



# Red Hat OpenShift Container Storage 4.8

## Deploying OpenShift Container Storage using IBM Z infrastructure

How to install and set up your IBM Z environment



# Red Hat OpenShift Container Storage 4.8 Deploying OpenShift Container Storage using IBM Z infrastructure

---

How to install and set up your IBM Z environment

## 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

Read this document for instructions on installing Red Hat OpenShift Container Storage 4.8 to use local storage on IBM Z infrastructure. While this document refers only to IBM Z, all information in it also applies to LinuxONE.

---

## 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. PREPARING TO DEPLOY OPENSIFT CONTAINER STORAGE</b> .....	<b>6</b>
1.1. REQUIREMENTS FOR INSTALLING OPENSIFT CONTAINER STORAGE USING LOCAL STORAGE DEVICES .....	6
<b>CHAPTER 2. DEPLOY OPENSIFT CONTAINER STORAGE USING LOCAL STORAGE DEVICES</b> .....	<b>7</b>
2.1. INSTALLING RED HAT OPENSIFT CONTAINER STORAGE OPERATOR .....	7
2.2. INSTALLING LOCAL STORAGE OPERATOR .....	8
2.3. FINDING AVAILABLE STORAGE DEVICES (OPTIONAL) .....	9
2.4. CREATING OPENSIFT CONTAINER STORAGE CLUSTER ON IBM Z .....	10
<b>CHAPTER 3. VERIFYING OPENSIFT CONTAINER STORAGE DEPLOYMENT FOR INTERNAL-ATTACHED DEVICES MODE</b> .....	<b>13</b>
3.1. VERIFYING THE STATE OF THE PODS .....	13
3.2. VERIFYING THE OPENSIFT CONTAINER STORAGE CLUSTER IS HEALTHY .....	14
3.3. VERIFYING THAT THE OPENSIFT CONTAINER STORAGE SPECIFIC STORAGE CLASSES EXIST .....	15
<b>CHAPTER 4. UNINSTALLING OPENSIFT CONTAINER STORAGE</b> .....	<b>16</b>
4.1. UNINSTALLING OPENSIFT CONTAINER STORAGE IN INTERNAL-ATTACHED DEVICES MODE .....	16
4.1.1. Removing local storage operator configurations .....	20
4.2. REMOVING MONITORING STACK FROM OPENSIFT CONTAINER STORAGE .....	23
4.3. REMOVING OPENSIFT CONTAINER PLATFORM REGISTRY FROM OPENSIFT CONTAINER STORAGE .....	26
4.4. REMOVING THE CLUSTER LOGGING OPERATOR FROM OPENSIFT CONTAINER STORAGE .....	27



## 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

Red Hat OpenShift Container Storage 4.8 supports deployment on existing Red Hat OpenShift Container Platform (RHOCP) IBM Z clusters in connected or disconnected environments along with out-of-the-box support for proxy environments.



### NOTE

Only internal OpenShift Container Storage clusters are supported on IBM Z. See [Planning your deployment](#) and [Preparing to deploy OpenShift Container Storage](#) for more information about deployment requirements.

To deploy OpenShift Container Storage, follow the appropriate deployment process for your environment:

- Internal Attached Devices mode
  - [Deploy using local storage devices](#)

# CHAPTER 1. PREPARING TO DEPLOY OPENSIFT CONTAINER STORAGE

When you deploy OpenShift Container Storage on OpenShift Container Platform using local storage devices, you can create internal cluster resources. This approach internally provisions base services. Then all applications can access additional storage classes.

Before you begin the deployment of Red Hat OpenShift Container Storage using local storage, ensure that your resource requirements are met. See [requirements for installing OpenShift Container Storage using local storage devices](#).

After you have addressed the above, follow these steps in the order given:

1. [Install the Red Hat OpenShift Container Storage Operator](#) .
2. [Install Local Storage Operator](#) .
3. [Find the available storage devices](#) .
4. [Create the OpenShift Container Storage cluster service on IBM Z](#) .

## 1.1. REQUIREMENTS FOR INSTALLING OPENSIFT CONTAINER STORAGE USING LOCAL STORAGE DEVICES

### Node requirements

The cluster must consist of at least three OpenShift Container Platform worker nodes with locally attached-storage devices on each of them.

- Each of the three selected nodes must have at least one raw block device available to be used by OpenShift Container Storage.
- The devices you use must be empty; the disks must not include physical volumes (PVs), volume groups (VGs), or logical volumes (LVs) remaining on the disk.

For more information, see the [Resource requirements](#) section in the Planning guide.

- For storage nodes, FCP storage devices are required.
- Multicloud Object Gateway is not supported.

### Minimum starting node requirements [Technology Preview]

An OpenShift Container Storage cluster is deployed with minimum configuration when the standard deployment resource requirement is not met.

For more information, see [Resource requirements](#) section in the Planning guide.

