Deploying OpenShift Container Storage using IBM Power Systems

How to install and set up your IBM Power Systems environment
Red Hat OpenShift Container Storage 4.8 Deploying OpenShift Container Storage using IBM Power Systems

How to install and set up your IBM Power Systems environment
Abstract

Read this document for instructions on installing Red Hat OpenShift Container Storage 4.8 to use local storage on IBM Power Systems.
# Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE ................................................. 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION .................................. 4

PREFACE .................................................................................. 5

CHAPTER 1. PREPARING TO DEPLOY OPENSFHT CONTAINER STORAGE ........................... 6
  1.1. REQUIREMENTS FOR INSTALLING OPENSFHT CONTAINER STORAGE USING LOCAL STORAGE DEVICES .......................... 6
  1.2. ENABLING KEY VALUE BACKEND PATH AND POLICY IN VAULT .................. 7

CHAPTER 2. DEPLOY OPENSFHT CONTAINER STORAGE USING LOCAL STORAGE DEVICES .............. 8
  2.1. INSTALLING LOCAL STORAGE OPERATOR ........................................... 8
  2.2. INSTALLING RED HAT OPENSFHT CONTAINER STORAGE OPERATOR .......... 8
  2.3. FINDING AVAILABLE STORAGE DEVICES ......................................... 9
  2.4. CREATING OPENSFHT CONTAINER STORAGE CLUSTER ON IBM POWER SYSTEMS .......... 11

CHAPTER 3. VERIFYING OPENSFHT CONTAINER STORAGE DEPLOYMENT FOR INTERNAL MODE .............................. 15
  3.1. VERIFYING THE STATE OF THE PODS .............................................. 15
  3.2. VERIFYING THE OPENSFHT CONTAINER STORAGE CLUSTER IS HEALTHY .......... 16
  3.3. VERIFYING THE MULTICLOUD OBJECT GATEWAY IS HEALTHY ................... 17
  3.4. VERIFYING THAT THE OPENSFHT CONTAINER STORAGE SPECIFIC STORAGE CLASSES EXIST .......... 17

CHAPTER 4. UNINSTALLING OPENSFHT CONTAINER STORAGE ..................................... 18
  4.1. UNINSTALLING OPENSFHT CONTAINER STORAGE IN INTERNAL MODE ............. 18
    4.1.1. Removing local storage operator configurations ................................... 23
  4.2. REMOVING MONITORING STACK FROM OPENSFHT CONTAINER STORAGE .......... 25
  4.3. REMOVING OPENSFHT CONTAINER PLATFORM REGISTRY FROM OPENSFHT CONTAINER STORAGE ........ 28
  4.4. REMOVING THE CLUSTER LOGGING OPERATOR FROM OPENSFHT CONTAINER STORAGE .......... 29
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 Power 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 Power Systems. 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:

- Internal-Attached Devices mode
  - Deploy using local storage devices
CHAPTER 1. PREPARING TO DEPLOY OPENShift CONTAINER STORAGE

Deploying OpenShift Container Storage on OpenShift Container Platform using local storage devices provided by IBM Power Systems enables you to create internal cluster resources. This approach internally provisions base services. Then, all applications can access additional storage classes.

NOTE

Only internal Openshift Container Storage clusters are supported on IBM Power Systems. See Planning your deployment for more information about deployment requirements.

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.

- On the external key management system (KMS),
  - Ensure that a policy with a token exists and the key value backend path in Vault is enabled. see enabling key value backend path and policy in vault.
  - Ensure that you are using signed certificates on your Vault servers.

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

1. Install Local Storage Operator.
2. Install the Red Hat OpenShift Container Storage Operator.
3. Finding available storage devices.

1.1. REQUIREMENTS FOR INSTALLING OPENSShift CONTAINER STORAGE USING LOCAL STORAGE DEVICES

Node requirements

- The cluster must consist of at least three OpenShift Container Platform worker nodes in the cluster 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 to be used must be empty, that is, there should be no persistent volumes (PVs), volume groups (VGs), or local volumes (LVs) remaining on the disks.
- You must have a minimum of three labeled nodes.
  - Each node that has local storage devices to be used by OpenShift Container Storage must have a specific label to deploy OpenShift Container Storage pods. To label the nodes, use the following command:

```bash
$ oc label nodes <NodeNames> cluster.ocs.openshift.io/openshift-storage=""
```
See the Resource requirements section in Planning guide.

