



Red Hat OpenShift Data Foundation 4.10

Deploying OpenShift Data Foundation on VMware vSphere

Instructions on deploying OpenShift Data Foundation using VMware vSphere
infrastructure

Red Hat OpenShift Data Foundation 4.10 Deploying OpenShift Data Foundation on VMware vSphere

Instructions on deploying OpenShift Data Foundation using VMware vSphere infrastructure

Legal Notice

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

Abstract

Read this document for instructions about how to install Red Hat OpenShift Data Foundation using Red Hat OpenShift Container Platform on 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 OPENSIFT DATA FOUNDATION	6
1.1. REQUIREMENTS FOR INSTALLING OPENSIFT DATA FOUNDATION USING LOCAL STORAGE DEVICES	6
CHAPTER 2. DEPLOY USING DYNAMIC STORAGE DEVICES	8
2.1. INSTALLING RED HAT OPENSIFT DATA FOUNDATION OPERATOR	8
2.2. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE TOKEN AUTHENTICATION METHOD	9
2.3. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE KUBERNETES AUTHENTICATION METHOD	10
2.4. CREATING MULTUS NETWORKS [TECHNOLOGY PREVIEW]	13
2.4.1. Creating network attachment definitions	13
2.5. CREATING AN OPENSIFT DATA FOUNDATION CLUSTER	15
CHAPTER 3. DEPLOY USING LOCAL STORAGE DEVICES	19
3.1. INSTALLING LOCAL STORAGE OPERATOR	19
3.2. INSTALLING RED HAT OPENSIFT DATA FOUNDATION OPERATOR	19
3.3. CREATING MULTUS NETWORKS [TECHNOLOGY PREVIEW]	21
3.3.1. Creating network attachment definitions	21
3.4. CREATING OPENSIFT DATA FOUNDATION CLUSTER ON VMWARE VSPHERE	23
CHAPTER 4. VERIFYING OPENSIFT DATA FOUNDATION DEPLOYMENT	28
4.1. VERIFYING THE STATE OF THE PODS	28
4.2. VERIFYING THE OPENSIFT DATA FOUNDATION CLUSTER IS HEALTHY	30
4.3. VERIFYING THE MULTICLOUD OBJECT GATEWAY IS HEALTHY	30
4.4. VERIFYING THAT THE OPENSIFT DATA FOUNDATION SPECIFIC STORAGE CLASSES EXIST	30
4.5. VERIFYING THE MULTUS NETWORKING	31
CHAPTER 5. DEPLOY STANDALONE MULTICLOUD OBJECT GATEWAY	34
5.1. DEPLOY STANDALONE MULTICLOUD OBJECT GATEWAY USING DYNAMIC STORAGE DEVICES	34
5.1.1. Installing Red Hat OpenShift Data Foundation Operator	34
5.1.2. Creating a standalone Multicloud Object Gateway	35
5.2. DEPLOY STANDALONE MULTICLOUD OBJECT GATEWAY USING LOCAL STORAGE DEVICES	37
5.2.1. Installing Local Storage Operator	37
5.2.2. Installing Red Hat OpenShift Data Foundation Operator	38
5.2.3. Creating a standalone Multicloud Object Gateway	39
CHAPTER 6. UNINSTALLING OPENSIFT DATA FOUNDATION	43
6.1. UNINSTALLING OPENSIFT DATA FOUNDATION IN INTERNAL MODE	43

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, 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 Data Foundation 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 Data Foundation clusters are supported on VMware vSphere. See [Planning your deployment](#) and [Preparing to deploy OpenShift Data Foundation](#) for more information about deployment requirements.

To deploy OpenShift Data Foundation, start with the requirements in the [Preparing to deploy OpenShift Data Foundation](#) 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](#)
 - [Deploy standalone Multicloud Object Gateway component](#)
- [External mode](#)

CHAPTER 1. PREPARING TO DEPLOY OPENSIFT DATA FOUNDATION

Deploying OpenShift Data Foundation 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 Data Foundation using dynamic or local storage, ensure that your resource requirements are met. See [Planning your deployment](#).

1. Optional: If you want to enable cluster-wide encryption using an external Key Management System (KMS) then follow the steps:
 - Ensure that you have a valid Red Hat OpenShift Data Foundation Advanced subscription. To know how subscriptions for OpenShift Data Foundation work, see [knowledgebase article on OpenShift Data Foundation subscriptions](#).
 - When the Token authentication method is selected for encryption then refer to [Enabling cluster-wide encryption with the Token authentication using KMS](#).
 - When the Kubernetes authentication method is selected for encryption then refer to [Enabling cluster-wide encryption with the Kubernetes authentication using KMS](#).
 - Ensure that you are using signed certificates on your Vault servers.
2. Minimum starting node requirements
An OpenShift Data Foundation cluster will be deployed with minimum configuration when the standard deployment resource requirement is not met. See [Resource requirements](#) section in Planning guide.
3. Regional-DR requirements [Developer preview]
Disaster Recovery features supported by Red Hat OpenShift Data Foundation require all of the following prerequisites in order to successfully implement a Disaster Recovery solution:
 - A valid Red Hat OpenShift Data Foundation Advanced subscription
 - A valid Red Hat Advanced Cluster Management for Kubernetes subscription
To know how subscriptions for OpenShift Data Foundation work, see [knowledgebase article on OpenShift Data Foundation subscriptions](#).

For detailed requirements, see [Regional-DR requirements](#) and [RHACM requirements](#).
4. For deploying using local storage devices, see [requirements for installing OpenShift Data Foundation using local storage devices](#). These are not applicable for deployment using dynamic storage devices.

1.1. REQUIREMENTS FOR INSTALLING OPENSIFT DATA FOUNDATION 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 Data Foundation.