## CHAPTER 2. DEPLOY OPENSIFT CONTAINER STORAGE USING LOCAL STORAGE DEVICES

Deploying OpenShift Container Storage on OpenShift Container Platform using local storage devices provides you with the option to create internal cluster resources. Follow this deployment method to use local storage to back persistent volumes for your OpenShift Container Platform applications.

Use this section to deploy OpenShift Container Storage on IBM Z infrastructure where OpenShift Container Platform is already installed.

### 2.1. INSTALLING RED HAT OPENSIFT CONTAINER STORAGE OPERATOR

You can install Red Hat OpenShift Container Storage Operator using the Red Hat OpenShift Container Platform Operator Hub.

#### Prerequisites

- Access to an OpenShift Container Platform cluster using an account with cluster-admin and operator installation permissions.
- You have at least three worker nodes in the Red Hat OpenShift Container Platform cluster.
- You have satisfied any additional requirements required. For more information, see [Planning your deployment](#).



#### NOTE

- When you need to override the cluster-wide default node selector for OpenShift Container Storage, you can use the following command to specify a blank node selector for the **openshift-storage** namespace (create openshift-storage namespace in this case):

```
$ oc annotate namespace openshift-storage openshift.io/node-selector=
```
- Taint a node as **infra** to ensure only Red Hat OpenShift Container Storage resources are scheduled on that node. This helps you save on subscription costs. For more information, see [How to use dedicated worker nodes for Red Hat OpenShift Container Storage](#) chapter in Managing and Allocating Storage Resources guide.

#### Procedure

1. Log in to OpenShift Web Console.
2. Click **Operators** → **OperatorHub**.
3. Search for **OpenShift Container Storage** from the list of operators and click on it.
4. Click **Install**.
5. Set the following options on the **Install Operator** page:
  - a. Channel as **stable-4.8**.

- b. Installation Mode as **A specific namespace on the cluster**
- c. Installed Namespace as **Operator recommended namespace openshift-storage**. If Namespace **openshift-storage** does not exist, it will be created during the operator installation.
- d. **Approval Strategy** as **Automatic** or **Manual**.
- e. Click **Install**.  
If you select **Automatic** updates, the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your operator without any intervention.  
  
If you select **Manual** updates, the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to have the operator updated to the new version.

### Verification step

- Verify that the **OpenShift Container Storage** Operator shows a green tick indicating successful installation.

## 2.2. INSTALLING LOCAL STORAGE OPERATOR

You can install the Local Storage Operator using the Red Hat OpenShift Container Platform Operator Hub.

### Prerequisite

- Access to an OpenShift Container Platform cluster using an account with cluster-admin and Operator installation permissions.

### Procedure

1. Log in to the OpenShift Web Console.
2. Click **Operators** → **OperatorHub**.
3. Type **local storage** in the **Filter by keyword...** box to search for **Local Storage** operator from the list of operators and click on it.
4. Click **Install**.
5. Set the following options on the **Install Operator** page:
  - a. Channel as **stable-4.8**.
  - b. Installation Mode as **A specific namespace on the cluster**
  - c. Installed Namespace as **Operator recommended namespace openshift-local-storage**.
  - d. Approval Strategy as **Automatic**.
6. Click **Install**.

### Verification step

- Verify that the Local Storage Operator shows the **Status** as **Succeeded**.

## 2.3. FINDING AVAILABLE STORAGE DEVICES (OPTIONAL)

This step is additional information and can be skipped as the disks are automatically discovered during storage cluster creation. Use this procedure to identify the device names for each of the three or more worker nodes that you have labeled with the OpenShift Container Storage label **cluster.ocs.openshift.io/openshift-storage=** before creating Persistent Volumes (PV) for IBM Z.

### Procedure

1. List and verify the name of the worker nodes with the OpenShift Container Storage label.

```
$ oc get nodes -l=cluster.ocs.openshift.io/openshift-storage=
```

Example output:

```
NAME          STATUS  ROLES  AGE   VERSION
bmworker01   Ready  worker 6h45m v1.16.2
bmworker02   Ready  worker 6h45m v1.16.2
bmworker03   Ready  worker 6h45m v1.16.2
```

2. Log in to each worker node that is used for OpenShift Container Storage resources and find the unique **by-id** device name for each available raw block device.

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

Example output:

```
$ oc debug node/bmworker01
Starting pod/bmworker01-debug ...
To use host binaries, run `chroot /host`
Pod IP: 10.0.135.71
If you don't see a command prompt, try pressing enter.
sh-4.2# chroot /host
sh-4.4# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0                7:0    0 500G  0 loop
sda                  8:0    0 120G  0 disk
|-sda1                8:1    0 384M  0 part /boot
`-sda4                8:4    0 119.6G 0 part
`-coreos-luks-root-nocrypt 253:0  0 119.6G 0 dm  /sysroot
sdb                  8:16   0 500G  0 disk
```

In this example, for **bmworker01**, the available local device is **sdb**.

