Red Hat OpenShift Data Foundation 4.15

Replacing nodes

Instructions for how to safely replace a node in an OpenShift Data Foundation cluster.

Last Updated: 2024-03-18

Red Hat OpenShift Data Foundation 4.15 Replacing nodes

Instructions for how to safely replace a node in an OpenShift Data Foundation cluster.

Legal Notice

Copyright © 2024 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 explains how to safely replace a node in a Red Hat OpenShift Data Foundation cluster.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 4

PREFACE 5

CHAPTER 1. OPENSHIFT DATA FOUNDATION DEPLOYED USING DYNAMIC DEVICES 6
1.1. OPENSHIFT DATA FOUNDATION DEPLOYED ON AWS 6
1.1.1. Replacing an operational AWS node on user-provisioned infrastructure 6
1.1.2. Replacing an operational AWS node on installer-provisioned infrastructure 8
1.1.3. Replacing a failed AWS node on user-provisioned infrastructure 10
1.1.4. Replacing a failed AWS node on installer-provisioned infrastructure 11
1.2. OPENSHIFT DATA FOUNDATION DEPLOYED ON VMWARE 13
1.2.1. Replacing an operational VMware node on user-provisioned infrastructure 13
1.2.2. Replacing an operational VMware node on installer-provisioned infrastructure 15
1.2.3. Replacing a failed VMware node on user-provisioned infrastructure 17
1.2.4. Replacing a failed VMware node on installer-provisioned infrastructure 19
1.3. OPENSHIFT DATA FOUNDATION DEPLOYED ON MICROSOFT AZURE 20
1.3.1. Replacing operational nodes on Azure installer-provisioned infrastructure 20
1.3.2. Replacing failed nodes on Azure installer-provisioned infrastructure 22
1.4. OPENSHIFT DATA FOUNDATION DEPLOYED ON GOOGLE CLOUD 24
1.4.1. Replacing operational nodes on Google Cloud installer-provisioned infrastructure 24
1.4.2. Replacing failed nodes on Google Cloud installer-provisioned infrastructure 25

CHAPTER 2. OPENSHIFT DATA FOUNDATION DEPLOYED USING LOCAL STORAGE DEVICES 28
2.1. REPLACING STORAGE NODES ON BARE METAL INFRASTRUCTURE 28
2.1.1. Replacing an operational node on bare metal user-provisioned infrastructure 28
2.1.2. Replacing a failed node on bare metal user-provisioned infrastructure 34
2.2. REPLACING STORAGE NODES ON IBM Z OR IBM® LINUXONE INFRASTRUCTURE 40
2.2.1. Replacing operational nodes on IBM Z or IBM® LinuxONE infrastructure 40
2.2.2. Replacing failed nodes on IBM Z or IBM® LinuxONE infrastructure 45
2.3. REPLACING STORAGE NODES ON IBM POWER INFRASTRUCTURE 46
2.3.1. Replacing an operational or failed storage node on IBM Power 46
2.4. REPLACING STORAGE NODES ON VMWARE INFRASTRUCTURE 52
2.4.1. Replacing an operational node on VMware user-provisioned infrastructure 53
2.4.2. Replacing an operational node on VMware installer-provisioned infrastructure 58
2.4.3. Replacing a failed node on VMware user-provisioned infrastructure 64
2.4.4. Replacing a failed node on VMware installer-provisioned infrastructure 70

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Do let us know how we can make it better.

To give feedback, create a Bugzilla ticket:

1. Go to the Bugzilla website.
2. In the **Component** section, choose **documentation**.
3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
4. Click **Submit Bug**.

PREFACE

For OpenShift Data Foundation, node replacement can be performed proactively for an operational node and reactively for a failed node for the following deployments:

- For Amazon Web Services (AWS)
  - User-provisioned infrastructure
  - Installer-provisioned infrastructure
- For VMware
  - User-provisioned infrastructure
  - Installer-provisioned infrastructure
- For Microsoft Azure
  - Installer-provisioned infrastructure
- For local storage devices
  - Bare metal
  - VMware
  - IBM Power
- For replacing your storage nodes in external mode, see Red Hat Ceph Storage documentation .

CHAPTER 1. OPENSHIFT DATA FOUNDATION DEPLOYED USING DYNAMIC DEVICES

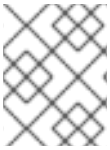
1.1. OPENSHIFT DATA FOUNDATION DEPLOYED ON AWS

- To replace an operational node, see:
  - Section 1.1.1, “Replacing an operational AWS node on user-provisioned infrastructure” .
  - Section 1.1.2, “Replacing an operational AWS node on installer-provisioned infrastructure” .
- To replace a failed node, see:
  - Section 1.1.3, “Replacing a failed AWS node on user-provisioned infrastructure” .
  - Section 1.1.4, “Replacing a failed AWS node on installer-provisioned infrastructure” .

1.1.1. Replacing an operational AWS node on user-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.



NOTE

When replacing an AWS node on user-provisioned infrastructure, the new node needs to be created in the same AWS zone as the original node.

Procedure

1. Identify the node that you need to replace.
2. Mark the node as unschedulable:

```
$ oc adm cordon <node_name>
```

<node_name>

Specify the name of node that you need to replace.

3. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

IMPORTANT

This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.

4. Delete the node:

```
$ oc delete nodes <node_name>
```

5. Create a new Amazon Web Service (AWS) machine instance with the required infrastructure. See Platform requirements.

6. Create a new OpenShift Container Platform node using the new AWS machine instance.

7. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

8. Approve all the required OpenShift Container Platform CSRs for the new node:

```
$ oc adm certificate approve <certificate_name>
```

<certificate_name>
Specify the name of the CSR.

9. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.

10. Apply the OpenShift Data Foundation label to the new node using one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**.

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>
Specify the name of the new node.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:

- **csi-cephfsplugin-***
- **csi-rbdplugin-***

3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support .

1.1.2. Replacing an operational AWS node on installer-provisioned infrastructure

Procedure

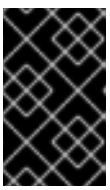
1. Log in to the OpenShift Web Console, and click **Compute → Nodes**.
2. Identify the node that you need to replace. Take a note of its **Machine Name**.
3. Mark the node as unschedulable:

```
$ oc adm cordon <node_name>
```

<node_name>
Specify the name of node that you need to replace.

4. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```



IMPORTANT

This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.

5. Click **Compute → Machines**. Search for the required machine.
6. Besides the required machine, click **Action menu (⋮) → Delete Machine**.

7. Click **Delete** to confirm that the machine is deleted. A new machine is automatically created.
8. Wait for the new machine to start and transition into **Running** state.

IMPORTANT

This activity might take at least 5 - 10 minutes or more.

9. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
10. Apply the OpenShift Data Foundation label to the new node:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**.

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

b. Display the list of available block devices:

```
$ lsblk
```

Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support.

1.1.3. Replacing a failed AWS node on user-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the Amazon Web Service (AWS) machine instance of the node that you need to replace.
2. Log in to AWS, and terminate the AWS machine instance that you identified.
3. Create a new AWS machine instance with the required infrastructure. See Platform requirements.
4. Create a new OpenShift Container Platform node using the new AWS machine instance.
5. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

6. Approve all the required OpenShift Container Platform CSRs for the new node:

```
$ oc adm certificate approve <certificate_name>
```

<certificate_name>

Specify the name of the CSR.

7. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
8. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**.

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Execute the following command to apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
6. If the verification steps fail, contact Red Hat Support .

1.1.4. Replacing a failed AWS node on installer-provisioned infrastructure

Procedure

1. Log in to the OpenShift Web Console, and click **Compute → Nodes**.
2. Identify the faulty node, and click on its **Machine Name**.

3. Click **Actions → Edit Annotations**, and click **Add More**.
4. Add **machine.openshift.io/exclude-node-draining**, and click **Save**.
5. Click **Actions → Delete Machine**, and click **Delete**.
6. A new machine is automatically created, wait for new machine to start.