NOTE

Make sure that the devices have a unique **by-id** device name for each available raw block device.

- 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.

Regional-DR requirements [Developer preview]

Disaster Recovery features supported by Red Hat OpenShift Data Foundation require all of the following prerequisites in order to successfully implement a Disaster Recovery solution:

- A valid Red Hat OpenShift Data Foundation Advanced subscription
- A valid Red Hat Advanced Cluster Management for Kubernetes subscription

To know how subscriptions for OpenShift Data Foundation work, see [knowledgebase article on OpenShift Data Foundation subscriptions](#).

For detailed requirements, see [Regional-DR requirements](#) and [RHACM requirements](#).

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 and in the same data center. This solution is not recommended for deployments stretching over multiple data centers. Instead, consider Metro-DR as a first option for no data loss DR solution deployed over multiple data centers with low latency networks.

For detailed requirements and instructions, see [Configuring OpenShift Data Foundation for stretch cluster](#).



NOTE

Flexible scaling and Arbiter both cannot be enabled at the same time as they have conflicting scaling logic. With Flexible scaling, you can add one node at a time to your OpenShift Data Foundation cluster. Whereas in an Arbiter cluster, you need to add at least one node in each of the two data zones.

Minimum starting node requirements

An OpenShift Data Foundation 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 Data Foundation 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 Data Foundation 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 Data Foundation](#) chapter before proceeding with the below steps for deploying using dynamic storage devices:

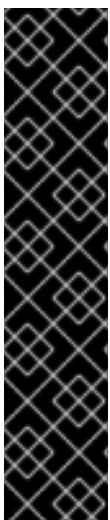
1. [Install the Red Hat OpenShift Data Foundation Operator](#) .
2. [Create the OpenShift Data Foundation Cluster](#) .

2.1. INSTALLING RED HAT OPENSIFT DATA FOUNDATION OPERATOR

You can install Red Hat OpenShift Data Foundation 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 must have at least three worker nodes in the Red Hat OpenShift Container Platform cluster.
- For additional resource requirements, see the [Planning your deployment](#) guide.



IMPORTANT

- When you need to override the cluster-wide default node selector for OpenShift Data Foundation, you can use the following command in the 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 Data Foundation 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 Data Foundation](#) chapter in the *Managing and Allocating Storage Resources* guide.

Procedure

1. Log in to the OpenShift Web Console.
2. Click **Operators** → **OperatorHub**.
3. Scroll or type **OpenShift Data Foundation** into the **Filter by keyword** box to find the **OpenShift Data Foundation** Operator.
4. Click **Install**.
5. Set the following options on the **Install Operator** page:
 - a. Update Channel as **stable-4.10**.
 - 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 is created during the operator installation.
 - d. Select Approval Strategy as **Automatic** or **Manual**.

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

If you select **Manual** updates, then the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to update the Operator to a newer version.
 - e. Ensure that the **Enable** option is selected for the **Console plugin**.
 - f. Click **Install**.

Verification steps

- After the operator is successfully installed, a pop-up with a message, **Web console update is available** appears on the user interface. Click **Refresh web console** from this pop-up for the console changes to reflect.
- In the Web Console:
 - Navigate to Installed Operators and verify that the **OpenShift Data Foundation** Operator shows a green tick indicating successful installation.
 - Navigate to **Storage** and verify if **Data Foundation** dashboard is available.

2.2. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE TOKEN AUTHENTICATION METHOD

To enable the key value backend path and policy in Vault for the Token authentication, follow the procedure:

Prerequisites

- Administrator access to Vault.
- A valid Red Hat OpenShift Data Foundation Advanced subscription. For more information, see the [knowledgebase article on OpenShift Data Foundation subscriptions](#).

- Carefully, select 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=odf kv
```

For Vault KV secret engine API, version 2:

```
$ vault secrets enable -path=odf 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 "odf/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}
path "sys/mounts" {
  capabilities = ["read"]
}' | vault policy write odf -
```

3. Create a token matching the above policy.

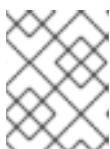
```
$ vault token create -policy=odf -format json
```

2.3. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE KUBERNETES AUTHENTICATION METHOD

You can enable the Kubernetes authentication method for cluster-wide encryption using the Key Management System (KMS).

Prerequisites

- Administrator access to Vault.
- A valid Red Hat OpenShift Data Foundation Advanced subscription. For more information, see the [knowledgebase article on OpenShift Data Foundation subscriptions](#).
- The OpenShift Data Foundation operator must be installed from the Operator Hub.
- Select a unique path name as the backend **path** that follows the naming convention since it cannot be changed later.



NOTE

Use of Vault Namespaces is not supported with the Kubernetes authentication method in OpenShift Data Foundation 4.10

Procedure

1. Create a service account:

```
$ oc -n openshift-storage create serviceaccount <serviceaccount_name>
```

where, **<serviceaccount_name>** specifies the name of the service account.

For example:

```
$ oc -n openshift-storage create serviceaccount odf-vault-auth
```

2. Create **clusterrolebindings** and **clusterroles**:

```
$ oc -n openshift-storage create clusterrolebinding vault-tokenreview-binding --
clusterrole=system:auth-delegator --serviceaccount=openshift-
storage: <serviceaccount_name>
```

For example:

```
$ oc -n openshift-storage create clusterrolebinding vault-tokenreview-binding --
clusterrole=system:auth-delegator --serviceaccount=openshift-storage:odf-vault-auth
```

3. Depending on the OpenShift Container Platform version, perform one of the following:

- For OpenShift Container Platform 4.10:
 - Identify the secret name associated with the serviceaccount (SA) created above.