3. Identify the unique ID for each of the devices selected in Step 2.

```
sh-4.4#ls -l /dev/disk/by-id/ | grep sdb
lrwxrwxrwx. 1 root root 9 Feb 3 16:49 scsi-360050763808104bc2800000000000259 ->
../sdb
lrwxrwxrwx. 1 root root 9 Feb 3 16:49 scsi-SIBM_2145_00e020412f0aXX00 -> ../sdb
lrwxrwxrwx. 1 root root 9 Feb 3 16:49 scsi-0x60050763808104bc2800000000000259 ->
../sdb
```

-

In the above example, the ID for the local device **sdb**

```
scsi-0x60050763808104bc2800000000000259
```

4. Repeat the above step to identify the device ID for all the other nodes that have the storage devices to be used by OpenShift Container Storage. See this [Knowledge Base article](#) for more details.

## 2.4. CREATING OPENSIFT CONTAINER STORAGE CLUSTER ON IBM Z

Use this procedure to create storage cluster on IBM Z.

### Prerequisites

- Ensure that all the requirements in the [Requirements for installing OpenShift Container Storage using local storage devices](#) section are met.
- You must have three worker nodes with the same storage type and size attached to each node (for example, 200 GB) to use local storage devices on IBM Z or LinuxONE.

### Procedure

1. Log into the OpenShift Web Console.
2. Click **Operators** → **Installed Operators** to view all the installed operators. Ensure that the **Project** selected is **openshift-storage**.
3. Click **OpenShift Container Storage**.
4. Click **Create Instance** link of Storage Cluster.
5. Select **Internal-Attached devices** for the **Select Mode**. By default, Internal is selected.
6. Create a storage cluster using the wizard that includes disk discovery, storage class creation, and storage cluster creation.  
You are prompted to install the Local Storage Operator if it is not already installed. Click **Install** and install the operator as described in [Installing Local Storage Operator](#).

### Discover disks

You can discover a list of potentially usable disks on the selected nodes. Block disks and partitions that are not in use and available for provisioning persistent volumes (PVs) are discovered.

- a. Choose one of the following:
  - **All nodes** to discover disks in all the nodes.
  - **Select nodes** to discover disks from a subset of the listed nodes.  
To find specific worker nodes in the cluster, you can filter nodes on the basis of Name or Label. Name allows you to search by name of the node and Label allows you to search by selecting the predefined label.

If the nodes selected do not match the OpenShift Container Storage cluster

requirement of an aggregated 30 CPUs and 72 GiB of RAM, a minimal cluster will be deployed. For minimum starting node requirements, see [Resource requirements](#) section in Planning guide.



#### NOTE

If the nodes to be selected are tainted and not discovered in the wizard, follow the steps provided in the [Red Hat Knowledgebase Solution](#) as a workaround to add tolerations for Local Storage Operator resources.

- b. Click **Next**.

### Create Storage Class

You can create a dedicated storage class to consume storage by filtering a set of storage volumes.

- a. Enter the **Volume Set Name**.
- b. Enter the **Storage Class Name** By default, the volume set name appears for the storage class name.
- c. The nodes selected for disk discovery in the earlier step are displayed in the **Filter Disks** section. Choose one of the following:
  - **All nodes** to select all the nodes for which you discovered the devices.
  - **Select nodes** to select a subset of the nodes for which you discovered the devices. To find specific worker nodes in the cluster, you can filter nodes on the basis of Name or Label. Name allows you to search by name of the node and Label allows you to search by selecting the predefined label.

It is recommended that the worker nodes are spread across three different physical nodes, racks or failure domains for high availability.




#### NOTE

Ensure OpenShift Container Storage rack labels are aligned with physical racks in the datacenter to prevent a double node failure at the failure domain level.

- d. Select the required **Disk Type**. The following options are available:

All	Selects all types of disks present on the nodes. By default, this option is selected.
SSD/NVME E	Selects only SSD NVME type of disks.
HDD	Selects only HDD type of disks.

- e. In the **Advanced** section, you can set the following:

Volume Mode	Block is selected by default.
Disk Size	<p>Minimum and maximum available size of the device that needs to be included.</p>  <p><b>NOTE</b></p> <p>You must set a minimum size of 100GB for the device.</p>
Max Disk Limit	This indicates the maximum number of PVs that can be created on a node. If this field is left empty, then PVs are created for all the available disks on the matching nodes.

- f. (Optional) You can view the selected capacity of the disks on the selected nodes using the *Select Capacity* chart. This chart might take a few minutes to reflect the disks that are discovered in the previous step.

You can click on the **Nodes** and **Disks** links on the chart to bring up the list of nodes and disks to view more details.

- g. Click **Next**.
- h. Click **Yes** in the message alert to confirm the creation of the storage class. After the local volume set and storage class are created, it is not possible to go back to the step.

### Create Storage Cluster

- a. Select the required storage class. You might need to wait a couple of minutes for the storage nodes corresponding to the selected storage class to get populated. The nodes corresponding to the storage class are displayed based on the storage class that you selected from the drop down list.
- b. Click **Next**.
- c. (Optional) In the Encryption section, set the toggle to **Enabled** to enable data encryption on the cluster. Security configuration:
- d. Click **Save**.
- e. Click **Next** to review your storage cluster.
- f. Click **Create**. The **Create** button is enabled only when a minimum of three nodes are selected. A new storage cluster with the number of volumes equal to the total number of devices attached to each node is created. The default configuration uses a replication factor of 1.

