Updating a cluster using the web console

TABLE OF CONTENTS

You can update, or upgrade, an OpenShift Container Platform cluster by using the web console. The following steps update a cluster within a minor version. You can use the same instructions for updating a cluster between minor versions.

> **NOTEa**
> Use the web console or `oc adm upgrade channel <channel>` to change the update channel. You can follow the steps in Updating a cluster using the CLI to complete the update after you change to a 4.9 channel.

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 update fails and you must restore your cluster to a previous state.

  OpenShift Container Platform 4.9 requires an update from etcd version 3.4 to 3.5. If the etcd Operator halts the update, an alert is triggered. To clear this alert, cancel the update with the following command:

  ```
  $ oc adm upgrade --clear
  ```

- Ensure all Operators previously installed through Operator Lifecycle Manager (OLM) are updated to their latest version in their latest channel. Updating the Operators ensures they have a valid update path when the default OperatorHub catalogs switch from the current minor version to the next during a cluster update. See Upgrading installed Operators for more information.

- Ensure that all machine config pools (MCPs) are running and not paused. Nodes associated with a paused MCP are skipped during the update process. You can pause the MCPs if you are performing a canary rollout update strategy.

- If your cluster uses manually maintained credentials, ensure that the Cloud Credential Operator (CCO) is in an upgradeable state. For more information, see *Upgrading clusters with manually maintained credentials* for AWS, Azure, or GCP.

- If your cluster uses manually maintained credentials with the AWS Secure Token Service (STS), obtain a copy of the `ccoctl` utility from the release image being updated to and use it to process any updated credentials. For more information, see *Upgrading an OpenShift Container Platform cluster configured for manual mode with STS*.

- Review the list of APIs that were removed in Kubernetes 1.22, migrate any affected components to use the new API version, and provide the administrator acknowledgment. For more information, see Preparing to update to OpenShift Container Platform 4.9.

> **IMPORTANTa**
>
> - When an update is failing to complete, the Cluster Version Operator (CVO) reports the status of any blocking components while attempting to reconcile the update. Rolling your cluster back to a previous version is not supported. If your update is failing to complete, contact Red Hat support.
>
> - Using the `unsupportedConfigOverrides` section to modify the configuration of an Operator is unsupported and might block cluster updates. You must remove this setting before you can update your cluster.

Additional resources

- Support policy for unmanaged Operators

Performing a canary rollout update

In some specific use cases, you might want a more controlled update process where you do not want specific nodes updated concurrently with the rest of the cluster. These use cases include, but are not limited to:

- You have mission-critical applications that you do not want unavailable during the update. You can slowly test the applications on your nodes in small batches after the update.
- You have a small maintenance window that does not allow the time for all nodes to be updated, or you have multiple maintenance windows.

The rolling update process is **not** a typical update workflow. With larger clusters, it can be a time-consuming process that requires you execute multiple commands. This complexity can result in errors that can affect the entire cluster. It is recommended that you carefully consider whether your organization wants to use a rolling update and carefully plan the implementation of the process before you start.

The rolling update process described in this topic involves:

- Creating one or more custom machine config pools (MCPs).
- Labeling each node that you do not want to update immediately to move those nodes to the custom MCPs.
- Pausing those custom MCPs, which prevents updates to those nodes.
- Performing the cluster update.
- Unpausing one custom MCP, which triggers the update on those nodes.
- Testing the applications on those nodes to make sure the applications work as expected on those newly-updated nodes.
- Optionally removing the custom labels from the remaining nodes in small batches and testing the applications on those nodes.

> **NOTEa**
>
> Pausing an MCP prevents the Machine Config Operator from applying any configuration changes on the associated nodes. Pausing an MCP also prevents any automatically-rotated certificates from being pushed to the associated nodes, including the automatic CA rotation of the `kube-apiserver-to-kubelet-signer` CA certificate. If the MCP is paused when the `kube-apiserver-to-kubelet-signer` CA certificate expires and the MCO attempts to automatically renew the certificate, the new certificate is created but not applied across the nodes in the respective machine config pool. This causes failure in multiple `oc` commands, including but not limited to `oc debug`, `oc logs`, `oc exec`, and `oc attach`.
>
> ```
>                                                              kube-apiserver-to-
> kubelet-signer
> ```

If you want to use the canary rollout update process, see Performing a canary rollout update

If you want to use the canary rollout update process, see Performing a canary rollout update.

Pausing a MachineHealthCheck resource

During the upgrade process, nodes in the cluster might become temporarily unavailable. In the case of worker nodes, the machine health check might identify such nodes as unhealthy and reboot them. To avoid rebooting such nodes, pause all the `MachineHealthCheck` resources before updating the cluster.

