Upgrading OpenShift Dedicated
Abstract

This document provides information about upgrading OpenShift Dedicated clusters.
Table of Contents

CHAPTER 1. PREPARING TO UPGRADE OPENSIFT DEDICATED TO 4.9 .......................... 3
  1.1. ADMINISTRATOR ACKNOWLEDGMENT WHEN UPGRADING TO OPENSIFT 4.9 3
  1.2. REMOVED KUBERNETES APIS 3
  1.3. EVALUATING YOUR CLUSTER FOR REMOVED APIS 4
    1.3.1. Reviewing alerts to identify uses of removed APIs 4
    1.3.2. Using APIRequestCount to identify uses of removed APIs 4
    1.3.3. Using APIRequestCount to identify which workloads are using the removed APIs 5
  1.4. MIGRATING INSTANCES OF REMOVED APIS 6

CHAPTER 2. OPENSIFT DEDICATED CLUSTER UPGRADES ..................................... 7
  2.1. UNDERSTANDING OPENSIFT DEDICATED CLUSTER UPGRADES 7
    2.1.1. Automatic upgrades 7
    2.1.2. Manual upgrades 7
    2.1.3. Upgrade notifications 7
  2.2. AUTOMATICALLY UPGRADING YOUR CLUSTER THROUGH RED HAT OPENSIFT CLUSTER MANAGER 8
  2.3. MANUALLY UPGRADING YOUR CLUSTER THROUGH RED HAT OPENSIFT CLUSTER MANAGER 8
CHAPTER 1. PREPARING TO UPGRADE OPENSHIFT DEDICATED TO 4.9

Upgrading your OpenShift Dedicated clusters to OpenShift 4.9 requires you to evaluate and migrate your APIs as the latest version of Kubernetes has removed a significant number of APIs.

Before you can upgrade your OpenShift Dedicated clusters, you must update the required tools to the appropriate version.

1.1. ADMINISTRATOR ACKNOWLEDGMENT WHEN UPGRADING TO OPENSHIFT 4.9

OpenShift Dedicated 4.9 uses Kubernetes 1.22, which removed a significant number of deprecated v1beta1 APIs.

OpenShift Dedicated 4.8.14 introduced a requirement that an administrator must provide a manual acknowledgment before the cluster can be upgraded from OpenShift Dedicated 4.8 to 4.9. This is to help prevent issues after upgrading to OpenShift Dedicated 4.9, where APIs that have been removed are still in use by workloads, tools, or other components running on or interacting with the cluster. Administrators must evaluate their cluster for any APIs in use that will be removed and migrate the affected components to use the appropriate new API version. After this is done, the administrator can provide the administrator acknowledgment.

All OpenShift Dedicated 4.8 clusters require this administrator acknowledgment before they can be upgraded to OpenShift Dedicated 4.9.

1.2. REMOVED KUBERNETES APIS

OpenShift Dedicated 4.9 uses Kubernetes 1.22, which removed the following deprecated v1beta1 APIs. You must migrate manifests and API clients to use the v1 API version. For more information about migrating removed APIs, see the Kubernetes documentation.

Table 1.1. v1beta1 APIs removed from Kubernetes 1.22

<table>
<thead>
<tr>
<th>Resource</th>
<th>API</th>
<th>Notable changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>APIService</td>
<td>apiregistration.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>CertificateSigningRequest</td>
<td>certificates.k8s.io/v1beta1</td>
<td>Yes</td>
</tr>
<tr>
<td>ClusterRole</td>
<td>rbac.authorization.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>ClusterRoleBinding</td>
<td>rbac.authorization.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>CSI Driver</td>
<td>storage.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>CSI Node</td>
<td>storage.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>CustomResourceDefinition</td>
<td>apiextensions.k8s.io/v1beta1</td>
<td>Yes</td>
</tr>
<tr>
<td>Resource</td>
<td>API</td>
<td>Notable changes</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Ingress</td>
<td>extensions/v1beta1</td>
<td>Yes</td>
</tr>
<tr>
<td>Ingress</td>
<td>networking.k8s.io/v1beta1</td>
<td>Yes</td>
</tr>
<tr>
<td>IngressClass</td>
<td>networking.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>Lease</td>
<td>coordination.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>LocalSubjectAccessReview</td>
<td>authorization.k8s.io/v1beta1</td>
<td>Yes</td>
</tr>
<tr>
<td>MutatingWebhookConfiguration</td>
<td>admissionregistration.k8s.io/v1beta1</td>
<td>Yes</td>
</tr>
<tr>
<td>PriorityClass</td>
<td>scheduling.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>Role</td>
<td>rbac.authorization.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>RoleBinding</td>
<td>rbac.authorization.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>SelfSubjectAccessReview</td>
<td>authorization.k8s.io/v1beta1</td>
<td>Yes</td>
</tr>
<tr>
<td>StorageClass</td>
<td>storage.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>SubjectAccessReview</td>
<td>authorization.k8s.io/v1beta1</td>
<td>Yes</td>
</tr>
<tr>
<td>TokenReview</td>
<td>authentication.k8s.io/v1beta1</td>
<td>No</td>
</tr>
<tr>
<td>ValidatingWebhookConfiguration</td>
<td>admissionregistration.k8s.io/v1beta1</td>
<td>Yes</td>
</tr>
<tr>
<td>VolumeAttachment</td>
<td>storage.k8s.io/v1beta1</td>
<td>No</td>
</tr>
</tbody>
</table>

