Red Hat OpenShift Data Foundation 4.12

Replacing devices

Instructions for safely replacing operational or failed devices

Last Updated: 2023-08-15

Red Hat OpenShift Data Foundation 4.12 Replacing devices

Instructions for safely replacing operational or failed devices

Legal Notice

Abstract

This document explains how to safely replace storage devices for Red Hat OpenShift Data Foundation.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

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

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

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

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the Bugzilla website.
  2. In the **Component** section, choose **documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

PREFACE

Depending on the type of your deployment, you can choose one of the following procedures to replace a storage device:

- For dynamically created storage clusters deployed on AWS, see:
  - Section 1.1, “Replacing operational or failed storage devices on AWS user-provisioned infrastructure”.
  - Section 1.2, “Replacing operational or failed storage devices on AWS installer-provisioned infrastructure”.
- For dynamically created storage clusters deployed on VMware, see Section 2.1, “Replacing operational or failed storage devices on VMware infrastructure”.
- For dynamically created storage clusters deployed on Red Hat Virtualization, see Section 3.1, “Replacing operational or failed storage devices on Red Hat Virtualization installer-provisioned infrastructure”.
- For dynamically created storage clusters deployed on Microsoft Azure, see Section 4.1, “Replacing operational or failed storage devices on Azure installer-provisioned infrastructure”.
- For storage clusters deployed using local storage devices, see:
  - Section 5.1, “Replacing operational or failed storage devices on clusters backed by local storage devices”.
  - Section 5.2, “Replacing operational or failed storage devices on IBM Power” .
  - Section 5.3, “Replacing operational or failed storage devices on IBM Z or LinuxONE infrastructure”.

NOTE

OpenShift Data Foundation does not support heterogeneous OSD sizes.

CHAPTER 1. DYNAMICALLY PROVISIONED OPENSHIFT DATA FOUNDATION DEPLOYED ON AWS

1.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON AWS USER-PROVISIONED INFRASTRUCTURE

When you need to replace a device in a dynamically created storage cluster on an AWS user-provisioned infrastructure, you must replace the storage node. For information about how to replace nodes, see:

- Replacing an operational AWS node on user-provisioned infrastructure .
- Replacing a failed AWS node on user-provisioned infrastructure .

1.2. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON AWS INSTALLER-PROVISIONED INFRASTRUCTURE

When you need to replace a device in a dynamically created storage cluster on an AWS installer-provisioned infrastructure, you must replace the storage node. For information about how to replace nodes, see:

- Replacing an operational AWS node on installer-provisioned infrastructure .
- Replacing a failed AWS node on installer-provisioned infrastructure .

CHAPTER 2. DYNAMICALLY PROVISIONED OPENSHIFT DATA FOUNDATION DEPLOYED ON VMWARE

2.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON VMWARE INFRASTRUCTURE

Create a new Persistent Volume Claim (PVC) on a new volume, and remove the old object storage device (OSD) when one or more virtual machine disks (VMDK) needs to be replaced in OpenShift Data Foundation which is deployed dynamically on VMware infrastructure.

Prerequisites

- Ensure that the data is resilient.
  - In the OpenShift Web Console, click **Storage → Data Foundation**.
  - Click the **Storage Systems** tab, and then click **ocs-storagecluster-storagesystem**.
  - In the **Status card** of **Block and File** dashboard, under the **Overview** tab, verify that *Data Resiliency* has a green tick mark.

Procedure

1. Identify the OSD that needs to be replaced and the OpenShift Container Platform node that has the OSD scheduled on it.

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

   Example output:

```
rook-ceph-osd-0-6d77d6c7c6-m8xj6   0/1   CrashLoopBackOff   0   24h   10.129.0.16
compute-2   <none>          <none>
rook-ceph-osd-1-85d99fb95f-2svc7   1/1   Running            0   24h   10.128.2.24   compute-
0   <none>          <none>
rook-ceph-osd-2-6c66cdb977-jp542   1/1   Running            0   24h   10.130.0.18   compute-
1   <none>          <none>
```

   In this example, **rook-ceph-osd-0-6d77d6c7c6-m8xj6** needs to be replaced and **compute-2** is the OpenShift Container platform node on which the OSD is scheduled.

   NOTE

   If the OSD to be replaced is healthy, the status of the pod will be **Running**.

2. Scale down the OSD deployment for the OSD to be replaced.
   Each time you want to replace the OSD, update the **osd_id_to_remove** parameter with the OSD ID, and repeat this step.

```
$ osd_id_to_remove=0
```

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

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:

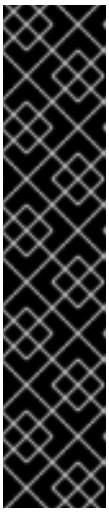
```
deployment.extensions/rook-ceph-osd-0 scaled
```

3. Verify that the **rook-ceph-osd** pod is terminated.

```
$ oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
```

Example output:

```
No resources found.
```



IMPORTANT

If the **rook-ceph-osd** pod is in **terminating** state, use the **force** option to delete the pod.

```
$ oc delete pod rook-ceph-osd-0-6d77d6c7c6-m8xj6 --force --grace-period=0
```

Example output:

```
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
  pod "rook-ceph-osd-0-6d77d6c7c6-m8xj6" force deleted
```

4. Remove the old OSD from the cluster so that you can add a new OSD.

   a. Delete any old **ocs-osd-removal** jobs.

