Red Hat OpenShift Container Storage 4.8

Deploying OpenShift Container Storage on VMware vSphere

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

Last Updated: 2022-05-18

Red Hat OpenShift Container Storage 4.8 Deploying OpenShift Container Storage on VMware vSphere

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

Legal Notice

Abstract

Read this document for instructions on installing Red Hat OpenShift Container Storage 4.8 on Red Hat OpenShift Container Platform VMware vSphere clusters.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

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

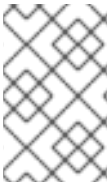
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

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

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

PREFACE

Red Hat OpenShift Container Storage 4.8 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 OPENSHIFT 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. Optional: If you want to enable cluster-wide encryption using an 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 section 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.
- 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 OPENSHIFT CONTAINER STORAGE USING LOCAL STORAGE DEVICES

Node requirements

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

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

For more information, see the Resource requirements section in the Planning guide.

Arbiter stretch cluster requirements [Technology Preview]

In this case, a single cluster is stretched across two zones with a third zone as the location for the arbiter. This is a technology preview feature that is currently intended for deployment in the OpenShift Container Platform on-premises.

For detailed requirements and instructions, see Configuring OpenShift Container Storage for Metro-DR stretch cluster.

NOTE

You cannot enable both flexible scaling and arbiter as they have conflicting scaling logic. With flexing scaling, you can add one node at a time to the OpenShift Container Storage cluster. Whereas, in an arbiter cluster, you need to add at least one node in each of the 2 data zones.

Minimum starting node requirements [Technology Preview]

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

For more information, see Resource requirements section in the Planning guide.

CHAPTER 2. DEPLOY 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:

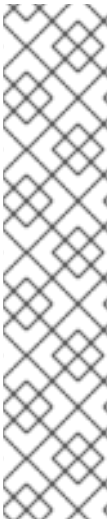
1. Install the Red Hat OpenShift Container Storage Operator .
2. Create the OpenShift Container Storage Cluster Service .

2.1. INSTALLING RED HAT OPENSHIFT CONTAINER STORAGE OPERATOR

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

Prerequisites

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



NOTE

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

```
$ oc annotate namespace openshift-storage openshift.io/node-selector=
```

- Taint a node as **infra** to ensure only Red Hat OpenShift Container Storage resources are scheduled on that node. This helps you save on subscription costs. For more information, see How to use dedicated worker nodes for Red Hat OpenShift Container Storage chapter in Managing and Allocating Storage Resources guide.

Procedure

1. Log in to OpenShift Web Console.
2. Click **Operators → OperatorHub**.
3. Search for **OpenShift Container Storage** from the list of operators and click on it.
4. Click **Install**.
5. Set the following options on the **Install Operator** page:
    a. Channel as **stable-4.8**.
    b. Installation Mode as **A specific namespace on the cluster**
    c. Installed Namespace as **Operator recommended namespace openshift-storage**. If Namespace **openshift-storage** does not exist, it will be created during the operator installation.
    d. **Approval Strategy** as **Automatic** or **Manual**.
    e. Click **Install**.
    If you select **Automatic** updates, the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your operator without any intervention.

    If you select **Manual** updates, the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to have the operator updated to the new version.

Verification step

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

2.2. CREATING MULTUS NETWORKS

OpenShift Container Platform uses the Multus CNI plug-in to allow chaining of CNI plug-ins. During cluster installation, you can configure your default pod network. The default network handles all ordinary network traffic for the cluster. You can define an additional network based on the available CNI plug-ins and attach one or more of these networks to your pods. To attach additional network interfaces to a pod, you must create configurations that define how the interfaces are attached. You can specify each interface by using a **NetworkAttachmentDefinition** custom resource (CR). A CNI configuration inside each of the **NetworkAttachmentDefinition** defines how that interface is created.

OpenShift Container Storage uses the CNI plug-in called **macvlan**. Creating a macvlan-based additional network allows pods on a host to communicate with other hosts and pods on those hosts by using a physical network interface. Each pod that is attached to a macvlan-based additional network is provided a unique MAC address.

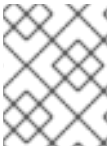
2.2.1. Creating network attachment definitions

To utilize Multus, an already working cluster with the correct networking configuration is required. For more information, see Recommended network configuration and requirements for a Multus configuration. The **NetworkAttachmentDefinition** (NAD) created now is later available to be selected during the storage cluster installation. This is the reason they must be created before the storage cluster.

