Red Hat OpenShift Container Storage 4.6

Replacing nodes

How to prepare replacement nodes and replace failed nodes

Last Updated: 2022-03-10

Red Hat OpenShift Container Storage 4.6 Replacing nodes

How to prepare replacement nodes and replace failed nodes

Legal Notice

Abstract

This document explains how to safely replace a node in a Red Hat OpenShift Container Storage cluster.

Table of Contents

PREFACE

For OpenShift Container Storage, 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
- For Microsoft Azure
  - Installer-provisioned infrastructure
- For local storage devices
  - Bare metal
  - Amazon EC2 I3
  - VMware
  - IBM Power Systems
  - IBM Z or LinuxONE
- For replacing your storage nodes in external mode, see Red Hat Ceph Storage documentation .

CHAPTER 1. OPENSHIFT CONTAINER STORAGE DEPLOYED ON AWS

1.1. REPLACING AN OPERATIONAL AWS NODE ON USER-PROVISIONED INFRASTRUCTURE

Perform this procedure to replace an operational node on AWS user-provisioned infrastructure.

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

Procedure

1. Identify the node that needs to be replaced.
2. Mark the node as unschedulable using the following command:

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

3. Drain the node using the following command:

```
$ oc adm drain <node_name> --force --delete-local-data --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.

4. Delete the node using the following command:

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

5. Create a new 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 certificate signing requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

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

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

9. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.
10. Apply the OpenShift Container Storage label to the new node.

    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

    - Execute the following command to apply the OpenShift Container Storage label to the new node:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **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 data 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 selected host(s).

      ```
      $ oc debug node/<node name>
      $ chroot /host
      ```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

      ```
      $ lsblk
      ```

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

1.2. REPLACING AN OPERATIONAL AWS NODE ON INSTALLER-PROVISIONED INFRASTRUCTURE

Use this procedure to replace an operational node on AWS installer-provisioned infrastructure (IPI).

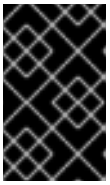
Procedure

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-local-data --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 Container 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 Container Storage label to the new node:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

1.3. REPLACING A FAILED AWS NODE ON USER-PROVISIONED INFRASTRUCTURE

Perform this procedure to replace a failed node which is not operational on AWS user-provisioned infrastructure (UPI) for OpenShift Container Storage.

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

Procedure

1. Identify the AWS machine instance of the node that needs to be replaced.
2. Log in to AWS and terminate the identified AWS machine instance.
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 certificate signing requests (CSRs) related to OpenShift Container Platform that are in **Pending** state:

```
$ oc get csr
```

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

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

7. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.
8. Apply the OpenShift Container 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 Container Storage label to the new node:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

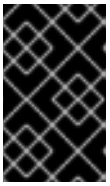
6. If verification steps fail, contact Red Hat Support .

1.4. REPLACING A FAILED AWS NODE ON INSTALLER-PROVISIONED INFRASTRUCTURE

Perform this procedure to replace a failed node which is not operational on AWS installer-provisioned infrastructure (IPI) for OpenShift Container Storage.

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 Container 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 Container Storage label to the new node:

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

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

CHAPTER 2. OPENSHIFT CONTAINER STORAGE DEPLOYED ON VMWARE

2.1. REPLACING AN OPERATIONAL VMWARE NODE ON USER-PROVISIONED INFRASTRUCTURE

Perform this procedure to replace an operational node on VMware user-provisioned infrastructure (UPI).

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

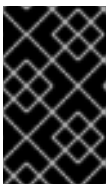
Procedure

1. Identify the node and its VM that needs to be replaced.
2. Mark the node as unschedulable using the following command:

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

3. Drain the node using the following command:

```
$ oc adm drain <node_name> --force --delete-local-data --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.

4. Delete the node using the following command:

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

5. Log in to vSphere and terminate the identified VM.

IMPORTANT

VM should be deleted only from the inventory and not from the disk.

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

```
$ oc get csr
```

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

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

10. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.

11. Apply the OpenShift Container 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 Container Storage label to the new node:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

2.2. REPLACING A FAILED VMWARE NODE ON USER-PROVISIONED INFRASTRUCTURE

Perform this procedure to replace a failed node on VMware user-provisioned infrastructure (UPI).

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

Procedure

1. Identify the node and its VM that needs to be replaced.
2. Delete the node using the following command:

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

3. Log in to vSphere and terminate the identified VM.

IMPORTANT

VM should be deleted only from the inventory and not from the disk.

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

```
$ oc get csr
```

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

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

8. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.
9. Apply the OpenShift Container 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 Container Storage label to the new node:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

CHAPTER 3. OPENSHIFT CONTAINER STORAGE DEPLOYED ON MICROSOFT AZURE

3.1. REPLACING OPERATIONAL NODES ON AZURE INSTALLER-PROVISIONED INFRASTRUCTURE

Use this procedure to replace an operational node on Azure installer-provisioned infrastructure (IPI).