To expand the capacity of the initial cluster, see [Scaling Storage](#) guide.

### Verification steps

See [Verifying your OpenShift Container Storage installation](#) .



## CHAPTER 3. VERIFYING OPENSIFT CONTAINER STORAGE DEPLOYMENT FOR INTERNAL-ATTACHED DEVICES MODE

Use this section to verify that OpenShift Container Storage is deployed correctly.

### 3.1. VERIFYING THE STATE OF THE PODS

To verify that the pods of OpenShift Containers Storage are in running state, follow the below procedure:

#### Procedure

1. Log in to OpenShift Web Console.
2. Click **Workloads** → **Pods** from the left pane of the OpenShift Web Console.
3. Select **openshift-storage** from the **Project** drop down list.  
For more information on the expected number of pods for each component and how it varies depending on the number of nodes, see [Table 3.1, "Pods corresponding to OpenShift Container storage cluster"](#).
4. Click on the **Running** and **Completed** tabs to verify that the pods are running and in a completed state:

**Table 3.1. Pods corresponding to OpenShift Container storage cluster**

Component	Corresponding pods
OpenShift Container Storage Operator	<ul style="list-style-type: none"> <li>● <b>ocs-operator-*</b> (1 pod on any worker node)</li> <li>● <b>ocs-metrics-exporter-*</b> (1 pod on any worker node)</li> </ul>
Rook-ceph Operator	<ul style="list-style-type: none"> <li>● <b>rook-ceph-operator-*</b> (1 pod on any worker node)</li> </ul>
Multicloud Object Gateway	<ul style="list-style-type: none"> <li>● <b>noobaa-operator-*</b> (1 pod on any worker node)</li> <li>● <b>noobaa-core-*</b> (1 pod on any storage node)</li> <li>● <b>nooba-db-*</b> (1 pod on any storage node)</li> <li>● <b>noobaa-endpoint-*</b> (1 pod on any storage node)</li> </ul>
MON	<ul style="list-style-type: none"> <li>● <b>rook-ceph-mon-*</b> (3 pods distributed across storage nodes)</li> </ul>

Component	Corresponding pods
MGR	<b>rook-ceph-mgr-*</b> (1 pod on any storage node)
MDS	<b>rook-ceph-mds-ocs-storagecluster-cephfilesystem-*</b> (2 pods distributed across storage nodes)
RGW	<b>rook-ceph-rgw-ocs-storagecluster-cephobjectstore-*</b> (1 pod on any storage node)
CSI	<ul style="list-style-type: none"> <li>● <b>cephfs</b> <ul style="list-style-type: none"> <li>○ <b>csi-cephfsplugin-*</b> (1 pod on each worker node)</li> <li>○ <b>csi-cephfsplugin-provisioner-*</b> (2 pods distributed across storage nodes)</li> </ul> </li> <li>● <b>rbd</b> <ul style="list-style-type: none"> <li>○ <b>csi-rbdplugin-*</b> (1 pod on each worker node)</li> <li>○ <b>csi-rbdplugin-provisioner-*</b> (2 pods distributed across storage nodes)</li> </ul> </li> </ul>
rook-ceph-crashcollector	<b>rook-ceph-crashcollector-*</b> (1 pod on each storage node)
OSD	<ul style="list-style-type: none"> <li>● <b>rook-ceph-osd-*</b> (1 pod for each device)</li> <li>● <b>rook-ceph-osd-prepare-ocs-deviceset-*</b> (1 pod for each device)</li> </ul>

## 3.2. VERIFYING THE OPENSIFT CONTAINER STORAGE CLUSTER IS HEALTHY

To verify that the cluster of OpenShift Container Storage is healthy, follow the steps in the procedure.

### Procedure

1. Click **Storage → Overview** and click the **Block and File** tab.
2. In the **Status card**, verify that *Storage Cluster* and *Data Resiliency* has a green tick mark.
3. In the **Details card**, verify that the cluster information is displayed.

For more information on the health of the OpenShift Container Storage clusters using the Block and File dashboard, see [Monitoring OpenShift Container Storage](#).

### 3.3. VERIFYING THAT THE OPENSIFT CONTAINER STORAGE SPECIFIC STORAGE CLASSES EXIST

To verify the storage classes exists in the cluster, follow the steps in the procedure.

#### Procedure

1. Click **Storage** → **Storage Classes** from the OpenShift Web Console.
2. Verify that the following storage classes are created with the OpenShift Container Storage cluster creation:
  - **ocs-storagecluster-ceph-rbd**
  - **ocs-storagecluster-cephfs**
  - **openshift-storage.noobaa.io**
  - **ocs-storagecluster-ceph-rgw**

## CHAPTER 4. UNINSTALLING OPENSIFT CONTAINER STORAGE

### 4.1. UNINSTALLING OPENSIFT CONTAINER STORAGE IN INTERNAL-ATTACHED DEVICES MODE

Use the steps in this section to uninstall OpenShift Container Storage.

