



# Red Hat OpenShift Container Storage 4.7

## Replacing devices

Instructions for safely replacing operational or failed devices



## Red Hat OpenShift Container Storage 4.7 Replacing devices

---

Instructions for safely replacing operational or failed devices

## Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This document explains how to safely replace storage devices for Red Hat OpenShift Container Storage.

---

## Table of Contents

<b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....	<b>3</b>
<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b> .....	<b>4</b>
<b>PREFACE</b> .....	<b>5</b>
<b>CHAPTER 1. DYNAMICALLY PROVISIONED OPENSIFT CONTAINER STORAGE DEPLOYED ON AWS</b> ...	<b>6</b>
1.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON AWS USER-PROVISIONED INFRASTRUCTURE	6
1.2. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON AWS INSTALLER-PROVISIONED INFRASTRUCTURE	6
<b>CHAPTER 2. DYNAMICALLY PROVISIONED OPENSIFT CONTAINER STORAGE DEPLOYED ON VMWARE</b>	<b>7</b>
2.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON VMWARE INFRASTRUCTURE	7
<b>CHAPTER 3. DYNAMICALLY PROVISIONED OPENSIFT CONTAINER STORAGE DEPLOYED ON RED HAT VIRTUALIZATION</b> .....	<b>12</b>
3.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON RED HAT VIRTUALIZATION INSTALLER- PROVISIONED INFRASTRUCTURE	12
<b>CHAPTER 4. DYNAMICALLY PROVISIONED OPENSIFT CONTAINER STORAGE DEPLOYED ON MICROSOFT AZURE</b> .....	<b>17</b>
4.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON AZURE INSTALLER-PROVISIONED INFRASTRUCTURE	17
<b>CHAPTER 5. OPENSIFT CONTAINER STORAGE DEPLOYED USING LOCAL STORAGE DEVICES</b> .....	<b>18</b>
5.1. REPLACING FAILED STORAGE DEVICES ON AMAZON EC2 INFRASTRUCTURE	18
5.2. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON CLUSTERS BACKED BY LOCAL STORAGE DEVICES	18
5.3. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON IBM POWER SYSTEMS	24
5.4. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON IBM Z OR LINUXONE INFRASTRUCTURE	32



## 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. As the Component, use **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 failed storage devices on Amazon EC2 infrastructure”](#)
  - [Section 5.2, “Replacing operational or failed storage devices on clusters backed by local storage devices”](#)
  - [Section 5.3, “Replacing operational or failed storage devices on IBM Power Systems”](#)
  - [Section 5.4, “Replacing operational or failed storage devices on IBM Z or LinuxONE infrastructure”](#)



### NOTE

OpenShift Container Storage does not support heterogeneous OSD sizes.

# CHAPTER 1. DYNAMICALLY PROVISIONED OPENSIFT CONTAINER STORAGE 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 OPENSIFT CONTAINER STORAGE DEPLOYED ON VMWARE

### 2.1. REPLACING OPERATIONAL OR FAILED STORAGE DEVICES ON VMWARE INFRASTRUCTURE

Use this procedure when one or more virtual machine disks (VMDK) needs to be replaced in OpenShift Container Storage which is deployed dynamically on VMware infrastructure. This procedure helps to 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.
  - On the OpenShift Web console, navigate to **Storage → Overview**.
  - Under **Persistent Storage** in the **Status** card, confirm that the *Data Resiliency* has a green tick mark.

#### Procedure

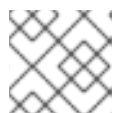
1. Identify the OSD that needs to be replaced and the OpenShift Container Platform node that has the OSD scheduled on it.

```
$ 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, repeat this step by updating the **osd\_id\_to\_remove** parameter with the OSD ID.

```
$ 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.
```



#### NOTE

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 a new OSD can be added.
  - 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. Change 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} |oc create -n openshift-storage -f -
```

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

**WARNING**

This step results in OSD being completely removed from the cluster. Ensure that the correct value of **osd\_id\_to\_remove** is provided.

5. 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-job -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-job -n openshift-storage --tail=-1'
```

6. 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 Container Storage nodes.

- a. Get 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'
```

For example:

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

- b. For each of the nodes identified in step #1, do the following:

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

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

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

```
sh-4.4# dmsetup ls| grep <pvc name>
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt (253:0)
```

- iii. Remove the mapped device.