1.2. ENABLING KEY VALUE BACKEND PATH AND POLICY IN VAULT

Prerequisites

- Administrator access to Vault.
- Choose a unique path name as the backend path that follows the naming convention since it cannot be changed later.

Procedure

1. Enable the Key/Value (KV) backend path in Vault.
   For Vault KV secret engine API, version 1:
   
   ```bash
   $ vault secrets enable -path=ocs kv
   ```
   For Vault KV secret engine API, version 2:
   
   ```bash
   $ vault secrets enable -path=ocs kv-v2
   ```

2. Create a policy to restrict users to perform a write or delete operation on the secret using the following commands:
   
   ```bash
   echo 'path "ocs/*/" {
       capabilities = ["create", "read", "update", "delete", "list"]
   }
   path "sys/mounts" {
       capabilities = ["read"]
   }' vault policy write ocs
   ```

3. Create a token matching the above policy:
   
   ```bash
   $ vault token create -policy=ocs -format json
   ```
CHAPTER 2. DEPLOY OPENSSHIFT CONTAINER STORAGE USING LOCAL STORAGE DEVICES

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

Follow the below steps in the order given:

1. Install Local Storage Operator.
2. Install the Red Hat OpenShift Container Storage Operator.
3. Finding available storage devices.

2.1. INSTALLING LOCAL STORAGE OPERATOR

Use this procedure to install the Local Storage Operator from the Operator Hub before creating OpenShift Container Storage clusters on local storage devices.

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. Update 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.
7. Verify that the Local Storage Operator shows the Status as Succeeded.

2.2. INSTALLING RED HAT OPENSIFT CONTAINER STORAGE OPERATOR


For information about the hardware and software requirements, see Planning your deployment.
Prerequisites

- Access to an OpenShift Container Platform cluster using an account with cluster-admin and Operator installation permissions.
- You must have at least three worker nodes in the RHOCP cluster.

**NOTE**

- When you need to override the cluster-wide default node selector for OpenShift Container Storage, you can use the following command in command line interface 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. Navigate in the left pane of the OpenShift Web Console to click Operators → OperatorHub.
2. Scroll or type a keyword into the Filter by keyword box to search for OpenShift Container Storage Operator.
3. Click Install on the OpenShift Container Storage operator page.
4. On the Install Operator page, the following required options are selected by default:
   a. Update 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. Select Approval Strategy as Automatic or Manual.
   e. Click Install.
      - If you selected Automatic updates, then the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without any intervention.
      - If you selected Manual updates, then 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 steps

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

2.3. FINDING AVAILABLE STORAGE DEVICES
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 PVs for IBM Power Systems.

**Procedure**

1. List and verify the name of the worker nodes with the OpenShift Container Storage label.
   
   ```bash
   $ oc get nodes -l cluster.ocs.openshift.io/openshift-storage=
   
   Example output:
   
<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>ROLES</th>
<th>AGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>worker-0</td>
<td>Ready</td>
<td>worker</td>
<td>2d11h</td>
<td>v1.21.1+f36aa36</td>
</tr>
<tr>
<td>worker-1</td>
<td>Ready</td>
<td>worker</td>
<td>2d11h</td>
<td>v1.21.1+f36aa36</td>
</tr>
<tr>
<td>worker-2</td>
<td>Ready</td>
<td>worker</td>
<td>2d11h</td>
<td>v1.21.1+f36aa36</td>
</tr>
</tbody>
</table>
   
   **Note:** If you don't see a command prompt, try pressing enter.
   
   `sh-4.4#`  
   `sh-4.4# lsblk`

2. Log in to each worker node that is used for OpenShift Container Storage resources and find the name of the additional disk that you have attached while deploying OpenShift Container Platform.

   ```bash
   $ oc debug node/<node name>
   
   Example output:
   
   $ oc debug node/worker-0  
   Starting pod/worker-0-debug ...  
   To use host binaries, run `chroot /host`  
   Pod IP: 192.168.0.63  
   If you don't see a command prompt, try pressing enter.  
   `sh-4.4#`  
   `sh-4.4# lsblk`

   Example output:

<table>
<thead>
<tr>
<th>NAME</th>
<th>MAJ:MIN RM</th>
<th>SIZE</th>
<th>RO</th>
<th>TYPE</th>
<th>MOUNTPOINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>loop1</td>
<td>7:1</td>
<td>500G</td>
<td>0</td>
<td>loop</td>
<td></td>
</tr>
<tr>
<td>sda</td>
<td>8:0</td>
<td>500G</td>
<td>0</td>
<td>disk</td>
<td></td>
</tr>
<tr>
<td>sdb</td>
<td>8:16</td>
<td>120G</td>
<td>0</td>
<td>disk</td>
<td></td>
</tr>
<tr>
<td>sdb1</td>
<td>8:17</td>
<td>4M</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sdb3</td>
<td>8:19</td>
<td>384M</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sdb4</td>
<td>8:20</td>
<td>119.6G</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sdc</td>
<td>8:32</td>
<td>500G</td>
<td>0</td>
<td>disk</td>
<td></td>
</tr>
<tr>
<td>sdd</td>
<td>8:48</td>
<td>120G</td>
<td>0</td>
<td>disk</td>
<td></td>
</tr>
<tr>
<td>sdd1</td>
<td>8:49</td>
<td>4M</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sdd3</td>
<td>8:51</td>
<td>384M</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sdd4</td>
<td>8:52</td>
<td>119.6G</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sde</td>
<td>8:64</td>
<td>500G</td>
<td>0</td>
<td>disk</td>
<td></td>
</tr>
<tr>
<td>sdf</td>
<td>8:80</td>
<td>120G</td>
<td>0</td>
<td>disk</td>
<td></td>
</tr>
<tr>
<td>sdf1</td>
<td>8:81</td>
<td>4M</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sdf3</td>
<td>8:83</td>
<td>384M</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sdf4</td>
<td>8:84</td>
<td>119.6G</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sdg</td>
<td>8:96</td>
<td>500G</td>
<td>0</td>
<td>disk</td>
<td></td>
</tr>
<tr>
<td>sdh</td>
<td>8:112</td>
<td>120G</td>
<td>0</td>
<td>disk</td>
<td></td>
</tr>
<tr>
<td>sdh1</td>
<td>8:113</td>
<td>4M</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
<tr>
<td>sdh3</td>
<td>8:115</td>
<td>384M</td>
<td>0</td>
<td>part</td>
<td></td>
</tr>
</tbody>
</table>
In this example, for worker-0, the available local devices of 500G are **sda, sdc, sde, sdg, sdi, sdk, sdm, sdo**.

3. Repeat the above step for all the other worker 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 POWER SYSTEMS

### Prerequisites

- Ensure that all the requirements in the Requirements for installing OpenShift Container Storage using local storage devices section are met.

- You must have a minimum of three worker nodes with the same storage type and size attached to each node (for example, 200 GB SSD) to use local storage devices on IBM Power Systems.

- Verify your OpenShift Container Platform worker nodes are labeled for OpenShift Container Storage:

  ```
  oc get nodes -l cluster.ocs.openshift.io/openshift-storage -o jsonpath='{range .items[*]}{.metadata.name}{"\n"}'
  ```

  To identify storage devices on each node, refer to Finding available storage devices.

### Procedure

1. Log into the OpenShift Web Console.

2. In openshift-local-storage namespace Click Operators → Installed Operators to view the installed operators.
3. Click the **Local Storage** installed operator.