IMPORTANT

This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.

7. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
8. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**.

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

9. Optional: If the failed Amazon Web Service (AWS) instance is not removed automatically, terminate the instance from the AWS console.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      $ oc debug node/<node_name>
      ```

      ```
      $ chroot /host
      ```

   b. Display the list of available block devices:

      ```
      $ lsblk
      ```

      Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support .

1.2. OPENSHIFT DATA FOUNDATION DEPLOYED ON VMWARE

- To replace an operational node, see:
  - Section 1.2.1, “Replacing an operational VMware node on user-provisioned infrastructure” .
  - Section 1.2.2, “Replacing an operational VMware node on installer-provisioned infrastructure”.
- To replace a failed node, see:
  - Section 1.2.3, “Replacing a failed VMware node on user-provisioned infrastructure” .
  - Section 1.2.4, “Replacing a failed VMware node on installer-provisioned infrastructure” .

1.2.1. Replacing an operational VMware node on user-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node and its Virtual Machine (VM) that you need replace.
2. Mark the node as unschedulable:

   ```
   $ oc adm cordon <node_name>
   ```

   <node_name>
   Specify the name of node that you need to replace.

3. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

IMPORTANT

This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.

4. Delete the node:

```
$ oc delete nodes <node_name>
```

5. Log in to VMware vSphere, and terminate the VM that you identified:

IMPORTANT

Delete the VM only from the inventory and not from the disk.

6. Create a new VM on VMware vSphere with the required infrastructure. See Platform requirements.
7. Create a new OpenShift Container Platform worker node using the new VM.
8. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

9. Approve all the required OpenShift Container Platform CSRs for the new node:

```
$ oc adm certificate approve <certificate_name>
```

<certificate_name>

Specify the name of the CSR.

10. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
11. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**.

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
6. If the verification steps fail, contact Red Hat Support .

1.2.2. Replacing an operational VMware node on installer-provisioned infrastructure

Procedure

1. Log in to the OpenShift Web Console, and click **Compute → Nodes**.
2. Identify the node that you need to replace. Take a note of its **Machine Name**.
3. Mark the node as unschedulable:

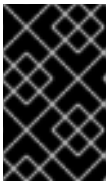
```
$ oc adm cordon <node_name>
```

<node_name>

Specify the name of node that you need to replace.

4. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```



IMPORTANT

This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.

5. Click **Compute → Machines**. Search for the required machine.
6. Besides the required machine, click **Action menu (⋮) → Delete Machine**.
7. Click **Delete** to confirm the machine is deleted. A new machine is automatically created.
8. Wait for the new machine to start and transition into **Running** state.

IMPORTANT

This activity might take at least 5 - 10 minutes or more.

9. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
10. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**.

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:

- **csi-cephfsplugin-***

- **csi-rbdplugin-***

3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   $ oc debug node/<node_name>
   ```

   ```
   $ chroot /host
   ```

   b. Display the list of available block devices:

   ```
   $ lsblk
   ```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support .

1.2.3. Replacing a failed VMware node on user-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

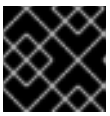
1. Identify the node and its Virtual Machine (VM) that you need to replace.
2. Delete the node:

```
$ oc delete nodes <node_name>
```

<node_name>

Specify the name of node that you need to replace.

3. Log in to VMware vSphere and terminate the VM that you identified.



IMPORTANT

Delete the VM only from the inventory and not from the disk.

4. Create a new VM on VMware vSphere with the required infrastructure. See Platform requirements.
5. Create a new OpenShift Container Platform worker node using the new VM.
6. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

7. Approve all the required OpenShift Container Platform CSRs for the new node:

```
$ oc adm certificate approve <certificate_name>
```

<certificate_name>
Specify the name of the CSR.

8. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
9. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**.

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>
Specify the name of the new node.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      $ oc debug node/<node_name>
      ```

      ```
      $ chroot /host
      ```

   b. Display the list of available block devices:

      ```
      $ lsblk
      ```

      Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support .

1.2.4. Replacing a failed VMware node on installer-provisioned infrastructure

Procedure

1. Log in to the OpenShift Web Console, and click **Compute → Nodes**.
2. Identify the faulty node, and click on its **Machine Name**.
3. Click **Actions → Edit Annotations**, and click **Add More**.
4. Add **machine.openshift.io/exclude-node-draining**, and click **Save**.
5. Click **Actions → Delete Machine**, and click **Delete**.
6. A new machine is automatically created. Wait for te new machine to start.

IMPORTANT

This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.

7. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
8. Apply the OpenShift Data Foundation label to the new node using any one of the following:

   From the user interface

   a. For the new node, click **Action Menu (⋮) → Edit Labels**.

   b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

   From the command-line interface

   - Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

9. Optional: If the failed Virtual Machine (VM) is not removed automatically, terminate the VM from VMware vSphere.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support .

1.3. OPENSHIFT DATA FOUNDATION DEPLOYED ON MICROSOFT AZURE

1.3.1. Replacing operational nodes on Azure installer-provisioned infrastructure

Procedure

1. Log in to the OpenShift Web Console, and click **Compute → Nodes**.
2. Identify the node that you need to replace. Take a note of its **Machine Name**.
3. Mark the node as unschedulable:

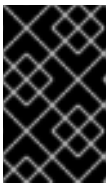
```
$ oc adm cordon <node_name>
```

<node_name>

Specify the name of node that you need to replace.

4. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```



IMPORTANT

This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.

5. Click **Compute → Machines**. Search for the required machine.
6. Besides the required machine, click the **Action menu (⋮) → Delete Machine**.
7. Click **Delete** to confirm the machine is deleted. A new machine is automatically created.
8. Wait for the new machine to start and transition into **Running** state.

IMPORTANT

This activity might take at least 5 - 10 minutes or more.

9. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
10. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**.

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Execute the following command to apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads→ Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

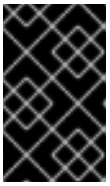
```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
6. If the verification steps fail, contact Red Hat Support .

1.3.2. Replacing failed nodes on Azure installer-provisioned infrastructure

Procedure

1. Log in to the OpenShift Web Console, and click **Compute → Nodes**.
2. Identify the faulty node, and click on its **Machine Name**.
3. Click **Actions → Edit Annotations**, and click **Add More**.
4. Add **machine.openshift.io/exclude-node-draining**, and click **Save**.
5. Click **Actions → Delete Machine**, and click **Delete**.
6. A new machine is automatically created. Wait for the new machine to start.



IMPORTANT

This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.

7. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
8. Apply the OpenShift Data Foundation label to the new node using any one of the following:

   From the user interface

   a. For the new node, click **Action Menu (⋮) → Edit Labels**.

   b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

   From the command-line interface

   - Apply the OpenShift Data Foundation label to the new node:

   ```
   $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
   ```

   <new_node_name>

   Specify the name of the new node.

9. Optional: If the failed Azure instance is not removed automatically, terminate the instance from the Azure console.

Verification steps

1. Verify that the new node is present in the output:

   ```
   $ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
   ```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:

   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***

3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that new the Object Storage Device (OSD) pods are running on the replacement node:

   ```
   $ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
   ```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

   ```
   $ oc debug node/<node_name>
   ```

```
$ chroot /host
```

b. Display the list of available block devices:

```
$ lsblk
```

Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support.