1.3. EVALUATING YOUR CLUSTER FOR REMOVED APIs

There are several methods to help administrators identify where APIs that will be removed are in use. However, OpenShift Dedicated cannot identify all instances, especially workloads that are idle or external tools that are used. It is the responsibility of the administrator to properly evaluate all workloads and other integrations for instances of removed APIs.

1.3.1. Reviewing alerts to identify uses of removed APIs

The **APIRemovedInNextReleaseInUse** alert tells you that there are removed APIs in use on your cluster. If this alert is firing in your cluster, review the alert; take action to clear the alert by migrating manifests and API clients to use the new API version. You can use the **APIRequestCount** API to get more information about which APIs are in use and which workloads are using removed APIs.

1.3.2. Using APIRequestCount to identify uses of removed APIs
You can use the `APIServerCount` API to track API requests and review if any of them are using one of the removed APIs.

**Prerequisites**

- You must have access to the cluster as a user with the `cluster-admin` role.

**Procedure**

- Run the following command and examine the `REMOVEDINRELEASE` column of the output to identify the removed APIs that are currently in use:

  ```bash
  $ oc get apirequestcounts
  ```

**Example output**

<table>
<thead>
<tr>
<th>NAME</th>
<th>REMOVEDINRELEASE</th>
<th>REQUESTSINCURRENTHOUR</th>
<th>REQUESTSINLAST24H</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloudcredentials.v1.operator.openshift.io</td>
<td>32</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>ingresses.v1.networking.k8s.io</td>
<td>28</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>ingresses.v1beta1.extensions</td>
<td>1.22</td>
<td>16</td>
<td>66</td>
</tr>
<tr>
<td>ingresses.v1beta1.networking.k8s.io</td>
<td>1.22</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>installplans.v1alpha1.operators.coreos.com</td>
<td>93</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE**

You can safely ignore the following entries that appear in the results:

- `system:serviceaccount:kube-system:generic-garbage-collector` appears in the results because it walks through all registered APIs searching for resources to remove.

- `system:kube-controller-manager` appears in the results because it walks through all resources to count them while enforcing quotas.

You can also use `-o jsonpath` to filter the results:

```bash
$ oc get apirequestcounts -o jsonpath="{range .items[?(@.status.removedInRelease!="")]}{.status.removedInRelease}{{"\t"}}{.metadata.name}{{"\n"}}{end}
```

**Example output**

```
1.22 certificatesigningrequests.v1beta1.certificates.k8s.io
1.22 ingresses.v1beta1.extensions
1.22 ingresses.v1beta1.networking.k8s.io
```

**1.3.3. Using APIServerCount to identify which workloads are using the removed APIs**

You can examine the `APIServerCount` resource for a given API version to help identify which workloads are using the API.
Prerequisites

- You must have access to the cluster as a user with the `cluster-admin` role.

Procedure

- Run the following command and examine the `username` and `userAgent` fields to help identify the workloads that are using the API:

  ```
  $ oc get apirequestcounts <resource>.<version>.<group> -o yaml
  
  For example:
  
  $ oc get apirequestcounts ingresses.v1beta1.networking.k8s.io -o yaml
  
  You can also use `-o jsonpath` to extract the `username` values from an `APIRequestCount` resource:
  
  ```
  $ oc get apirequestcounts ingresses.v1beta1.networking.k8s.io -o jsonpath='{range ..username}{"$"}{\"n\"}{end}' | sort | uniq
  ```

  Example output