As detailed in the Planning Guide, the Multus networks you create depend on the number of available network interfaces you have for OpenShift Container Storage traffic. It is possible to separate all of the storage traffic onto one of two interfaces (one interface used for default OpenShift SDN) or to further segregate storage traffic into client storage traffic (public) and storage replication traffic (private or cluster).

The following is an example **NetworkAttachmentDefinition** for all storage traffic, public and cluster, on the same interface. It requires one additional interface on all schedulable nodes (OpenShift default SDN on separate network interface).

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-public-cluster
  namespace: openshift-storage
spec:
  config: '{
   "cniVersion": "0.3.1",
   "type": "macvlan",
   "master": "ens2",
   "mode": "bridge",
   "ipam": {
        "type": "whereabouts",
        "range": "192.168.1.0/24"
   }
  }'
```



NOTE

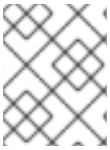
All network interface names must be the same on all nodes attached to Multus network (that is, **ens2** for **ocs-public-cluster**).

The following is an example **NetworkAttachmentDefinitions** for storage traffic on separate Multus networks, public for client storage traffic and cluster for replication traffic. It requires two additional interfaces on OpenShift nodes hosting OSD pods and one additional interface on all other schedulable nodes (OpenShift default SDN on separate network interface).

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-public
  namespace: openshift-storage
spec:
  config: '{
   "cniVersion": "0.3.1",
   "type": "macvlan",
   "master": "ens2",
   "mode": "bridge",
   "ipam": {
        "type": "whereabouts",
        "range": "192.168.1.0/24"
   }
  }'
```

Example **NetworkAttachmentDefinition**:

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-cluster
  namespace: openshift-storage
spec:
  config: '{
   "cniVersion": "0.3.1",
   "type": "macvlan",
   "master": "ens3",
   "mode": "bridge",
   "ipam": {
         "type": "whereabouts",
         "range": "192.168.2.0/24"
   }
  }'
```



NOTE

All network interface names must be the same on all nodes attached to Multus networks (that is, **ens2** for **ocs-public** and **ens3** for **ocs-cluster**).

2.3. CREATING AN OPENSHIFT 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.
- (Optional) If you want to use thick-provisioned storage for flexibility, you must create a storage class with **zeroedthick** or **eagerzeroedthick** disk format. For information, see VMware vSphere object definition.
- If you want to use the technology preview feature of multus support, before deployment you must create network attachment definitions (NADs) that later will be attached to the cluster. For more information, see Multi network plug-in (Multus) support and Creating network attachment definitions.

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. Select **Capacity and nodes**

   a. Select **Storage Class**.
   By default, it is set to **thin**. If you have created a storage class with **zeroedthick** or **eagerzeroedthick** disk format for thick-provisioned storage, then that storage class is listed in addition to the default, **thin** storage class.

   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) Set **Security and network** 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.

   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. (Optional) 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**.

7. Select **Default** (SDN) if you are using a single network or **Custom** (Multus) Network if you plan on using multiple network interfaces.

   a. Select a **Public Network Interface** from drop down.

   b. Select a **Cluster Network Interface** from drop down.

NOTE

If only using one additional network interface select the single NetworkAttachementDefinition (i.e. ocs-public-cluster) for the Public Network Interface and leave the Cluster Network Interface blank.

8. 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**.

```
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 .

4. To verify the multi network plug-in (Multus), see Verifying the Multus plug-in.

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

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

Prerequisite

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

Procedure

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

Verification step

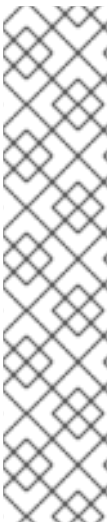
- Verify that the Local Storage Operator shows the **Status** as **Succeeded**.

3.2. INSTALLING RED HAT OPENSHIFT CONTAINER STORAGE OPERATOR

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

Prerequisites

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



NOTE

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

```
$ oc annotate namespace openshift-storage openshift.io/node-selector=
```

- Taint a node as **infra** to ensure only Red Hat OpenShift Container Storage resources are scheduled on that node. This helps you save on subscription costs. For more information, see How to use dedicated worker nodes for Red Hat OpenShift Container Storage chapter in Managing and Allocating Storage Resources guide.