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-local-data --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 Container 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 Container Storage label to the new node:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

3.2. REPLACING FAILED NODES ON AZURE INSTALLER-PROVISIONED INFRASTRUCTURE

Perform this procedure to replace a failed node which is not operational on Azure installer-provisioned infrastructure (IPI) for OpenShift Container Storage.

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 Container 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 Container Storage label to the new node:

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

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

CHAPTER 4. OPENSHIFT CONTAINER STORAGE DEPLOYED USING LOCAL STORAGE DEVICES

4.1. REPLACING STORAGE NODES ON BARE METAL INFRASTRUCTURE

- To replace an operational node, see Section 4.1.1, "Replacing an operational node on bare metal user-provisioned infrastructure"
- To replace a failed node, see Section 4.1.2, "Replacing a failed node on bare metal user-provisioned infrastructure"

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

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.
- If you upgraded to OpenShift Container Storage 4.6 from a previous version instead of performing a fresh installation, ensure that you have completed Post-update configuration changes.

Procedure

1. Identify the node and get labels on the node to be replaced.

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

2. Identify the **mon** (if any) and OSDs 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 node as unschedulable.

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

5. Drain the node.

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

6. Delete the node.

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

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

8. Create a new OpenShift Container Platform node using the new bare metal machine.

9. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

```
$ oc get csr
```

10. Approve all required OpenShift Container Platform 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 Container 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 Container Storage 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 Container Storage 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.

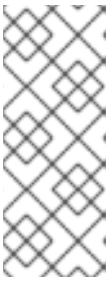
```
$ oc project openshift-storage
```

16. Remove the failed OSD from the cluster.

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
```

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-failed-osd-id -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-failed-osd_id -n openshift-storage --
tail=-1
```

18. 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
local-pv-d6bf175b  1490Gi  RWO  Delete  Released  openshift-storage/ocs-deviceset-0-
data-0-6c5pw  localblock  2d22h  compute-1
```

    b. Delete the PV.

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

    For example:

```
# oc delete pv local-pv-d6bf175b
persistentvolume "local-pv-d9c5cbd6" deleted
```

19. Delete the **crashcollector** pod deployment.

```
$ 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. Execute the following command and 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 Container Storage 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-c-64556f7659-c2ngc                  1/1     Running     0          6h14m
rook-ceph-mon-d-7c8b74dc4d-tt6hd                  1/1     Running     0          4h24m
rook-ceph-mon-e-57fb8c657-wg5f2                   1/1     Running     0          162m
```

   OSD and Mon 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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

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

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

- If you upgraded to OpenShift Container Storage 4.6 from a previous version instead of performing a fresh installation, ensure that you have completed Post-update configuration changes.

Procedure

1. Identify the node and get labels on the node to be replaced.

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

2. Identify the **mon** (if any) and OSDs 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 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-local-data --ignore-daemonsets
```

7. Delete the node.

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

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

9. Create a new OpenShift Container Platform node using the new bare metal machine.

10. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

```
$ oc get csr
```

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

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

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

13. Apply the OpenShift Container 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 Container Storage label to the new node:

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

14. 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 Container Storage 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.

15. 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
```

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

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

17. Remove the failed OSD from the cluster.

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
```

18. 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-failed-osd-id -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-failed-osd_id -n openshift-storage --tail=-1
```

19. 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
    local-pv-d6bf175b  1490Gi  RWO  Delete  Released  openshift-storage/ocs-deviceset-0-data-0-6c5pw  localblock  2d22h  compute-1
    ```

    b. Delete the PV.

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

    For example:

    ```
    # oc delete pv local-pv-d6bf175b
    persistentvolume "local-pv-d9c5cbd6" deleted
    ```

20. Delete the **crashcollector** pod deployment.

    ```
    $ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
    ```

21. 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. Execute the following command and 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 Container Storage 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-c-64556f7659-c2ngc   1/1   Running   0   6h14m
rook-ceph-mon-d-7c8b74dc4d-tt6hd   1/1   Running   0   4h24m
rook-ceph-mon-e-57fb8c657-wg5f2    1/1   Running   0   162m
```

OSD and Mon 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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

4.2. REPLACING STORAGE NODES ON IBM Z OR LINUXONE INFRASTRUCTURE

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

- Section 4.2.1, “Replacing operational nodes on IBM Z or LinuxONE infrastructure”
- Section 4.2.2, “Replacing failed nodes on IBM Z or LinuxONE infrastructure”

4.2.1. Replacing operational nodes on IBM Z or LinuxONE infrastructure

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

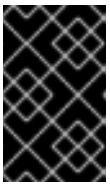
Procedure

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

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

5. Drain the node using the following command:

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```



IMPORTANT

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

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

IMPORTANT

This activity may take at least 5-10 minutes.

10. Click **Compute → Nodes**, confirm if the new node is in **Ready** state.
11. Apply the OpenShift Container 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 Container Storage label to the new node:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **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 data 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 selected host(s).