```
$ VAULT_SA_SECRET_NAME=$(oc -n openshift-storage get sa <SA_NAME> -o
jsonpath="{.secrets[*]['name']}") | grep -o "[^[:space:]]*-token-[^[:space:]]*"

```

For example:

```
$ VAULT_SA_SECRET_NAME=$(oc -n openshift-storage get sa odf-vault-auth -o
jsonpath="{.secrets[*]['name']}") | grep -o "[^[:space:]]*-token-[^[:space:]]*"

```

- For OpenShift Container Platform 4.11:
 - Create a secret for the **serviceaccount** token and CA certificate.

```
$ cat <<EOF | oc create -f -
apiVersion: v1
kind: Secret
metadata:
  name: odf-vault-auth-token
  namespace: openshift-storage
annotations:
  kubernetes.io/service-account.name: <serviceaccount_name>
type: kubernetes.io/service-account-token
data: {}
EOF

```

where, **<serviceaccount_name>** is the service account created in the earlier step.

```
$ VAULT_SA_SECRET_NAME=odf-vault-auth-token
```

4. Get the token and the CA certificate from the secret.

```
$ SA_JWT_TOKEN=$(oc -n openshift-storage get secret "$VAULT_SA_SECRET_NAME" -o
jsonpath="{.data.token}" | base64 --decode; echo)
$ SA_CA_CERT=$(oc -n openshift-storage get secret "$VAULT_SA_SECRET_NAME" -o
jsonpath="{.data['ca.crt']}" | base64 --decode; echo)
```

5. Retrieve the OCP cluster endpoint.

```
$ OCP_HOST=$(oc config view --minify --flatten -o jsonpath="{.clusters[0].cluster.server}")
```

6. Fetch the service account issuer.

```
$ oc proxy &
$ proxy_pid=$!
$ issuer="$( curl --silent http://127.0.0.1:8001/.well-known/openid-configuration | jq -r
.issuer)"
$ kill $proxy_pid
```

7. Use the information collected in the steps above to setup the Kubernetes authentication method in Vault as shown below.

```
$ vault auth enable kubernetes
$ vault write auth/kubernetes/config \
  token_reviewer_jwt="$SA_JWT_TOKEN" \
  kubernetes_host="$OCP_HOST" \
  kubernetes_ca_cert="$SA_CA_CERT" \
  issuer="$issuer"
```

IMPORTANT

To configure Kubernetes authentication method in Vault when the issuer is empty.

```
$ vault write auth/kubernetes/config \
  token_reviewer_jwt="$SA_JWT_TOKEN" \
  kubernetes_host="$OCP_HOST" \
  kubernetes_ca_cert="$SA_CA_CERT"
```

8. Enable the Key/Value (KV) backend path in Vault.
For Vault KV secret engine API, version 1.

```
$ vault secrets enable -path=odf kv
```

For Vault KV secret engine API, version 2.

```
$ vault secrets enable -path=odf kv-v2
```

9. Create a policy to restrict users to perform a write or delete operation on the secret:


```

echo '
path "odf/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}
path "sys/mounts" {
  capabilities = ["read"]
}' | vault policy write odf -

```

10. Generate the roles:

```

$ vault write auth/kubernetes/role/odf-rook-ceph-op \
  bound_service_account_names=rook-ceph-system,rook-ceph-osd,noobaa \
  bound_service_account_namespaces=openshift-storage \
  policies=odf \
  ttl=1440h

```

The role **odf-rook-ceph-op** is later used while you configure the KMS connection details during the creation of the storage system.