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

   Example output:

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

   b. Navigate to the **openshift-storage** project.

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

   c. Remove the old OSD from the cluster.

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

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

WARNING

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

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

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

6. Ensure that the OSD removal is completed.

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

   Example output:

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

IMPORTANT

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

For example:

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

7. If encryption was enabled at the time of install, remove **dm-crypt** managed **device-mapper** mapping from the OSD devices that are removed from the respective OpenShift Data Foundation nodes.

   a. Get the PVC name(s) of the replaced OSD(s) from the logs of **ocs-osd-removal-job** pod.

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1  |egrep -i ‘pvc|deviceset’
```

      Example output:

```
2021-05-12 14:31:34.666000 I | cephosd: removing the OSD PVC "ocs-deviceset-xxxx-xxx-xxx-xxx"
```

   b. For each of the previously identified nodes, do the following:

      i. Create a **debug** pod and **chroot** to the host on the storage node.

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

<node name>

Is the name of the node.

```
$ chroot /host
```

ii. Find a relevant device name based on the PVC names identified in the previous step.

```
$ dmsetup ls| grep <pvc name>
```

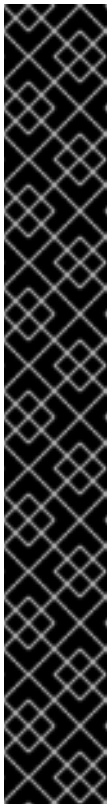
<pvc name>

Is the name of the PVC.
Example output:

```
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt (253:0)
```

iii. Remove the mapped device.

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt
```



IMPORTANT

If the above command gets stuck due to insufficient privileges, run the following commands:

- Press **CTRL+Z** to exit the above command.
- Find the PID of the process which was stuck.

```
$ ps -ef | grep crypt
```

- Terminate the process using the **kill** command.

```
$ kill -9 <PID>
```

<PID>

Is the process ID.

- Verify that the device name is removed.

```
$ dmsetup ls
```

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

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

Example output:

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

NOTE

When using an external key management system (KMS) with data encryption, the old OSD encryption key can be removed from the Vault server as it is now an orphan key.

Verfication steps

1. Verify that there is a new OSD running.

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd
```

Example output:

```
rook-ceph-osd-0-5f7f4747d4-snshw                 1/1     Running     0          4m47s
rook-ceph-osd-1-85d99fb95f-2svc7                 1/1     Running     0          1d20h
rook-ceph-osd-2-6c66cdb977-jp542                 1/1     Running     0          1d20h
```

2. Verify that there is a new PVC created which is in **Bound** state.

```
$ oc get -n openshift-storage pvc
```

Example output:

```
NAME                      STATUS   VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS    AGE
ocs-deviceset-0-0-2s6w4   Bound    pvc-7c9bcaf7-de68-40e1-95f9-0b0d7c0ae2fc   512Gi
RWO           thin           5m
ocs-deviceset-1-0-q8fwh   Bound    pvc-9e7e00cb-6b33-402e-9dc5-b8df4fd9010f   512Gi
RWO           thin           1d20h
ocs-deviceset-2-0-9v8lq   Bound    pvc-38cdfcee-ea7e-42a5-a6e1-aaa6d4924291   512Gi
RWO           thin           1d20h
```

3. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   a. Identify the nodes where the new OSD pods are running.

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-pod-name>
```

<OSD-pod-name>

Is the name of the OSD pod.
For example:

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

Example output:

```
NODE
compute-1
```

b. For each of the nodes identified in the previous step, do the following:

i. Create a debug pod and open a chroot environment for the selected host(s).

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

<node name>

Is the name of the node.

```
$ chroot /host
```

ii. Check for the **crypt** keyword beside the **ocs-deviceset** name(s).

```
$ lsblk
```

4. Log in to OpenShift Web Console and view the storage dashboard.

CHAPTER 3. DYNAMICALLY PROVISIONED OPENSHIFT DATA FOUNDATION DEPLOYED ON RED HAT VIRTUALIZATION

3.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON RED HAT VIRTUALIZATION INSTALLER-PROVISIONED INFRASTRUCTURE

Create a new Persistent Volume Claim (PVC) on a new volume, and remove the old object storage device (OSD).

Prerequisites

- Ensure that the data is resilient.
  - In the OpenShift Web Console, click **Storage → Data Foundation**.
  - Click the **Storage Systems** tab, and then click **ocs-storagecluster-storagesystem**.
  - In the **Status card** of **Block and File** dashboard, under the **Overview** tab, verify that *Data Resiliency* has a green tick mark.