1.4. OPENSHIFT DATA FOUNDATION DEPLOYED ON GOOGLE CLOUD

1.4.1. Replacing operational nodes on Google Cloud installer-provisioned infrastructure

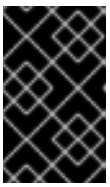
Procedure

1. Log in to OpenShift Web Console and click **Compute → Nodes**.
2. Identify the node that needs to be replaced. Take a note of its **Machine Name**.
3. Mark the node as unschedulable using the following command:

```
$ oc adm cordon <node_name>
```

4. Drain the node using the following command:

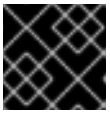
```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```



IMPORTANT

This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

5. Click **Compute → Machines**. Search for the required machine.
6. Besides the required machine, click the **Action menu (⋮) → Delete Machine**.
7. Click **Delete** to confirm the machine deletion. A new machine is automatically created.
8. Wait for new machine to start and transition into **Running** state.



IMPORTANT

This activity may take at least 5-10 minutes or more.

9. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.
10. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From User interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**

b. Add **cluster.ocs.openshift.io/openshift-storage** and click **Save**.

From Command line interface

- Execute the following command to apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that the new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
6. If the verification steps fail, contact Red Hat Support .

1.4.2. Replacing failed nodes on Google Cloud installer-provisioned infrastructure

Procedure

1. Log in to OpenShift Web Console and click **Compute → Nodes**.
2. Identify the faulty node and click on its **Machine Name**.
3. Click **Actions → Edit Annotations**, and click **Add More**.
4. Add **machine.openshift.io/exclude-node-draining** and click **Save**.
5. Click **Actions → Delete Machine**, and click **Delete**.
6. A new machine is automatically created, wait for new machine to start.

IMPORTANT

This activity may take at least 5-10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when the new node is labeled and functional.

7. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.
8. Apply the OpenShift Data Foundation label to the new node using any one of the following:

   From the web user interface

   a. For the new node, click **Action Menu (⋮) → Edit Labels**

   b. Add **cluster.ocs.openshift.io/openshift-storage** and click **Save**.

   From the command line interface

   - Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

   <new_node_name>

   Specify the name of the new node.

9. Optional: If the failed Google Cloud instance is not removed automatically, terminate the instance from Google Cloud console.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.

4. Verify that new the Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support.

CHAPTER 2. OPENSHIFT DATA FOUNDATION DEPLOYED USING LOCAL STORAGE DEVICES

2.1. REPLACING STORAGE NODES ON BARE METAL INFRASTRUCTURE

- To replace an operational node, see Section 2.1.1, "Replacing an operational node on bare metal user-provisioned infrastructure".
- To replace a failed node, see Section 2.1.2, "Replacing a failed node on bare metal user-provisioned infrastructure".

2.1.1. Replacing an operational node on bare metal user-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node, and get the labels on the node that you need to replace:

   ```
   $ oc get nodes --show-labels | grep <node_name>
   ```

   <node_name>
   Specify the name of node that you need to replace.

2. Identify the monitor pod (if any), and OSDs that are running in the node that you need to replace:

   ```
   $ oc get pods -n openshift-storage -o wide | grep -i <node_name>
   ```

3. Scale down the deployments of the pods identified in the previous step:
   For example:

   ```
   $ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
   ```

   ```
   $ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
   ```

   ```
   $ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
   --replicas=0 -n openshift-storage
   ```

4. Mark the node as unschedulable:

   ```
   $ oc adm cordon <node_name>
   ```

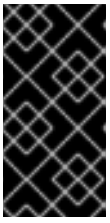
5. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

6. Delete the node:

```
$ oc delete node <node_name>
```

7. Get a new bare-metal machine with the required infrastructure. See Installing on bare metal.



IMPORTANT

For information about how to replace a master node when you have installed OpenShift Data Foundation on a three-node OpenShift compact bare-metal cluster, see the Backup and Restore guide in the OpenShift Container Platform documentation.

8. Create a new OpenShift Container Platform node using the new bare-metal machine.
9. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

10. Approve all the required OpenShift Container Platform CSRs for the new node:

```
$ oc adm certificate approve <certificate_name>
```

<certificate_name>

Specify the name of the CSR.

11. Click **Compute → Nodes** in the OpenShift Web Console. Confirm that the new node is in **Ready** state.
12. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

13. Identify the namespace where OpenShift local storage operator is installed, and assign it to the **local_storage_project** variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

14. Add a new worker node to the **localVolumeDiscovery** and **localVolumeSet**.

    a. Update the **localVolumeDiscovery** definition to include the new node, and remove the failed node:

    ```
    # oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
    ```

    Example output:

    ```
    [...]
      nodeSelector:
       nodeSelectorTerms:
         - matchExpressions:
             - key: kubernetes.io/hostname
               operator: In
               values:
               - server1.example.com
               - server2.example.com
               #- server3.example.com
               - newnode.example.com
    [...]
    ```

    Remember to save before exiting the editor.

    In this example, **server3.example.com** is removed, and **newnode.example.com** is the new node.

    b. Determine the **localVolumeSet** to edit:

    ```
    # oc get -n $local_storage_project localvolumeset
    ```

    Example output:

    ```
    NAME         AGE
    localblock   25h
    ```

    c. Update the **localVolumeSet** definition to include the new node, and remove the failed node:

    ```
    # oc edit -n $local_storage_project localvolumeset localblock
    ```

Example output:

```
[...]
  nodeSelector:
   nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - server1.example.com
          - server2.example.com
          #- server3.example.com
          - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the this example, **server3.example.com** is removed and **newnode.example.com** is the new node.

15. Verify that the new **localblock** Persistent Volume (PV) is available:

```
$oc get pv | grep localblock | grep Available
```

Example output:

```
local-pv-551d950    512Gi    RWO    Delete  Available
localblock    26s
```

16. Navigate to the **openshift-storage** project:

```
$ oc project openshift-storage
```

17. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required:

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the **rook-ceph-osd** prefix.
You can add comma separated OSD IDs in the command to remove more than one OSD, for example, **FAILED_OSD_IDS=0,1,2**.

The **FORCE_OSD_REMOVAL** value must be changed to **true** in clusters that only have three OSDs, or clusters with insufficient space to restore all three replicas of the data after the OSD is removed.

18. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.
A status of **Completed** confirms that the OSD removal job succeeded.

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

19. Ensure that the OSD removal is completed.

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

Example output:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

IMPORTANT

If the **ocs-osd-removal-job** fails, and the pod is not in the expected **Completed** state, check the pod logs for further debugging:

For example:

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

20. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC):

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

If there is a PV in **Released** state, delete it:

```
# oc delete pv <persistent_volume>
```

For example:

```
# oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

21. Identify the **crashcollector** pod deployment:

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing **crashcollector** pod deployment, delete it:

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

22. Delete the **ocs-osd-removal-job**:

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all other required OpenShift Data Foundation pods are in **Running** state. Ensure that the new incremental **mon** is created, and is in the **Running** state:

```
$ oc get pod -n openshift-storage | grep mon
```

Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66        2/2     Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8       2/2     Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh       2/2     Running
0          4m8s
```

OSD and monitor pod might take several minutes to get to the **Running** state.

4. Verify that new OSD pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support .

2.1.2. Replacing a failed node on bare metal user-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node, and get the labels on the node that you need to replace:

```
$ oc get nodes --show-labels | grep <node_name>
```

<node_name>

Specify the name of node that you need to replace.

2. Identify the monitor pod (if any), and OSDs that are running in the node that you need to replace:

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Scale down the deployments of the pods identified in the previous step:
For example:

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. Mark the node as unschedulable:

```
$ oc adm cordon <node_name>
```

5. Remove the pods which are in **Terminating** state:

```
$ oc get pods -A -o wide | grep -i <node_name> |  awk '{if ($4 == "Terminating") system ("oc -
n " $1 " delete pods " $2  " --grace-period=0 " " --force ")}'
```

6. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. Delete the node:

```
$ oc delete node <node_name>
```

8. Get a new bare-metal machine with the required infrastructure. See Installing on bare metal.

IMPORTANT

For information about how to replace a master node when you have installed OpenShift Data Foundation on a three-node OpenShift compact bare-metal cluster, see the Backup and Restore guide in the OpenShift Container Platform documentation.

9. Create a new OpenShift Container Platform node using the new bare-metal machine.
10. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

11. Approve all the required OpenShift Container Platform CSRs for the new node:

```
$ oc adm certificate approve <certificate_name>
```

<certificate_name>

Specify the name of the CSR.