#### Uninstall Annotations

Annotations on the Storage Cluster are used to change the behavior of the uninstall process. To define the uninstall behavior, the following two annotations have been introduced in the storage cluster:

- **uninstall.ocs.openshift.io/cleanup-policy: delete**
- **uninstall.ocs.openshift.io/mode: graceful**

The below table provides information on the different values that can be used with these annotations:

**Table 4.1. uninstall.ocs.openshift.io uninstall annotations descriptions**

Annotation	Value	Default	Behavior
cleanup-policy	delete	Yes	Rook cleans up the physical drives and the <b>DataDirHostPath</b>
cleanup-policy	retain	No	Rook does <b>not</b> clean up the physical drives and the <b>DataDirHostPath</b>
mode	graceful	Yes	Rook and NooBaa <b>pauses</b> the uninstall process until the PVCs and the OBCs are removed by the administrator/user
mode	forced	No	Rook and NooBaa proceeds with uninstall even if PVCs/OBCs provisioned using Rook and NooBaa exist respectively.

You can change the cleanup policy or the uninstall mode by editing the value of the annotation by using the following commands:

```
$ oc -n openshift-storage annotate storagecluster ocs-storagecluster
uninstall.ocs.openshift.io/cleanup-policy="retain" --overwrite
storagecluster.ocs.openshift.io/ocs-storagecluster annotated
```

```
$ oc -n openshift-storage annotate storagecluster ocs-storagecluster
uninstall.ocs.openshift.io/mode="forced" --overwrite
storagecluster.ocs.openshift.io/ocs-storagecluster annotated
```

## Prerequisites

- Ensure that the OpenShift Container Storage cluster is in a healthy state. The uninstall process can fail when some of the pods are not terminated successfully due to insufficient resources or nodes. In case the cluster is in an unhealthy state, contact Red Hat Customer Support before uninstalling OpenShift Container Storage.
- Ensure that applications are not consuming persistent volume claims (PVCs) or object bucket claims (OBCs) using the storage classes provided by OpenShift Container Storage.
- If any custom resources (such as custom storage classes, cephblockpools) were created by the admin, they must be deleted by the admin after removing the resources which consumed them.

## Procedure

1. Delete the volume snapshots that are using OpenShift Container Storage.

- a. List the volume snapshots from all the namespaces.

```
$ oc get volumesnapshot --all-namespaces
```

- b. From the output of the previous command, identify and delete the volume snapshots that are using OpenShift Container Storage.

```
$ oc delete volumesnapshot <VOLUME-SNAPSHOT-NAME> -n <NAMESPACE>
```

2. Delete PVCs and OBCs that are using OpenShift Container Storage.

In the default uninstall mode (graceful), the uninstaller waits till all the PVCs and OBCs that use OpenShift Container Storage are deleted.

If you wish to delete the Storage Cluster without deleting the PVCs beforehand, you may set the uninstall mode annotation to "forced" and skip this step. Doing so will result in orphan PVCs and OBCs in the system.

- a. Delete OpenShift Container Platform monitoring stack PVCs using OpenShift Container Storage.

See [Removing monitoring stack from OpenShift Container Storage](#)

- b. Delete OpenShift Container Platform Registry PVCs using OpenShift Container Storage.

[Removing OpenShift Container Platform registry from OpenShift Container Storage](#)

- c. Delete OpenShift Container Platform logging PVCs using OpenShift Container Storage.

[Removing the cluster logging operator from OpenShift Container Storage](#)

3. Delete the Storage Cluster object and wait for the removal of the associated resources.

```
$ oc delete -n openshift-storage storagecluster --all --wait=true
```

4. Check for cleanup pods if the **uninstall.ocs.openshift.io/cleanup-policy** was set to **delete** (default) and ensure that their status is **Completed**.

```
$ oc get pods -n openshift-storage | grep -i cleanup
NAME                                READY STATUS RESTARTS AGE
cluster-cleanup-job-<xx>             0/1   Completed 0      8m35s
cluster-cleanup-job-<yy>             0/1   Completed 0      8m35s
cluster-cleanup-job-<zz>             0/1   Completed 0      8m35s
```

5. Confirm that the directory **/var/lib/rook** is now empty. This directory will be empty only if the **uninstall.ocs.openshift.io/cleanup-policy** annotation was set to **delete**(default).

```
$ for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o jsonpath='{.items[*].metadata.name}'); do oc debug node/${i} -- chroot /host ls -l /var/lib/rook; done
```

6. If encryption was enabled at the time of install, remove **dm-crypt** managed **device-mapper** mapping from OSD devices on all the OpenShift Container Storage nodes.
  - a. Create a **debug** pod and **chroot** to the host on the storage node.

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

- b. Get Device names and make note of the OpenShift Container Storage devices.

```
$ dmsetup ls
ocs-deviceset-0-data-0-57snx-block-dmccrypt (253:1)
```

- c. Remove the mapped device.