   ```
   $ oc debug node/<node name>
   $ chroot /host
   ```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

   ```
   $ lsblk
   ```

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

4.2.2. Replacing failed nodes on IBM Z or LinuxONE infrastructure

Perform this procedure to replace a failed node which is not operational on IBM Z or LinuxONE infrastructure for OpenShift Container Storage.

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. 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 Container Storage 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

- Execute the following command to apply the OpenShift Container Storage label to the new node:

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

9. Execute the following command and 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
```

10. Click **Workloads → Pods**, confirm that at least the following pods on the new node are in **Running** state:
    - **csi-cephfsplugin-***
    - **csi-rbdplugin-***
11. Verify that all other required OpenShift Container Storage pods are in **Running** state.
12. 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
```

13. (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 previous step, do the following:
    a. Create a debug pod and open a chroot environment for the selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

    b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

14. If verification steps fail, contact Red Hat Support .

4.3. REPLACING STORAGE NODES ON AMAZON EC2 INFRASTRUCTURE

- To replace an operational Amazon EC2 node on user-provisioned and installer provisioned infrastructures, see:
  - Section 4.3.1, “Replacing an operational Amazon EC2 node on user-provisioned infrastructure”

- Section 4.3.2, "Replacing an operational Amazon EC2 node on installer-provisioned infrastructure"

- To replace a failed Amazon EC2 node on user-provisioned and installer provisioned infrastructures, see:
  - Section 4.3.3, "Replacing a failed Amazon EC2 node on user-provisioned infrastructure"
  - Section 4.3.4, "Replacing a failed Amazon EC2 node on installer-provisioned infrastructure"

4.3.1. Replacing an operational Amazon EC2 node on user-provisioned infrastructure

Perform this procedure to replace an operational node on Amazon EC2 I3 user-provisioned infrastructure (UPI).

IMPORTANT

Replacing storage nodes in Amazon EC2 I3 infrastructure is a Technology Preview feature. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

Procedure

1. Identify the node and get labels on the node to be replaced.

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

2. Identify the mon (if any) and OSDs 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. Drain the node.

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

6. Delete the node.

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

7. Create a new Amazon EC2 I3 machine instance with the required infrastructure. See Supported Infrastructure and Platforms.

8. Create a new OpenShift Container Platform node using the new Amazon EC2 I3 machine instance.

9. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

```
$ oc get csr
```

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

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

11. Click **Compute → Nodes** in the OpenShift web console. Confirm if the new node is in **Ready** state.

12. Apply the OpenShift Container 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 Container Storage label to the new node:

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

13. Add the local storage devices available in the new worker node to the OpenShift Container Storage StorageCluster.

    a. Add the new disk entries to LocalVolume CR.
    Edit **LocalVolume** CR. You can either remove or comment out the failed device **/dev/disk/by-id/{id}** and add the new **/dev/disk/by-id/{id}**.

```
$ oc get -n local-storage localvolume
```

    Example output:

```
NAME          AGE
local-block   25h
```

```
$ oc edit -n local-storage localvolume local-block
```

Example output:

```
[...]
    storageClassDevices:
    - devicePaths:
      - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC
      - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84FE3E9
      - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE4
      - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441464EP
  #   - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84F43E7
  #   - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE8
      - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9
      - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4
      storageClassName: localblock
      volumeMode: Block
[...]
```

Make sure to save the changes after editing the CR.

You can see that in this CR the below two new devices using by-id have been added.

- **nvme-Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9**
- **nvme-Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4**

b. Display PVs with **localblock**.

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

Example output:

```
local-pv-3646185e   2328Gi  RWO    Delete     Available
localblock  9s
local-pv-3933e86    2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-2-1-v9jp4   localblock  5h1m
local-pv-8176b2bf   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-0-0-nvs68   localblock  5h1m
local-pv-ab7cabb3   2328Gi  RWO    Delete     Available
localblock  9s
local-pv-ac52e8a    2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-1-0-knrgr   localblock  5h1m
```

```
local-pv-b7e6fd37   2328Gi  RWO    Delete    Bound    openshift-storage/ocs-deviceset-2-0-rdm7m   localblock  5h1m
local-pv-cb454338   2328Gi  RWO    Delete    Bound    openshift-storage/ocs-deviceset-0-1-h9hfm   localblock  5h1m
local-pv-da5e3175   2328Gi  RWO    Delete    Bound    openshift-storage/ocs-deviceset-1-1-g97lq   localblock  5h
...
```

14. Delete the storage resources associated with the failed node.

    a. Identify the DeviceSet associated with the OSD to be replaced.

    ```
    $ osd_id_to_remove=0
    $ 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-0**.

    Example output:

    ```
    ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
    ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
    ```

    b. Identify the PV associated with the PVC.

    ```
    $ oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
    ```

    where, **x**, **y**, and **pvc-suffix** are the values in the DeviceSet identified in an earlier step.

    Example output:

    ```
    NAME                      STATUS    VOLUME        CAPACITY   ACCESS MODES   STORAGECLASS   AGE
    ocs-deviceset-0-0-nvs68   Bound     local-pv-8176b2bf   2328Gi     RWO            localblock     4h49m
    ```

    In this example, the associated PV is **local-pv-8176b2bf**.