```

$ vault write auth/kubernetes/role/odf-rook-ceph-osd \
  bound_service_account_names=rook-ceph-osd \
  bound_service_account_namespaces=openshift-storage \
  policies=odf \
  ttl=1440h

```

2.4. CREATING MULTUS NETWORKS [TECHNOLOGY PREVIEW]

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 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 Data Foundation 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.



IMPORTANT

Multus support is a Technology Preview feature that is only supported and has been tested on bare metal and VMWare deployments. 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](#).

2.4.1. Creating network attachment definitions

To utilize Multus, an already working cluster with the correct networking configuration is required, see [Recommended network configuration and requirements for a Multus configuration](#) . The newly created NetworkAttachmentDefinition (NAD) can 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 Data Foundation 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 the nodes attached to the Multus network (that is, **ens2** for **ocs-public-cluster**).

The following is an example **NetworkAttachmentDefinition** 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 the nodes attached to the Multus networks (that is, **ens2** for **ocs-public**, and **ens3** for **ocs-cluster**).

2.5. CREATING AN OPENSIFT DATA FOUNDATION CLUSTER

Create an OpenShift Data Foundation cluster after you install the OpenShift Data Foundation operator.

Prerequisites

- The OpenShift Data Foundation operator must be installed from the Operator Hub. For more information, see [Installing OpenShift Data Foundation Operator](#).
- 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 [Installing on 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 are later attached to the cluster. For more information, see [Multi network plug-in \(Multus\) support](#) and [Creating network attachment definitions](#).

Procedure

1. In the OpenShift Web Console, click **Operators → Installed Operators** to view all the installed operators.
Ensure that the **Project** selected is **openshift-storage**.
2. Click on the **OpenShift Data Foundation** operator, and then click **Create StorageSystem**.
3. In the **Backing storage** page, select the following:
 - a. Select **Full Deployment** for the **Deployment type** option.
 - b. Select the **Use an existing StorageClass** option.
 - c. Select the **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.
 - d. Click **Next**.
4. In the **Capacity and nodes** page, provide the necessary information:
 - a. Select a value for **Requested Capacity** from the dropdown list. It is set to **2 TiB** by default.

**NOTE**

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

- b. In the **Select Nodes** section, select at least three available nodes.
 - c. Optional: Select the **Taint nodes** checkbox to dedicate the selected nodes for OpenShift Data Foundation.
Spread the worker nodes across three different physical nodes, racks, or failure domains for high availability.

Use vCenter anti-affinity to align OpenShift Data Foundation 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 Data Foundation cluster requirement of the aggregated 30 CPUs and 72 GiB of RAM, a minimal cluster is deployed. For minimum starting node requirements, see [Resource requirements](#) section in Planning guide.

Select the **Taint nodes** checkbox to make selected nodes dedicated for OpenShift Data Foundation.
 - d. Click **Next**.
5. Optional: In the **Security and network** page, configure the following based on your requirements:
 - a. To enable encryption, select **Enable data encryption for block and file storage**
 - b. Select either one or both the encryption levels:
 - **Cluster-wide encryption**
Encrypts the entire cluster (block and file).

- **StorageClass encryption**
Creates 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. Select an **Authentication Method**.

Using Token authentication method

- Enter a unique **Connection Name**, host **Address** of the Vault server ('https://<hostname or ip>'), **Port** number and **Token**.
- Expand **Advanced Settings** to enter additional settings and certificate details based on your **Vault** configuration:
 - Enter the Key Value secret path in **Backend Path** that is dedicated and unique to OpenShift Data Foundation.
 - Optional: Enter **TLS Server Name** and **Vault Enterprise Namespace**
 - Upload the respective PEM encoded certificate file to provide the **CA Certificate**, **Client Certificate** and **Client Private Key**.
 - Click **Save**.

Using Kubernetes authentication method

- Enter a unique Vault **Connection Name**, host **Address** of the Vault server ('https://<hostname or ip>'), **Port** number and **Role** name.
- Expand **Advanced Settings** to enter additional settings and certificate details based on your **Vault** configuration:
 - Enter the Key Value secret path in **Backend Path** that is dedicated and unique to OpenShift Data Foundation.
 - Optional: Enter **TLS Server Name** and **Authentication Path** if applicable.
 - Upload the respective PEM encoded certificate file to provide the **CA Certificate**, **Client Certificate** and **Client Private Key**.
 - Click **Save**.

- d. Select **Default (SDN)** if you are using a single network or **Custom (Multus)** Network if you plan to use multiple network interfaces.
- i. Select a **Public Network Interface** from the dropdown.
 - ii. Select a **Cluster Network Interface** from the dropdown.

**NOTE**

If you are using only one additional network interface, select the single **NetworkAttachmentDefinition** that is, **ocs-public-cluster** for the **Public Network Interface** and leave the **Cluster Network Interface** blank.

- e. Click **Next**.
6. In the **Review and create** page, review the configuration details. To modify any configuration settings, click **Back**.
7. Click **Create StorageSystem**.

Verification steps

- To verify the final Status of the installed storage cluster:
 - a. In the OpenShift Web Console, navigate to **Installed Operators** → **OpenShift Data Foundation** → **Storage System** → **ocs-storagecluster-storagesystem** → **Resources**.
 - b. Verify that **Status** of **StorageCluster** is **Ready** and has a green tick mark next to it.
 1. To verify that all components for OpenShift Data Foundation are successfully installed, see [Verifying your OpenShift Data Foundation deployment](#).
 2. To verify the multi networking (Multus), see [Verifying the Multus networking](#).

Additional resources

To enable Overprovision Control alerts, refer to [Alerts](#) in Monitoring guide.

CHAPTER 3. DEPLOY USING LOCAL STORAGE DEVICES

Deploying OpenShift Data Foundation 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 Data Foundation on VMware where OpenShift Container Platform is already installed.

Also, ensure that you have addressed the requirements in [Preparing to deploy OpenShift Data Foundation](#) chapter before proceeding with the next steps.

1. [Installing Local Storage Operator](#)
2. [Install the Red Hat OpenShift Data Foundation Operator](#) .
3. [Create the OpenShift Data Foundation Cluster](#) .

3.1. INSTALLING LOCAL STORAGE OPERATOR

Install the Local Storage Operator from the Operator Hub before creating Red Hat OpenShift Data Foundation clusters on local storage devices.

Procedure

1. Log in to the OpenShift Web Console.
2. Click **Operators** → **OperatorHub**.
3. Type **local storage** in the **Filter by keyword** box to find the **Local Storage Operator** from the list of operators and click on it.