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

**NOTE**

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

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

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

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

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

- Verify that the device name is removed.

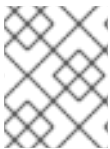
```
$ dmsetup ls
```

7. 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.

**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
```

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 node(s) where the new OSD pod(s) are running.

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

For example:

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

- b. For each of the nodes identified in previous step, do the following:
  - i. Create a debug pod and open a chroot environment for the selected host(s).

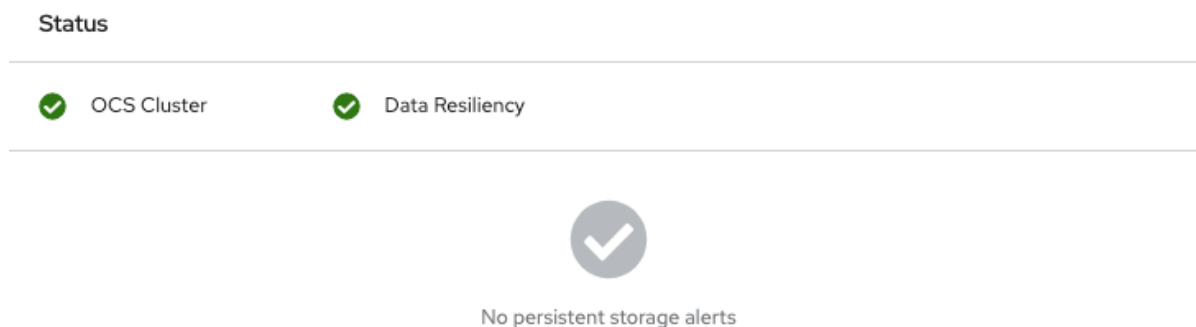
```
$ oc debug node/<node name>
$ chroot /host
```

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

```
$ lsblk
```

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

**Figure 2.1. OSD status in OpenShift Container Platform storage dashboard after device replacement**



## CHAPTER 3. DYNAMICALLY PROVISIONED OPENSIFT CONTAINER STORAGE DEPLOYED ON RED HAT VIRTUALIZATION

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

Use this procedure when one or more virtual machine disks (VMDK) needs to be replaced in OpenShift Container Storage which is deployed on Red Hat Virtualization infrastructure. This procedure helps to 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.
  - On the OpenShift Web console, navigate to **Storage → Overview**.
  - Under **Persistent Storage** in the **Status** card, confirm that the *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, repeat this step by updating the **osd\_id\_to\_remove** parameter with the OSD ID.

```
$ 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.
```



#### NOTE

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 a new OSD can be added.

- 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. Change 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} |oc create -n openshift-storage -f -
```

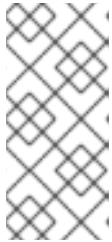
You can remove more than one OSD by adding comma separated OSD IDs in the command. (For example: `FAILED_OSD_IDS=0,1,2`)

**WARNING**

This step results in OSD being completely removed from the cluster. Ensure that the correct value of **osd\_id\_to\_remove** is provided.

5. 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-job -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-job -n openshift-storage --tail=-1'
```

6. 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 Container Storage nodes.

- a. Get 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'
```

For example:

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

- b. For each of the nodes identified in step #1, do the following:

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

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

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

```
sh-4.4# dmsetup ls| grep <pvc name>
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt (253:0)
```

- iii. Remove the mapped device.

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



## NOTE

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

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

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

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

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

- Verify that the device name is removed.

```
$ dmsetup ls
```

7. 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.

