Red Hat OpenShift Container Storage 4.7

Deploying OpenShift Container Storage on VMware vSphere

How to install OpenShift Container Storage on Red Hat OpenShift Container Platform VMware vSphere clusters
Abstract

Read this document for instructions on installing Red Hat OpenShift Container Storage 4.7 on Red Hat OpenShift Container Platform VMware vSphere clusters.
# Table of Contents

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

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

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

CHAPTER 1. PREPARING TO DEPLOY OPENSHIFT CONTAINER STORAGE ................. 6
  1.1. ENABLING FILE SYSTEM ACCESS FOR CONTAINERS ON RED HAT ENTERPRISE LINUX BASED NODES 6
  1.2. ENABLING KEY VALUE BACKEND PATH AND POLICY IN VAULT 7
  1.3. REQUIREMENTS FOR INSTALLING OPENSHIFT CONTAINER STORAGE USING LOCAL STORAGE DEVICES 8

CHAPTER 2. DEPLOY USING DYNAMIC STORAGE DEVICES .................................. 9
  2.1. INSTALLING RED HAT OPENSHIFT CONTAINER STORAGE OPERATOR 9
  2.2. CREATING AN OPENSHIFT CONTAINER STORAGE CLUSTER SERVICE IN INTERNAL MODE 10

CHAPTER 3. DEPLOY USING LOCAL STORAGE DEVICES ..................................... 14
  3.1. INSTALLING LOCAL STORAGE OPERATOR 14
  3.2. INSTALLING RED HAT OPENSHIFT CONTAINER STORAGE OPERATOR 14
  3.3. CREATING OPENSHIFT CONTAINER STORAGE CLUSTER ON VMWARE 16

CHAPTER 4. VERIFYING OPENSHIFT CONTAINER STORAGE DEPLOYMENT FOR INTERNAL MODE ... 21
  4.1. VERIFYING THE STATE OF THE PODS 21
  4.2. VERIFYING THE OPENSHIFT CONTAINER STORAGE CLUSTER IS HEALTHY 23
  4.3. VERIFYING THE MULTICLOUD OBJECT GATEWAY IS HEALTHY 23
  4.4. VERIFYING THAT THE OPENSHIFT CONTAINER STORAGE SPECIFIC STORAGE CLASSES EXIST 24

CHAPTER 5. UNINSTALLING OPENSHIFT CONTAINER STORAGE ........................... 25
  5.1. UNINSTALLING OPENSHIFT CONTAINER STORAGE IN INTERNAL MODE 25
    5.1.1. Removing local storage operator configurations 30
  5.2. REMOVING MONITORING STACK FROM OPENSHIFT CONTAINER STORAGE 31
  5.3. REMOVING OPENSHIFT CONTAINER PLATFORM REGISTRY FROM OPENSHIFT CONTAINER STORAGE 34
  5.4. REMOVING THE CLUSTER LOGGING OPERATOR FROM OPENSHIFT CONTAINER STORAGE 35
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.
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**.
Red Hat OpenShift Container Storage 4.7 supports deployment on existing Red Hat OpenShift Container Platform (RHOCP) vSphere clusters in connected or disconnected environments along with out-of-the-box support for proxy environments.

**NOTE**

Both internal and external OpenShift Container Storage clusters are supported on VMware vSphere. See Planning your deployment and Preparing to deploy OpenShift Container Storage for more information about deployment requirements.

To deploy OpenShift Container Storage, start with the requirements in the Preparing to deploy OpenShift Container Storage chapter and then follow any one of the below deployment process for your environment:

- **Internal mode**
  - Deploy using dynamic storage devices
  - Deploy using local storage devices

- **External mode**
CHAPTER 1. PREPARING TO DEPLOY OPENSOURCE CONTAINER STORAGE

Deploying OpenShift Container Storage on OpenShift Container Platform using dynamic or local storage devices provides you with the option to create internal cluster resources. This will result in the internal provisioning of the base services, which helps to make additional storage classes available to applications.