4. Set the following options on the **Install Operator** page:
 - a. Update channel as either **4.10** or **stable**.
 - b. Installation mode as **A specific namespace on the cluster**
 - c. Installed Namespace as **Operator recommended namespace openshift-local-storage**.
 - d. Update approval as **Automatic**.
5. Click **Install**.

Verification steps

- Verify that the Local Storage Operator shows a green tick indicating successful installation.

3.2. INSTALLING RED HAT OPENSIFT DATA FOUNDATION OPERATOR

You can install Red Hat OpenShift Data Foundation 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 must have at least three worker nodes in the Red Hat OpenShift Container Platform cluster.
- For additional resource requirements, see the [Planning your deployment](#) guide.



IMPORTANT

- When you need to override the cluster-wide default node selector for OpenShift Data Foundation, you can use the following command in the 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 Data Foundation 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 Data Foundation](#) chapter in the *Managing and Allocating Storage Resources* guide.

Procedure

1. Log in to the OpenShift Web Console.
2. Click **Operators** → **OperatorHub**.
3. Scroll or type **OpenShift Data Foundation** into the **Filter by keyword** box to find the **OpenShift Data Foundation** Operator.
4. Click **Install**.
5. Set the following options on the **Install Operator** page:
 - a. Update Channel as **stable-4.10**.
 - 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 is created during the operator installation.
 - d. Select Approval Strategy as **Automatic** or **Manual**.
If you select **Automatic** updates, then the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without any intervention.

If you select **Manual** updates, then the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to update the Operator to a newer version.
 - e. Ensure that the **Enable** option is selected for the **Console plugin**.
 - f. Click **Install**.

Verification steps

- After the operator is successfully installed, a pop-up with a message, **Web console update is available** appears on the user interface. Click **Refresh web console** from this pop-up for the console changes to reflect.
- In the Web Console:
 - Navigate to Installed Operators and verify that the **OpenShift Data Foundation Operator** shows a green tick indicating successful installation.
 - Navigate to **Storage** and verify if **Data Foundation** dashboard is available.

3.3. CREATING MULTUS NETWORKS [TECHNOLOGY PREVIEW]

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 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 Data Foundation 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.



IMPORTANT

Multus support is a Technology Preview feature that is only supported and has been tested on bare metal and VMWare deployments. 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](#).

3.3.1. Creating network attachment definitions

To utilize Multus, an already working cluster with the correct networking configuration is required, see [Recommended network configuration and requirements for a Multus configuration](#). The newly created NetworkAttachmentDefinition (NAD) can 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 Data Foundation 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 the nodes attached to the Multus network (that is, **ens2** for **ocs-public-cluster**).

The following is an example **NetworkAttachmentDefinition** 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 the nodes attached to the Multus networks (that is, **ens2** for **ocs-public**, and **ens3** for **ocs-cluster**).

3.4. CREATING OPENSIFT DATA FOUNDATION CLUSTER ON VMWARE VSPHERE

VMware vSphere 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 Data Foundation 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 vSphere, 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 [Installing on 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 [Multi network plug-in \(Multus\) support](#) and [Creating network attachment definitions](#).

Procedure

1. In the OpenShift Web Console, click **Operators → Installed Operators** to view all the installed operators.
Ensure that the **Project** selected is **openshift-storage**.
2. Click on the **OpenShift Data Foundation** operator and then click **Create StorageSystem**.
3. In the Backing storage page, perform the following:
 - a. Select **Full Deployment** for the **Deployment type** option.

- b. Select the **Create a new StorageClass using the local storage devices** option.
- c. Click Next.

**NOTE**

You are prompted to install the Local Storage Operator if it is not already installed. Click **Install** and follow the procedure as described in [Installing Local Storage Operator](#).

4. In the Create local volume set page, provide the following information:
 - a. Enter a name for the **LocalVolumeSet** and the **StorageClass**.
By default, the local volume set name appears for the storage class name. You can change the name.
 - b. Select one of the following:
 - **Disks on all nodes** to use the available disks that match the selected filters on all nodes.
 - **Disks on selected nodes** to use the available disks that match the selected filters only on selected nodes.

**IMPORTANT**

- The flexible scaling feature is enabled only when the storage cluster that you created with 3 or more nodes are spread across fewer than the minimum requirement of 3 availability zones.
For information about flexible scaling, see [Add capacity using YAML](#) section in Scaling Storage guide.
- If the nodes selected do not match the OpenShift Data Foundation cluster requirement of an aggregated 30 CPUs and 72 GiB of RAM, a minimal cluster is deployed.
For minimum starting node requirements, see [Resource requirements](#) section in Planning guide.

- c. From the available list of **Disk Type**, select **SSD/NVMe**.
- d. Expand the **Advanced** section and set the following options:

Volume Mode	Block is selected by default.
Device Type	Select one or more device type from the dropdown 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 Disks 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.

- e. Click **Next**.
A pop-up to confirm the creation of LocalVolumeSet is displayed.
 - f. Click **Yes** to continue.
5. In the Capacity and nodes page, configure the following:
 - a. **Available raw capacity** is populated with the capacity value based on all the attached disks associated with the storage class. This takes some time to show up. The **Selected nodes** list shows the nodes based on the storage class.
 - b. Optional: Select the **Taint nodes** checkbox to dedicate the selected nodes for OpenShift Data Foundation.
 - c. Click **Next**.
 6. Optional: In the Security and network page, configure the following based on your requirement:
 - a. To enable encryption, select **Enable data encryption for block and file storage**
 - b. Select one of the following **Encryption level**:
 - **Cluster-wide encryption** to encrypt the entire cluster (block and file).
 - **StorageClass 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. Select an **Authentication Method**.

Using Token authentication method

- Enter a unique **Connection Name**, host **Address** of the Vault server ('https://<hostname or ip>'), **Port** number and **Token**.
- Expand **Advanced Settings** to enter additional settings and certificate details based on your **Vault** configuration:
 - Enter the Key Value secret path in **Backend Path** that is dedicated and unique to OpenShift Data Foundation.