## 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
```

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 node(s) where the new OSD pod(s) are running.

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

For example:

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

- b. For each of the nodes identified in previous step, do the following:
  - i. Create a debug pod and open a chroot environment for the selected host(s).

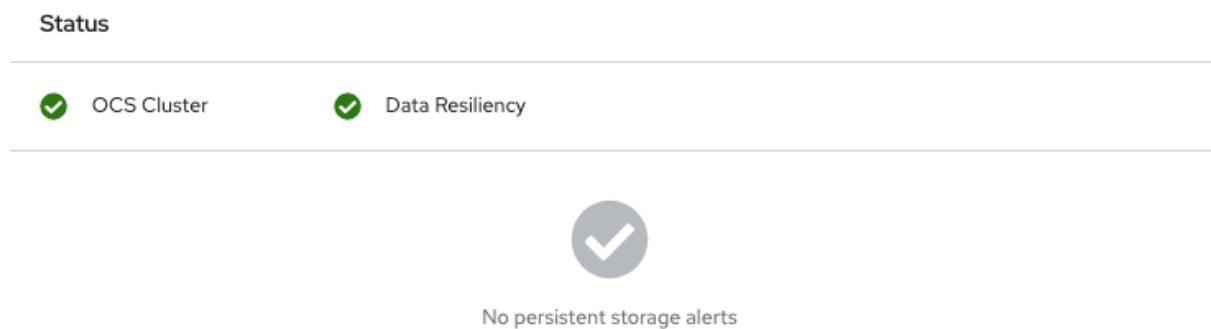
```
$ oc debug node/<node name>  
$ chroot /host
```

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

```
$ lsblk
```

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

**Figure 3.1. OSD status in OpenShift Container Platform storage dashboard after device replacement**



## CHAPTER 4. DYNAMICALLY PROVISIONED OPENSIFT CONTAINER STORAGE 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. OPENSIFT CONTAINER STORAGE DEPLOYED USING LOCAL STORAGE DEVICES

### 5.1. REPLACING FAILED STORAGE DEVICES ON AMAZON EC2 INFRASTRUCTURE

When you need to replace a storage device on an Amazon EC2 (storage-optimized I3) infrastructure, you must replace the storage node. For information about how to replace nodes, see [Replacing failed storage nodes on Amazon EC2 infrastructure](#).

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

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

- Bare metal
- VMware
- Red Hat Virtualization

Use this procedure when one or more underlying storage devices need to be replaced.

#### Prerequisites

- Red Hat recommends that replacement devices are configured with similar infrastructure and resources to the device being replaced.
- If you upgraded to OpenShift Container Storage 4.7 from a previous version and have not already created a **LocalVolumeSet** object to enable automatic provisioning of devices, do so now by following the procedure described in [Post-update configuration changes for clusters backed by local storage](#).
- If you upgraded to OpenShift Container Storage 4.7 from a previous version and have not already created the **LocalVolumeDiscovery** object, do so now by following the procedure described in [Post-update configuration changes for clusters backed by local storage](#).
- Ensure that the data is resilient.
  - On the OpenShift Web console, navigate to **Storage → Overview**.
  - Under **Persistent Storage** in the **Status** card, confirm that the *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.

- 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
```

- 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.
```



## NOTE

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
```

- Remove the old OSD from the cluster so that a new OSD can be added.
  - 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. Change 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} |oc create -n openshift-storage -f -
```

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

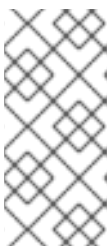


#### WARNING

This step results in OSD being completely removed from the cluster. Ensure that the correct value of **osd\_id\_to\_remove** is provided.

6. 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-job -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-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 Container Storage nodes.

- a. Get 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'
```

For example:



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

- b. For each of the nodes identified in step #1, do the following:
  - i. Create a **debug** pod and **chroot** to the host on the storage node.

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

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

```
sh-4.4# dmsetup ls| grep <pvc name>
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt (253:0)
```

- iii. Remove the mapped device.

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



#### NOTE

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

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

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

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

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

- Verify that the device name is removed.

```
$ dmsetup ls
```

8. Find the persistent volume (PV) that need to be deleted by the command:

```
$ 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
```

9. Delete the persistent volume.

```
$ oc delete pv local-pv-d6bf175b
```

10. Physically add a new device to the node.

- Use the following command to track provisioning of persistent volumes for devices that match the **deviceInclusionSpec**. It can take a few minutes to provision persistent volumes.

```
$ 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 persistent volume is provisioned, a new OSD pod is automatically created for the provisioned volume.

- 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

- 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
```

**NOTE**

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 node(s) where the new OSD pod(s) are running.