Before you begin the deployment of Red Hat OpenShift Container Storage using dynamic or local storage, ensure that your resource requirements are met. See Planning your deployment.

---

1. For Red Hat Enterprise Linux based hosts for worker nodes in a user provisioned infrastructure (UPI), enable the container access to the underlying file system. Follow the instructions on enable file system access for containers on Red Hat Enterprise Linux based nodes.

   **NOTE**
   
   Skip this step for Red Hat Enterprise Linux CoreOS (RHCOS).

2. 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.

3. Minimum starting node requirements [Technology Preview]
   
   An OpenShift Container Storage cluster will be deployed with minimum configuration when the standard deployment resource requirement is not met. See Resource requirements section in Planning guide.

4. For deploying using local storage devices, see requirements for installing OpenShift Container Storage using local storage devices. These are not applicable for deployment using dynamic storage devices.

---

1.1. ENABLING FILE SYSTEM ACCESS FOR CONTAINERS ON RED HAT ENTERPRISE LINUX BASED NODES

Deploying OpenShift Container Storage on an OpenShift Container Platform with worker nodes on a Red Hat Enterprise Linux base in a user provisioned infrastructure (UPI) does not automatically provide container access to the underlying Ceph file system.

   **NOTE**
   
   Skip this step for hosts based on Red Hat Enterprise Linux CoreOS (RHCOS).

---

**Procedure**

1. Log in to the Red Hat Enterprise Linux based node and open a terminal.

2. For each node in your cluster:
   
   a. Verify that the node has access to the rhel-7-server-extras-rpms repository.
# subscription-manager repos --list-enabled | grep rhel-7-server

If you do not see both `rhel-7-server-rpms` and `rhel-7-server-extras-rpms` in the output, or if there is no output, run the following commands to enable each repository.

```
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

b. Install the required packages.

```
# yum install -y policycoreutils container-selinux
```

c. Persistently enable container use of the Ceph file system in SELinux.

```
# setsebool -P container_use_cephfs on
```

1.2. ENABLING KEY VALUE BACKEND PATH AND POLICY IN VAULT

Prerequisites

- Administrator access to Vault.
- Carefully, 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:
   
   ```
   $ vault secrets enable -path=ocs kv
   ```
   
   For Vault KV secret engine API, version 2:
   
   ```
   $ 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:

   ```
   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:

   ```
   $ vault token create -policy=ocs -format json
   ```
1.3. REQUIREMENTS FOR INSTALLING OPENSOURCE 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.

See the Resource requirements section in Planning guide.

Arbiter stretch cluster requirements [Technology Preview]

- You need a minimum of five nodes in three zones. Two zones include two nodes per data-center zone while the third zone includes one node. A master node can be used for the arbiter zone.

- This solution is designed to be deployed where latencies do not exceed 4 milliseconds round-trip time (RTT) between locations. Contact Red Hat Customer Support if you are planning to deploy with higher latencies.

- Each node must be pre-labeled with its zone label. To label the nodes use the following command:

  ```
  $ oc label nodes <NodeNames> topology.kubernetes.io/zone='<label>'
  ```

  For example, you can label the nodes as follows:

  - **topology.kubernetes.io/zone=arbiter** to master or worker node
  - **topology.kubernetes.io/zone=datacenter1** to at least two worker nodes
  - **topology.kubernetes.io/zone=datacenter2** to at least two worker nodes

Minimum starting node requirements [Technology Preview]

An OpenShift Container Storage cluster will be deployed with minimum configuration when the standard deployment resource requirement is not met. See Resource requirements section in Planning guide.
CHAPTER 2. DEPLOY USING DYNAMIC STORAGE DEVICES

Deploying OpenShift Container Storage on OpenShift Container Platform using dynamic storage devices provided by VMware vSphere (disk format: thin) provides you with the option to create internal cluster resources. This will result in the internal provisioning of the base services, which helps to make additional storage classes available to applications.

NOTE

Both internal and external OpenShift Container Storage clusters are supported on VMware vSphere. See Planning your deployment for more information about deployment requirements.

Also, ensure that you have addressed the requirements in Preparing to deploy OpenShift Container Storage chapter before proceeding with the below steps for deploying using dynamic storage devices:

1. Install the Red Hat OpenShift Container Storage Operator.
2. Create the OpenShift Container Storage Cluster Service.

2.1. INSTALLING RED HAT OPENSHIFT CONTAINER STORAGE OPERATOR


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 RHOC cluster.
- For additional resource requirements, 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 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 web console to the 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-4.7**.
   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 the **OpenShift Container Storage** Operator shows a green tick indicating successful installation.

**Next steps**

- Create OpenShift Container Storage cluster.
  
  For information, see **Creating OpenShift Container Storage cluster in internal mode**.

---

### 2.2. CREATING AN OPENSIFT CONTAINER STORAGE CLUSTER SERVICE IN INTERNAL MODE

Use this procedure to create an OpenShift Container Storage Cluster Service after you install the OpenShift Container Storage operator.

**Prerequisites**

- The OpenShift Container Storage operator must be installed from the Operator Hub. For more information, see **Installing OpenShift Container Storage Operator using the Operator Hub**.

- For VMs on VMware, ensure the **disk.EnableUUID** option is set to **TRUE**. You need to have vCenter account privileges to configure the VMs. For more information, see **Required vCenter account privileges**. To set the **disk.EnableUUID** option, use the **Advanced** option of the **VM Options** in the **Customize hardware tab**. For more information, see **Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines in vSphere**.

**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 > Create Instance** link of Storage Cluster.

4. **Select Mode** is set to **Internal** by default.

5. In **Select capacity and nodes**,
   a. Select **Storage Class**. By default, it is set to **thin**.
   b. Select **Requested Capacity** from the drop down list. It is set to **2 TiB** by default. You can use the drop down to modify the capacity value.

   **NOTE**
   Once you select the initial storage capacity, cluster expansion is performed only using the selected usable capacity (3 times of raw storage).

   c. In the **Select Nodes** section, select at least three available nodes.
      Spread the worker nodes across three different physical nodes, racks or failure domains for high availability.

      Use vCenter anti-affinity to align OpenShift Container Storage rack labels with physical nodes and racks in the data center to avoid scheduling two worker nodes on the same physical chassis.

      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.

   d. Click **Next**.

6. (Optional) **Security configuration**
   a. Select the **Enable encryption** checkbox to encrypt block and file storage.
   b. 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.

   **IMPORTANT**
   Storage class encryption is a Technology Preview feature available only for RBD PVs. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

   For more information, see **Technology Preview Features Support Scope**.
c. Select the **Connect to an external key management service** checkbox. This is optional for cluster-wide encryption.

i. **Key Management Service Provider** is set to **Vault** by default.

ii. Enter **Vault Service Name**, host **Address** of Vault server (‘https://<hostname or ip>’), **Port number** and **Token**.

iii. Expand **Advanced Settings** to enter additional settings and certificate details based on your Vault configuration:

   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**.

d. Click **Next**.

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

8. Click **Create**.

9. Edit the configmap if Vault Key/Value (KV) secret engine API, version 2 is used for cluster-wide encryption with Key Management System (KMS).

   a. On the OpenShift Web Console, navigate to **Workloads → ConfigMaps**

   b. To view the KMS connection details, click **ocs-kms-connection-details**

   c. Edit the configmap.

      i. Click **Action menu ( ⋮ ) → Edit ConfigMap**

      ii. Set the **VAULT_BACKEND** parameter to **v2**.

      ```yaml
kind: ConfigMap
apiVersion: v1
metadata:
  name: ocs-kms-connection-details
[...]
data:
  KMS_PROVIDER: vault
  KMS_SERVICE_NAME: vault
[...]
  VAULT_BACKEND: v2
[...]
      ```

      iii. Click **Save**.

**Verification steps**

1. On the storage cluster details page, the storage cluster name displays a green tick next to it to indicate that the cluster was created successfully.

2. 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.

3. To verify that all components for OpenShift Container Storage are successfully installed, see [Verifying your OpenShift Container Storage installation](#) .
CHAPTER 3. DEPLOY 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. This will result in the internal provisioning of the base services, which helps to make additional storage classes available to applications.

Use this section to deploy OpenShift Container Storage on VMware where OpenShift Container Platform is already installed.

Also, ensure that you have addressed the requirements in Preparing to deploy OpenShift Container Storage chapter before proceeding with the next steps.

1. Installing Local Storage Operator
2. Install the Red Hat OpenShift Container Storage Operator.
3. Create the OpenShift Container Storage Cluster Service.

3.1. INSTALLING LOCAL STORAGE OPERATOR

Procedure

1. Log in to the OpenShift Web Console.
2. Click Operators → OperatorHub.
3. 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-4.7
   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
5. Click Install.
6. Verify that the Local Storage Operator shows the Status as Succeeded.

3.2. INSTALLING RED HAT OPENSOURCES CONTAINER STORAGE OPERATOR


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 RHOCP cluster.

• For additional resource requirements, 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 in command line interface to specify a blank node selector for the `openshift-storage` namespace (create openshift-storage namespace in this case):