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

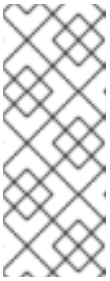
    ```
    $ oc project openshift-storage
    ```

    d. 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 -
    ```

    e. Verify that the OSD is 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
```

f. Delete the PV which was identified in earlier steps. In this example, the PV name is **local-pv-8176b2bf**.

```
$ oc delete pv local-pv-8176b2bf
```

Example output:

```
persistentvolume "local-pv-8176b2bf" deleted
```

15. Delete **crashcollector** pod deployment identified in an earlier step.

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

16. Delete the **ocs-osd-removal** job(s).

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

Example output:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **Running** state. Also, 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-64556f7659-c2ngc   1/1   Running   0   5h1m
rook-ceph-mon-b-7c8b74dc4d-tt6hd   1/1   Running   0   5h1m
rook-ceph-mon-d-57fb8c657-wg5f2    1/1   Running   0   27m
```

OSDs and mon's 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 data 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 selected host(s).

   ```
   $ oc debug node/<node name>
   $ chroot /host
   ```

   b. Run "lsblk" and check for the "crypt" keyword beside the **ocs-deviceset** name(s)

   ```
   $ lsblk
   ```

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

4.3.2. Replacing an operational Amazon EC2 node on installer-provisioned infrastructure

Use this procedure to replace an operational node on Amazon EC2 I3 installer-provisioned infrastructure (IPI).

IMPORTANT

Replacing storage nodes in Amazon EC2 I3 infrastructure is a Technology Preview feature. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

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. Get labels on the node to be replaced.

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

4. Identify the mon (if any) and OSDs that are running in the node to be replaced.

```
$ 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 nodes as unschedulable.

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

7. Drain the node.

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

8. Click **Compute → Machines**. Search for the required machine.
9. Besides the required machine, click the **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 may take at least 5-10 minutes or more.

12. Click **Compute → Nodes** in the OpenShift web console. Confirm if the new node is in **Ready** state.
13. Apply the OpenShift Container 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 Container Storage label to the new node:

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

14. Add the local storage devices available in the new worker node to the OpenShift Container Storage StorageCluster.

    a. Add the new disk entries to LocalVolume CR.
    Edit **LocalVolume** CR. You can either remove or comment out the failed device **/dev/disk/by-id/{id}** and add the new **/dev/disk/by-id/{id}**.

    ```
    $ oc get -n local-storage localvolume
    ```

    Example output:

    ```
    NAME          AGE
    local-block   25h
    ```

    ```
    $ oc edit -n local-storage localvolume local-block
    ```

    Example output:

    ```
    [...]
       storageClassDevices:
       - devicePaths:
         - /dev/disk/by-id/nvme-
    Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC
         - /dev/disk/by-id/nvme-
    Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84FE3E9
         - /dev/disk/by-id/nvme-
    Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE4
         - /dev/disk/by-id/nvme-
    Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441464EP
      #    - /dev/disk/by-id/nvme-
    Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84F43E7
      #    - /dev/disk/by-id/nvme-
    Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE8
         - /dev/disk/by-id/nvme-
    Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9
         - /dev/disk/by-id/nvme-
    Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4
         storageClassName: localblock
         volumeMode: Block
    [...]
    ```

    Make sure to save the changes after editing the CR.

    You can see that in this CR the below two new devices using by-id have been added.

    - **nvme-Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9**
    - **nvme-Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4**

    b. Display PVs with **localblock**.

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

    Example output:

    ```
    -
    ```

```
local-pv-3646185e   2328Gi  RWO    Delete     Available
localblock  9s
local-pv-3933e86    2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-2-1-v9jp4   localblock  5h1m
local-pv-8176b2bf   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-0-0-nvs68   localblock  5h1m
local-pv-ab7cabb3   2328Gi  RWO    Delete     Available
localblock  9s
local-pv-ac52e8a    2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-1-0-knrgr   localblock  5h1m
local-pv-b7e6fd37   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-2-0-rdm7m   localblock  5h1m
local-pv-cb454338   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-0-1-h9hfm   localblock  5h1m
local-pv-da5e3175   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-1-1-g97lq   localblock  5h
...
```

15. Delete the storage resources associated with the failed node.

    a. Identify the DeviceSet associated with the OSD to be replaced.

    ```
    $ osd_id_to_remove=0
    $ 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-0**.

    Example output:

    ```
    ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
    ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
    ```

    b. Identify the PV associated with the PVC.

    ```
    $ oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
    ```

    where, **x**, **y**, and **pvc-suffix** are the values in the DeviceSet identified in an earlier step.

    Example output:

    ```
    NAME                 STATUS       VOLUME       CAPACITY   ACCESS MODES
    STORAGECLASS   AGE
    ocs-deviceset-0-0-nvs68   Bound   local-pv-8176b2bf   2328Gi     RWO          localblock
    4h49m
    ```

    In this example, the associated PV is **local-pv-8176b2bf**.

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

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

    d. 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 -
```

e. Verify that the OSD is 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
```

f. Delete the PV which was identified in earlier steps. In this example, the PV name is **local-pv-8176b2bf**.