12. Click **Compute → Nodes** in the OpenShift Web Console. Confirm that the new node is in **Ready** state.
13. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

14. Identify the namespace where OpenShift local storage operator is installed, and assign it to the **local_storage_project** variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

15. Add a new worker node to the **localVolumeDiscovery** and **localVolumeSet**.

    a. Update the **localVolumeDiscovery** definition to include the new node, and remove the failed node:

    ```
    # oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
    ```

    Example output:

    ```
    [...]
      nodeSelector:
       nodeSelectorTerms:
         - matchExpressions:
             - key: kubernetes.io/hostname
               operator: In
               values:
               - server1.example.com
               - server2.example.com
               #- server3.example.com
               - newnode.example.com
    [...]
    ```

    Remember to save before exiting the editor.

    In this example, **server3.example.com** is removed, and **newnode.example.com** is the new node.

    b. Determine the **localVolumeSet** to edit:

    ```
    # oc get -n $local_storage_project localvolumeset
    ```

    Example output:

    ```
    NAME         AGE
    localblock   25h
    ```

    c. Update the **localVolumeSet** definition to include the new node, and remove the failed node:

    ```
    # oc edit -n $local_storage_project localvolumeset localblock
    ```

    Example output:

    ```
    [...]
      nodeSelector:
    ```

```
    nodeSelectorTerms:
     - matchExpressions:
         - key: kubernetes.io/hostname
           operator: In
           values:
           - server1.example.com
           - server2.example.com
           #- server3.example.com
           - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the this example, **server3.example.com** is removed and **newnode.example.com** is the new node.

16. Verify that the new **localblock** Persistent Volume (PV) is available:

```
$oc get pv | grep localblock | grep Available
```

Example output:

```
local-pv-551d950    512Gi    RWO    Delete  Available
localblock    26s
```

17. Navigate to the **openshift-storage** project:

```
$ oc project openshift-storage
```

18. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required:

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the **rook-ceph-osd** prefix.
You can add comma separated OSD IDs in the command to remove more than one OSD, for example, **FAILED_OSD_IDS=0,1,2**.

The **FORCE_OSD_REMOVAL** value must be changed to **true** in clusters that only have three OSDs, or clusters with insufficient space to restore all three replicas of the data after the OSD is removed.

19. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.
A status of **Completed** confirms that the OSD removal job succeeded.

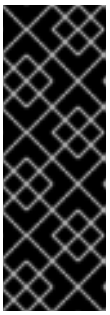
```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. Ensure that the OSD removal is completed.

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

Example output:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```



IMPORTANT

If the **ocs-osd-removal-job** fails, and the pod is not in the expected **Completed** state, check the pod logs for further debugging:

For example:

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

21. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC):

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

If there is a PV in **Released** state, delete it:

```
# oc delete pv <persistent_volume>
```

For example:

```
# oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

22. Identify the **crashcollector** pod deployment:

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing **crashcollector** pod deployment, delete it:

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

23. Delete the **ocs-osd-removal-job**:

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all other required OpenShift Data Foundation pods are in **Running** state. Ensure that the new incremental **mon** is created, and is in the **Running** state:

```
$ oc get pod -n openshift-storage | grep mon
```

   Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66        2/2     Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8       2/2     Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh       2/2     Running
0          4m8s
```

   OSD and monitor pod might take several minutes to get to the **Running** state.
4. Verify that new OSD pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

      Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
6. If the verification steps fail, contact Red Hat Support.

2.2. REPLACING STORAGE NODES ON IBM Z OR IBM® LINUXONE INFRASTRUCTURE

You can choose one of the following procedures to replace storage nodes:

- Section 2.2.1, “Replacing operational nodes on IBM Z or IBM® LinuxONE infrastructure” .
- Section 2.2.2, “Replacing failed nodes on IBM Z or IBM® LinuxONE infrastructure” .

2.2.1. Replacing operational nodes on IBM Z or IBM® LinuxONE infrastructure

Use this procedure to replace an operational node on IBM Z or IBM® LinuxONE infrastructure.

Procedure

1. Identify the node and get labels on the node to be replaced. Make a note of the rack label.

```
$ oc get nodes --show-labels | grep <node_name>
```

2. Identify the mon (if any) and object storage device (OSD) pods that are running in the node to be replaced.

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Scale down the deployments of the pods identified in the previous step.
For example:

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. Mark the nodes as unschedulable.

```
$ oc adm cordon <node_name>
```

5. Remove the pods which are in the **Terminating** state.

```
$ oc get pods -A -o wide | grep -i <node_name> |  awk '{if ($4 == "Terminating") system ("oc -
n " $1 " delete pods " $2  " --grace-period=0 " " --force ")}'
```

6. Drain the node.

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. Delete the node.

```
$ oc delete node <node_name>
```

8. Get a new IBM Z storage node as a replacement.

9. Check for certificate signing requests (CSRs) related to OpenShift Data Foundation that are in **Pending** state:

```
$ oc get csr
```

10. Approve all required OpenShift Data Foundation CSRs for the new node:

```
$ oc adm certificate approve <Certificate_Name>
```

11. Click **Compute → Nodes** in OpenShift Web Console, confirm if the new node is in **Ready** state.

12. Apply the **openshift-storage** label to the new node using any one of the following:

    From User interface

    a. For the new node, click **Action Menu (⋮) → Edit Labels**

    b. Add **cluster.ocs.openshift.io/openshift-storage** and click **Save**.

    From Command line interface

    - Execute the following command to apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

13. Add a new worker node to **localVolumeDiscovery** and **localVolumeSet**.

    a. Update the **localVolumeDiscovery** definition to include the new node and remove the failed node.

```
# oc edit -n local-storage-project localvolumediscovery auto-discover-devices
[...]
  nodeSelector:
   nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - server1.example.com
          - server2.example.com
          #- server3.example.com
          - newnode.example.com
[...]
```

    Remember to save before exiting the editor.

    In the above example, **server3.example.com** was removed and **newnode.example.com** is the new node.

    b. Determine which **localVolumeSet** to edit.
    Replace *local-storage-project* in the following commands with the name of your local storage project. The default project name is **openshift-local-storage** in OpenShift Data Foundation 4.6 and later. Previous versions use **local-storage** by default.

```
# oc get -n local-storage-project localvolumeset
NAME         AGE
localblock   25h
```

c. Update the **localVolumeSet** definition to include the new node and remove the failed node.

```
# oc edit -n local-storage-project localvolumeset localblock
[...]
   nodeSelector:
    nodeSelectorTerms:
     - matchExpressions:
         - key: kubernetes.io/hostname
           operator: In
           values:
           - server1.example.com
           - server2.example.com
           #- server3.example.com
           - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the above example, **server3.example.com** was removed and **newnode.example.com** is the new node.

14. Verify that the new **localblock** PV is available.

```
$ oc get pv | grep localblock
        CAPA- ACCESS RECLAIM                          STORAGE
NAME     CITY  MODES  POLICY  STATUS     CLAIM             CLASS      AGE
local-pv- 931Gi  RWO   Delete  Bound      openshift-storage/  localblock  25h
3e8964d3                        ocs-deviceset-2-0
                               -79j94
local-pv- 931Gi  RWO   Delete  Bound      openshift-storage/  localblock  25h
414755e0                        ocs-deviceset-1-0
                               -959rp
local-pv- 931Gi RWO Delete Available localblock 3m24s b481410
local-pv- 931Gi  RWO   Delete  Bound      openshift-storage/  localblock  25h
d9c5cbd6                       ocs-deviceset-0-0
                               -nvs68
```

15. Change to the **openshift-storage** project.

```
$ oc project openshift-storage
```

16. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

a. Identify the PVC as afterwards we need to delete PV associated with that specific PVC.