 - Optional: Enter **TLS Server Name** and **Vault Enterprise Namespace**
 - Upload the respective PEM encoded certificate file to provide the **CA Certificate**, **Client Certificate** and **Client Private Key**.
 - Click **Save**.

Using Kubernetes authentication method

- Enter a unique Vault **Connection Name**, host **Address** of the Vault server ('https://<hostname or ip>'), Port number and **Role** name.

- Expand **Advanced Settings** to enter additional settings and certificate details based on your **Vault** configuration:
 - Enter the Key Value secret path in **Backend Path** that is dedicated and unique to OpenShift Data Foundation.
 - Optional: Enter **TLS Server Name** and **Authentication Path** if applicable.
 - Upload the respective PEM encoded certificate file to provide the **CA Certificate**, **Client Certificate** and **Client Private Key**.
 - Click **Save**.
- d. Select one of the following:
 - Select **Default (SDN)** if you are using a single network.
 - Select **Custom (Multus)** if you are using multiple network interfaces.
 - i. Select a **Public Network Interface** from the dropdown.
 - ii. Select a **Cluster Network Interface** from the dropdown.



NOTE

If you are using only one additional network interface, select the single **NetworkAttachmentDefinition**, that is, **ocs-public-cluster** for the Public Network Interface and leave the Cluster Network Interface blank.

- e. Click **Next**.
7. In the Review and create page, review the configuration details.
 - To modify any configuration settings, click **Back** to go back to the previous configuration page.
 8. Click **Create StorageSystem**.

Verification steps

- To verify the final Status of the installed storage cluster:
 - a. In the OpenShift Web Console, navigate to **Installed Operators** → **OpenShift Data Foundation** → **Storage System** → **ocs-storagecluster-storagesystem** → **Resources**.
 - b. Verify that **Status** of **StorageCluster** is **Ready** and has a green tick mark next to it.
- To verify if flexible scaling is enabled on your storage cluster, perform the following steps (for arbiter mode, flexible scaling is disabled):
 1. In the OpenShift Web Console, navigate to **Installed Operators** → **OpenShift Data Foundation** → **Storage System** → **ocs-storagecluster-storagesystem** → **Resources**.

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 Data Foundation are successfully installed, see [Verifying your OpenShift Data Foundation deployment](#).
- To verify the multi networking (Multus), see [Verifying the Multus networking](#).

Additional resources

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

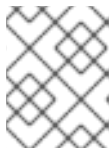
CHAPTER 4. VERIFYING OPENSIFT DATA FOUNDATION DEPLOYMENT

Use this section to verify that OpenShift Data Foundation is deployed correctly.

4.1. VERIFYING THE STATE OF THE PODS

Procedure

1. Click **Workloads** → **Pods** from the OpenShift Web Console.
2. Select **openshift-storage** from the **Project** drop-down list.



NOTE

If the **Show default projects** option is disabled, use the toggle button to list all the default projects.

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 Data Foundation cluster”](#).

3. Click the **Running** and **Completed** tabs to verify that the following pods are in **Running** and **Completed** state:

Table 4.1. Pods corresponding to OpenShift Data Foundation cluster

Component	Corresponding pods
OpenShift Data Foundation Operator	<ul style="list-style-type: none"> ● ocs-operator-* (1 pod on any worker node) ● ocs-metrics-exporter-* (1 pod on any worker node) ● odf-operator-controller-manager-* (1 pod on any worker node) ● odf-console-* (1 pod on any worker node) ● csi-addons-controller-manager-* (1 pod on any worker node)
Rook-ceph Operator	<p>rook-ceph-operator-*</p> <p>(1 pod on any worker node)</p>

Component	Corresponding pods
Multicloud Object Gateway	<ul style="list-style-type: none"> ● 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	<p>rook-ceph-mon-*</p> <p>(3 pods distributed across storage nodes)</p>
MGR	<p>rook-ceph-mgr-*</p> <p>(1 pod on any storage node)</p>
MDS	<p>rook-ceph-mds-ocs-storagecluster-cephfilesystem-*</p> <p>(2 pods distributed across storage nodes)</p>
RGW	<p>rook-ceph-rgw-ocs-storagecluster-cephobjectstore-* (1 pod on any storage node)</p>
CSI	<ul style="list-style-type: none"> ● cephfs <ul style="list-style-type: none"> ○ csi-cephfsplugin-* (1 pod on each worker node) ○ csi-cephfsplugin-provisioner-* (2 pods distributed across worker nodes) ● rbd <ul style="list-style-type: none"> ○ csi-rbdplugin-* (1 pod on each worker node) ○ csi-rbdplugin-provisioner-* (2 pods distributed across worker nodes)
rook-ceph-crashcollector	<p>rook-ceph-crashcollector-*</p> <p>(1 pod on each storage node)</p>

Component	Corresponding pods
OSD	<ul style="list-style-type: none"> ● rook-ceph-osd-* (1 pod for each device) ● rook-ceph-osd-prepare-ocs-deviceset-* (1 pod for each device)

4.2. VERIFYING THE OPENSIFT DATA FOUNDATION CLUSTER IS HEALTHY

Procedure

1. In the OpenShift Web Console, click **Storage** → **Data Foundation**.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
3. In the **Status** card of the **Block and File** tab, verify that *Storage Cluster* has a green tick.
4. In the **Details** card, verify that the cluster information is displayed.

For more information on the health of the OpenShift Data Foundation cluster using the **Block and File** dashboard, see [Monitoring OpenShift Data Foundation](#).

4.3. VERIFYING THE MULTICLOUD OBJECT GATEWAY IS HEALTHY

Procedure

1. In the OpenShift Web Console, click **Storage** → **Data Foundation**.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
 - a. In the **Status** card of the **Object** tab, verify that both *Object Service* and *Data Resiliency* have a green tick.
 - b. In the **Details** card, verify that the MCG information is displayed.

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

4.4. VERIFYING THAT THE OPENSIFT DATA FOUNDATION SPECIFIC STORAGE CLASSES EXIST

Procedure

1. Click **Storage** → **Storage Classes** from the left pane of the OpenShift Web Console.
2. Verify that the following storage classes are created with the OpenShift Data Foundation 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 your cluster, verify the Multus networking.

Procedure

Based on your Network configuration choices, the OpenShift Data Foundation operator will do one of the following:

- If only a single NetworkAttachmentDefinition (for example, **ocs-public-cluster**) was selected for the Public Network Interface, then the traffic between the application pods and the OpenShift Data Foundation cluster will happen on this network. Additionally the cluster will be self configured to also use this network for the replication and rebalancing traffic between OSDs.
- If both NetworkAttachmentDefinitions (for example, **ocs-public** and **ocs-cluster**) were selected for the Public Network Interface and the Cluster Network Interface respectively during the Storage Cluster installation, then client storage traffic will be on the public network and cluster network for the replication and rebalancing traffic between OSDs.

To verify the network configuration is correct, complete the following:

In the OpenShift console, navigate to **Installed Operators → OpenShift Data Foundation → Storage System → ocs-storagecluster-storagesystem → Resources → ocs-storagecluster**.

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:
    ipFamily: IPv4
    provider: multus
    selectors:
      cluster: openshift-storage/ocs-cluster
      public: openshift-storage/ocs-public
  [..]
```