Procedure

1. Identify the OSD that needs to be replaced and the OpenShift Container Platform node that has the OSD scheduled on it.

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

Example output:

```
rook-ceph-osd-0-6d77d6c7c6-m8xj6   0/1   CrashLoopBackOff   0   24h   10.129.0.16
compute-2   <none>          <none>
rook-ceph-osd-1-85d99fb95f-2svc7   1/1   Running            0   24h   10.128.2.24   compute-
0   <none>          <none>
rook-ceph-osd-2-6c66cdb977-jp542   1/1   Running            0   24h   10.130.0.18   compute-
1   <none>          <none>
```

In this example, **rook-ceph-osd-0-6d77d6c7c6-m8xj6** needs to be replaced and **compute-2** is the OpenShift Container platform node on which the OSD is scheduled.

NOTE

If the OSD to be replaced is healthy, the status of the pod will be **Running**.

2. Scale down the OSD deployment for the OSD to be replaced.
Each time you want to replace the OSD, update the **osd_id_to_remove** parameter with the OSD ID, and repeat this step.

```
$ osd_id_to_remove=0
```

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

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:

```
deployment.extensions/rook-ceph-osd-0 scaled
```

3. Verify that the **rook-ceph-osd** pod is terminated.

```
$ oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
```

Example output:

```
No resources found.
```

IMPORTANT

If the **rook-ceph-osd** pod is in **terminating** state, use the **force** option to delete the pod.

```
$ oc delete pod rook-ceph-osd-0-6d77d6c7c6-m8xj6 --force --grace-period=0
```

Example output:

```
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
  pod "rook-ceph-osd-0-6d77d6c7c6-m8xj6" force deleted
```

4. Remove the old OSD from the cluster so that you can add a new OSD.

   a. Delete any old **ocs-osd-removal** jobs.

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

   Example output:

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

   b. Navigate to the **openshift-storage** project.

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

   c. Remove the old OSD from the cluster.

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

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

WARNING

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

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

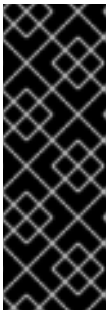
```
$ oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

6. Ensure that the OSD removal is completed.

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

   Example output:

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



IMPORTANT

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

For example:

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

7. If encryption was enabled at the time of install, remove **dm-crypt** managed **device-mapper** mapping from the OSD devices that are removed from the respective OpenShift Data Foundation nodes.

   a. Get the PVC name(s) of the replaced OSD(s) from the logs of **ocs-osd-removal-job** pod.

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

      Example output:

```
2021-05-12 14:31:34.666000 I | cephosd: removing the OSD PVC "ocs-deviceset-xxxx-xxx-xxx-xxx"
```

   b. For each of the previously identified nodes, do the following:

i. Create a **debug** pod and **chroot** to the host on the storage node.

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

<node name>

Is the name of the node.

```
$ chroot /host
```

ii. Find a relevant device name based on the PVC names identified in the previous step.

```
$ dmsetup ls| grep <pvc name>
```

<pvc name>

Is the name of the PVC.
Example output:

```
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt (253:0)
```

iii. Remove the mapped device.

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt
```

IMPORTANT

If the above command gets stuck due to insufficient privileges, run the following commands:

- Press **CTRL+Z** to exit the above command.
- Find the PID of the process which was stuck.

```
$ ps -ef | grep crypt
```

- Terminate the process using the **kill** command.

```
$ kill -9 <PID>
```

<PID>

Is the process ID.

- Verify that the device name is removed.

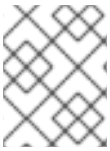
```
$ dmsetup ls
```

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

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

Example output:

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



NOTE

When using an external key management system (KMS) with data encryption, the old OSD encryption key can be removed from the Vault server as it is now an orphan key.

Verfication steps

1. Verify that there is a new OSD running.

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd
```

Example output:

```
rook-ceph-osd-0-5f7f4747d4-snshw   1/1   Running   0   4m47s
rook-ceph-osd-1-85d99fb95f-2svc7   1/1   Running   0   1d20h
rook-ceph-osd-2-6c66cdb977-jp542   1/1   Running   0   1d20h
```

2. Verify that there is a new PVC created which is in **Bound** state.

```
$ oc get -n openshift-storage pvc
```

3. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   a. Identify the nodes where the new OSD pods are running.

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-pod-name>
```

   <OSD-pod-name>

   Is the name of the OSD pod.
   For example:

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

   Example output:

```
NODE
compute-1
```

   b. For each of the previously identified nodes, do the following:

      i. Create a debug pod and open a chroot environment for the selected host(s).

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

<node name>

Is the name of the node.

```
$ chroot /host
```

ii. Check for the **crypt** keyword beside the **ocs-deviceset** name(s).

```
$ lsblk
```

4. Log in to OpenShift Web Console and view the storage dashboard.

CHAPTER 4. DYNAMICALLY PROVISIONED OPENSHIFT DATA FOUNDATION DEPLOYED ON MICROSOFT AZURE

4.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON AZURE INSTALLER-PROVISIONED INFRASTRUCTURE

When you need to replace a device in a dynamically created storage cluster on an Azure installer-provisioned infrastructure, you must replace the storage node. For information about how to replace nodes, see:

- Replacing operational nodes on Azure installer-provisioned infrastructure.
- Replacing failed nodes on Azure installer-provisioned infrastructures.