```
$ osd_id_to_remove=1
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} |
grep ceph.rook.io/pvc
```

where, **osd_id_to_remove** is the integer in the pod name immediately after the **rook-ceph-osd prefix**. In this example, the deployment name is **rook-ceph-osd-1**.

Example output:

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
    ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
```

In this example, the PVC name is **ocs-deviceset-localblock-0-data-0-g2mmc**.

b. Remove the failed OSD from the cluster.

```
$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_id_to_remove} |oc create -f -
```

You can remove more than one OSD by adding comma separated OSD IDs in the command. (For example: FAILED_OSD_IDS=0,1,2)

WARNING

This step results in OSD being completely removed from the cluster. Ensure that the correct value of **osd_id_to_remove** is provided.

17. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal** pod.
A status of **Completed** confirms that the OSD removal job succeeded.

```
# oc get pod -l job-name=ocs-osd-removal-osd_id_to_remove -n openshift-storage
```

NOTE

If **ocs-osd-removal** fails and the pod is not in the expected **Completed** state, check the pod logs for further debugging. For example:

```
# oc logs -l job-name=ocs-osd-removal-osd_id_to_remove -n openshift-storage --tail=-1
```

It may be necessary to manually cleanup the removed OSD as follows:

```
ceph osd crush remove osd.osd_id_to_remove
ceph osd rm osd_id_to_remove
ceph auth del osd.osd_id_to_remove
ceph osd crush rm osd_id_to_remove
```

18. Delete the PV associated with the failed node.

a. Identify the PV associated with the PVC.

The PVC name must be identical to the name that is obtained while removing the failed OSD from the cluster.

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
local-pv-5c9b8982 500Gi RWO Delete Released openshift-storage/ocs-deviceset-localblock-0-data-0-g2mmc localblock 24h worker-0
```

b. If there is a PV in **Released** state, delete it.

```
# oc delete pv <persistent-volume>
```

For example:

```
# oc delete pv local-pv-5c9b8982
persistentvolume "local-pv-5c9b8982" deleted
```

19. Identify the **crashcollector** pod deployment.

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing **crashcollector** pod deployment, delete it.

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

20. Delete the **ocs-osd-removal** job.

```
# oc delete job ocs-osd-removal-${osd_id_to_remove}
```

Example output:

```
job.batch "ocs-osd-removal-0" deleted
```

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If data encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

      ```
      $ oc debug node/<node_name>
      ```

      ```
      $ chroot /host
      ```

   b. Display the list of available block devices:

      ```
      $ lsblk
      ```

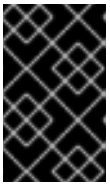
      Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support .

2.2.2. Replacing failed nodes on IBM Z or IBM® LinuxONE infrastructure

Procedure

1. Log in to the OpenShift Web Console, and click **Compute → Nodes**.
2. Identify the faulty node, and click on its **Machine Name**.
3. Click **Actions → Edit Annotations**, and click **Add More**.
4. Add **machine.openshift.io/exclude-node-draining**, and click **Save**.
5. Click **Actions → Delete Machine**, and click **Delete**.
6. A new machine is automatically created. Wait for new machine to start.



IMPORTANT

This activity might take at least 5 - 10 minutes or more. Ceph errors generated during this period are temporary and are automatically resolved when you label the new node, and it is functional.

7. Click **Compute → Nodes**. Confirm that the new node is in **Ready** state.
8. Apply the OpenShift Data Foundation label to the new node using any one of the following:

   From the user interface

   a. For the new node, click **Action Menu (⋮) → Edit Labels**.

   b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

   From the command-line interface

   - Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all the other required OpenShift Data Foundation pods are in **Running** state.
4. Verify that new Object Storage Device (OSD) pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If data encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
6. If the verification steps fail, contact Red Hat Support.

2.3. REPLACING STORAGE NODES ON IBM POWER INFRASTRUCTURE

For OpenShift Data Foundation, you can perform node replacement proactively for an operational node, and reactively for a failed node, for the deployments related to IBM Power.

2.3.1. Replacing an operational or failed storage node on IBM Power

Prerequisites

- Ensure that the replacement nodes are configured with the similar infrastructure and resources to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node, and get the labels on the node that you need to replace:

```
$ oc get nodes --show-labels | grep <node_name>
```

<node_name>

Specify the name of node that you need to replace.

2. Identify the **mon** (if any), and Object Storage Device (OSD) pods that are running in the node that you need to replace:

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Scale down the deployments of the pods identified in the previous step:
For example:

```
$ oc scale deployment rook-ceph-mon-a --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-1 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. Mark the node as unschedulable:

```
$ oc adm cordon <node_name>
```

5. Remove the pods which are in **Terminating** state:

```
$ oc get pods -A -o wide | grep -i <node_name> |  awk '{if ($4 == "Terminating") system ("oc -
n " $1 " delete pods " $2  " --grace-period=0 " " --force ")}'
```

6. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. Delete the node:

```
$ oc delete node <node_name>
```

8. Get a new IBM Power machine with the required infrastructure. See Installing a cluster on IBM Power.
9. Create a new OpenShift Container Platform node using the new IBM Power machine.

10. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

11. Approve all the required OpenShift Container Platform CSRs for the new node:

```
$ oc adm certificate approve <certificate_name>
```

<certificate_name>
Specify the name of the CSR.

12. Click **Compute → Nodes** in the OpenShift Web Console. Confirm that the new node is in **Ready** state.

13. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**.

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

a. Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=''
```

<new_node_name>
Specify the name of the new node.

14. Identify the namespace where OpenShift local storage operator is installed, and assign it to the **local_storage_project** variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

15. Add a newly added worker node to the **localVolume**.

a. Determine the **localVolume** you need to edit:

```
# oc get -n $local_storage_project localvolume
```

Example output:

```
NAME          AGE
localblock    25h
```

b. Update the **localVolume** definition to include the new node, and remove the failed node:

```
# oc edit -n $local_storage_project localvolume localblock
```

Example output:

```
[...]
   nodeSelector:
   nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          #- worker-0
          - worker-1
          - worker-2
          - worker-3
[...]
```

Remember to save before exiting the editor.

In the this example, **worker-0** is removed and **worker-3** is the new node.

16. Verify that the new **localblock** Persistent Volume (PV) is available:

```
$ oc get pv | grep localblock
```

Example output:

```
NAME              CAPACITY   ACCESSMODES RECLAIMPOLICY STATUS     CLAIM
STORAGECLASS              AGE
local-pv-3e8964d3   500Gi    RWO         Delete       Bound      ocs-deviceset-localblock-2-
data-0-mdbg9  localblock     25h
local-pv-414755e0   500Gi    RWO         Delete       Bound      ocs-deviceset-localblock-1-
data-0-4cslf  localblock     25h
local-pv-b481410    500Gi     RWO         Delete       Available
localblock     3m24s
local-pv-5c9b8982   500Gi    RWO         Delete       Bound      ocs-deviceset-localblock-0-
data-0-g2mmc  localblock     25h
```

17. Navigate to the **openshift-storage** project:

```
$ oc project openshift-storage
```

18. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required.

a. Identify the Persistent Volume Claim (PVC):

```
$ osd_id_to_remove=1
```

-

```
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${<osd_id_to_remove>}
| grep ceph.rook.io/pvc
```

where, ***<osd_id_to_remove>*** is the integer in the pod name immediately after the **rook-ceph-osd** prefix.

In this example, the deployment name is **rook-ceph-osd-1**.

Example output:

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
    ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
```

b. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required:

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the **rook-ceph-osd** prefix. You can add comma separated OSD IDs in the command to remove more than one OSD, for example, **FAILED_OSD_IDS=0,1,2**.
The **FORCE_OSD_REMOVAL** value must be changed to **true** in clusters that only have three OSDs, or clusters with insufficient space to restore all three replicas of the data after the OSD is removed.

WARNING

This step results in the OSD being completely removed from the cluster. Ensure that the correct value of **osd_id_to_remove** is provided.

19. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.
A status of **Completed** confirms that the OSD removal job has succeeded.

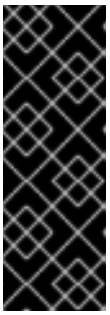
```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. Ensure that the OSD removal is completed.

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed
removal'
```

Example output:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```



IMPORTANT

If the **ocs-osd-removal-job** fails, and the pod is not in the expected **Completed** state, check the pod logs for further debugging.

For example:

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

21. Delete the PV associated with the failed node.

    a. Identify the PV associated with the PVC:

    ```
    # oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
    ```

    Example output:

    ```
    local-pv-5c9b8982 500Gi RWO Delete Released openshift-storage/ocs-deviceset-localblock-0-data-0-g2mmc localblock 24h worker-0
    ```

    The PVC name must be identical to the name that is obtained while removing the failed OSD from the cluster.

    b. If there is a PV in **Released** state, delete it:

    ```
    # oc delete pv <persistent_volume>
    ```

    For example:

    ```
    # oc delete pv local-pv-5c9b8982
    ```

    Example output:

    ```
    persistentvolume "local-pv-5c9b8982" deleted
    ```

22. Identify the **crashcollector** pod deployment:

    ```
    $ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
    ```

    If there is an existing **crashcollector** pod deployment, delete it:

    ```
    $ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
    ```

23. Delete the **ocs-osd-removal-job**:

    ```
    # oc delete -n openshift-storage job ocs-osd-removal-job
    ```

    Example output:

    ```
    job.batch "ocs-osd-removal-job" deleted
    ```

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all other required OpenShift Data Foundation pods are in **Running** state.
   Ensure that the new incremental **mon** is created and is in the **Running** state:

```
$ oc get pod -n openshift-storage | grep mon
```

   Example output:

```
rook-ceph-mon-b-74f6dc9dd6-4llzq                  1/1     Running     0          6h14m
rook-ceph-mon-c-74948755c-h7wtx                   1/1     Running     0          4h24m
rook-ceph-mon-d-598f69869b-4bv49                  1/1     Running     0          162m
```

   The OSD and monitor pod might take several minutes to get to the **Running** state.
4. Verify that the new OSD pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
6. If the verification steps fail, contact Red Hat Support .

2.4. REPLACING STORAGE NODES ON VMWARE INFRASTRUCTURE

- To replace an operational node, see:
  - Section 2.4.1, “Replacing an operational node on VMware user-provisioned infrastructure” .

- Section 2.4.2, “Replacing an operational node on VMware installer-provisioned infrastructure”.

- To replace a failed node,see:
  - Section 2.4.3, “Replacing a failed node on VMware user-provisioned infrastructure” .
  - Section 2.4.4, “Replacing a failed node on VMware installer-provisioned infrastructure” .

2.4.1. Replacing an operational node on VMware user-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node, and get the labels on the node that you need to replace:

```
$ oc get nodes --show-labels | grep <node_name>
```

<node_name>

Specify the name of node that you need to replace.

2. Identify the monitor pod (if any), and OSDs that are running in the node that you need to replace:

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Scale down the deployments of the pods identified in the previous step:
For example:

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. Mark the node as unschedulable:

```
$ oc adm cordon <node_name>
```

5. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

6. Delete the node:

```
$ oc delete node <node_name>
```

7. Log in to VMware vSphere and terminate the Virtual Machine (VM) that you have identified.
8. Create a new VM on VMware vSphere with the required infrastructure. See Infrastructure requirements.
9. Create a new OpenShift Container Platform worker node using the new VM.
10. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

11. Approve all the required OpenShift Container Platform CSRs for the new node:

```
$ oc adm certificate approve <certificate_name>
```

<certificate_name>

Specify the name of the CSR.

12. Click **Compute → Nodes** in the OpenShift Web Console. Confirm that the new node is in **Ready** state.
13. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

14. Identify the namespace where OpenShift local storage operator is installed, and assign it to the **local_storage_project** variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

15. Add a new worker node to the **localVolumeDiscovery** and **localVolumeSet**.

    a. Update the **localVolumeDiscovery** definition to include the new node, and remove the failed node:

    ```
    # oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
    ```

    Example output:

    ```
    [...]
      nodeSelector:
       nodeSelectorTerms:
         - matchExpressions:
             - key: kubernetes.io/hostname
               operator: In
               values:
               - server1.example.com
               - server2.example.com
               #- server3.example.com
               - newnode.example.com
    [...]
    ```

    Remember to save before exiting the editor.

    In this example, **server3.example.com** is removed, and **newnode.example.com** is the new node.

    b. Determine the **localVolumeSet** to edit:

    ```
    # oc get -n $local_storage_project localvolumeset
    ```

    Example output:

    ```
    NAME         AGE
    localblock   25h
    ```

    c. Update the **localVolumeSet** definition to include the new node, and remove the failed node:

    ```
    # oc edit -n $local_storage_project localvolumeset localblock
    ```

    Example output:

    ```
    [...]
      nodeSelector:
       nodeSelectorTerms:
         - matchExpressions:
             - key: kubernetes.io/hostname
               operator: In
               values:
    ```

```
            - server1.example.com
            - server2.example.com
            #- server3.example.com
            - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the this example, **server3.example.com** is removed and **newnode.example.com** is the new node.

16. Verify that the new **localblock** Persistent Volume (PV) is available:

```
$oc get pv | grep localblock | grep Available
```

Example output:

```
local-pv-551d950    512Gi    RWO    Delete  Available
localblock    26s
```

17. Navigate to the **openshift-storage** project:

```
$ oc project openshift-storage
```

18. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required:

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the **rook-ceph-osd** prefix.
You can add comma separated OSD IDs in the command to remove more than one OSD, for example, **FAILED_OSD_IDS=0,1,2**.

The **FORCE_OSD_REMOVAL** value must be changed to **true** in clusters that only have three OSDs, or clusters with insufficient space to restore all three replicas of the data after the OSD is removed.

19. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.
A status of **Completed** confirms that the OSD removal job succeeded.

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. Ensure that the OSD removal is completed.

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

Example output:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

IMPORTANT

If the **ocs-osd-removal-job** fails, and the pod is not in the expected **Completed** state, check the pod logs for further debugging:

For example:

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

21. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC):

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

If there is a PV in **Released** state, delete it:

```
# oc delete pv <persistent_volume>
```

For example:

```
# oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

22. Identify the **crashcollector** pod deployment:

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing **crashcollector** pod deployment, delete it:

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

23. Delete the **ocs-osd-removal-job**:

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all other required OpenShift Data Foundation pods are in **Running** state. Ensure that the new incremental **mon** is created, and is in the **Running** state:

```
$ oc get pod -n openshift-storage | grep mon
```

   Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66        2/2     Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8       2/2     Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh       2/2     Running
0          4m8s
```

   OSD and monitor pod might take several minutes to get to the **Running** state.
4. Verify that new OSD pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

      Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
6. If the verification steps fail, contact Red Hat Support.

2.4.2. Replacing an operational node on VMware installer-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with the similar infrastructure, resources, and disks to the node that you replace.

- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Log in to the OpenShift Web Console, and click **Compute → Nodes**.
2. Identify the node that you need to replace. Take a note of its **Machine Name**.
3. Get labels on the node:

```
$ oc get nodes --show-labels | grep <node_name>
```

<node_name>
Specify the name of node that you need to replace.

4. Identify the **mon** (if any), and Object Storage Devices (OSDs) that are running in the node:

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

5. Scale down the deployments of the pods that you identified in the previous step:
For example:

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

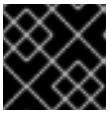
6. Mark the node as unschedulable:

```
$ oc adm cordon <node_name>
```

7. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

8. Click **Compute → Machines**. Search for the required machine.
9. Besides the required machine, click **Action menu (⋮) → Delete Machine**
10. Click **Delete** to confirm the machine deletion. A new machine is automatically created.
11. Wait for the new machine to start and transition into **Running** state.



IMPORTANT

This activity might take at least 5 - 10 minutes or more.

12. Click **Compute → Nodes** in the OpenShift Web Console. Confirm that the new node is in **Ready** state.

13. Physically add a new device to the node.

14. Apply the OpenShift Data Foundation label to the new node using any one of the following:

    From the user interface

    a. For the new node, click **Action Menu (⋮) → Edit Labels**

    b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

    From the command-line interface

    - Apply the OpenShift Data Foundation label to the new node:

    ```
    $ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
    ```

    <new_node_name>

    Specify the name of the new node.

15. Identify the namespace where the OpenShift local storage operator is installed, and assign it to the **local_storage_project** variable:

    ```
    $ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
    ```

    For example:

    ```
    $ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
    ```

    ```
    echo $local_storage_project
    ```

    Example output:

    ```
    openshift-local-storage
    ```

16. Add a new worker node to the **localVolumeDiscovery** and **localVolumeSet**.

    a. Update the **localVolumeDiscovery** definition to include the new node and remove the failed node.

    ```
    # oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
    ```

    Example output:

    ```
    [...]
      nodeSelector:
       nodeSelectorTerms:
        - matchExpressions:
            - key: kubernetes.io/hostname
              operator: In
              values:
              - server1.example.com
              - server2.example.com
    ```

```
            #- server3.example.com
            - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In this example, **server3.example.com** is removed, and **newnode.example.com** is the new node.

b. Determine the **localVolumeSet** you need to edit:

```
# oc get -n $local_storage_project localvolumeset
```

Example output:

```
NAME         AGE
localblock   25h
```

c. Update the **localVolumeSet** definition to include the new node and remove the failed node:

```
# oc edit -n $local_storage_project localvolumeset localblock
```

Example output:

```
[...]
  nodeSelector:
   nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - server1.example.com
          - server2.example.com
          #- server3.example.com
          - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In this example, **server3.example.com** is removed, and **newnode.example.com** is the new node.

17. Verify that the new **localblock** Persistent Volume (PV) is available:

```
$ oc get pv | grep localblock | grep Available
```

Example output:

```
local-pv-551d950    512Gi    RWO    Delete  Available
localblock    26s
```

18. Navigate to the **openshift-storage** project:

```
$ oc project openshift-storage
```

19. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required:

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the **rook-ceph-osd** prefix.
You can add comma separated OSD IDs in the command to remove more than one OSD, for example, **FAILED_OSD_IDS=0,1,2**.

The **FORCE_OSD_REMOVAL** value must be changed to **true** in clusters that only have three OSDs, or clusters with insufficient space to restore all three replicas of the data after the OSD is removed.

20. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.
A status of **Completed** confirms that the OSD removal job succeeded.

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

21. Ensure that the OSD removal is completed.

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

Example output:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

IMPORTANT

If the **ocs-osd-removal-job** fails and the pod is not in the expected **Completed** state, check the pod logs for further debugging.

For example:

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

22. Identify the PV associated with the Persistent Volume Claim (PVC):

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

If there is a PV in **Released** state, delete it:

```
# oc delete pv <persistent_volume>
```

For example:

```
# oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

23. Identify the **crashcollector** pod deployment:

```
$ oc get deployment --selector=app=rook-ceph-
crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing **crashcollector** pod deployment, delete it:

```
$ oc delete deployment --selector=app=rook-ceph-
crashcollector,node_name=<failed_node_name> -n openshift-storage
```

24. Delete the **ocs-osd-removal-job**:

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all other required OpenShift Data Foundation pods are in **Running** state. Ensure that the new incremental **mon** is created and is in the **Running** state.

```
$ oc get pod -n openshift-storage | grep mon
```

Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66        2/2     Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8       2/2     Running
```

```
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh        2/2     Running
0          4m8s
```

OSD and monitor pod might take several minutes to get to the **Running** state.

4. Verify that new OSD pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:

   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

   Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support.

2.4.3. Replacing a failed node on VMware user-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Identify the node, and get the labels on the node that you need to replace:

```
$ oc get nodes --show-labels | grep <node_name>
```

<node_name>
Specify the name of node that you need to replace.

2. Identify the monitor pod (if any), and OSDs that are running in the node that you need to replace:

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. Scale down the deployments of the pods identified in the previous step:
   For example:

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. Mark the node as unschedulable:

```
$ oc adm cordon <node_name>
```

5. Remove the pods which are in **Terminating** state:

```
$ oc get pods -A -o wide | grep -i <node_name> |  awk '{if ($4 == "Terminating") system ("oc -
n " $1 " delete pods " $2  " --grace-period=0 " " --force ")}'
```

6. Drain the node:

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. Delete the node:

```
$ oc delete node <node_name>
```

8. Log in to VMware vSphere and terminate the Virtual Machine (VM) that you have identified.

9. Create a new VM on VMware vSphere with the required infrastructure. See Infrastructure requirements.

10. Create a new OpenShift Container Platform worker node using the new VM.

11. Check for the Certificate Signing Requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

12. Approve all the required OpenShift Container Platform CSRs for the new node:

```
$ oc adm certificate approve <certificate_name>
```

<certificate_name>

Specify the name of the CSR.

13. Click **Compute → Nodes** in the OpenShift Web Console. Confirm that the new node is in **Ready** state.

14. Apply the OpenShift Data Foundation label to the new node using any one of the following:

From the user interface

a. For the new node, click **Action Menu (⋮) → Edit Labels**

b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

From the command-line interface

- Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

<new_node_name>

Specify the name of the new node.

15. Identify the namespace where OpenShift local storage operator is installed, and assign it to the **local_storage_project** variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

16. Add a new worker node to the **localVolumeDiscovery** and **localVolumeSet**.

a. Update the **localVolumeDiscovery** definition to include the new node, and remove the failed node:

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

Example output:

```
[...]
  nodeSelector:
   nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - server1.example.com
          - server2.example.com
          #- server3.example.com
          - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In this example, **server3.example.com** is removed, and **newnode.example.com** is the new node.

b. Determine the **localVolumeSet** to edit:

```
# oc get -n $local_storage_project localvolumeset
```

Example output:

```
NAME         AGE
localblock   25h
```

c. Update the **localVolumeSet** definition to include the new node, and remove the failed node:

```
# oc edit -n $local_storage_project localvolumeset localblock
```

Example output:

```
[...]
  nodeSelector:
   nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - server1.example.com
          - server2.example.com
          #- server3.example.com
          - newnode.example.com
[...]
```

Remember to save before exiting the editor.

In the this example, **server3.example.com** is removed and **newnode.example.com** is the new node.

17. Verify that the new **localblock** Persistent Volume (PV) is available:

```
$oc get pv | grep localblock | grep Available
```

Example output:

```
local-pv-551d950    512Gi    RWO    Delete  Available
localblock    26s
```

18. Navigate to the **openshift-storage** project:

```
$ oc project openshift-storage
```

19. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required:

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the **rook-ceph-osd** prefix.

You can add comma separated OSD IDs in the command to remove more than one OSD, for example, **FAILED_OSD_IDS=0,1,2**.

The **FORCE_OSD_REMOVAL** value must be changed to **true** in clusters that only have three OSDs, or clusters with insufficient space to restore all three replicas of the data after the OSD is removed.

20. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.
A status of **Completed** confirms that the OSD removal job succeeded.

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

21. Ensure that the OSD removal is completed.

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

Example output:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

IMPORTANT

If the **ocs-osd-removal-job** fails, and the pod is not in the expected **Completed** state, check the pod logs for further debugging:

For example:

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

22. Identify the Persistent Volume (PV) associated with the Persistent Volume Claim (PVC):

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b  1490Gi  RWO  Delete  Released  openshift-storage/ocs-deviceset-0-data-0-6c5pw  localblock  2d22h  compute-1
```

If there is a PV in **Released** state, delete it:

```
# oc delete pv <persistent_volume>
```

For example:

```
# oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

23. Identify the **crashcollector** pod deployment:

```
$ oc get deployment --selector=app=rook-ceph-
crashcollector,node_name=<failed_node_name> -n openshift-storage
```

If there is an existing **crashcollector** pod deployment, delete it:

```
$ oc delete deployment --selector=app=rook-ceph-
crashcollector,node_name=<failed_node_name> -n openshift-storage
```

24. Delete the **ocs-osd-removal-job**:

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all other required OpenShift Data Foundation pods are in **Running** state. Ensure that the new incremental **mon** is created, and is in the **Running** state:

```
$ oc get pod -n openshift-storage | grep mon
```

Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66        2/2     Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8       2/2     Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh       2/2     Running
0          4m8s
```

OSD and monitor pod might take several minutes to get to the **Running** state.