```bash
$ 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 web console to the 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-4.7**.
   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 the OpenShift Container Storage Operator shows a green tick indicating successful installation.

**Next steps**

• Create OpenShift Container Storage cluster.
3.3. CREATING OPENSHIFT CONTAINER STORAGE CLUSTER ON VMWARE

VMware supports the following three types of local storage:

- Virtual machine disk (VMDK)
- Raw device mapping (RDM)
- VMDirectPath I/O

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 to use local storage devices on VMware.
- For VMs on VMware, ensure the disk.EnableUUID option is set to TRUE. You need to have vCenter account privileges to configure the VMs. For more information, see Required vCenter account privileges. To set the disk.EnableUUID option, use the Advanced option of the VM Options in the Customize hardware tab. For more information, see Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines in vSphere.

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 > Create Instance link of Storage Cluster.
4. Choose Select Mode as Internal-Attached devices.

NOTE

You are prompted to install the Local Storage Operator if it is not already installed. Click Install and follows procedure as described in Installing Local Storage Operator.

5. Discover disks
   a. Choose one of the following:
      - All nodes to discover disks in all the nodes.
      - Select nodes to discover disks from a subset of available nodes.
IMPORTANT

For arbiter, do not select All nodes option. Instead, use Select nodes option to select the labeled nodes with attached storage device(s) from data-center zones.

b. Click Next.

6. 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. You can also change the name.

c. The nodes selected for disk discovery in the previous 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. Spread the worker nodes across three different physical nodes, racks or failure domains for high availability.

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.

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.

d. Select SSD/NVME Disk Type from the available list.

e. Expand the Advanced section and set the following options:

<table>
<thead>
<tr>
<th>Volume Mode</th>
<th>Block is selected by default.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Type</td>
<td>Select one or more disk type from the drop down list.</td>
</tr>
<tr>
<td>Disk Size</td>
<td>Set a minimum size of 100GB for the device and maximum available size of the device that needs to be included.</td>
</tr>
</tbody>
</table>
f. Click Next. A pop-up to confirm creation of the new storage class is displayed.

g. Click Yes to continue.

7. Set Storage and nodes

a. Select Storage Class. By default, the new storage class created in the previous step is selected.

b. (Optional) Select Enable arbiter checkbox if you want to use the stretch clusters. This option is available only when all the prerequisites for arbiter are fulfilled and the Selected Nodes are populated. For more details, see Arbiter stretch cluster requirements [Technology Preview].

   - Select the arbiter zone from the available drop down list.

c. Selected Nodes shows the nodes selected in the previous step. This list takes a few minutes to reflect the disks that were discovered in the previous step.

d. Click Next.

