



OpenShift Container Platform 4.2

Updating clusters

Updating OpenShift Container Platform 4.2 clusters

OpenShift Container Platform 4.2 Updating clusters

Updating OpenShift Container Platform 4.2 clusters

Legal Notice

Copyright © 2019 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 4.2 clusters. In version 4.2, 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	3
1.1. ABOUT THE OPENSIFT CONTAINER PLATFORM UPDATE SERVICE	3
1.2. UNDERSTANDING OPENSIFT CONTAINER PLATFORM UPGRADE CHANNELS	4
1.3. UPDATING A CLUSTER BY USING THE WEB CONSOLE	4
CHAPTER 2. UPDATING A CLUSTER WITHIN A MINOR VERSION FROM THE WEB CONSOLE	6
2.1. ABOUT THE OPENSIFT CONTAINER PLATFORM UPDATE SERVICE	6
2.2. UNDERSTANDING OPENSIFT CONTAINER PLATFORM UPGRADE CHANNELS	6
2.3. UPDATING A CLUSTER BY USING THE WEB CONSOLE	7
CHAPTER 3. UPDATING A CLUSTER WITHIN A MINOR VERSION BY USING THE CLI	8
3.1. ABOUT THE OPENSIFT CONTAINER PLATFORM UPDATE SERVICE	8
3.2. UNDERSTANDING OPENSIFT CONTAINER PLATFORM UPGRADE CHANNELS	8
3.3. UPDATING A CLUSTER BY USING THE CLI	9
CHAPTER 4. UPDATING A CLUSTER THAT INCLUDES RHEL COMPUTE MACHINES	12
4.1. ABOUT THE OPENSIFT CONTAINER PLATFORM UPDATE SERVICE	12
4.2. UNDERSTANDING OPENSIFT CONTAINER PLATFORM UPGRADE CHANNELS	12
4.3. UPDATING A CLUSTER BY USING THE WEB CONSOLE	13
4.4. (OPTIONAL) ADDING HOOKS TO PERFORM ANSIBLE TASKS ON RHEL MACHINES	14
4.4.1. About Ansible hooks for upgrades	14
4.4.2. Configuring the Ansible inventory file to use hooks	14
4.4.3. Available hooks for RHEL compute machines	15
4.5. UPDATING RHEL COMPUTE MACHINES IN YOUR CLUSTER	15

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 4.2 channel.

Prerequisites

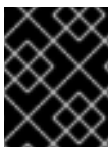
- Access to the cluster as a user with **admin** privileges. See [Using RBAC to define and apply permissions](#).

1.1. ABOUT THE OPENSIFT 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.

1.2. UNDERSTANDING OPENSIFT CONTAINER PLATFORM UPGRADE CHANNELS

In OpenShift Container Platform 4.1, Red Hat introduced the concept of upgrade channels for recommending the appropriate upgrade versions to your cluster. Upgrade channels separate upgrade strategies and also are used to control the cadence of updates. Channels are tied to a minor version of OpenShift Container Platform. For instance, OpenShift Container Platform 4.2 channels will never include an upgrade to a 4.3 release. This ensures administrators make an explicit decision to upgrade to the next minor version of OpenShift Container Platform. Channels only control updates and have no impact on the version of the cluster you install; the **openshift-install** binary for a given patch level of OpenShift Container Platform always installs that patch level.

See [OpenShift 4.2 Upgrades phased roll out](#) for more information on the types of updates and upgrade channels.

1.3. UPDATING A CLUSTER BY USING THE WEB CONSOLE

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

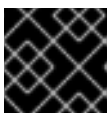
You can find information about available OpenShift Container Platform advisories and updates [in the errata section](#) of the Customer Portal.

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-4.2**.



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. 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.

Prerequisites

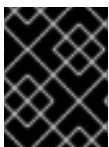
- Access to the cluster as a user with **admin** privileges. See [Using RBAC to define and apply permissions](#).

2.1. ABOUT THE OPENSIFT 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.

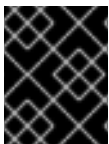
To allow the OpenShift Container Platform update service to provide only compatible updates, a release verification pipeline exists to drive automation. Each release artifact is verified for compatibility with supported cloud platforms and system architectures as well as other component packages. After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

2.2. UNDERSTANDING OPENSIFT CONTAINER PLATFORM UPGRADE CHANNELS

In OpenShift Container Platform 4.1, Red Hat introduced the concept of upgrade channels for recommending the appropriate upgrade versions to your cluster. Upgrade channels separate upgrade strategies and also are used to control the cadence of updates. Channels are tied to a minor version of

OpenShift Container Platform. For instance, OpenShift Container Platform 4.2 channels will never include an upgrade to a 4.3 release. This ensures administrators make an explicit decision to upgrade to the next minor version of OpenShift Container Platform. Channels only control updates and have no impact on the version of the cluster you install; the **openshift-install** binary for a given patch level of OpenShift Container Platform always installs that patch level.

See [OpenShift 4.2 Upgrades phased roll out](#) for more information on the types of updates and upgrade channels.