4. On the **Operator Details** page, click the **Local Volume** link.

5. Click **Create Local Volume**

6. Click on **YAML view** for configuring Local Volume.

7. Define a **LocalVolume** custom resource for block PVs using the following YAML.

   ```yaml
   apiVersion: local.storage.openshift.io/v1
   kind: LocalVolume
   metadata:
     name: localblock
     namespace: openshift-local-storage
   spec:
     logLevel: Normal
     managementState: Managed
     nodeSelector:
       nodeSelectorTerms:
         - matchExpressions:
             - key: kubernetes.io/hostname
               operator: In
               values:
                 - worker-0
                 - worker-1
                 - worker-2
       storageClassDevices:
         - devicePaths:
             - /dev/sda
       storageClassName: localblock
       volumeMode: Block
   ``

   The above definition selects **sda** local device from the **worker-0, worker-1** and **worker-2** nodes. The **localblock** storage class is created and persistent volumes are provisioned from **sda**.

   **IMPORTANT**

   Specify appropriate values of nodeSelector as per your environment. The device name should be same on all the worker nodes. You can also specify more than one devicePaths.

8. Click **Create**.

9. Confirm whether **diskmaker-manager** pods and **Persistent Volumes** are created.

   a. For **Pods**

      i. Click **Workloads → Pods** from the left pane of the OpenShift Web Console.

      ii. Select **openshift-local-storage** from the **Project** drop down list.

      iii. Check if there are **diskmaker-manager** pods for each of the worker node that you used while creating LocalVolume CR.
b. For Persistent Volumes

i. Click Storage → PersistentVolumes from the left pane of the OpenShift Web Console.

ii. Check the Persistent Volumes with the name `local-pv-*`. Number of Persistent Volumes will be equivalent to the product of number of worker nodes and number of storage devices provisioned while creating localVolume CR.

**IMPORTANT**

The flexible scaling feature gets enabled on creating a storage cluster with 3 or more nodes spread across fewer than the minimum requirement of 3 availability zones. This feature is available only for the new deployments of OpenShift Container Storage 4.7 clusters and does not support the upgraded clusters. For information about flexible scaling, see Scaling Storage Guide

10. Click Operators → Installed Operators from the left pane of the OpenShift Web Console to view the installed operators.

11. Select `openshift-storage` from the Project drop down list.

12. Click the OpenShift Container Storage installed operator.

13. On the Operator Details page, click the Storage Cluster link.

14. Click Create Storage Cluster.

   a. Select Internal-Attached devices for the Select Mode.

   b. Click on Storage and Nodes

   c. Select the required storage class.

   d. The nodes corresponding to the storage class are displayed based on the storage class that you selected from the drop down.

   e. Click Next.

   f. (Optional) Set Security and network configuration

      i. Select the Enable encryption checkbox to encrypt block and file storage.

      ii. Choose any one or both Encryption level:

         - Cluster-wide encryption to encrypt the entire cluster (block and file).

         - Storage class encryption to create encrypted persistent volume (block only) using encryption enabled storage class.

      iii. Select the Connect to an external key management service checkbox. This is optional for cluster-wide encryption.

         A. Key Management Service Provider is set to Vault by default.

         B. Enter Vault Service Name, host Address of Vault server (‘https://<hostname or ip>’), Port number and Token.
iv. Expand **Advanced Settings** to enter additional settings and certificate details:
   A. Enter the Key Value secret path in **Backend Path** that is dedicated and unique to OpenShift Container Storage.
   
   B. Enter **TLS Server Name** and **Vault Enterprise Namespace**
   
   C. Provide **CA Certificate**, **Client Certificate** and **Client Private Key** by uploading the respective PEM encoded certificate file.
   
   D. Click **Save**.

   g. Click **Next**.

   h. Review the configurations details. To modify any configuration settings, click **Back** to go back to the previous configuration page.

   i. Click **Create**.

**Verification steps**

- Verify that the final **Status** of the installed storage cluster shows as Phase: Ready with a green tick mark.
  
  - Click **Operators** → **Installed Operators** → **Storage Cluster** link to view the storage cluster installation status.
  
  - Alternatively, when you are on the Operator **Details** tab, you can click on the **Storage Cluster** tab to view the status.

- To verify if flexible scaling is enabled on your storage cluster, perform the following steps:
  
  1. Click **ocs-storagecluster** in **Storage Cluster** tab.
  
  2. In the YAML tab, search for the keys **flexibleScaling** in **spec** section and **failureDomain** in **status** section. If **flexible scaling** is true and **failureDomain** is set to host, flexible scaling feature is enabled.