8. (Optional) Security configuration

a. Select Enable encryption checkbox to encrypt block and file storage.

b. Choose one of the following 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.

   **IMPORTANT**

   Storage class encryption is a Technology Preview feature available only for RBD PVs. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

   For more information, see Technology Preview Features Support Scope .

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

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

   ii. Enter Vault Service Name, host Address of Vault server ('https://<hostname or ip>”), Port number and Token.
iii. Expand **Advanced Settings** to enter additional settings and certificate details based on your Vault configuration:

   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**.

d. Click **Next**.

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

10. Click **Create**.

11. Edit the configmap if Vault Key/Value (KV) secret engine API, version 2 is used for cluster-wide encryption with Key Management System (KMS).

   a. On the OpenShift Web Console, navigate to **Workloads → ConfigMaps**

   b. To view the KMS connection details, click **ocs-kms-connection-details**.

   c. Edit the configmap.

      i. Click **Action menu (…)** → **Edit ConfigMap**

      ii. Set the **VAULT_BACKEND** parameter to **v2**.

```yaml
kind: ConfigMap
apiVersion: v1
metadata:
  name: ocs-kms-connection-details
[...]
data:
  KMS_PROVIDER: vault
  KMS_SERVICE_NAME: vault
[...]
  VAULT_BACKEND: v2
[...]
```

iii. Click **Save**.

**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 (for arbiter mode, flexible scaling is disabled):

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.

```
spec:
  flexibleScaling: true
  ...
status:
  failureDomain: host
```

For arbiter mode of deployment:

1. Click `ocs-storagecluster` in Storage Cluster tab.

2. In the YAML tab, search for the `arbiter` key in `spec` section and ensure 'enable' is set to 'true'.

```
spec:
  arbiter:
    enable: true
  ...
nodeTopologies:
  arbiterLocation: arbiter #arbiter zone
storageDeviceSets:
  - config: {}
    count: 1
    ...
    replica: 4
status:
  conditions:
  ...
  failureDomain: zone
```

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 Scaling Storage guide.
CHAPTER 4. VERIFYING OPENSSHIFT CONTAINER STORAGE DEPLOYMENT FOR INTERNAL MODE

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

4.1. 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 4.1, "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 4.1. 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-*</td>
</tr>
<tr>
<td></td>
<td>(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>- nooba-db-* (1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>- noobaa-endpoint-* (1 pod on any storage node)</td>
</tr>
</tbody>
</table>
## Red Hat OpenShift Container Storage 4.7 Deploying OpenShift Container Storage on VMware vSphere

<table>
<thead>
<tr>
<th>Component</th>
<th>Corresponding pods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MON</strong></td>
<td><strong>rook-ceph-mon-</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>(3 pods distributed across storage nodes)</td>
</tr>
<tr>
<td></td>
<td>(For arbiter, 5 pods are distributed across 3 zones, 2 per data-center zones and 1 in arbiter zone)</td>
</tr>
<tr>
<td><strong>MGR</strong></td>
<td><strong>rook-ceph-mgr-</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>(1 pod on any storage node)</td>
</tr>
<tr>
<td><strong>MDS</strong></td>
<td><strong>rook-ceph-mds-ocs-storagecluster-cephfilesystem-</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>(2 pods distributed across storage nodes)</td>
</tr>
<tr>
<td></td>
<td>(For arbiter, 2 pods are distributed across 2 data-center zones)</td>
</tr>
<tr>
<td><strong>RGW</strong></td>
<td><strong>rook-ceph-rgw-ocs-storagecluster-cephobjectstore-</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>(1 pod on any storage node)</td>
</tr>
<tr>
<td></td>
<td>(For arbiter, 2 pods are distributed across 2 data-center zones)</td>
</tr>
<tr>
<td><strong>CSI</strong></td>
<td><strong>cephfs</strong></td>
</tr>
<tr>
<td></td>
<td>- <strong>csi-cephfsplugin-</strong>&lt;sup&gt;*&lt;/sup&gt; (1 pod on each worker node)</td>
</tr>
<tr>
<td></td>
<td>- <strong>csi-cephfsplugin-provisioner-</strong>&lt;sup&gt;*&lt;/sup&gt; (2 pods distributed across storage nodes)</td>
</tr>
<tr>
<td></td>
<td>- <strong>rbd</strong></td>
</tr>
<tr>
<td></td>
<td>- <strong>csi-rbdplugin-</strong>&lt;sup&gt;*&lt;/sup&gt; (1 pod on each worker node)</td>
</tr>
<tr>
<td></td>
<td>- <strong>csi-rbdplugin-provisioner-</strong>&lt;sup&gt;*&lt;/sup&gt; (2 pods distributed across storage nodes)</td>
</tr>
<tr>
<td><strong>rook-ceph-crashcollector</strong></td>
<td><strong>rook-ceph-crashcollector-</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>(1 pod on each storage node)</td>
</tr>
<tr>
<td></td>
<td>(For arbiter, 1 pod on each storage node and 1 pod in arbiter zone)</td>
</tr>
</tbody>
</table>
4.2. VERIFYING THE OPENShift CONTAINER STORAGE CLUSTER IS HEALTHY