2.3. UPDATING A CLUSTER BY USING THE WEB CONSOLE

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

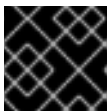
You can find information about available OpenShift Container Platform advisories and updates [in the errata section](#) of the Customer Portal.

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-4.2**.



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. 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. ABOUT THE OPENSIFT 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.

3.2. UNDERSTANDING OPENSIFT CONTAINER PLATFORM UPGRADE CHANNELS

In OpenShift Container Platform 4.1, Red Hat introduced the concept of upgrade channels for recommending the appropriate upgrade versions to your cluster. Upgrade channels separate upgrade strategies and also are used to control the cadence of updates. Channels are tied to a minor version of OpenShift Container Platform. For instance, OpenShift Container Platform 4.2 channels will never include an upgrade to a 4.3 release. This ensures administrators make an explicit decision to upgrade to

the next minor version of OpenShift Container Platform. Channels only control updates and have no impact on the version of the cluster you install; the **openshift-install** binary for a given patch level of OpenShift Container Platform always installs that patch level.

See [OpenShift 4.2 Upgrades phased roll out](#) for more information on the types of updates and upgrade channels.

3.3. 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 version of the OpenShift Command-line Interface (CLI), commonly known as **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:

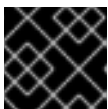
```
$ oc get clusterversion
```

NAME	VERSION	AVAILABLE	PROGRESSING	SINCE	STATUS
version	4.2.0	True	False	158m	Cluster version is 4.2.0

2. Review the current update channel information and confirm that your channel is set to **stable-4.2**:

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

```
{
  "channel": "stable-4.2",
  "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 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-4.2",
  "clusterID": "990f7ab8-109b-4c95-8480-2bd1deec55ff",
  "desiredUpdate": {
    "force": false,
    "image": "quay.io/openshift-release-dev/ocp-
release@sha256:9c5f0df8b192a0d7b46cd5f6a4da2289c155fd5302dec7954f8f06c878160b8b",
    "version": "4.2.1" 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.

- 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

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.

Prerequisites

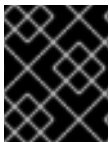
- Access to the cluster as a user with **admin** privileges. See [Using RBAC to define and apply permissions](#).

4.1. ABOUT THE OPENSIFT 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.

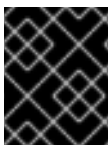
To allow the OpenShift Container Platform update service to provide only compatible updates, a release verification pipeline exists to drive automation. Each release artifact is verified for compatibility with supported cloud platforms and system architectures as well as other component packages. After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

4.2. UNDERSTANDING OPENSIFT CONTAINER PLATFORM UPGRADE CHANNELS

In OpenShift Container Platform 4.1, Red Hat introduced the concept of upgrade channels for recommending the appropriate upgrade versions to your cluster. Upgrade channels separate upgrade

strategies and also are used to control the cadence of updates. Channels are tied to a minor version of OpenShift Container Platform. For instance, OpenShift Container Platform 4.2 channels will never include an upgrade to a 4.3 release. This ensures administrators make an explicit decision to upgrade to the next minor version of OpenShift Container Platform. Channels only control updates and have no impact on the version of the cluster you install; the **openshift-install** binary for a given patch level of OpenShift Container Platform always installs that patch level.

See [OpenShift 4.2 Upgrades phased roll out](#) for more information on the types of updates and upgrade channels.

4.3. UPDATING A CLUSTER BY USING THE WEB CONSOLE

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

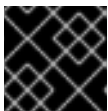
You can find information about available OpenShift Container Platform advisories and updates [in the errata section](#) of the Customer Portal.

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-4.2**.



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. 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.

4.4. (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.

4.4.1. 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 your 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.

4.4.2. Configuring the Ansible inventory file to use hooks

You define the hooks to use when you update the Red Hat Enterprise Linux (RHEL) compute, or 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.

4.4.3. 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.

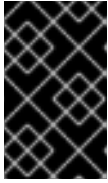
Hook name	Description
openshift_node_pre_cordon_hook	<ul style="list-style-type: none"> Runs before each node is cordoned. This hook runs against each node in serial. If a task must run against a different host, the task must use delegate_to or local_action.
openshift_node_pre_upgrade_hook	<ul style="list-style-type: none"> Runs after each node is cordoned but before it is updated. This hook runs against each node in serial. If a task must run against a different host, the task must use delegate_to or local_action.
openshift_node_pre_uncordon_hook	<ul style="list-style-type: none"> Runs after each node is updated but before it is uncordoned. This hook runs against each node in serial. If a task must run against a different host, they task must use delegate_to or local_action.
openshift_node_post_upgrade_hook	<ul style="list-style-type: none"> Runs after each node uncordoned. It is the last node update action. This hook runs against each node in serial. If a task must run against a different host, the task must use delegate_to or local_action.

4.5. 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. Review your Ansible inventory file at `/<path>/inventory/hosts` and ensure that all of your compute, or worker, machines are 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-worker-0.example.com
mycluster-worker-1.example.com
mycluster-worker-2.example.com
mycluster-worker-3.example.com
```

If all of your RHEL compute machines are not listed in the **[workers]** section, you must move them to that section.

2. 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.