```
$ oc delete pv local-pv-8176b2bf
```

Example output:

```
persistentvolume "local-pv-8176b2bf" deleted
```

16. Delete **crashcollector** pod deployment identified in an earlier step.

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

a. Delete the **rook-ceph-operator**.

```
$ oc delete -n openshift-storage pod rook-ceph-operator-6f74fb5bff-2d982
```

Example output:

```
pod "rook-ceph-operator-6f74fb5bff-2d982" deleted
```

b. Verify that the **rook-ceph-operator** pod is restarted.

```
$ oc get -n openshift-storage pod -l app=rook-ceph-operator
```

Example output:

```
NAME                                READY   STATUS    RESTARTS   AGE
rook-ceph-operator-6f74fb5bff-7mvrq   1/1     Running   0          66s
```

Creation of the new OSD may take several minutes after the operator starts.

17. Delete the **ocs-osd-removal** job(s).

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

Example output:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **Running** state. Also, 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-64556f7659-c2ngc   1/1   Running   0   5h1m
rook-ceph-mon-b-7c8b74dc4d-tt6hd   1/1   Running   0   5h1m
rook-ceph-mon-d-57fb8c657-wg5f2    1/1   Running   0   27m
```

OSDs and mon's 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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run "lsblk" and check for the "crypt" keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

4.3.3. Replacing a failed Amazon EC2 node on user-provisioned infrastructure

The ephemeral storage of Amazon EC2 I3 for OpenShift Container Storage might cause data loss when there is an instance power off. Use this procedure to recover from such an instance power off on Amazon EC2 infrastructure.

IMPORTANT

Replacing storage nodes in Amazon EC2 I3 infrastructure is a Technology Preview feature. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

Procedure

1. Identify the node and get labels on the node to be replaced.

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

2. Identify the mon (if any) and OSDs 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 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-local-data --ignore-daemonsets
```

7. Delete the node.

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

8. Create a new Amazon EC2 I3 machine instance with the required infrastructure. See Supported Infrastructure and Platforms.

9. Create a new OpenShift Container Platform node using the new Amazon EC2 I3 machine instance.

10. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

```
$ oc get csr
```

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

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

12. Click **Compute → Nodes** in the OpenShift web console. Confirm if the new node is in **Ready** state.

13. Apply the OpenShift Container 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 Container Storage label to the new node:

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

14. Add the local storage devices available in the new worker node to the OpenShift Container Storage StorageCluster.

    a. Add the new disk entries to LocalVolume CR.
    Edit **LocalVolume** CR. You can either remove or comment out the failed device **/dev/disk/by-id/{id}** and add the new **/dev/disk/by-id/{id}**.

```
$ oc get -n local-storage localvolume
```

    Example output:

```
NAME         AGE
local-block  25h
```

```
$ oc edit -n local-storage localvolume local-block
```

    Example output:

```
[...]
   storageClassDevices:
   - devicePaths:
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84FE3E9
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE4
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441464EP
  #   - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84F43E7
  #   - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE8
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4
     storageClassName: localblock
     volumeMode: Block
[...]
```

Make sure to save the changes after editing the CR.

You can see that in this CR the below two new devices using by-id have been added.

- **nvme-Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9**
- **nvme-Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4**

b. Display PVs with **localblock**.

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

Example output:

```
local-pv-3646185e   2328Gi  RWO    Delete     Available
localblock  9s
local-pv-3933e86    2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-2-1-v9jp4   localblock  5h1m
local-pv-8176b2bf   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-0-0-nvs68   localblock  5h1m
local-pv-ab7cabb3   2328Gi  RWO    Delete     Available
localblock  9s
local-pv-ac52e8a    2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-1-0-knrgr   localblock  5h1m
local-pv-b7e6fd37   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-2-0-rdm7m   localblock  5h1m
local-pv-cb454338   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-0-1-h9hfm   localblock  5h1m
local-pv-da5e3175   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-1-1-g97lq   localblock  5h
...
```

15. Delete the storage resources associated with the failed node.

    a. Identify the DeviceSet associated with the OSD to be replaced.

    ```
    $ osd_id_to_remove=0
    $ 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-0**.

    Example output:

    ```
    ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
    ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
    ```

    b. Identify the PV associated with the PVC.

    ```
    $ oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
    ```

    where, **x**, **y**, and **pvc-suffix** are the values in the DeviceSet identified in an earlier step.

    Example output:

    ```
    NAME                     STATUS    VOLUME        CAPACITY   ACCESS MODES
    STORAGECLASS   AGE
    ocs-deviceset-0-0-nvs68  Bound   local-pv-8176b2bf  2328Gi     RWO           localblock
    4h49m
    ```

    In this example, the associated PV is **local-pv-8176b2bf**.

    c. Change into the **openshift-storage** project.

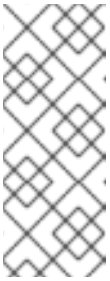
    ```
    $ oc project openshift-storage
    ```

    d. Remove the failed OSD from the cluster.

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

    e. Verify that the OSD is 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
    ```

f. Delete the PV which was identified in earlier steps. In this example, the PV name is **local-pv-8176b2bf**.