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}'
```

Sample output:

```
{"ipFamily":"IPv4","provider":"multus","selectors":{"cluster":"openshift-storage/ocs-cluster","public":"openshift-storage/ocs-public"}}
```

Confirm the OSD pods are using correct network

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 will connect to both Multus public and cluster networks if both are created. All other OCS pods will 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}'
```

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": {}
}]
```

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}' | jq -r '.[].name'
```

Sample output:

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

CHAPTER 5. DEPLOY STANDALONE MULTICLOUD OBJECT GATEWAY

Deploying only the Multicloud Object Gateway component with the OpenShift Data Foundation provides the flexibility in deployment and helps to reduce the resource consumption. You can deploy the Multicloud Object Gateway component either using dynamic storage devices or using the local storage devices.

5.1. DEPLOY STANDALONE MULTICLOUD OBJECT GATEWAY USING DYNAMIC STORAGE DEVICES

Use this section to deploy only the standalone Multicloud Object Gateway component, which involves the following steps:

- Installing Red Hat OpenShift Data Foundation Operator
- Creating standalone Multicloud Object Gateway

5.1.1. Installing Red Hat OpenShift Data Foundation Operator

You can install Red Hat OpenShift Data Foundation 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 must have at least three worker nodes in the Red Hat OpenShift Container Platform cluster.
- For additional resource requirements, see the [Planning your deployment](#) guide.



IMPORTANT

- When you need to override the cluster-wide default node selector for OpenShift Data Foundation, you can use the following command in the 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 Data Foundation 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 Data Foundation](#) chapter in the *Managing and Allocating Storage Resources* guide.

Procedure

1. Log in to the OpenShift Web Console.
2. Click **Operators** → **OperatorHub**.

3. Scroll or type **OpenShift Data Foundation** into the **Filter by keyword** box to find the **OpenShift Data Foundation Operator**.
4. Click **Install**.
5. Set the following options on the **Install Operator** page:
 - a. Update Channel as **stable-4.10**.
 - 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 is created during the operator installation.
 - d. Select Approval Strategy as **Automatic** or **Manual**.
If you select **Automatic** updates, then the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without any intervention.

If you select **Manual** updates, then the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to update the Operator to a newer version.
 - e. Ensure that the **Enable** option is selected for the **Console plugin**.
 - f. Click **Install**.

Verification steps

- After the operator is successfully installed, a pop-up with a message, **Web console update is available** appears on the user interface. Click **Refresh web console** from this pop-up for the console changes to reflect.
- In the Web Console:
 - Navigate to Installed Operators and verify that the **OpenShift Data Foundation Operator** shows a green tick indicating successful installation.
 - Navigate to **Storage** and verify if **Data Foundation** dashboard is available.

5.1.2. Creating a standalone Multicloud Object Gateway

You can create only the standalone Multicloud Object Gateway component while deploying OpenShift Data Foundation.

Prerequisites

- Ensure that the OpenShift Data Foundation Operator is installed.

Procedure

1. In the OpenShift Web Console, click **Operators** → **Installed Operators** to view all the installed operators.
Ensure that the **Project** selected is **openshift-storage**.
2. Click **OpenShift Data Foundation** operator and then click **Create StorageSystem**.

3. In the **Backing storage** page, select the following:
 - a. Select **Multicloud Object Gateway** for **Deployment type**.
 - b. Select the **Use an existing StorageClass** option.
 - c. Click **Next**.
4. Optional: In the **Security** page, select **Connect to an external key management service**
 - a. **Key Management Service Provider** is set to **Vault** by default.
 - b. Enter Vault **Service Name**, host Address of Vault server ('https:// <hostname or ip>'), **Port number**, and **Token**.
 - c. Expand **Advanced Settings** to enter additional settings and certificate details based on your **Vault** configuration:
 - i. Enter the Key Value secret path in the **Backend Path** that is dedicated and unique to OpenShift Data Foundation.
 - ii. Optional: Enter **TLS Server Name** and **Vault Enterprise Namespace**
 - iii. Upload the respective PEM encoded certificate file to provide the **CA Certificate**, **Client Certificate**, and **Client Private Key**.
 - iv. Click **Save**.
 - d. Click **Next**.
5. In the **Review and create** page, review the configuration details:
To modify any configuration settings, click **Back**.
6. Click **Create StorageSystem**.

Verification steps

Verifying that the OpenShift Data Foundation cluster is healthy

1. In the OpenShift Web Console, click **Storage → Data Foundation**.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
 - a. In the **Status card** of the **Object** tab, verify that both *Object Service* and *Data Resiliency* have a green tick.
 - b. In the **Details** card, verify that the MCG information is displayed.

Verifying the state of the pods

1. Click **Workloads → Pods** from the OpenShift Web Console.
2. Select **openshift-storage** from the **Project** drop-down list and verify that the following pods are in **Running** state.

**NOTE**

If the **Show default projects** option is disabled, use the toggle button to list all the default projects.