- Click **Home → Overview** from the left pane of the OpenShift Web Console and click **Persistent Storage** tab.

- In the **Status card**, verify that **OCS Cluster** and **Data Resiliency** has a green tick mark as shown in the following image:

  ![Health status card in Persistent Storage Overview Dashboard](image)

  **Figure 4.1. Health status card in Persistent Storage Overview Dashboard**

<table>
<thead>
<tr>
<th>Status</th>
<th>OCS Cluster</th>
<th>Data Resiliency</th>
</tr>
</thead>
</table>

- In the **Details card**, verify that the cluster information is displayed as follows:

  **Service Name**  
  OpenShift Container Storage

  **Cluster Name**  
  ocs-storagecluster

  **Provider**  
  VSphere

  **Mode**  
  Internal

  **Version**  
  ocs-operator-4.7.0

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

4.3. VERIFYING THE MULTICLOUD OBJECT GATEWAY IS HEALTHY

- Click **Home → Overview** from the left pane of the OpenShift Web Console and click the **Object Service** tab.

- In the **Status card**, verify that both **Object Service** and **Data Resiliency** are in **Ready** state (green tick).
In the **Details card**, verify that the MCG information is displayed as follows:

**Service Name**
OpenShift Container Storage

**System Name**
Multicloud Object Gateway
RADOS Object Gateway

**Provider**
VSpere

**Version**
ocs-operator-4.7.0

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

### 4.4. VERIFYING THAT THE OPENSШIFT CONTAINER STORAGE SPECIFIC STORAGE CLASSES EXIST

To verify the storage classes exists in the cluster:

- Click **Storage → Storage Classes** from the left pane of the OpenShift Web Console.