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-0-data-0-57snx-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 namespace and wait till the deletion is complete. You will need to switch to another project if **openshift-storage** is the active project.  
For example:

```
$ oc project default
$ oc delete project openshift-storage
```

The project is deleted if the following command returns a NotFound error.

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



#### NOTE

While uninstalling OpenShift Container Storage, if **namespace** is not deleted completely and remains in **Terminating** state, perform the steps in [Troubleshooting and deleting remaining resources during Uninstall](#) to identify objects that are blocking the namespace from being terminated.

8. Delete local storage operator configurations if you have deployed OpenShift Container Storage using local storage devices. See [Removing local storage operator configurations](#).
9. Unlabel the storage nodes.

```
$ oc label nodes --all cluster.ocs.openshift.io/openshift-storage-
$ oc label nodes --all topology.rook.io/rack-
```

10. Remove the OpenShift Container Storage taint if the nodes were tainted.

```
$ oc adm taint nodes --all node.ocs.openshift.io/storage-
```

11. Confirm all PVs provisioned using OpenShift Container Storage are deleted. If there is any PV left in the **Released** state, delete it.

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

12. Delete the Multicloud Object Gateway storageclass.

```
$ oc delete storageclass openshift-storage.noobaa.io --wait=true --timeout=5m
```

13. Remove **CustomResourceDefinitions**.

```
$ oc delete crd backingstores.noobaa.io bucketclasses.noobaa.io
cephblockpools.ceph.rook.io cephclusters.ceph.rook.io cephfilesystems.ceph.rook.io
cephnfses.ceph.rook.io cephobjectstores.ceph.rook.io cephobjectstoreusers.ceph.rook.io
noobaas.noobaa.io ocsinitializations.ocs.openshift.io storageclusters.ocs.openshift.io
cephclients.ceph.rook.io cephobjectrealms.ceph.rook.io cephobjectzonegroups.ceph.rook.io
cephobjectzones.ceph.rook.io cephrbdmirrors.ceph.rook.io --wait=true --timeout=5m
```

14. To ensure that OpenShift Container Storage is uninstalled completely, on the OpenShift Container Platform Web Console,

- a. Click **Storage**.
- b. Verify that **Overview** no longer appears under Storage.

## 4.1.1. Removing local storage operator configurations

Use the instructions in this section only if you have deployed OpenShift Container Storage using local storage devices.



### NOTE

For OpenShift Container Storage deployments only using **localvolume** resources, go directly to step 8.

### Procedure

1. Identify the **LocalVolumeSet** and the corresponding **StorageClassName** being used by OpenShift Container Storage.

```
$ oc get localvolumesets.local.storage.openshift.io -n openshift-local-storage
```

2. Set the variable `SC` to the **StorageClass** providing the **LocalVolumeSet**.

```
$ export SC="<StorageClassName>"
```

3. List and note the devices to be cleaned up later. In order to list the device ids of the disks, please follow the procedure mentioned here, See [Find the available storage devices](#).

Example output:

```
/dev/disk/by-id/scsi-360050763808104bc28000000000000eb
/dev/disk/by-id/scsi-360050763808104bc28000000000000ef
/dev/disk/by-id/scsi-360050763808104bc28000000000000f3
```

4. Delete the **LocalVolumeSet**.

```
$ oc delete localvolumesets.local.storage.openshift.io <name-of-volumeset> -n openshift-local-storage
```

5. Delete the local storage PVs for the given **StorageClassName**.

```
$ oc get pv | grep $SC | awk '{print $1}' | xargs oc delete pv
```

6. Delete the **StorageClassName**.

```
$ oc delete sc $SC
```

7. Delete the symlinks created by the **LocalVolumeSet**.

```
[[ ! -z $SC ]] && for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o jsonpath='{.items[*].metadata.name}'); do oc debug node/${i} -- chroot /host rm -rfv /mnt/local-storage/${SC}; done
```

8. Delete **LocalVolumeDiscovery**.

```
$ oc delete localvolumediscovery.local.storage.openshift.io/auto-discover-devices -n openshift-local-storage
```



9. Removing **LocalVolume** resources (if any).

Use the following steps to remove the **LocalVolume** resources that were used to provision PVs in the current or previous OpenShift Container Storage version. Also, ensure that these resources are not being used by other tenants on the cluster.

For each of the local volumes, do the following:

- a. Identify the **LocalVolume** and the corresponding **StorageClassName** being used by OpenShift Container Storage.