  ```
  user1
  user2
  app:serviceaccount:delta
  ```

1.4. MIGRATING INSTANCES OF REMOVED APIS

For information on how to migrate removed Kubernetes APIs, see the [Deprecated API Migration Guide](#) in the Kubernetes documentation.


CHAPTER 2. OPENSHEET DEDICATED CLUSTER UPGRADES

You can schedule automatic or manual upgrade policies to update the version of your OpenShift Dedicated clusters. Upgrading OpenShift Dedicated clusters can be done through Red Hat OpenShift Cluster Manager or OpenShift Cluster Manager CLI.

2.1. UNDERSTANDING OPENSHEET DEDICATED CLUSTER UPGRADES

When upgrades are made available for your OpenShift Dedicated cluster, you can upgrade to the newest version through Red Hat OpenShift Cluster Manager or OpenShift Cluster Manager CLI. You can set your upgrade policies on existing clusters or during cluster creation, and upgrades can be scheduled to occur automatically or manually.

Red Hat Site Reliability Engineers (SRE) will provide a curated list of available versions for your OpenShift Dedicated clusters. For each cluster you will be able to review the full list of available releases, as well as the corresponding release notes. OpenShift Cluster Manager will enable installation of clusters at the latest supported versions, and upgrades can be canceled at any time.

You can also set a grace period for how long PodDisruptionBudget protected workloads are respected during upgrades. After this grace period, any workloads protected by PodDisruptionBudget that have not been successfully drained from a node, will be forcibly deleted.

![NOTE]

All Kubernetes objects and PVs in each OpenShift Dedicated cluster are backed up as part of the OpenShift Dedicated service. Application and application data backups are not a part of the OpenShift Dedicated service. Ensure you have a backup policy in place for your applications and application data prior to scheduling upgrades.

2.1.1. Automatic upgrades

Upgrades can be scheduled to occur automatically on a day and time specified by the cluster owner or administrator. Upgrades occur on a weekly basis, unless an upgrade is unavailable for that week.

![NOTE]

Automatic upgrade policies are optional and if they are not set, the upgrade policies default to manual.

2.1.2. Manual upgrades

If you opt for manual upgrades, you are responsible for updating your cluster. If your cluster version falls too far behind, it will transition to a limited support status. For more information on OpenShift life cycle policies, see OpenShift Dedicated update life cycle.

2.1.3. Upgrade notifications

From OpenShift Cluster Manager console you can view your cluster’s history from the Overview tab. The Upgrade states can be viewed in the service log under the Cluster history heading.

Every change of state also triggers an email notification to the cluster owner and subscribed users. You will receive email notifications for the following events:

- An upgrade has been scheduled.
• An upgrade has started.
• An upgrade has completed.
• An upgrade has been canceled.

NOTE
For automatic upgrades, you will also receive email notifications before the upgrade occurs based on the following cadence:

• 2 week notice
• 1 week notice
• 1 day notice

2.2. AUTOMATICALLY UPGRADING YOUR CLUSTER THROUGH RED HAT OPENSHIFT CLUSTER MANAGER

You can use OpenShift Cluster Manager to automatically upgrade your OpenShift Dedicated cluster on a weekly basis. Based on upstream changes, there might be times when no updates are released. Therefore, no upgrade occurs for that week.

Procedure

1. From {console-red-hat}, select your cluster from the clusters list.
2. Click the Upgrade settings tab to access the upgrade operator.
3. To schedule automatic upgrades, select Automatic.
4. Specify the day of the week and the time you want your cluster to upgrade.
5. Click Save.
6. Optional: Set a grace period for Node draining by selecting a designated amount of time from the drop down list. A 1 hour grace period is set by default.
7. To edit an existing automatic upgrade policy, edit the preferred day or start time from the Upgrade Settings tab. Click Save.
8. To cancel an automatic upgrade policy, switch the upgrade method to manual from the Upgrade Settings tab. Click Save.

On the Upgrade settings tab, the Upgrade status box indicates that an upgrade is scheduled. The date and time of the next scheduled update is listed.

2.3. MANUALLY UPGRADING YOUR CLUSTER THROUGH RED HAT OPENSHIFT CLUSTER MANAGER

You can use OpenShift Cluster Manager to manually upgrade your OpenShift Dedicated cluster.

Procedure
1. From [console-red-hat], select your cluster from the clusters list.

2. Click the Upgrade settings tab to access the upgrade operator. You can also update your cluster from the Overview tab by clicking Update next to the cluster version under the Details heading.

3. Click Update in the Update Status box.

4. Select the version you want to upgrade your cluster to. Recommended cluster upgrades will be noted in the UI. To learn more about each available upgrade version, click View release notes.

5. Click Next.

6. To schedule your upgrade:
   - Click Upgrade now to upgrade within the next hour.
   - Click Schedule a different time and specify the date and time that you want the cluster to upgrade.

7. Click Next.

8. Review the upgrade policy and click Confirm upgrade.

9. A confirmation appears when the cluster upgrade has been scheduled. Click Close.

10. Optional: Set a grace period for Node draining by selecting a designated amount of time from the drop down list. A 1 hour grace period is set by default.

From the Overview tab, next to the cluster version, the UI notates that the upgrade has been scheduled. Click View details to view the upgrade details. If you need to cancel the scheduled upgrade, you can click Cancel this upgrade from the View Details pop-up.

The same upgrade details are available on the Upgrade settings tab under the Upgrade status box. If you need to cancel the scheduled upgrade, you can click Cancel this upgrade from the Upgrade status box.

---

**WARNING**

In the event that a CVE or other critical issue to OpenShift Dedicated is found, all clusters are upgraded within 48 hours of the fix being released. You are notified when the fix is available and informed that the cluster will be automatically upgraded at your latest preferred start time before the 48 hour window closes. You can also upgrade manually at any time before the automatic upgrade starts.