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

For example:

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

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

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

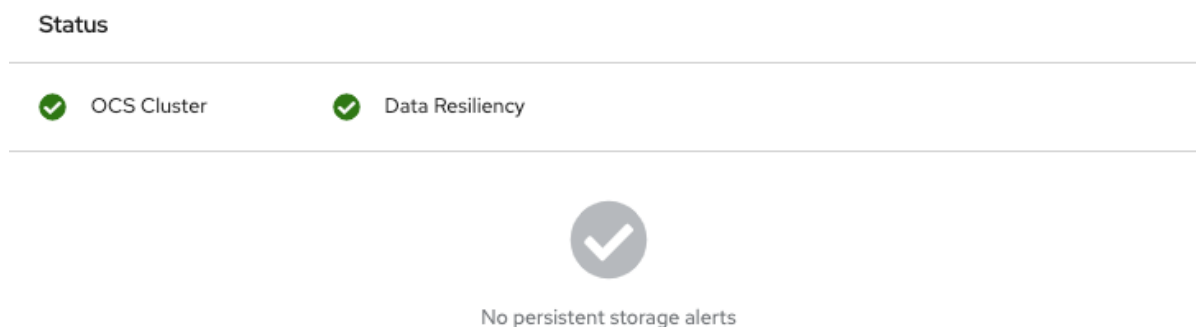
```
$ oc debug node/<node name>
$ chroot /host
```

- ii. Run "lsblk" and 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.

**Figure 5.1. OSD status in OpenShift Container Platform storage dashboard after device replacement**



#### 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 POWER SYSTEMS

You can replace an object storage device (OSD) in OpenShift Container Storage deployed using local storage devices on IBM Power Systems. Use this procedure when an underlying storage device needs to be replaced.

### Prerequisites

- Ensure that the data is resilient.
  - On the OpenShift Web console, navigate to **Storage → Overview**.
  - Under **Persistent Storage** in the **Status** card, confirm that the *Data Resiliency* has a green tick mark.

### Procedure

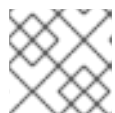
1. Identify the OSD that needs to be replaced and the OpenShift Container Platform node that has the OSD scheduled on it.

```
$ 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:

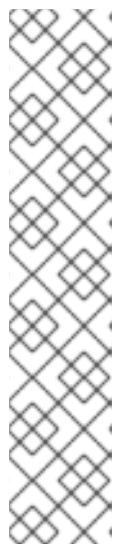
```
deployment.apps/rook-ceph-osd-0 scaled
```

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.
```



#### NOTE

If the **rook-ceph-osd** pod is in **terminating** state, 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
```

1. Remove the old OSD from the cluster so that a new OSD can be added.
  - 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 PVC name is **ocs-deviceset-localblock-0-data-0-64xjl**.

- b. 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
```

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

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

- d. Remove the old OSD from the cluster

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

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

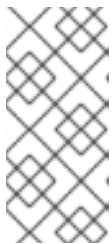


### WARNING

This step results in OSD being completely removed from the cluster. Make sure that the correct value of **osd\_id\_to\_remove** is provided.

2. 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 completed successfully.

```
$ oc get pod -l job-name=ocs-osd-removal-job -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-job -n openshift-storage --tail=-1
```

3. Delete the persistent volume claim (PVC) resources associated with the OSD to be replaced.

- 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 an step 4(a).

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**.

- b. Identify the name of the device to be replaced.

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

where, **pvc-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**.

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

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

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

Example output:

```
Mounted 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**.

- d. Delete the **osd-prepare** pod before removing the associated PVC.

```
$ oc delete -n openshift-storage pod rook-ceph-osd-prepare-ocs-deviceset-<x>-<y>-
<pvc-suffix>-<pod-suffix>
```

where, **x**, **y**, **pvc-suffix**, and **pod-suffix** are the values in the **osd-prepare** pod name identified in an earlier step.

Example output:

```
pod "rook-ceph-osd-prepare-ocs-deviceset-localblock-0-data-0-64knzkc" deleted
```

- e. Delete the PVC associated with the OSD to be replaced.