```
$ oc delete pv local-pv-8176b2bf
```

Example output:

```
persistentvolume "local-pv-8176b2bf" deleted
```

16. Delete **crashcollector** pod deployment identified in an earlier step.

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

17. Delete the **ocs-osd-removal** job(s).

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

Example output:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **Running** state. Also, 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-64556f7659-c2ngc   1/1   Running   0   5h1m
rook-ceph-mon-b-7c8b74dc4d-tt6hd   1/1   Running   0   5h1m
rook-ceph-mon-d-57fb8c657-wg5f2    1/1   Running   0   27m
```

OSDs and mon's 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 data 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 selected host(s).

   ```
   $ oc debug node/<node name>
   $ chroot /host
   ```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

   ```
   $ lsblk
   ```

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

4.3.4. Replacing a failed Amazon EC2 node on installer-provisioned infrastructure

The ephemeral storage of Amazon EC2 I3 for OpenShift Container Storage might cause data loss when there is an instance power off. Use this procedure to recover from such an instance power off on Amazon EC2 infrastructure.

IMPORTANT

Replacing storage nodes in Amazon EC2 I3 infrastructure is a Technology Preview feature. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.

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. Get the labels on the node to be replaced.

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

4. Identify the mon (if any) and OSDs that are running in the node to be replaced.

   ```
   $ 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-local-data --ignore-daemonsets
```

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

IMPORTANT

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

13. Click **Compute → Nodes** in the OpenShift web console. Confirm if the new node is in **Ready** state.
14. Apply the OpenShift Container 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 Container Storage label to the new node:

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

15. Add the local storage devices available in the new worker node to the OpenShift Container Storage StorageCluster.

    a. Add the new disk entries to LocalVolume CR.

Edit **LocalVolume** CR. You can either remove or comment out the failed device **/dev/disk/by-id/{id}** and add the new **/dev/disk/by-id/{id}**.

```
$ oc get -n local-storage localvolume
```

Example output:

```
NAME          AGE
local-block   25h
```

```
$ oc edit -n local-storage localvolume local-block
```

Example output:

```
[...]
   storageClassDevices:
   - devicePaths:
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84FE3E9
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE4
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441464EP
  #   - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84F43E7
  #   - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE8
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9
     - /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4
     storageClassName: localblock
     volumeMode: Block
[...]
```

Make sure to save the changes after editing the CR.

You can see that in this CR the below two new devices using by-id have been added.

- **nvme-Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9**
- **nvme-Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4**

b. Display PVs with **localblock**.

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

Example output:

```
local-pv-3646185e   2328Gi  RWO     Delete      Available
localblock  9s
local-pv-3933e86    2328Gi  RWO     Delete      Bound       openshift-storage/ocs-
```

```
deviceset-2-1-v9jp4   localblock  5h1m
local-pv-8176b2bf   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-0-0-nvs68   localblock  5h1m
local-pv-ab7cabb3   2328Gi  RWO    Delete     Available
localblock  9s
local-pv-ac52e8a   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-1-0-knrgr   localblock  5h1m
local-pv-b7e6fd37   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-2-0-rdm7m   localblock  5h1m
local-pv-cb454338   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-0-1-h9hfm   localblock  5h1m
local-pv-da5e3175   2328Gi  RWO    Delete     Bound      openshift-storage/ocs-
deviceset-1-1-g97lq   localblock  5h
...
```

16. Delete the storage resources associated with the failed node.

    a. Identify the DeviceSet associated with the OSD to be replaced.

    ```
    $ osd_id_to_remove=0
    $ 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-0**.

    Example output:

    ```
    ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
    ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
    ```

    b. Identify the PV associated with the PVC.

    ```
    $ oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
    ```

    where, **x**, **y**, and **pvc-suffix** are the values in the DeviceSet identified in an earlier step.

    Example output:

    ```
    NAME                  STATUS    VOLUME       CAPACITY   ACCESS MODES
    STORAGECLASS   AGE
    ocs-deviceset-0-0-nvs68   Bound   local-pv-8176b2bf   2328Gi     RWO           localblock
    4h49m
    ```

    In this example, the associated PV is **local-pv-8176b2bf**.

    c. Change into the **openshift-storage** project.

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

    d. Remove the failed OSD from the cluster.

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

e. Verify that the OSD is 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
```

f. Delete the PV which was identified in earlier steps. In this example, the PV name is **local-pv-8176b2bf**.