Prerequisites

- Install the OpenShift CLI (`oc`).

Procedure

1. To list all the available `MachineHealthCheck` resources that you want to pause, run the following command:

```
$ oc get machinehealthcheck -n openshift-machine-api
```

2. To pause the machine health checks, add the `cluster.x-k8s.io/paused=""` annotation to the `MachineHealthCheck` resource. Run the following command:

```
$ oc -n openshift-machine-api annotate mhc <mhc-name> cluster.x-k8s.io/paused="
```

The annotated `MachineHealthCheck` resource resembles the following YAML file:

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineHealthCheck
metadata:
  name: example
  namespace: openshift-machine-api
  annotations:
    cluster.x-k8s.io/paused: ""
spec:
  selector:
    matchLabels:
      role: worker
  unhealthyConditions:
  - type:    "Ready"
    status:  "Unknown"
    timeout: "300s"
  - type:    "Ready"
    status:  "False"
    timeout: "300s"
  maxUnhealthy: "40%"
```

```
status:
  currentHealthy: 5
  expectedMachines: 5
```

IMPORTANTa

Resume the machine health checks after updating the cluster. To resume the check, remove the pause annotation from the `MachineHealthCheck` resource by running the following command:

```
$ oc -n openshift-machine-api annotate mhc <mhc-name> cluster.x-k8
```

About updating single node OpenShift Container Platform

You can update, or upgrade, a single-node OpenShift Container Platform cluster by using either the console or CLI.

However, note the following limitations:

- The prerequisite to pause the `MachineHealthCheck` resources is not required because there is no other node to perform the health check.
- Restoring a single-node OpenShift Container Platform cluster using an etcd backup is not officially supported. However, it is good practice to perform the etcd backup in case your upgrade fails. If your control plane is healthy, you might be able to restore your cluster to a previous state by using the backup.
- Updating a single-node OpenShift Container Platform cluster requires downtime and can include an automatic reboot. The amount of downtime depends on the update payload, as described in the following scenarios:
    - If the update payload contains an operating system update, which requires a reboot, the downtime is significant and impacts cluster management and user workloads.
    - If the update contains machine configuration changes that do not require a reboot, the downtime is less, and the impact on the cluster management and user workloads is lessened. In this case, the node draining step is skipped with single-node OpenShift Container Platform because there is no other node in the cluster to reschedule the workloads to.

    - If the update payload does not contain an operating system update or machine configuration changes, a short API outage occurs and resolves quickly.

> **IMPORTANTa**
> There are conditions, such as bugs in an updated package, that can cause the single node to not restart after a reboot. In this case, the update does not rollback automatically.

Additional resources

- For information on which machine configuration changes require a reboot, see the note in Understanding the Machine Config Operator.

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.
- Pause all `MachineHealthCheck` resources.

Procedure

1. From the web console, click **Administration** → **Cluster Settings** and review the contents of the **Details** tab.
2. For production clusters, ensure that the **Channel** is set to the correct channel for the version that you want to update to, such as `stable-4.9`.

    > **IMPORTANTa**
    > 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.
    - **Select channel** indicates the cluster version that your cluster is running or is updating to.
3. Select a version to update to and click **Save**.

    The Input channel **Update status** changes to **Update to <product-version> in progress**, and you can review the progress of the cluster update by watching the progress bars for the Operators and nodes.

progress bars for the Operators and nodes.

> **NOTEa**
>
> If you are upgrading your cluster to the next minor version, like version 4.y to 4.(y+1), it is recommended to confirm your nodes are upgraded before deploying workloads that rely on a new feature. Any pools with worker nodes that are not yet updated are displayed on the **Cluster Settings** page.

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.

Changing the update server by using the web console

Changing the update server is optional. If you have an OpenShift Update Service (OSUS) installed and configured locally, you must set the URL for the server as the `upstream` to use the local server during updates.

Procedure

1. Navigate to **Administration** → **Cluster Settings**, click **version**.

2. Click the **YAML** tab and then edit the `upstream` parameter value:

   Example output

```
...
spec:
  clusterID: db93436d-7b05-42cc-b856-43e11ad2d31a
  upstream: '<update-server-url>' 1
...
```

   1 The `<update-server-url>` variable specifies the URL for the update server.

   The default `upstream` is `https://api.openshift.com/api/upgrades_info/v1/graph` `https://api.openshift.com/api/upgrades_info/v1/graph`.

3. Click **Save**.