```
$ oc delete -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:

```
persistentvolumeclaim "ocs-deviceset-localblock-0-data-0-64xjl" deleted
```

4. Delete the PV associated with the device to be replaced, which was identified in earlier steps. In this example, the PV name is **local-pv-8137c873**.

```
$ oc delete pv local-pv-8137c873
```

Example output:

```
persistentvolume "local-pv-8137c873" deleted
```

5. Replace the old device and use the new device to create a new OpenShift Container Platform PV.
  - a. Log in to 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 **LocalVolumeSet** CR, and remove or comment out the device **/dev/disk** that is to be replaced.

```
$ oc get -n openshift-local-storage localvolumeset
NAME      AGE
localblock 25h
```

6. Log in to 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 ..
```



### IMPORTANT

For new deployments of OpenShift Container Storage 4.5 or later, LVM is not in use, **ceph-volume** raw mode is in play instead. Therefore, additional validation is not needed and you can proceed to the next step.

7. Replace the device with the new device.
8. Log back into the correct OpenShift Container Platform node and identify the device name for the new drive. The device name must change unless you are reseating 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**.

9. After the new **/dev/disk** is available ,it will be auto detected by localvolumeset.
10. 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
```

11. Create new OSD for new device.
  - a. Deploy the new OSD by restarting the **rook-ceph-operator** to force operator reconciliation.
    - i. 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
```

- ii. 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**.

- iii. 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.

- 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
```

## Verification steps

- Verify that there is a new OSD running and a new PVC created.

```
$ 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
```

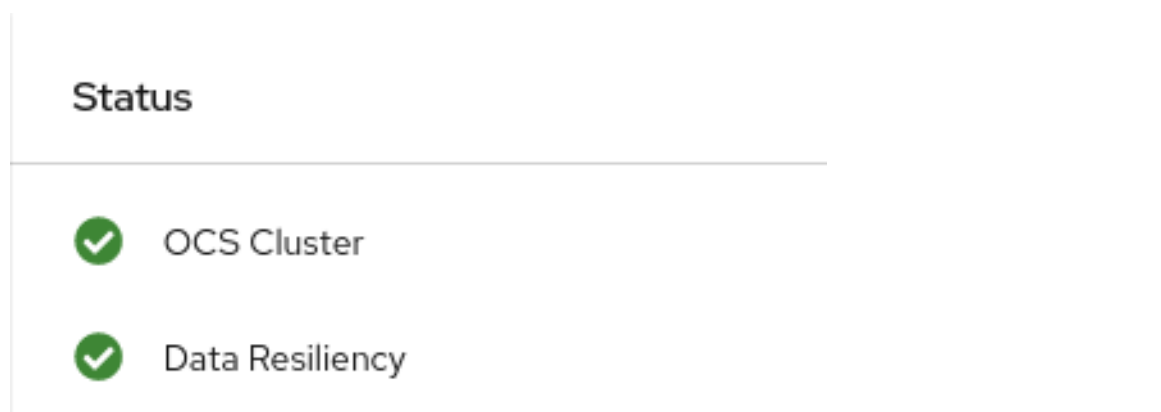
```
$ oc get -n openshift-storage pvc | grep localblock
```

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
```

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

**Figure 5.2. OSD status in OpenShift Container Platform storage dashboard after device replacement**



## 5.4. 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 SCSI disks.

IBM Z or LinuxONE supports SCSI FCP disk logical units (SCSI disks) as persistent storage devices from external disk storage. A SCSI disk can be identified by 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](https://www.ibm.com/support/knowledgecenter/SSB27U_6.4.0/com.ibm.zvm.v640.hcpa5/scsiover.html)

### Prerequisites

- Ensure that the data is resilient.
  - On the OpenShift Web console, navigate to **Storage → Overview**.
  - Under **Persistent Storage** in the **Status** card, confirm that the *Data Resiliency* has a green tick mark.

### Procedure

1. List all the disks with the following command.

```
$ lszdev
```

Example output:

```
TYPE      ID
zfcplib  0.0.8204                yes yes
zfcplib  0.0.8204:0x102107630b1b5060:0x4001402900000000 yes no  sda sg0
zfcplib  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 **zfcplib** 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, it can be replaced 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.

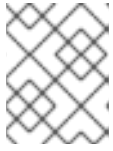
```
$ chzdev -d scsi-id
```

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

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

3. Append a new SCSI disk with the following command:

```
$ 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 zfcplib
TYPE      ID                                     ON PERS NAMES
zfcplib  0.0.8204:0x102107630b1b5060:0x4001402900000000  yes no  sda sg0
zfcplib  0.0.8204:0x500507630b1b50a4:0x4001302a00000000  yes yes sdb sg1
```