```
$ oc delete pv local-pv-8176b2bf
```

Example output:

```
persistentvolume "local-pv-8176b2bf" deleted
```

17. Delete **crashcollector** pod deployment identified in an earlier step.

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

18. Delete the **ocs-osd-removal** job(s).

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

Example output:

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

Verification steps

1. Execute the following command and 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 Container Storage pods are in **Running** state. Also, 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-64556f7659-c2ngc   1/1   Running   0   5h1m
rook-ceph-mon-b-7c8b74dc4d-tt6hd   1/1   Running   0   5h1m
rook-ceph-mon-d-57fb8c657-wg5f2    1/1   Running   0   27m
```

OSDs and mon's 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 data 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 selected host(s).

   ```
   $ oc debug node/<node name>
   $ chroot /host
   ```

   b. Run "lsblk" and check for the "crypt" keyword beside the **ocs-deviceset** name(s)

   ```
   $ lsblk
   ```

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

4.4. REPLACING STORAGE NODES ON VMWARE INFRASTRUCTURE

- To replace an operational node, see Section 4.4.1, "Replacing an operational node on VMware user-provisioned infrastructure"
- To replace a failed node, see Section 4.4.2, "Replacing a failed node on VMware user-provisioned infrastructure"

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

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.
- If you upgraded to OpenShift Container Storage 4.6 from a previous version instead of performing a fresh installation, ensure that you have completed Post-update configuration changes.

Procedure

1. Identify the node and get labels on the node to be replaced.

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

2. Identify the **mon** (if any) and OSDs 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 node as unschedulable.

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

5. Drain the node.

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

6. Delete the node.

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

7. Log in to vSphere and terminate the identified VM.

8. Create a new VM on VMware with the required infrastructure. See Supported Infrastructure and Platforms.

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

10. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

```
$ oc get csr
```

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

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

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

13. Apply the OpenShift Container 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 Container Storage label to the new node:

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

14. 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 Container Storage 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.

15. 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
```

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

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

17. Remove the failed OSD from the cluster.

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
```

18. 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-failed-osd-id -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-failed-osd_id -n openshift-storage --
tail=-1
```

19. 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
local-pv-d6bf175b  1490Gi  RWO  Delete  Released  openshift-storage/ocs-deviceset-0-
data-0-6c5pw  localblock  2d22h  compute-1
```

    b. Delete the PV.

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

For example:

```
# oc delete pv local-pv-d6bf175b
persistentvolume "local-pv-d9c5cbd6" deleted
```

20. Delete the **crashcollector** pod deployment.

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

21. 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. Execute the following command and 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 Container Storage 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-c-64556f7659-c2ngc   1/1   Running   0   6h14m
rook-ceph-mon-d-7c8b74dc4d-tt6hd   1/1   Running   0   4h24m
rook-ceph-mon-e-57fb8c657-wg5f2    1/1   Running   0   162m
```

OSD and Mon 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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

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

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure, resources, and disks to the node being replaced.
- You must be logged into the OpenShift Container Platform (RHOCP) cluster.
- If you upgraded to OpenShift Container Storage 4.6 from a previous version instead of performing a fresh installation, ensure that you have completed Post-update configuration changes.

Procedure

1. Identify the node and get labels on the node to be replaced.

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

2. Identify the **mon** (if any) and OSDs 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 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-local-data --ignore-daemonsets
```

7. Delete the node.

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

8. Log in to vSphere and terminate the identified VM.

9. Create a new VM on VMware with the required infrastructure. See Supported Infrastructure and Platforms.

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

11. Check for certificate signing requests (CSRs) related to OpenShift Container Platform that are in Pending state:

```
$ oc get csr
```

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

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

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

14. Apply the OpenShift Container 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 Container Storage label to the new node:

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

15. 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 Container Storage 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.

16. 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-d9c5cbd6  931Gi  RWO  Delete  Bound  openshift-storage/ocs-deviceset-0-0-nvs68  localblock  25h
```

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

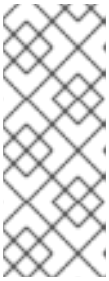
```
$ oc project openshift-storage
```

18. Remove the failed OSD from the cluster.

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=failed-osd-id1,failed-osd-id2 | oc create -f -
```

19. 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-failed-osd-id -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-failed-osd_id -n openshift-storage --tail=-1
```

20. 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
    local-pv-d6bf175b  1490Gi  RWO  Delete  Released  openshift-storage/ocs-deviceset-0-data-0-6c5pw  localblock  2d22h  compute-1
    ```

    b. Delete the PV.

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

    For example:

    ```
    # oc delete pv local-pv-d6bf175b
    persistentvolume "local-pv-d9c5cbd6" deleted
    ```

21. Delete the **crashcollector** pod deployment.

```
$ 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 job ocs-osd-removal-${osd_id_to_remove}
```

Example output:

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

Verification steps

1. Execute the following command and 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 Container Storage 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-c-64556f7659-c2ngc   1/1   Running   0   6h14m
rook-ceph-mon-d-7c8b74dc4d-tt6hd   1/1   Running   0   4h24m
rook-ceph-mon-e-57fb8c657-wg5f2    1/1   Running   0   162m
```

OSD and Mon 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 data 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 selected host(s).

```
$ oc debug node/<node name>
$ chroot /host
```

   b. Run “lsblk” and check for the “crypt” keyword beside the **ocs-deviceset** name(s)

```
$ lsblk
```

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

4.5. REPLACING STORAGE NODES ON IBM POWER SYSTEMS INFRASTRUCTURE

For OpenShift Container Storage, node replacement can be performed proactively for an operational node and reactively for a failed node for the IBM Power Systems related deployments.