CHAPTER 5. OPENSHIFT DATA FOUNDATION DEPLOYED USING LOCAL STORAGE DEVICES

5.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON CLUSTERS BACKED BY LOCAL STORAGE DEVICES

You can replace an object storage device (OSD) in OpenShift Data Foundation deployed using local storage devices on the following infrastructures:

- Bare metal
- VMware
- Red Hat Virtualization

NOTE

One or more underlying storage devices may need to be replaced.

Prerequisites

- Red Hat recommends that replacement devices are configured with similar infrastructure and resources to the device being replaced.
- Ensure that the data is resilient.
  - In the OpenShift Web Console, click **Storage → Data Foundation**.
  - Click the **Storage Systems** tab, and then click **ocs-storagecluster-storagesystem**.
  - In the **Status card** of **Block and File** dashboard, under the **Overview** tab, verify that *Data Resiliency* has a green tick mark.

Procedure

1. Remove the underlying storage device from relevant worker node.
2. Verify that relevant OSD Pod has moved to CrashLoopBackOff state.
   Identify the OSD that needs to be replaced and the OpenShift Container Platform node that has the OSD scheduled on it.

   ```
   $ oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
   ```

   Example output:

   ```
   rook-ceph-osd-0-6d77d6c7c6-m8xj6   0/1   CrashLoopBackOff   0   24h   10.129.0.16
   compute-2   <none>          <none>
   rook-ceph-osd-1-85d99fb95f-2svc7   1/1   Running            0   24h   10.128.2.24   compute-
   0   <none>          <none>
   rook-ceph-osd-2-6c66cdb977-jp542   1/1   Running            0   24h   10.130.0.18   compute-
   1   <none>          <none>
   ```

In this example, **rook-ceph-osd-0-6d77d6c7c6-m8xj6** needs to be replaced and **compute-2** is the OpenShift Container platform node on which the OSD is scheduled.

3. Scale down the OSD deployment for the OSD to be replaced.

```
$ osd_id_to_remove=0
```

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

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:

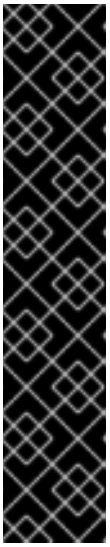
```
deployment.extensions/rook-ceph-osd-0 scaled
```

4. Verify that the **rook-ceph-osd** pod is terminated.

```
$ oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
```

Example output:

```
No resources found in openshift-storage namespace.
```



IMPORTANT

If the **rook-ceph-osd** pod is in **terminating** state for more than a few minutes, use the **force** option to delete the pod.

```
$ oc delete -n openshift-storage pod rook-ceph-osd-0-6d77d6c7c6-m8xj6 --grace-period=0 --force
```

Example output:

```
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
 pod "rook-ceph-osd-0-6d77d6c7c6-m8xj6" force deleted
```

5. Remove the old OSD from the cluster so that you can add a new OSD.

   a. Delete any old **ocs-osd-removal** jobs.

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

   Example output:

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

   b. Navigate to the **openshift-storage** project.

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

c. Remove the old OSD from the cluster.

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

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

WARNING

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

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

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

7. Ensure that the OSD removal is completed.

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

Example output:

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

IMPORTANT

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

For example:

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

8. If encryption was enabled at the time of install, remove **dm-crypt** managed **device-mapper** mapping from the OSD devices that are removed from the respective OpenShift Data Foundation nodes.

a. Get the Persistent Volume Claim (PVC) name(s) of the replaced OSD(s) from the logs of **ocs-osd-removal-job** pod.

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

Example output:

```
2021-05-12 14:31:34.666000 I | cephosd: removing the OSD PVC "ocs-deviceset-xxxx-
xxx-xxx-xxx"
```

b. For each of the previously identified nodes, do the following:

i. Create a **debug** pod and **chroot** to the host on the storage node.

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

<node name>

Is the name of the node.

```
$ chroot /host
```

ii. Find the relevant device name based on the PVC names identified in the previous step.

```
$ dmsetup ls| grep <pvc name>
```

<pvc name>

Is the name of the PVC.
Example output:

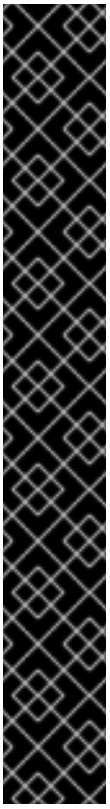
```
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt (253:0)
```

iii. Remove the mapped device.

```
$ cryptsetup luksClose --debug --verbose <ocs-deviceset-name>
```

<ocs-deviceset-name>

Is the name of the relevant device based on the PVC names identified in the previous step.



