path to the folder to store the pull secret in and a name for the JSON file that you create.

The contents of the file resemble the following example:

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

4. Edit the new file and add a section that describes your registry to it:

```
  "auths": {
    "<mirror_registry>": { 1
      "auth": "<credentials>", 2
```

```
    "email": "you@example.com"
  },
```

1. For **<mirror_registry>**, specify the registry domain name, and optionally the port, that your mirror registry uses to serve content. For example, **registry.example.com** or **registry.example.com:5000**

2. For **<credentials>**, specify the base64-encoded user name and password for the mirror registry.

The file resembles the following example:

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

1. Edit the new file and add a section that describes your registry to it:

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

1. For **<mirror_registry>**, specify the registry domain name, and optionally the port, that your mirror registry uses to serve content. For example, **registry.example.com** or **registry.example.com:5000**

2. For **<credentials>**, specify the base64-encoded user name and password for the mirror registry.

The file resembles the following example:

```
{
  "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"
    },
    "<mirror_registry>": {
      "auth": "<credentials>",
      "email": "you@example.com"
    },
    "registry.redhat.io": {
      "auth": "NTE3Njg5Nj...",
      "email": "you@example.com"
    }
  }
}
```

## 5.3. MIRRORING THE OPENSHIFT CONTAINER PLATFORM IMAGE REPOSITORY

Before you upgrade a cluster on infrastructure that you provision in a restricted network, you must mirror the required container images into that environment. You can also use this procedure in unrestricted networks to ensure your clusters only use container images that have satisfied your organizational controls on external content.

### Procedure

1. Review the OpenShift Container Platform upgrade paths to confirm that there is an upgrade path between your current and intended cluster versions.

2. Set the required environment variables:

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

1 For **<release_version>**, specify the tag that corresponds to the version of OpenShift Container Platform to which you want to upgrade, such as **4.5.0**.

2. 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.
3. For **<repository_name>**, specify the name of the repository to create in your registry, such as **ocp4/openshift4**.
4. The repository to mirror. For a production release, you must specify **openshift-release-dev**.
5. For **<path_to_pull_secret>**, specify the absolute path to and file name of the pull secret for your mirror registry that you created.
6. For a production release, you must specify **ocp-release**.
7. For **<server_architecture>**, specify the architecture of the server, such as **x86_64**.
8. For **<path>**, specify the path to the directory to host the mirrored images.

3. Review the images and configuration manifests to mirror:

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

4. Mirror the version images to the internal container registry.
   - If your mirror host does not have internet access, take the following actions:
     i. Connect the removable media to a system that is connected to the internet.
     ii. Mirror the images and configuration manifests 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}-
${ARCHITECTURE}
```

     iii. 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}
```

   - If the local container registry is connected to the mirror host, directly push the release images to the local registry:

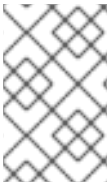
```
$ oc adm release mirror -a ${LOCAL_SECRET_JSON} --
from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-
${ARCHITECTURE} \
  --to=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}
```

## 5.4. CREATING THE IMAGE CONFIGMAP

Before you update your cluster, you must manually create a ConfigMap that contains the signatures of the release images that you use. This signature allows the Cluster Version Operator (CVO) to verify that the release images have not been modified by comparing the expected and actual image signatures.

### 5.4.1. Creating the ConfigMap for image signature verification by using the oc CLI

Before you update your cluster, you must manually create a ConfigMap that contains the signatures of the release images that you use. This signature allows the Cluster Version Operator (CVO) to verify that the release images have not been modified by comparing the expected and actual image signatures.



NOTE

If you are upgrading from a release prior to version 4.4.8, you must use the manual method for creating the ConfigMap instead of this procedure. The commands that this procedure uses are not in earlier versions of the **oc** command-line interface (CLI).

**Prerequisites**

- Install the OpenShift Command-line Interface (CLI), commonly known as **oc**, version 4.4.8 or later.

**Procedure**

1. Obtain the image signature for the version that you are upgrading to from either mirror.openshift.com or Google Cloud Storage (GCS) .
2. Use **oc** command-line interface (CLI) to log into the cluster that you are upgrading.
3. Apply the mirrored release image signature ConfigMap to the connected cluster:

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

1 For **<image_signature_file>**, specify the path and name of the file, for example, **mirror/config/signature-sha256-81154f5c03294534.yaml**.

### 5.4.2. Creating an image signature ConfigMap manually

Create and apply the image signature ConfigMap to the cluster that you want to update.

NOTE

You must perform following steps each time that you update a cluster.

**Procedure**

1. Review the OpenShift Container Platform upgrade paths knowledge base article to determine a valid upgrade path for your cluster.
2. Add the version to the **OCP_RELEASE_NUMBER** environment variable:

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

1 For **<release_version>**, specify the tag that corresponds to the version of OpenShift Container Platform you want to update the cluster, such as **4.4.0**.

3. Add the system architecture for your cluster to **ARCHITECTURE** environment variable:

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

For **server_architecture**, specify the architecture of the server, such as **x86_64**.

4. Get the release image digest from Quay:

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

5. Set the digest algorithm:

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

6. Set the digest signature:

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

7. Get the image signature from mirror.openshift.com website.

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

8. Create the ConfigMap:

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

9. Apply the ConfigMap to the cluster to update:

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

## 5.5. UPGRADING THE RESTRICTED NETWORK CLUSTER

Update the restricted network cluster to the OpenShift Container Platform version that you downloaded the release images for.

**Prerequisites**

- You mirrored the images for the new release to your registry.
- You applied the release image signature ConfigMap for the new release to your cluster.
- You obtained the sha256 sum value for the release from the image signature ConfigMap.
- Install the OpenShift Command-line Interface (CLI), commonly known as **oc**, version 4.4.8 or later.