```yaml
spec:
  flexibleScaling: true
  […]
status:
  failureDomain: host
```

- To verify that all components for OpenShift Container Storage are successfully installed, see **Verifying your OpenShift Container Storage installation**.

**Additional resources**

- To expand the capacity of the initial cluster, see the **Scaling Storage** guide.
CHAPTER 3. VERIFYING OPENSIFT CONTAINER STORAGE DEPLOYMENT FOR INTERNAL MODE

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

3.1. VERIFYING THE STATE OF THE PODS

To determine if OpenShift Container storage is deployed successfully, you can verify that the pods are in **Running** state.

**Procedure**

1. Click Workloads → Pods from the left pane of the OpenShift Web Console.

2. 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".

3. Verify that the following pods are in running and completed state by clicking on the Running and the Completed tabs:

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

<table>
<thead>
<tr>
<th>Component</th>
<th>Corresponding pods</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenShift Container Storage Operator</td>
<td>ocs-operator-* (1 pod on any worker node)</td>
</tr>
<tr>
<td></td>
<td>ocs-metrics-exporter-*</td>
</tr>
<tr>
<td>Rook-ceph Operator</td>
<td>rook-ceph-operator-* (1 pod on any worker node)</td>
</tr>
<tr>
<td>Multicloud Object Gateway</td>
<td>noobaa-operator-* (1 pod on any worker node)</td>
</tr>
<tr>
<td></td>
<td>noobaa-core-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>noobaa-db-pg-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>noobaa-endpoint-* (1 pod on any storage node)</td>
</tr>
<tr>
<td>MON</td>
<td>rook-ceph-mon-* (3 pods on each storage node)</td>
</tr>
<tr>
<td>MGR</td>
<td>rook-ceph-mgr-* (1 pod on any storage node)</td>
</tr>
<tr>
<td>Component</td>
<td>Corresponding pods</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MDS</td>
<td><code>rook-ceph-mds-ocs-storagecluster-cephfilesystem-*</code> (2 pods distributed across storage node)</td>
</tr>
<tr>
<td>RGW</td>
<td><code>rook-ceph-rgw-ocs-storagecluster-cephobjectstore-*</code> (1 pod on any storage node)</td>
</tr>
<tr>
<td>CSI</td>
<td>• cephfs&lt;br&gt;  o <code>csi-cephfsplugin-*</code> (1 pod on each worker node)&lt;br&gt;  o <code>csi-cephfsplugin-provisioner-*</code> (2 pods distributed across storage nodes)&lt;br&gt;• rbd&lt;br&gt;  o <code>csi-rbdplugin-*</code> (1 pod on each worker node)&lt;br&gt;  o <code>csi-rbdplugin-provisioner-*</code> (2 pods distributed across storage nodes)</td>
</tr>
<tr>
<td><code>rook-ceph-crashcollector</code></td>
<td><code>rook-ceph-crashcollector-*</code> (1 pod on each storage node)</td>
</tr>
<tr>
<td>OSD</td>
<td>• <code>rook-ceph-osd-*</code> (1 pod for each device)&lt;br&gt;• <code>rook-ceph-osd-prepare-ocs-deviceset-*</code> (1 pod for each device)</td>
</tr>
</tbody>
</table>

### 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 THE MULTICLOUD OBJECT GATEWAY IS HEALTHY

To verify that the OpenShift Container Storage Multicloud Object Gateway is healthy, follow the steps in the procedure.

Procedure

1. Click **Storage → Overview** from the OpenShift Web Console and click the **Object** tab.

2. In the **Status card**, verify that both **Object Service** and **Data Resiliency** are in **Ready** state (green tick).

3. In the **Details card**, verify that the Multicloud Object Gateway information is displayed.

For more information on the health of the OpenShift Container Storage cluster using the object service dashboard, see [Monitoring OpenShift Container Storage](#).

3.4. VERIFYING THAT THE OPENSHIFT 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 OPENSFiGHT CONTAINER STORAGE

4.1. UNINSTALLING OPENSFiGHT CONTAINER STORAGE IN INTERNAL 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 used with these annotations:

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

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

```bash
$ 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 until 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.

      For more information, see Removing monitoring stack from OpenShift Container Storage

   b. Delete OpenShift Container Platform Registry PVCs using OpenShift Container Storage.
      For more information, see Removing OpenShift Container Platform registry from OpenShift Container Storage

   c. Delete OpenShift Container Platform logging PVCs using OpenShift Container Storage.
      For more information, see Removing the cluster logging operator from OpenShift Container Storage

   d. Delete other PVCs and OBCs provisioned using OpenShift Container Storage.