IMPORTANT

If the above command gets stuck due to insufficient privileges, run the following commands:

- Press **CTRL+Z** to exit the above command.
- Find the PID of the process which was stuck.

```
$ ps -ef | grep crypt
```

- Terminate the process using the **kill** command.

```
$ kill -9 <PID>
```

<PID>

Is the process ID.

- Verify that the device name is removed.

```
$ dmsetup ls
```

9. Find the persistent volume (PV) that need to be deleted.

```
$ oc get pv -L kubernetes.io/hostname | grep <storageclass-name> | grep Released
```

Example output:

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

10. Delete the PV.

```
$ oc delete pv <pv_name>
```

11. Physically add a new device to the node.

12. Track the provisioning of PVs for the devices that match the **deviceInclusionSpec**. It can take a few minutes to provision the PVs.

```
$ oc -n openshift-local-storage describe localvolumeset localblock
```

Example output:

```
[...]
Status:
  Conditions:
    Last Transition Time:         2020-11-17T05:03:32Z
    Message:                      DiskMaker: Available, LocalProvisioner: Available
    Status:                       True
    Type:                         DaemonSetsAvailable
    Last Transition Time:         2020-11-17T05:03:34Z
```

```
    Message:                    Operator reconciled successfully.
    Status:                     True
    Type:                       Available
  Observed Generation:          1
  Total Provisioned Device Count: 4
Events:
Type    Reason      Age         From              Message
----    ------      ----        ----              -------
Normal  Discovered  2m30s (x4   localvolumeset-    node.example.com -
        NewDevice   over 2m30s)  symlink-controller  found possible
                                                    matching disk,
                                                    waiting 1m to claim

Normal  FoundMatch  89s (x4     localvolumeset-    node.example.com -
        ingDisk     over 89s)   symlink-controller  symlinking matching
                                                    disk
```

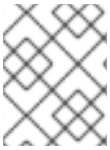
Once the PV is provisioned, a new OSD pod is automatically created for the PV.

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

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

Example output:

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



NOTE

When using an external key management system (KMS) with data encryption, the old OSD encryption key can be removed from the Vault server as it is now an orphan key.

Verification steps

1. Verify that there is a new OSD running.

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd
```

Example output:

```
rook-ceph-osd-0-5f7f4747d4-snshw    1/1     Running     0          4m47s
rook-ceph-osd-1-85d99fb95f-2svc7    1/1     Running     0          1d20h
rook-ceph-osd-2-6c66cdb977-jp542    1/1     Running     0          1d20h
```

IMPORTANT

If the new OSD does not show as **Running** after a few minutes, restart the **rook-ceph-operator** pod to force a reconciliation.

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

Example output:

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

2. Verify that a new PVC is created.

```
$ oc get -n openshift-storage pvc | grep localblock
```

Example output:

```
ocs-deviceset-0-0-c2mqb   Bound   local-pv-b481410    1490Gi   RWO   localblock   5m
ocs-deviceset-1-0-959rp   Bound   local-pv-414755e0   1490Gi   RWO   localblock   1d20h
ocs-deviceset-2-0-79j94   Bound   local-pv-3e8964d3   1490Gi   RWO   localblock   1d20h
```

3. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   a. Identify the nodes where the new OSD pods are running.

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-pod-name>
```

<OSD-pod-name>

Is the name of the OSD pod.
For example:

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

Example output:

```
NODE
compute-1
```

   b. For each of the nodes identified in the previous step, do the following:

      i. Create a debug pod and open a chroot environment for the selected host(s).

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

<node name>

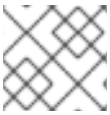
Is the name of the node.

```
$ chroot /host
```

ii. Check for the **crypt** keyword beside the **ocs-deviceset** name(s).

```
$ lsblk
```

4. Log in to OpenShift Web Console and check the OSD status on the storage dashboard.



NOTE

A full data recovery may take longer depending on the volume of data being recovered.

5.2. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON IBM POWER

You can replace an object storage device (OSD) in OpenShift Data Foundation deployed using local storage devices on IBM Power.

NOTE

One or more underlying storage devices may need to be replaced.

Prerequisites

- Red Hat recommends that replacement devices are configured with similar infrastructure and resources to the device being replaced.
- Ensure that the data is resilient.
  - In the OpenShift Web Console, click **Storage → Data Foundation**.
  - Click the **Storage Systems** tab, and then click **ocs-storagecluster-storagesystem**.
  - In the **Status card** of **Block and File** dashboard, under the **Overview** tab, verify that *Data Resiliency* has a green tick mark.

Procedure

1. Identify the OSD that needs to be replaced and the OpenShift Container Platform node that has the OSD scheduled on it.