4.5.1. Replacing an operational or failed storage node on IBM Power Systems

Prerequisites

- Red Hat recommends that replacement nodes are configured with similar infrastructure and resources to the node being replaced.
- You must be logged into OpenShift Container Platform (RHOCP) cluster.

Procedure

1. Check the labels on the failed node and make note of the rack label.

```
$ oc get nodes --show-labels | grep failed-node-name
```

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

```
$ oc get pods -n openshift-storage -o wide | grep -i failed-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=failed-node-name --replicas=0 -n openshift-storage
```

4. Mark the failed node so that it cannot be scheduled for work.

```
$ oc adm cordon failed-node-name
```

5. Drain the failed node of existing work.

```
$ oc adm drain failed-node-name --force --delete-local-data --ignore-daemonsets
```

NOTE

If the failed node is not connected to the network, remove the pods running on it by using the command:

```
$ oc get pods -A -o wide | grep -i failed-node-name |  awk '{if ($4 == "Terminating") system ("oc -n " $1 " delete pods " $2  " --grace-period=0 " " --force ")}'
$ oc adm drain failed-node-name --force --delete-local-data --ignore-daemonsets
```

6. Delete the failed node.

```
$ oc delete node failed-node-name
```

7. Get a new IBM Power machine with required infrastructure. See Installing a cluster on IBM Power Systems.

8. Create a new OpenShift Container Platform node using the new IBM Power Systems machine.

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

```
$ oc get csr
```

10. Approve all required OpenShift Container Storage 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 Container Storage label to the new node using your preferred interface:

- **From OpenShift web console**

  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. Execute the following command to apply the OpenShift Container Storage label to the new node:

```
$ oc label node new-node-name cluster.ocs.openshift.io/openshift-storage=""
```

13. Add a newly added worker node to localVolumeSet.

a. 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 Container Storage 4.6 and later. Previous versions use **local-storage** by default.

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

b. 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:
            #- worker-0
            - worker-1
            - worker-2
            - worker-3
[...]
```

Remember to save before exiting the editor.

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

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

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

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

16. Remove the failed OSD from the cluster.

    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},{osd_id_to_remove2} | oc create -f -
    ```

17. Verify that the OSD is 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
```

18. Delete the PV associated with the failed node.

    a. Identify the PV associated with the PVC.

    ```
    # oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
    ```

    where, **x**, **y**, and **pvc-suffix** are the values in the **DeviceSet** identified in the previous step.

    For example:

    ```
    # oc get -n openshift-storage pvc ocs-deviceset-localblock-0-data-0-g2mmc
    NAME                                  STATUS   VOLUME                 CAPACITY   ACCESS MODES   STORAGECLASS   AGE
    ocs-deviceset-localblock-0-data-0-g2mmc   Bound   local-pv-5c9b8982   500Gi      RWO            localblock     24h
    ```

    In this example, the associated PV is **local-pv-5c9b8982**.

    b. Delete the PV.

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

    For example:

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

19. Delete the **crashcollector** pod deployment.

    ```
    $ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
    ```

20. Deploy the new OSD by restarting the **rook-ceph-operator** to force operator reconciliation.

    ```
    # oc get -n openshift-storage pod -l app=rook-ceph-operator
    ```

    Example output:

    ```
    NAME                                  READY   STATUS    RESTARTS   AGE
    rook-ceph-operator-77758ddc74-dlwn2   1/1     Running   0          1d20h
    ```

    a. Delete the **rook-ceph-operator**.

```
# oc delete -n openshift-storage pod rook-ceph-operator-77758ddc74-dlwn2
```

Example output:

```
pod "rook-ceph-operator-77758ddc74-dlwn2" deleted
```

21. Verify that the **rook-ceph-operator** pod is restarted.

```
# oc get -n openshift-storage pod -l app=rook-ceph-operator
```

Example output:

```
NAME                                  READY   STATUS    RESTARTS   AGE
rook-ceph-operator-77758ddc74-wqf25   1/1     Running   0          66s
```

Creation of the new OSD and **mon** might take several minutes after the operator restarts.

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

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

For example:

```
# oc delete job ocs-osd-removal-1
job.batch "ocs-osd-removal-1" deleted
```

Verification steps

- Execute the following command and 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
```

- Click **Workloads → Pods**, confirm that at least the following pods on the new node are in **Running** state:
  - **csi-cephfsplugin-***
  - **csi-rbdplugin-***
- Verify that all other required OpenShift Container Storage pods are in **Running** state.
  - Make sure 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
```

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

- If verification steps fail, contact Red Hat Support .