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

- b. Set the variable LV to the name of the LocalVolume and variable SC to the name of the StorageClass

For example:

```
$ LV=local-block
$ SC=localblock
```

- c. List and note the devices to be cleaned up later.

```
$ oc get localvolume -n openshift-local-storage $LV -o jsonpath='{
.spec.storageClassDevices[].devicePaths[]}'
```

Example output:

```
/dev/sdb
/dev/sdc
/dev/sdd
/dev/sde
```

- d. Delete the local volume resource.

```
$ oc delete localvolume -n openshift-local-storage --wait=true $LV
```

- e. Delete the remaining PVs and StorageClasses if they exist.

```
$ oc delete pv -l storage.openshift.com/local-volume-owner-name=${LV} --wait --
timeout=5m
$ oc delete storageclass $SC --wait --timeout=5m
```

- f. Clean up the artifacts from the storage nodes for that resource.

```
$ [[ ! -z $SC ]] && for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o
jsonpath='{.items[*].metadata.name}'); do oc debug node/${i} -- chroot /host rm -rfv
/mnt/local-storage/${SC}/; done
```

Example output:

```
Starting pod/node-xxx-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
Starting pod/node-yyy-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
Starting pod/node-zzz-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
```

10. Wipe the disks for each of the local volumesets or local volumes listed in step 1 and 8 respectively so that they can be reused.

- a. List the storage nodes.

```
oc get nodes -l cluster.ocs.openshift.io/openshift-storage=
```

Example output:

```
NAME      STATUS  ROLES  AGE   VERSION
node-xxx  Ready  worker  4h45m v1.18.3+6c42de8
node-yyy  Ready  worker  4h46m v1.18.3+6c42de8
node-zzz  Ready  worker  4h45m v1.18.3+6c42de8
```

- b. Obtain the node console and execute **chroot /host** command when the prompt appears.

```
$ oc debug node/node-xxx
Starting pod/node-xxx-debug ...
To use host binaries, run `chroot /host`
Pod IP: w.x.y.z
If you don't see a command prompt, try pressing enter.
sh-4.2# chroot /host
```

- c. Store the disk paths in the **DISKS** variable within quotes. For the list of disk paths, see step 3 and step 8.c for local volumeset and local volume respectively.

Example output:

```
sh-4.4# DISKS="/dev/disk/by-id/scsi-360050763808104bc28000000000000eb
/dev/disk/by-id/scsi-360050763808104bc28000000000000ef /dev/disk/by-id/scsi-
360050763808104bc28000000000000f3 "
or
sh-4.2# DISKS="/dev/sdb /dev/sdc /dev/sdd /dev/sde "
```

- d. Run **sgdisk --zap-all** on all the disks.

```
sh-4.4# for disk in $DISKS; do sgdisk --zap-all $disk;done
```

Example output:

```

Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.

```

- e. Exit the shell and repeat for the other nodes.

```

sh-4.4# exit
exit
sh-4.2# exit
exit
Removing debug pod ...

```

11. Delete the **openshift-local-storage** namespace and wait till the deletion is complete. You will need to switch to another project if the **openshift-local-storage** namespace is the active project.

For example:

```

$ oc project default
$ oc delete project openshift-local-storage --wait=true --timeout=5m

```

The project is deleted if the following command returns a **NotFound** error.

```

$ oc get project openshift-local-storage

```

## 4.2. REMOVING MONITORING STACK FROM OPENSIFT CONTAINER STORAGE

Use this section to clean up the monitoring stack from OpenShift Container Storage.

The PVCs that are created as a part of configuring the monitoring stack are in the **openshift-monitoring** namespace.

### Prerequisites

- PVCs are configured to use OpenShift Container Platform monitoring stack. For information, see [configuring monitoring stack](#).

### Procedure

1. List the pods and PVCs that are currently running in the **openshift-monitoring** namespace.

```

$ oc get pod,pvc -n openshift-monitoring
NAME                                READY STATUS  RESTARTS  AGE

```

```

pod/alertmanager-main-0      3/3  Running 0      8d
pod/alertmanager-main-1      3/3  Running 0      8d
pod/alertmanager-main-2      3/3  Running 0      8d
pod/cluster-monitoring-
operator-84457656d-pkrxm     1/1  Running 0      8d
pod/grafana-79ccf6689f-2ll28 2/2  Running 0      8d
pod/kube-state-metrics-
7d86fb966-rvd9w             3/3  Running 0      8d
pod/node-exporter-25894      2/2  Running 0      8d
pod/node-exporter-4dsd7      2/2  Running 0      8d
pod/node-exporter-6p4zc      2/2  Running 0      8d
pod/node-exporter-jbjvg      2/2  Running 0      8d
pod/node-exporter-jj4t5      2/2  Running 0     6d18h
pod/node-exporter-k856s      2/2  Running 0     6d18h
pod/node-exporter-rf8gn      2/2  Running 0      8d
pod/node-exporter-rmb5m      2/2  Running 0     6d18h
pod/node-exporter-zj7kx      2/2  Running 0      8d
pod/openshift-state-metrics-
59dbd4f654-4clng           3/3  Running 0      8d
pod/prometheus-adapter-
5df5865596-k8dzn           1/1  Running 0     7d23h
pod/prometheus-adapter-
5df5865596-n2gj9           1/1  Running 0     7d23h
pod/prometheus-k8s-0         6/6  Running 1      8d
pod/prometheus-k8s-1         6/6  Running 1      8d
pod/prometheus-operator-
55cfb858c9-c4zd9           1/1  Running 0     6d21h
pod/telemeter-client-
78fc8fc97d-2rgfp           3/3  Running 0      8d