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

Example output:

```
rook-ceph-osd-0-86bf8cdc8-4nb5t   0/1     crashLoopBackOff   0   24h   10.129.2.26
worker-0    <none>      <none>
rook-ceph-osd-1-7c99657cfb-jdzvz   1/1     Running   0         24h     10.128.2.46     worker-1
```

```
<none>      <none>
rook-ceph-osd-2-5f9f6dfb5b-2mnw9   1/1     Running   0          24h   10.131.0.33   worker-2
<none>      <none>
```

In this example, **rook-ceph-osd-0-86bf8cdc8-4nb5t** needs to be replaced and **worker-0** is the RHOCP node on which the OSD is scheduled.

NOTE

If the OSD to be replaced is healthy, the status of the pod will be **Running**.

2. Scale down the OSD deployment for the OSD to be replaced.

```
$ osd_id_to_remove=0
```

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

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:

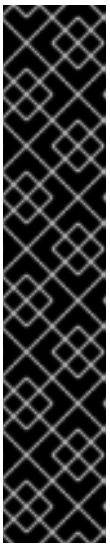
```
deployment.extensions/rook-ceph-osd-0 scaled
```

3. Verify that the **rook-ceph-osd** pod is terminated.

```
$ oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
```

Example output:

```
No resources found in openshift-storage namespace.
```



IMPORTANT

If the **rook-ceph-osd** pod is in **terminating** state for more than a few minutes, use the **force** option to delete the pod.

```
$ oc delete -n openshift-storage pod rook-ceph-osd-0-86bf8cdc8-4nb5t --
grace-period=0 --force
```

Example output:

```
warning: Immediate deletion does not wait for confirmation that the running
resource has been terminated. The resource may continue to run on the
cluster indefinitely.
  pod "rook-ceph-osd-0-86bf8cdc8-4nb5t" force deleted
```

4. Remove the old OSD from the cluster so that you can add a new OSD.

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

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

Example output:

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-64xjl
    ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-64xjl
```

In this example, the Persistent Volume Claim (PVC) name is **ocs-deviceset-localblock-0-data-0-64xjl**.

b. Identify the Persistent Volume (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-localblock-0-data-0-64xjl   Bound    local-pv-8137c873    256Gi      RWO
localblock     24h
```

In this example, the associated PV is **local-pv-8137c873**.

c. Identify the name of the device to be replaced.

```
$ oc get pv local-pv-<pv-suffix> -o yaml | grep path
```

where, ***pv-suffix*** is the value in the PV name identified in an earlier step.

Example output:

```
path: /mnt/local-storage/localblock/vdc
```

In this example, the device name is **vdc**.

d. Identify the **prepare-pod** associated with the OSD to be replaced.

```
$ oc describe -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix> | grep Used
```

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

Example output:

```
Used By:    rook-ceph-osd-prepare-ocs-deviceset-localblock-0-data-0-64knzkc
```

In this example, the **prepare-pod** name is **rook-ceph-osd-prepare-ocs-deviceset-localblock-0-data-0-64knzkc**.

e. Delete any old **ocs-osd-removal** jobs.

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

Example output:

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

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

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

g. Remove the old OSD from the cluster.

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

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

WARNING

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

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

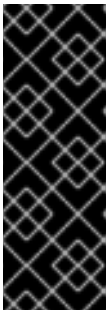
```
$ oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

6. Ensure that the OSD removal is completed.

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

Example output:

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



IMPORTANT

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

For example:

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

7. If encryption was enabled at the time of install, remove **dm-crypt** managed **device-mapper** mapping from the OSD devices that are removed from the respective OpenShift Data Foundation nodes.

   a. Get the PVC name(s) of the replaced OSD(s) from the logs of **ocs-osd-removal-job** pod.

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

   Example output:

   ```
   2021-05-12 14:31:34.666000 I | cephosd: removing the OSD PVC "ocs-deviceset-xxxx-xxx-xxx-xxx"
   ```

   b. For each of the previously identified nodes, do the following:

      i. Create a **debug** pod and **chroot** to the host on the storage node.

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

      <node name>

      Is the name of the node.

      ```
      $ chroot /host
      ```

      ii. Find the relevant device name based on the PVC names identified in the previous step.

      ```
      $ dmsetup ls| grep <pvc name>
      ```

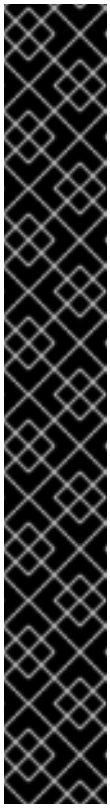
      <pvc name>

      Is the name of the PVC.
      Example output:

      ```
      ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt (253:0)
      ```

      iii. Remove the mapped device.

      ```
      $ cryptsetup luksClose --debug --verbose ocs-deviceset-xxx-xxx-xxx-xxx-block-dmcrypt
      ```



IMPORTANT

If the above command gets stuck due to insufficient privileges, run the following commands:

- Press **CTRL+Z** to exit the above command.
- Find the PID of the process which was stuck.

```
$ ps -ef | grep crypt
```

- Terminate the process using the **kill** command.

```
$ kill -9 <PID>
```

<PID>

Is the process ID.

- Verify that the device name is removed.

```
$ dmsetup ls
```

8. Find the PV that need to be deleted.

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

Example output:

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

9. Delete the PV.