**Procedure**

- Update the cluster:

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

1 The **<sha256_sum_value>** value is the sha256 sum value for the release from the image signature ConfigMap, for example, **@sha256:81154f5c03294534e1eaf0319bef7a601134f891689ccede5d705ef659aa8c92**

If you use an **ImageContentSourcePolicy** for the mirror registry, you can use the canonical registry name instead of **LOCAL_REGISTRY**.

## 5.6. CONFIGURING IMAGE REGISTRY REPOSITORY MIRRORING

Setting up container registry repository mirroring lets you:

- Configure your OpenShift Container Platform cluster to redirect requests to pull images from a repository on a source image registry and have it resolved by a repository on a mirrored image registry.
- Identify multiple mirrored repositories for each target repository, to make sure that if one mirror is down, another can be used.

Here are some of the attributes of repository mirroring in OpenShift Container Platform:

- Image pulls are resilient to registry downtimes
- Clusters in restricted networks can request to pull images from critical locations (such as quay.io) and have registries behind a company firewall provide the requested images.
- A particular order of registries is tried when an image pull request is made, with the permanent registry typically being the last one tried.
- The mirror information you enter is added to the **/etc/containers/registries.conf** file on every node in the OpenShift Container Platform cluster.
- When a node makes a request for an image from the source repository, it tries each mirrored repository in turn until it finds the requested content. If all mirrors fail, the cluster tries the source repository. Upon success, the image is pulled to the node.

Setting up repository mirroring can be done in the following ways:

- At OpenShift Container Platform installation time: By pulling container images needed by OpenShift Container Platform and then bringing those images behind your company's firewall, you can install OpenShift Container Platform into a datacenter that is in a restricted network. See Mirroring the OpenShift Container Platform image repository for details.

- After OpenShift Container Platform installation time: Even if you don't configure mirroring during OpenShift Container Platform installation, you can do so later using the **ImageContentSourcePolicy** object.

The following procedure provides a post-installation mirror configuration, where you create an **ImageContentSourcePolicy** object that identifies:

- The source of the container image repository you want to mirror

- A separate entry for each mirror repository you want to offer the content requested from the source repository.

### Prerequisites

- Access to the cluster as a user with the **cluster-admin** role.

### Procedure

1. Configure mirrored repositories. To do that, you can either:

   - Set up a mirrored repository with Red Hat Quay, as described in Red Hat Quay Repository Mirroring. Using Red Hat Quay allows you to copy images from one repository to another and also automatically sync those repositories repeatedly over time.

   - Use a tool such as **skopeo** to copy images manually from the source directory to the mirrored repository.
     For example, after installing the skopeo RPM package on a Red Hat Enterprise Linux (RHEL 7 or RHEL 8) system, use the **skopeo** command as shown in this example:

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

     In this example, you have a container image registry that is named **example.io** with an image repository named **example** to which you want to copy the **ubi8/ubi-minimal** image from **registry.access.redhat.com**. After you create the registry, you can configure your OpenShift Container Platform cluster to redirect requests made of the source repository to the mirrored repository.

2. Log in to your OpenShift Container Platform cluster.

3. Create an **ImageContentSourcePolicy** file (for example, **registryrepomirror.yaml**), replacing the source and mirrors with those of your own registry and repository pairs and images:

   ```
   apiVersion: operator.openshift.io/v1alpha1
   kind: ImageContentSourcePolicy
   metadata:
   ```

```
  name: ubi8repo
spec:
  repositoryDigestMirrors:
  - mirrors:
    - example.io/example/ubi-minimal (1)
    source: registry.access.redhat.com/ubi8/ubi-minimal (2)
  - mirrors:
    - example.com/example/ubi-minimal
    source: registry.access.redhat.com/ubi8/ubi-minimal
```

(1) Indicates the name of the image registry and repository

(2) Indicates the registry and repository containing the content that is mirrored

4. Create the new **ImageContentSourcePolicy**:

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

After the **ImageContentSourcePolicy** is created, the new settings are deployed to each node and shortly start using the mirrored repository for requests to the source repository.

5. To check that the mirrored configuration worked, go to one of your nodes. For example:

   a. List your nodes:

```
$ oc get node
```

**Example output**

```
NAME                           STATUS                     ROLES    AGE  VERSION
ip-10-0-137-44.ec2.internal    Ready                      worker   7m   v1.16.2
ip-10-0-138-148.ec2.internal   Ready                      master   11m  v1.16.2
ip-10-0-139-122.ec2.internal   Ready                      master   11m  v1.16.2
ip-10-0-147-35.ec2.internal    Ready,SchedulingDisabled   worker   7m   v1.16.2
ip-10-0-153-12.ec2.internal    Ready                      worker   7m   v1.16.2
ip-10-0-154-10.ec2.internal    Ready                      master   11m  v1.16.2
```

You can see that scheduling on each worker node is disabled as the change is being applied.

   b. Start the debugging process:

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

**Example output**

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

   c. Access the node's files:

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

d. Check the **/etc/containers/registries.conf** file to make sure the changes were made:

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

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

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

e. Pull an image digest to the node from the source and check if it is actually resolved by the mirror. The **ImageContentSourcePolicy** supports image digests only, not image tags.

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

### Troubleshooting repository mirroring

If the repository mirroring procedure does not work as described, use the following information about how repository mirroring works to help troubleshoot the problem.

- The first working mirror is used to supply the pulled image.
- The main registry will only be used if no other mirror works.
- From the system context, the **Insecure** flags are used as fallback.
- The format of the **/etc/containers/registries** file has changed recently. It is now version 2 and in TOML format.