```

```

NAME                                STATUS  VOLUME
CAPACITY ACCESS MODES STORAGECLASS  AGE
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-0 Bound pvc-0d519c4f-
15a5-11ea-baa0-026d231574aa 40Gi  RWO          ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-1 Bound pvc-
0d5a9825-15a5-11ea-baa0-026d231574aa 40Gi  RWO          ocs-storagecluster-ceph-
rbd 8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-2 Bound pvc-
0d6413dc-15a5-11ea-baa0-026d231574aa 40Gi  RWO          ocs-storagecluster-ceph-
rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-0 Bound pvc-0b7c19b0-
15a5-11ea-baa0-026d231574aa 40Gi  RWO          ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-1 Bound pvc-0b8aed3f-
15a5-11ea-baa0-026d231574aa 40Gi  RWO          ocs-storagecluster-ceph-rbd 8d

```

2. Edit the monitoring **configmap**.

```
$ oc -n openshift-monitoring edit configmap cluster-monitoring-config
```

3. Remove any **config** sections that reference the OpenShift Container Storage storage classes as shown in the following example and save it.

**Before editing**

```
.  
. .  
apiVersion: v1  
data:  
  config.yaml: |  
    alertmanagerMain:  
      volumeClaimTemplate:  
        metadata:  
          name: my-alertmanager-claim  
        spec:  
          resources:  
            requests:  
              storage: 40Gi  
          storageClassName: ocs-storagecluster-ceph-rbd  
  prometheusK8s:  
    volumeClaimTemplate:  
      metadata:  
        name: my-prometheus-claim  
      spec:  
        resources:  
          requests:  
            storage: 40Gi  
        storageClassName: ocs-storagecluster-ceph-rbd  
kind: ConfigMap  
metadata:  
  creationTimestamp: "2019-12-02T07:47:29Z"  
  name: cluster-monitoring-config  
  namespace: openshift-monitoring  
  resourceVersion: "22110"  
  selfLink: /api/v1/namespaces/openshift-monitoring/configmaps/cluster-monitoring-config  
  uid: fd6d988b-14d7-11ea-84ff-066035b9efa8  
. . .
```

After editing

```

.
.
.
apiVersion: v1
data:
  config.yaml: |
kind: ConfigMap
metadata:
  creationTimestamp: "2019-11-21T13:07:05Z"
  name: cluster-monitoring-config
  namespace: openshift-monitoring
  resourceVersion: "404352"
  selfLink: /api/v1/namespaces/openshift-monitoring/configmaps/cluster-monitoring-config
  uid: d12c796a-0c5f-11ea-9832-063cd735b81c
.
.
.

```

In this example, **alertmanagerMain** and **prometheusK8s** monitoring components are using the OpenShift Container Storage PVCs.

4. Delete relevant PVCs. Make sure you delete all the PVCs that are consuming the storage classes.

```
$ oc delete -n openshift-monitoring pvc <pvc-name> --wait=true --timeout=5m
```

### 4.3. REMOVING OPENSIFT CONTAINER PLATFORM REGISTRY FROM OPENSIFT CONTAINER STORAGE

Use this section to clean up OpenShift Container Platform registry from OpenShift Container Storage. If you want to configure an alternative storage, see [image registry](#)

The PVCs that are created as a part of configuring OpenShift Container Platform registry are in the **openshift-image-registry** namespace.

#### Prerequisites

- The image registry should have been configured to use an OpenShift Container Storage PVC.

#### Procedure

1. Edit the **configs.imageregistry.operator.openshift.io** object and remove the content in the **storage** section.

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Before editing

```

.
.
.
managementState: Managed
storage:
  pvc:
    claim: registry-cephfs-rwx-pvc
.
.
.

```

#### After editing

```

.
.
.
managementState: Removed
storage:
  emptyDir: {}
.
.
.

```

In this example, the PVC is called **registry-cephfs-rwx-pvc**, which is now safe to delete.

2. Delete the PVC.

```
$ oc delete pvc <pvc-name> -n openshift-image-registry --wait=true --timeout=5m
```

## 4.4. REMOVING THE CLUSTER LOGGING OPERATOR FROM OPENSIFT CONTAINER STORAGE

To clean the cluster logging operator from the OpenShift Container Storage, follow the steps in the procedure.

The PVCs created as a part of configuring cluster logging operator are in the **openshift-logging** namespace.

### Prerequisites

- The cluster logging instance must be configured to use OpenShift Container Storage PVCs.

### Procedure

1. Remove the **ClusterLogging** instance in the namespace.

```
$ oc delete clusterlogging instance -n openshift-logging --wait=true --timeout=5m
```

- 

The PVCs in the **openshift-logging** namespace are now safe to delete.

2. Delete PVCs.

```
█ $ oc delete pvc <pvc-name> -n openshift-logging --wait=true --timeout=5m
```