```
$ oc delete pv <pv-name>
```

<pv-name>

Is the name of the PV.

10. Replace the old device and use the new device to create a new OpenShift Container Platform PV.

    a. Log in to the OpenShift Container Platform node with the device to be replaced. In this example, the OpenShift Container Platform node is **worker-0**.

```
$ oc debug node/worker-0
```

Example output:

```
Starting pod/worker-0-debug ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.88.21
If you don't see a command prompt, try pressing enter.
# chroot /host
```

-

b. Record the /**dev**/**disk** that is to be replaced using the device name, **vdc**, identified earlier.

```
# ls -alh /mnt/local-storage/localblock
```

Example output:

```
total 0
drwxr-xr-x. 2 root root 17 Nov  18 15:23 .
drwxr-xr-x. 3 root root 24 Nov  18 15:23 ..
lrwxrwxrwx. 1 root root  8 Nov  18 15:23 vdc -> /dev/vdc
```

c. Find the name of the **LocalVolume** CR, and remove or comment out the device /**dev**/**disk** that is to be replaced.

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

Example output:

```
NAME         AGE
localblock   25h
```

```
# oc edit -n openshift-local-storage localvolume localblock
```

Example output:

```
[...]
    storageClassDevices:
    - devicePaths:
   #   - /dev/vdc
      storageClassName: localblock
      volumeMode: Block
[...]
```

Make sure to save the changes after editing the CR.

11. Log in to the OpenShift Container Platform node with the device to be replaced and remove the old **symlink**.

```
$ oc debug node/worker-0
```

Example output:

```
Starting pod/worker-0-debug ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.88.21
If you don't see a command prompt, try pressing enter.
# chroot /host
```

a. Identify the old **symlink** for the device name to be replaced. In this example, the device name is **vdc**.

```
# ls -alh /mnt/local-storage/localblock
```

Example output:

```
total 0
drwxr-xr-x. 2 root root 17 Nov  18 15:23 .
drwxr-xr-x. 3 root root 24 Nov  18 15:23 ..
lrwxrwxrwx. 1 root root  8 Nov  18 15:23 vdc -> /dev/vdc
```

b. Remove the **symlink**.

```
# rm /mnt/local-storage/localblock/vdc
```

c. Verify that the **symlink** is removed.

```
# ls -alh /mnt/local-storage/localblock
```

Example output:

```
total 0
drwxr-xr-x. 2 root root 6 Nov 18 17:11 .
drwxr-xr-x. 3 root root 24 Nov 18 15:23 ..
```

12. Replace the old device with the new device.

13. Log back into the correct OpenShift Cotainer Platform node and identify the device name for the new drive. The device name must change unless you are resetting the same device.

```
# lsblk
```

Example output:

```
NAME                         MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
vda                          252:0    0   40G  0 disk
|-vda1                       252:1    0    4M  0 part
|-vda2                       252:2    0  384M  0 part /boot
`-vda4                       252:4    0 39.6G  0 part
  `-coreos-luks-root-nocrypt 253:0    0 39.6G  0 dm   /sysroot
vdb                          252:16   0  512B  1 disk
vdd                          252:32   0  256G  0 disk
```

In this example, the new device name is **vdd**.

14. After the new **/dev/disk** is available, you can add a new disk entry to the LocalVolume CR.

a. Edit the LocalVolume CR and add the new **/dev/disk**.
In this example, the new device is **/dev/vdd**.

```
# oc edit -n openshift-local-storage localvolume localblock
```

Example output:

```
[...]
    storageClassDevices:
    - devicePaths:
```

```
    #  - /dev/vdc
      - /dev/vdd
      storageClassName: localblock
      volumeMode: Block
[...]
```

Make sure to save the changes after editing the CR.

15. Verify that there is a new PV in **Available** state and of the correct size.

```
$ oc get pv | grep 256Gi
```

Example output:

```
local-pv-1e31f771  256Gi  RWO  Delete  Bound  openshift-storage/ocs-deviceset-
localblock-2-data-0-6xhkf  localblock  24h
local-pv-ec7f2b80  256Gi  RWO  Delete  Bound  openshift-storage/ocs-deviceset-
localblock-1-data-0-hr2fx  localblock  24h
local-pv-8137c873  256Gi  RWO  Delete  Available
localblock  32m
```

16. Create a new OSD for the new device.
Deploy the new OSD. You need to restart the **rook-ceph-operator** to force operator reconciliation.

    a. Identify the name of the **rook-ceph-operator**.

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

    Example output:

    ```
    NAME                                READY  STATUS   RESTARTS  AGE
    rook-ceph-operator-85f6494db4-sg62v  1/1    Running  0         1d20h
    ```

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

    ```
    $ oc delete -n openshift-storage pod rook-ceph-operator-85f6494db4-sg62v
    ```

    Example output:

    ```
    pod "rook-ceph-operator-85f6494db4-sg62v" deleted
    ```

    In this example, the rook-ceph-operator pod name is **rook-ceph-operator-85f6494db4-sg62v**.

    c. 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-85f6494db4-wx9xx  1/1    Running  0         50s
    ```

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

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

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

Example output:

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

NOTE

When using an external key management system (KMS) with data encryption, the old OSD encryption key can be removed from the Vault server as it is now an orphan key.

Verfication steps

1. Verify that there is a new OSD running.

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd
```