Procedure

1. Log in to OpenShift Web Console.
2. Click **Operators → OperatorHub**.
3. Search for **OpenShift Container Storage** from the list of operators and click on it.
4. Click **Install**.
5. Set the following options on the **Install Operator** page:
   a. Channel as **stable-4.8**.
   b. Installation Mode as **A specific namespace on the cluster**.
   c. Installed Namespace as **Operator recommended namespace openshift-storage**. If Namespace **openshift-storage** does not exist, it will be created during the operator installation.
   d. **Approval Strategy** as **Automatic** or **Manual**.
   e. Click **Install**.

If you select **Automatic** updates, the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your operator without any intervention.

If you select **Manual** updates, the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to have the operator updated to the new version.

Verification step

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

3.3. CREATING MULTUS NETWORKS

OpenShift Container Platform uses the Multus CNI plug-in to allow chaining of CNI plug-ins. During cluster installation, you can configure your default pod network. The default network handles all ordinary network traffic for the cluster. You can define an additional network based on the available CNI plug-ins and attach one or more of these networks to your pods. To attach additional network interfaces to a pod, you must create configurations that define how the interfaces are attached. You can specify each interface by using a **NetworkAttachmentDefinition** custom resource (CR). A CNI configuration inside each of the **NetworkAttachmentDefinition** defines how that interface is created.

OpenShift Container Storage uses the CNI plug-in called **macvlan**. Creating a macvlan-based additional network allows pods on a host to communicate with other hosts and pods on those hosts by using a physical network interface. Each pod that is attached to a macvlan-based additional network is provided a unique MAC address.

3.3.1. Creating network attachment definitions

To utilize Multus, an already working cluster with the correct networking configuration is required. For more information, see Recommended network configuration and requirements for a Multus configuration. The **NetworkAttachmentDefinition** (NAD) created now is later available to be selected during the storage cluster installation. This is the reason they must be created before the storage cluster.

As detailed in the Planning Guide, the Multus networks you create depend on the number of available network interfaces you have for OpenShift Container Storage traffic. It is possible to separate all of the storage traffic onto one of two interfaces (one interface used for default OpenShift SDN) or to further segregate storage traffic into client storage traffic (public) and storage replication traffic (private or cluster).

The following is an example **NetworkAttachmentDefinition** for all storage traffic, public and cluster, on the same interface. It requires one additional interface on all schedulable nodes (OpenShift default SDN on separate network interface).

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-public-cluster
  namespace: openshift-storage
spec:
  config: '{
   "cniVersion": "0.3.1",
   "type": "macvlan",
   "master": "ens2",
```

```
 "mode": "bridge",
 "ipam": {
      "type": "whereabouts",
      "range": "192.168.1.0/24"
 }
}'
```

NOTE

All network interface names must be the same on all nodes attached to Multus network (that is, **ens2** for **ocs-public-cluster**).

The following is an example **NetworkAttachmentDefinitions** for storage traffic on separate Multus networks, public for client storage traffic and cluster for replication traffic. It requires two additional interfaces on OpenShift nodes hosting OSD pods and one additional interface on all other schedulable nodes (OpenShift default SDN on separate network interface).

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-public
  namespace: openshift-storage
spec:
  config: '{
   "cniVersion": "0.3.1",
   "type": "macvlan",
   "master": "ens2",
   "mode": "bridge",
   "ipam": {
        "type": "whereabouts",
        "range": "192.168.1.0/24"
   }
  }'
```

Example **NetworkAttachmentDefinition**:

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-cluster
  namespace: openshift-storage
spec:
  config: '{
   "cniVersion": "0.3.1",
   "type": "macvlan",
   "master": "ens3",
   "mode": "bridge",
   "ipam": {
        "type": "whereabouts",
        "range": "192.168.2.0/24"
   }
  }'
```

NOTE

All network interface names must be the same on all nodes attached to Multus networks (that is, **ens2** for **ocs-public** and **ens3** for **ocs-cluster**).

3.4. 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.
- If you want to use the technology preview feature of multus support, before deployment you must create network attachment definitions (NADs) that later will be attached to the cluster. For more information, see Multus support and Creating network attachment definitions .

Procedure

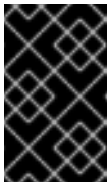
1. Log into the OpenShift Web Console.
2. Click **Operators → Installed Operators** to view all the installed operators.
   Ensure that the **Project** selected is **openshift-storage**.
3. Click **OpenShift Container Storage** > **Create Instance** link of Storage Cluster.
4. Choose **Select Mode** 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 using arbiter mode, do not select the **All nodes** option. Instead, use the **Select nodes** option to select the labeled nodes with attached storage device(s) from the two data-center zones.

b. Click **Next**.

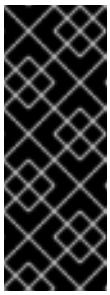
6. Create Storage class.
   You can create a dedicated storage class to consume storage by filtering a set of storage volumes.

   a. Enter the **Local 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 By** section. Choose one of the following:

   - **Disks on all nodes** to select all the nodes for which you discovered the devices.
   - **Disks on 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 to add tolerations for Local Storage Operator resources.

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

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

Volume Mode	Block is selected by default.
Device Type	Select one or more disk type from the drop down list.

Disk Size	Set a minimum size of 100GB for the device and maximum available size of the device that needs to be included.
Maximum Disk Limit	This indicates the maximum number of PVs that can be created on a node. If this field is left empty, then PVs are created for all the available disks on the matching nodes.

f. Click **Next**. A pop-up to confirm creation of the new storage class is displayed.

g. Click **Yes** to continue.

7. Set **Capacity and nodes**

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

   b. **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.

   c. Click **Next**.

8. (Optional) Set **Security and network** 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.

   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. (Optional) Enter **TLS Server Name** and **Vault Enterprise Namespace**.

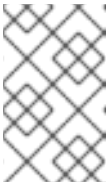
         C. Provide **CA Certificate**, **Client Certificate** and **Client Private Key** by uploading the respective PEM encoded certificate file.

         D. Click **Save**.

9. Select **Default** (SDN) if you are using a single network or **Custom** (Multus) Network if you plan on using multiple network interfaces.

    a. Select a **Public Network Interface** from drop down.

    b. Select a **Cluster Network Interface** from drop down.



NOTE

If only using one additional network interface select the single NetworkAttachementDefinition (i.e. ocs-public-cluster) for the Public Network Interface and leave the Cluster Network Interface blank.

10. Click **Next**.

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

12. Click **Create**.

13. 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**.

```
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
     ```

- 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 OPENSHIFT CONTAINER STORAGE DEPLOYMENT

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

4.1. VERIFYING THE STATE OF THE PODS

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

Procedure

1. Log in to OpenShift Web Console.
2. Click **Workloads → Pods** from the left pane of the OpenShift Web Console.
3. Select **openshift-storage** from the **Project** drop down list.
   For more information on the expected number of pods for each component and how it varies depending on the number of nodes, see Table 4.1, "Pods corresponding to OpenShift Container storage cluster".
4. Click on the **Running** and **Completed** tabs to verify that the pods are running and in a completed state:

Table 4.1. Pods corresponding to OpenShift Container storage cluster

Component	Corresponding pods
OpenShift Container Storage Operator	• **ocs-operator-*** (1 pod on any worker node) • **ocs-metrics-exporter-***
Rook-ceph Operator	**rook-ceph-operator-*** (1 pod on any worker node)
Multicloud Object Gateway	• **noobaa-operator-*** (1 pod on any worker node) • **noobaa-core-*** (1 pod on any storage node) • **noobaa-db-pg-*** (1 pod on any storage node) • **noobaa-endpoint-*** (1 pod on any storage node)
MON	**rook-ceph-mon-*** (3 pods distributed across storage nodes)

Component	Corresponding pods
MGR	**rook-ceph-mgr-*** (1 pod on any storage node)
MDS	**rook-ceph-mds-ocs-storagecluster-cephfilesystem-*** (2 pods distributed across storage nodes)
RGW	**rook-ceph-rgw-ocs-storagecluster-cephobjectstore-*** (1 pod on any storage node)

CSI	• **cephfs**
	  ◦ **csi-cephfsplugin-*** (1 pod on each worker node)
	  ◦ **csi-cephfsplugin-provisioner-*** (2 pods distributed across worker nodes)
	• **rbd**
	  ◦ **csi-rbdplugin-*** (1 pod on each worker node)
	  ◦ **csi-rbdplugin-provisioner-*** (2 pods distributed across worker nodes)

rook-ceph-crashcollector	**rook-ceph-crashcollector-*** (1 pod on each storage node)
OSD	• **rook-ceph-osd-*** (1 pod for each device) • **rook-ceph-osd-prepare-ocs-deviceset-*** (1 pod for each device)

4.2. VERIFYING THE OPENSHIFT 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 .

4.3. 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 .

4.4. 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**

4.5. VERIFYING THE MULTUS NETWORKING

To determine if Multus is working in the cluster, verify the Multus networking.