4. Verify that new OSD pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

For each of the new nodes identified in the previous step, do the following:

a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

b. Display the list of available block devices:

```
$ lsblk
```

Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.

6. If the verification steps fail, contact Red Hat Support.

2.4.4. Replacing a failed node on VMware installer-provisioned infrastructure

Prerequisites

- Ensure that the replacement nodes are configured with the similar infrastructure, resources, and disks to the node that you replace.
- You must be logged into the OpenShift Container Platform cluster.

Procedure

1. Log in to the OpenShift Web Console, and click **Compute → Nodes**.
2. Identify the node that you need to replace. Take a note of its **Machine Name**.
3. Get the labels on the node:

```
$ oc get nodes --show-labels | grep _<node_name>_
```

<node_name>
Specify the name of node that you need to replace.

4. Identify the **mon** (if any) and Object Storage Devices (OSDs) that are running in the node:

```
$ oc get pods -n openshift-storage -o wide | grep -i _<node_name>_
```

5. Scale down the deployments of the pods identified in the previous step:
For example:

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

6. Mark the node as unschedulable:

```
$ oc adm cordon _<node_name>_
```

7. Remove the pods which are in **Terminating** state:

```
$ oc get pods -A -o wide | grep -i _<node_name>_ |  awk '{if ($4 == "Terminating") system
("oc -n " $1 " delete pods " $2  " --grace-period=0 " " --force ")}'
```

8. Drain the node:

```
$ oc adm drain _<node_name>_ --force --delete-emptydir-data=true --ignore-daemonsets
```

9. Click **Compute → Machines**. Search for the required machine.
10. Besides the required machine, click **Action menu (⋮) → Delete Machine**
11. Click **Delete** to confirm the machine is deleted. A new machine is automatically created.
12. Wait for the new machine to start and transition into **Running** state.

IMPORTANT

This activity might take at least 5 - 10 minutes or more.

13. Click **Compute → Nodes** in the OpenShift Web Console. Confirm that the new node is in **Ready** state.
14. Physically add a new device to the node.
15. Apply the OpenShift Data Foundation label to the new node using any one of the following:

    From the user interface

    a. For the new node, click **Action Menu (⋮) → Edit Labels**

    b. Add **cluster.ocs.openshift.io/openshift-storage**, and click **Save**.

    From the command-line interface

    - Apply the OpenShift Data Foundation label to the new node:

```
$ oc label node _<new_node_name>_ cluster.ocs.openshift.io/openshift-storage=""
```

    <new_node_name>

    Specify the name of the new node.

16. Identify the namespace where the OpenShift local storage operator is installed, and assign it to the **local_storage_project** variable:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

    For example:

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

Example output:

```
openshift-local-storage
```

17. Add a new worker node to the **localVolumeDiscovery** and **localVolumeSet**.

    a. Update the **localVolumeDiscovery** definition to include the new node and remove the failed node:

    ```
    # oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
    ```

    Example output:

    ```
    [...]
      nodeSelector:
       nodeSelectorTerms:
         - matchExpressions:
             - key: kubernetes.io/hostname
               operator: In
               values:
               - server1.example.com
               - server2.example.com
               #- server3.example.com
               - **newnode.example.com**
    [...]
    ```

    Remember to save before exiting the editor.

    In this example, **server3.example.com** is removed and **newnode.example.com** is the new node.

    b. Determine the **localVolumeSet** you need to edit.

    ```
    # oc get -n $local_storage_project localvolumeset
    ```

    Example output:

    ```
    NAME         AGE
    localblock   25h
    ```

    c. Update the **localVolumeSet** definition to include the new node and remove the failed node:

    ```
    # oc edit -n $local_storage_project localvolumeset localblock
    ```

    Example output:

    ```
    [...]
      nodeSelector:
    ```

```
    nodeSelectorTerms:
     - matchExpressions:
         - key: kubernetes.io/hostname
           operator: In
           values:
           - server1.example.com
           - server2.example.com
           #- server3.example.com
           - **newnode.example.com**
[...]
```

Remember to save before exiting the editor.

In this example, **server3.example.com** is removed and **newnode.example.com** is the new node.

18. Verify that the new **localblock** PV is available:

```
$ oc get pv | grep localblock | grep Available
```

Example output:

```
local-pv-551d950    512Gi    RWO    Delete  Available
localblock    26s
```

19. Navigate to the **openshift-storage** project:

```
$ oc project openshift-storage
```

20. Remove the failed OSD from the cluster. You can specify multiple failed OSDs if required:

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> | oc create -f -
```

<failed_osd_id>

Is the integer in the pod name immediately after the **rook-ceph-osd** prefix.
You can add comma separated OSD IDs in the command to remove more than one OSD, for example, **FAILED_OSD_IDS=0,1,2**.

The **FORCE_OSD_REMOVAL** value must be changed to **true** in clusters that only have three OSDs, or clusters with insufficient space to restore all three replicas of the data after the OSD is removed.

21. Verify that the OSD was removed successfully by checking the status of the **ocs-osd-removal-job** pod.
A status of **Completed** confirms that the OSD removal job succeeded.

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

22. Ensure that the OSD removal is completed.

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

Example output:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

IMPORTANT

If the **ocs-osd-removal-job** fails and the pod is not in the expected **Completed** state, check the pod logs for further debugging:

For example:

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

23. Identify the PV associated with the Persistent Volume Claim (PVC):

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

Example output:

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

If there is a PV in **Released** state, delete it:

```
# oc delete pv _<persistent_volume>_
```

For example:

```
# oc delete pv local-pv-d6bf175b
```

Example output:

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

24. Identify the **crashcollector** pod deployment:

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=_<failed_node_name>_ -n openshift-storage
```

If there is an existing **crashcollector** pod deployment, delete it:

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=_<failed_node_name>_ -n openshift-storage
```

25. Delete the **ocs-osd-removal-job**:

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

Example output:

```
job.batch "ocs-osd-removal-job" deleted
```

Verification steps

1. Verify that the new node is present in the output:

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. Click **Workloads → Pods**. Confirm that at least the following pods on the new node are in **Running** state:
   - **csi-cephfsplugin-***
   - **csi-rbdplugin-***
3. Verify that all other required OpenShift Data Foundation pods are in **Running** state. Ensure that the new incremental **mon** is created, and is in the **Running** state:

```
$ oc get pod -n openshift-storage | grep mon
```

   Example output:

```
rook-ceph-mon-a-cd575c89b-b6k66        2/2     Running
0          38m
rook-ceph-mon-b-6776bc469b-tzzt8       2/2     Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh       2/2     Running
0          4m8s
```

   OSD and monitor pod might take several minutes to get to the **Running** state.
4. Verify that new OSD pods are running on the replacement node:

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.
   For each of the new nodes identified in the previous step, do the following:
   a. Create a debug pod and open a chroot environment for the one or more selected hosts:

```
$ oc debug node/<node_name>
```

```
$ chroot /host
```

   b. Display the list of available block devices:

```
$ lsblk
```

      Check for the **crypt** keyword beside the one or more **ocs-deviceset** names.
6. If the verification steps fail, contact Red Hat Support.