      - Given below is a sample script to identify the PVCs and OBCs provisioned using OpenShift Container Storage. The script ignores the PVCs that are used internally by OpenShift Container Storage.
#!/bin/bash

RBD_PROVISIONER="openshift-storage.rbd.csi.ceph.com"
CEPHFS_PROVISIONER="openshift-storage.cephfs.csi.ceph.com"
NOOBAA_PROVISIONER="openshift-storage.noobaa.io/obc"
RGW_PROVISIONER="openshift-storage.ceph.rook.io/bucket"

NOOBAA_DB_PVC="noobaa-db"
NOOBAA_BACKINGSTORE_PVC="noobaa-default-backing-store-noobaa-pvc"

# Find all the OCS StorageClasses
OCS_STORAGECLASSES=$(oc get storageclasses | grep -e "$RBD_PROVISIONER" -e "$CEPHFS_PROVISIONER" -e "$NOOBAA_PROVISIONER" -e "$RGW_PROVISIONER" | awk '{print $1}')

# List PVCs in each of the StorageClasses
for SC in $OCS_STORAGECLASSES
   do
      echo
      "=====================================================================
      =="
      echo "$SC StorageClass PVCs and OBCs"
      echo
      "=====================================================================
      =="
      oc get pvc --all-namespaces --no-headers 2>/dev/null | grep $SC | grep -v -e "$NOOBAA_DB_PVC" -e "$NOOBAA_BACKINGSTORE_PVC"
      oc get obc --all-namespaces --no-headers 2>/dev/null | grep $SC
      echo
done

NOTE
Omit RGW_PROVISIONER for cloud platforms.

- Delete the OBCs.
  
  $ oc delete obc <obc name> -n <project name>

- Delete the PVCs.
  
  $ oc delete pvc <pvc name> -n <project-name>

NOTE
Ensure that you have removed any custom backing stores, bucket classes, etc., created in the cluster.

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`.

   ```bash
   $ 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).

   ```bash
   $ 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.

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

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

   ```bash
   $ dmsetup ls
   ocs-deviceset-localblock-0-data-1x4r7g-block-dmcrypt (253:1)
   ```

   c. Remove the mapped device.

   ```bash
   $ cryptsetup luksClose --debug --verbose ocs-deviceset-localblock-0-data-1x4r7g-block-dmcrypt
   ```

   **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.

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

   - Terminate the process using `kill` command.

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

   - Verify that the device name is removed.

     ```bash
     $ 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 --wait=true --timeout=5m
   ```
   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 the 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
    ```

    ```
    $ oc delete crd backingstores.noobaa.io bucketclasses.noobaa.io
    cephblockpools.ceph.rook.io cephclusters.ceph.rook.io cephfilesystems.ceph.rook.io
    cephfnses.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**.
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.

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

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

3. Delete the `LocalVolumeSet`.

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

4. Delete the local storage PVs for the given `StorageClassName`.

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

5. Delete the `StorageClassName`.

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

6. Delete the symlinks created by the `LocalVolumeSet`.

   ```bash
   [[ ! -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
   ```

7. Delete `LocalVolumeDiscovery`.

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

8. 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. Cancel the local volume pod.s
   b. Wait for the local volumes to be deleted
   c. Verify that `Overview` no longer appears under Storage.
a. Identify the LocalVolume and the corresponding StorageClassName being used by OpenShift Container Storage.

b. Set the variable LV to the name of the LocalVolume and variable SC to the name of the StorageClass
   For example:

   ```
   $ LV=localblock
   $ SC=localblock
   ```

c. Delete the local volume resource.

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

d. 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
   ```

e. 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/sda'
   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/sda'
   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/sda'
   removed directory '/mnt/local-storage/localblock'
   Removing debug pod ...
   ```

9. Delete the openshift-local-storage namespace and wait until 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 OPENSHIFT 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.

```bash
$ 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-2l28     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-59dbd4654-4clng 3/3     Running   0          8d
pod/prometheus-adapter-5df5865596-k8dz80 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-55c8b858c9-c4zd9 1/1     Running   0          6d21h
pod/telemeter-client-
```
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
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 OPENSSHIFT CONTAINER PLATFORM REGISTRY FROM OPENSHEET 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
In this example, the PVC is called `registry-cephfs-rwx-pvc`, which is now safe to delete.

2. Delete the PVC.

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

### 4.4. REMOVING THE CLUSTER LOGGING OPERATOR FROM OPENSHIFT 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.

   ```bash
   $ 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