Procedure

1. Based on your network configuration choices, the OpenShift Container Storage operator does one of the following:
   - If only a single **NetworkAttachmentDefinition** (for example, **ocs-public-cluster**) is selected for the Public Network Interface, the traffic between the application pods and the OpenShift Container Storage cluster happens on this network. Additionally, the cluster is also self configured to use this network for the replication and rebalancing traffic between OSDs.

- If both **NetworkAttachmentDefinitions** (for example, **ocs-public** and **ocs-cluster**) are selected for the Public Network Interface and the Cluster Network Interface respectively during the storage cluster installation, then client storage traffic is on the public network and cluster network is for the replication and rebalancing traffic between OSDs.

2. To verify the network configuration is correct, follow the steps:

   a. In the OpenShift console, click **Installed Operators → Storage Cluster → ocs-storagecluster**.

   b. In the YAML tab, search for **network** in the **spec** section and ensure the configuration is correct for your network interface choices. This example is for separating the client storage traffic from the storage replication traffic.
   Sample output:

```
[..]
spec:
   [..]
   network:
   provider: multus
   selectors:
     cluster: openshift-storage/ocs-cluster
     public: openshift-storage/ocs-public
   [..]
```

3. To verify the network configuration is correct using the command line interface, run the following commands:

```
$ oc get storagecluster ocs-storagecluster \
-n openshift-storage \
-o=jsonpath='{.spec.network}{"\n"}'
```

   Sample output:

```
{"provider":"multus","selectors":{"cluster":"openshift-storage/ocs-cluster","public":"openshift-storage/ocs-public"}}
```

4. Confirm the OSD pods are using correct network:

   a. In the **openshift-storage** namespace use one of the OSD pods to verify the pod has connectivity to the correct networks. This example is for separating the client storage traffic from the storage replication traffic.

NOTE

Only the OSD pods connects to both Multus public and cluster networks if both are created. All other OCS pods connect to the Multus public network.

```
$ oc get -n openshift-storage $(oc get pods -n openshift-storage -o name -l app=rook-ceph-osd | grep 'osd-0') -
o=jsonpath='{.metadata.annotations.k8s\.v1\.cni\.cncf\.io/network-status}{"\n"}'
```

   Sample output:

```
[{
    "name": "openshift-sdn",
    "interface": "eth0",
    "ips": [
        "10.129.2.30"
    ],
    "default": true,
    "dns": {}
},{
    "name": "openshift-storage/ocs-cluster",
    "interface": "net1",
    "ips": [
        "192.168.2.1"
    ],
    "mac": "e2:04:c6:81:52:f1",
    "dns": {}
},{
    "name": "openshift-storage/ocs-public",
    "interface": "net2",
    "ips": [
        "192.168.1.1"
    ],
    "mac": "ee:a0:b6:a4:07:94",
    "dns": {}
}]
```

5. To confirm the OSD pods are using correct network using the command line interface, run the following command (requires the jq utility):

```
$ oc get -n openshift-storage $(oc get pods -n openshift-storage -o name -l app=rook-ceph-osd | grep 'osd-0') -o=jsonpath='{.metadata.annotations.k8s\.v1\.cni\.cncf\.io/network-status}{"\n"}' | jq -r '.[].name'
```

Sample output:

```
openshift-sdn
openshift-storage/ocs-cluster
openshift-storage/ocs-public
```

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 5.1. uninstall.ocs.openshift.io uninstall annotations descriptions

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

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

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

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

Prerequisites

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

Procedure

1. Delete the volume snapshots that are using OpenShift Container Storage.
   a. List the volume snapshots from all the namespaces.

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

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

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

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

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

   a. Delete OpenShift Container Platform monitoring stack PVCs using OpenShift Container Storage.
      For more information, see Section 5.2, “Removing monitoring stack from OpenShift Container Storage”.

   b. Delete OpenShift Container Platform Registry PVCs using OpenShift Container Storage.
      For more information, see Section 5.3, “Removing OpenShift Container Platform registry from OpenShift Container Storage”.

   c. Delete OpenShift Container Platform logging PVCs using OpenShift Container Storage.
      For more information, see Section 5.4, “Removing the cluster logging operator from OpenShift Container Storage”.

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

      - Following script is 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**.

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

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

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

6. If encryption was enabled at the time of install, remove **dm-crypt** managed **device-mapper** mapping from OSD devices on all the OpenShift Container Storage nodes.