Example output:

```
rook-ceph-osd-0-76d8fb97f9-mn8qz   1/1     Running   0          23m
rook-ceph-osd-1-7c99657cfb-jdzvz   1/1     Running   1          25h
rook-ceph-osd-2-5f9f6dfb5b-2mnw9   1/1     Running   0          25h
```

2. Verify that a new PVC is created.

```
$ oc get -n openshift-storage pvc | grep localblock
```

Example output:

```
ocs-deviceset-localblock-0-data-0-q4q6b   Bound    local-pv-8137c873       256Gi     RWO
localblock         10m
ocs-deviceset-localblock-1-data-0-hr2fx   Bound    local-pv-ec7f2b80       256Gi     RWO
localblock         1d20h
ocs-deviceset-localblock-2-data-0-6xhkf   Bound    local-pv-1e31f771       256Gi     RWO
localblock         1d20h
```

3. Optional: If cluster-wide encryption is enabled on the cluster, verify that the new OSD devices are encrypted.

   a. Identify the nodes where the new OSD pods are running.

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-pod-name>
```

<OSD-pod-name>

Is the name of the OSD pod.
For example:

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

Example output:

```
NODE
compute-1
```

b. For each of the previously identified nodes, do the following:

i. Create a debug pod and open a chroot environment for the selected host(s).

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

<node name>

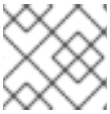
Is the name of the node.

```
$ chroot /host
```

ii. Check for the **crypt** keyword beside the **ocs-deviceset** name(s).

```
$ lsblk
```

4. Log in to OpenShift Web Console and check the status card in the OpenShift Data Foundation dashboard under Storage section.



NOTE

A full data recovery may take longer depending on the volume of data being recovered.

5.3. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON IBM Z OR LINUXONE INFRASTRUCTURE

You can replace operational or failed storage devices on IBM Z or LinuxONE infrastructure with new Small Computer System Interface (SCSI) disks.

IBM Z or LinuxONE supports SCSI FCP disk logical units (SCSI disks) as persistent storage devices from external disk storage. You can identify a SCSI disk using its FCP Device number, two target worldwide port names (WWPN1 and WWPN2), and the logical unit number (LUN). For more information, see https://www.ibm.com/support/knowledgecenter/SSB27U_6.4.0/com.ibm.zvm.v640.hcpa5/scsiover.html

Prerequisites

- Ensure that the data is resilient.
  - In the OpenShift Web Console, click **Storage → Data Foundation**.
  - Click the **Storage Systems** tab, and then click **ocs-storagecluster-storagesystem**.
  - In the **Status card** of **Block and File** dashboard, under the **Overview** tab, verify that *Data Resiliency* has a green tick mark.

Procedure

1. List all the disks.

```
$ lszdev
```

Example output:

```
TYPE        ID
zfcp-host   0.0.8204                                    yes  yes
zfcp-lun    0.0.8204:0x102107630b1b5060:0x4001402900000000 yes  no    sda sg0
zfcp-lun    0.0.8204:0x500407630c0b50a4:0x3002b03000000000  yes  yes   sdb sg1
qeth        0.0.bdd0:0.0.bdd1:0.0.bdd2                     yes  no    encbdd0
generic-ccw 0.0.0009                                       yes  no
```

A SCSI disk is represented as a **zfcp-lun** with the structure ***<device-id>:<wwpn>:<lun-id>*** in the **ID** section. The first disk is used for the operating system. If one storage device fails, you can replace it with a new disk.

2. Remove the disk.
Run the following command on the disk, replacing ***scsi-id*** with the SCSI disk identifier of the disk to be replaced:

```
$ chzdev -d scsi-id
```

For example, the following command removes one disk with the device ID **0.0.8204**, the WWPN **0x500507630a0b50a4**, and the LUN **0x4002403000000000**:

```
$ chzdev -d 0.0.8204:0x500407630c0b50a4:0x3002b03000000000
```

3. Append a new SCSI disk.

```
$ chzdev -e 0.0.8204:0x500507630b1b50a4:0x4001302a00000000
```

NOTE

The device ID for the new disk must be the same as the disk to be replaced. The new disk is identified with its WWPN and LUN ID.

4. List all the FCP devices to verify the new disk is configured.

```
$ lszdev zfcp-lun
```

Example output:

```
TYPE        ID                                      ON   PERS  NAMES
zfcp-lun    0.0.8204:0x102107630b1b5060:0x4001402900000000 yes  no    sda sg0
zfcp-lun    0.0.8204:0x500507630b1b50a4:0x4001302a00000000  yes  yes   sdb sg1
```