- 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 5. UNINSTALLING OPENSHIFT CONTAINER STORAGE

5.1. UNINSTALLING OPENSHIFT 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 DataDirHostPath</td>
</tr>
<tr>
<td>cleanup-policy</td>
<td>retain</td>
<td>No</td>
<td>Rook does not clean up the physical drives and the DataDirHostPath</td>
</tr>
<tr>
<td>mode</td>
<td>graceful</td>
<td>Yes</td>
<td>Rook and NooBaa pauses 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:

```
$ oc annotate storagecluster ocs-storagecluster uninstall.ocs.openshift.io/cleanup-policy="retain" --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.
      
      ```bash
      $ 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.
      
      ```bash
      $ 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.
   
      See Section 5.2, “Removing monitoring stack from OpenShift Container Storage”

   b. Delete OpenShift Container Platform Registry PVCs using OpenShift Container Storage.
      See Section 5.3, “Removing OpenShift Container Platform registry from OpenShift Container Storage”

   c. Delete OpenShift Container Platform logging PVCs using OpenShift Container Storage.
      See Section 5.4, “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.
      
      ```bash
      #!/bin/bash
      ```
RBD_PROVISIONER="openshift-storage.rbd.csi.ceph.com"
CEPHFS_PROVISIONER="openshift-storage.cephfs.csi.ceph.com"
NOOBA PROVISIONER="openshift-storage.noobaa.io/obc"
RGW_PROVISIONER="openshift-storage.ceph.rook.io/bucket"

NOOBA_DB_PVC="noobaa-db"
NOOBA_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 "NOOBA PROVISIONER" -e "RGW_PROVISIONER" | awk '{print $1}')

# List PVCs in each of the StorageClasses
for SC in $OCS_STORAGECLASSES
do
echo "=====================================================================
== $SC StorageClass PVCs and OBCs==
=====================================================================
="
echo "$SC StorageClass PVCs and OBCs"
[...]
"=====================================================================
=="
[...]
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.
$ oc get pods -n openshift-storage | grep -i cleanup
<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster-cleanup-job-&lt;xx&gt;</td>
<td>0/1</td>
<td>Completed</td>
<td>0</td>
<td>8m35s</td>
</tr>
<tr>
<td>cluster-cleanup-job-&lt;yy&gt;</td>
<td>0/1</td>
<td>Completed</td>
<td>0</td>
<td>8m35s</td>
</tr>
<tr>
<td>cluster-cleanup-job-&lt;zz&gt;</td>
<td>0/1</td>
<td>Completed</td>
<td>0</td>
<td>8m35s</td>
</tr>
</tbody>
</table>

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-dmcrypt (253:1)

   c. Remove the mapped device.

   $ cryptsetup luksClose --debug --verbose ocs-deviceset-0-data-0-57snx-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.
     
     $ 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:
The project is deleted if the following command returns a NotFound error.

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.

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

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

12. Delete the Multicloud Object Gateway storageclass.


14. To ensure that OpenShift Container Storage is uninstalled completely, on the OpenShift Container Platform Web Console,
   a. Click Home → Overview to access the dashboard.
   b. Verify that the Persistent Storage and Object Service tabs no longer appear next to the Cluster tab.
5.1.1. 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. 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=local-block
   $ SC=localblock
   ```

c. Delete the local volume resource.
   
   ```
   $ oc delete localvolume -n 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/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 ...
   ```

5.2. REMOVING MONITORING STACK FROM OPENSOURCE 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-2ili28  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-k8dzn 1/1     Running   0          7d23h
pod/prometheus-adapter-5df5865596-n2gij9 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-55cfe858c9-c4zd9 1/1     Running   0          6d21h
pod/telemeter-client-78fc8fc97d-2rgfp 3/3     Running   0          8d

NAME                             STATUS   VOLUME                                      AGE
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-0 Bound pvc-0d519c4f-15a5-11ea-ba0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-1 Bound pvc-0d5a9825-15a5-11ea-ba0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-2 Bound pvc-0d6413dc-15a5-11ea-ba0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-0 Bound pvc-0b7c19b0-15a5-11ea-ba0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-1 Bound pvc-0b8aed3f-15a5-11ea-ba0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
2. Edit the monitoring `configmap`.

```bash
$ 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**

```yaml
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**

```yaml
```

CHAPTER 5. UNINSTALLING OPENSHIFT CONTAINER STORAGE
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
```

### 5.3. REMOVING OPENSHIFT CONTAINER PLATFORM REGISTRY FROM OPENSHIFT 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
After editing

```yaml
storage:
pvc:
  claim: registry-cephfs-rwx-pvc
```

+ 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`

5.4. REMOVING THE CLUSTER LOGGING OPERATOR FROM OPENSIGHT CONTAINER STORAGE

Use this section to clean up the cluster logging operator from OpenShift Container Storage.

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

**Prerequisites**

- The cluster logging instance should have been 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.

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