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

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

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

```
$ dmsetup ls
ocs-deviceset-0-data-0-57snx-block-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 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>
    ```

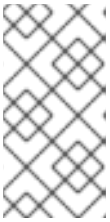
12. Delete the Multicloud Object Gateway storageclass.

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

13. Remove **CustomResourceDefinitions**.

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

14. Optional: To ensure that the vault keys are deleted permanently you need to manually delete the metadata associated with the vault key.



NOTE

Execute this step only if Vault Key/Value (KV) secret engine API, version 2 is used for cluster-wide encryption with Key Management System (KMS) since the vault keys are marked as deleted and not permanently deleted during the uninstallation of OpenShift Container Storage. You can always restore it later if required.

a. List the keys in the vault.

```
$ vault kv list <backend_path>
```

<backend_path>

Is the path in the vault where the encryption keys are stored.
For example:

```
$ vault kv list kv-v2
```

Example output:

```
Keys
-----
NOOBAA_ROOT_SECRET_PATH/
rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
rook-ceph-osd-encryption-key-ocs-deviceset-thin-1-data-0sq227
rook-ceph-osd-encryption-key-ocs-deviceset-thin-2-data-0xzszb
```

b. List the metadata associated with the vault key.

```
$ vault kv get kv-v2/<key>
```

For the Multicloud Object Gateway (MCG) key:

```
$ vault kv get kv-v2/NOOBAA_ROOT_SECRET_PATH/<key>
```

<key>

Is the encryption key.
For Example:

```
$ vault kv get kv-v2/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
```

Example output:

```
====== Metadata ======
Key              Value
---              -----
created_time     2021-06-23T10:06:30.650103555Z
deletion_time    2021-06-23T11:46:35.045328495Z
destroyed        false
version          1
```

c. Delete the metadata.

```
$ vault kv metadata delete kv-v2/<key>
```

For the MCG key:

```
$ vault kv metadata delete kv-v2/NOOBAA_ROOT_SECRET_PATH/<key>
```

<key>

Is the encryption key.
For Example:

```
$ vault kv metadata delete kv-v2/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
```

Example output:

```
Success! Data deleted (if it existed) at: kv-v2/metadata/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
```

d. Repeat these steps to delete the metadata associated with all the vault keys.

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

a. Click **Storage**.

b. Verify that **Overview** no longer appears under Storage.

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 using only **localvolume** resources, refer 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**.

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

3. Delete the **LocalVolumeSet**.

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

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

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

5. Delete the **StorageClassName**.

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

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

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

7. Delete **LocalVolumeDiscovery**.

```
$ 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 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 in the cluster.

   For each of the local volumes, perform 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 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.

```
$ oc get pod,pvc -n openshift-monitoring
NAME                          READY   STATUS    RESTARTS   AGE
pod/alertmanager-main-0       3/3     Running   0          8d
pod/alertmanager-main-1       3/3     Running   0          8d
pod/alertmanager-main-2       3/3     Running   0          8d
pod/cluster-monitoring-
operator-84457656d-pkrxm      1/1     Running   0          8d
pod/grafana-79ccf6689f-2ll28  2/2     Running   0          8d
pod/kube-state-metrics-
7d86fb966-rvd9w               3/3     Running   0          8d
pod/node-exporter-25894       2/2     Running   0          8d
pod/node-exporter-4dsd7       2/2     Running   0          8d
pod/node-exporter-6p4zc       2/2     Running   0          8d
pod/node-exporter-jbjvg       2/2     Running   0          8d
pod/node-exporter-jj4t5       2/2     Running   0          6d18h
pod/node-exporter-k856s       2/2     Running   0          6d18h
pod/node-exporter-rf8gn       2/2     Running   0          8d
pod/node-exporter-rmb5m       2/2     Running   0          6d18h
```

```
pod/node-exporter-zj7kx          2/2     Running   0          8d
pod/openshift-state-metrics-
59dbd4f654-4clng                 3/3     Running   0          8d
pod/prometheus-adapter-
5df5865596-k8dzn                 1/1     Running   0          7d23h
pod/prometheus-adapter-
5df5865596-n2gj9                 1/1     Running   0          7d23h
pod/prometheus-k8s-0             6/6     Running   1          8d
pod/prometheus-k8s-1             6/6     Running   1          8d
pod/prometheus-operator-
55cfb858c9-c4zd9                 1/1     Running   0          6d21h
pod/telemeter-client-
78fc8fc97d-2rgfp                 3/3     Running   0          8d

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

2. Edit the monitoring **configmap**.

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

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

   Before editing

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

After editing

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

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

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

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

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

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

After editing

```
.
.
.
storage:
  emptyDir: {}
.
.
.
```

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

2. Delete the PVC.

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

5.4. 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.

```
$ 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
```