Component	Corresponding pods
OpenShift Data Foundation Operator	<ul style="list-style-type: none"> ● ocs-operator-* (1 pod on any worker node) ● ocs-metrics-exporter-* (1 pod on any worker node) ● odf-operator-controller-manager-* (1 pod on any worker node) ● odf-console-* (1 pod on any worker node) ● csi-addons-controller-manager-* (1 pod on any worker node)
Rook-ceph Operator	<p>rook-ceph-operator-*</p> <p>(1 pod on any worker node)</p>
Multicloud Object Gateway	<ul style="list-style-type: none"> ● noobaa-operator-* (1 pod on any worker node) ● noobaa-core-* (1 pod on any worker node) ● noobaa-db-pg-* (1 pod on any worker node) ● noobaa-endpoint-* (1 pod on any worker node)

5.2. DEPLOY STANDALONE MULTICLOUD OBJECT GATEWAY USING LOCAL STORAGE DEVICES

Use this section to deploy only the standalone Multicloud Object Gateway component, which involves the following steps:

- Installing the Local Storage Operator
- Installing Red Hat OpenShift Data Foundation Operator
- Creating standalone Multicloud Object Gateway

5.2.1. Installing Local Storage Operator

Install the Local Storage Operator from the Operator Hub before creating Red Hat OpenShift Data Foundation clusters on local storage devices.

Procedure

1. Log in to the OpenShift Web Console.

2. Click **Operators** → **OperatorHub**.
3. Type **local storage** in the **Filter by keyword** box to find the **Local Storage Operator** from the list of operators and click on it.
4. Set the following options on the **Install Operator** page:
 - a. Update channel as either **4.10** or **stable**.
 - b. Installation mode as **A specific namespace on the cluster**
 - c. Installed Namespace as **Operator recommended namespace openshift-local-storage**.
 - d. Update approval as **Automatic**.
5. Click **Install**.

Verification steps

- Verify that the Local Storage Operator shows a green tick indicating successful installation.

5.2.2. Installing Red Hat OpenShift Data Foundation Operator

You can install Red Hat OpenShift Data Foundation 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 must have at least three worker nodes in the Red Hat OpenShift Container Platform cluster.
- For additional resource requirements, see the [Planning your deployment](#) guide.



IMPORTANT

- When you need to override the cluster-wide default node selector for OpenShift Data Foundation, you can use the following command in the 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 Data Foundation 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 Data Foundation](#) chapter in the *Managing and Allocating Storage Resources* guide.

Procedure

1. Log in to the OpenShift Web Console.

2. Click **Operators** → **OperatorHub**.
3. Scroll or type **OpenShift Data Foundation** into the **Filter by keyword** box to find the **OpenShift Data Foundation Operator**.
4. Click **Install**.
5. Set the following options on the **Install Operator** page:
 - a. Update Channel as **stable-4.10**.
 - 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 is created during the operator installation.
 - d. Select Approval Strategy as **Automatic** or **Manual**.
If you select **Automatic** updates, then the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without any intervention.

If you select **Manual** updates, then the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to update the Operator to a newer version.
 - e. Ensure that the **Enable** option is selected for the **Console plugin**.
 - f. Click **Install**.

Verification steps

- After the operator is successfully installed, a pop-up with a message, **Web console update is available** appears on the user interface. Click **Refresh web console** from this pop-up for the console changes to reflect.
- In the Web Console:
 - Navigate to Installed Operators and verify that the **OpenShift Data Foundation Operator** shows a green tick indicating successful installation.
 - Navigate to **Storage** and verify if **Data Foundation** dashboard is available.

5.2.3. Creating a standalone Multicloud Object Gateway

You can create only the standalone Multicloud Object Gateway component while deploying OpenShift Data Foundation.

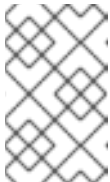
Prerequisites

- Ensure that the OpenShift Data Foundation Operator is installed.

Procedure

1. In the OpenShift Web Console, click **Operators** → **Installed Operators** to view all the installed operators.
Ensure that the **Project** selected is **openshift-storage**.

2. Click **OpenShift Data Foundation** operator and then click **Create StorageSystem**.
3. In the **Backing storage** page, select the following:
 - a. Select **Multicloud Object Gateway** for **Deployment type**.
 - b. Select the **Create a new StorageClass using the local storage devices** option.
 - c. Click **Next**.

**NOTE**

You are prompted to install the Local Storage Operator if it is not already installed. Click **Install**, and follow the procedure as described in [Installing Local Storage Operator](#).

4. In the **Create local volume set** page, provide the following information:
 - a. Enter a name for the **LocalVolumeSet** and the **StorageClass**.
By default, the local volume set name appears for the storage class name. You can change the name.
 - b. Choose one of the following:
 - **Disks on all nodes**
Uses the available disks that match the selected filters on all the nodes.
 - **Disks on selected nodes**
Uses the available disks that match the selected filters only on the selected nodes.
 - c. From the available list of **Disk Type**, select **SSD/NVMe**.
 - d. Expand the **Advanced** section and set the following options:

Volume Mode	Filesystem is selected by default. Always ensure that Filesystem is selected for Volume Mode .
Device Type	Select one or more device type from the dropdown 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 Disks 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.

- e. Click **Next**.
A pop-up to confirm the creation of LocalVolumeSet is displayed.
 - f. Click **Yes** to continue.
5. In the **Capacity and nodes** page, configure the following:

- a. **Available raw capacity** is populated with the capacity value based on all the attached disks associated with the storage class. This takes some time to show up. The **Selected nodes** list shows the nodes based on the storage class.
 - b. Click **Next**.
6. Optional: In the **Security** page, select **Connect to an external key management service**
- a. **Key Management Service Provider** is set to **Vault** by default.
 - b. Enter Vault **Service Name**, host Address of Vault server ('https:// <hostname or ip>'), **Port number**, and **Token**.
 - c. Expand **Advanced Settings** to enter additional settings and certificate details based on your **Vault** configuration:
 - i. Enter the Key Value secret path in the **Backend Path** that is dedicated and unique to OpenShift Data Foundation.
 - ii. Optional: Enter **TLS Server Name** and **Vault Enterprise Namespace**
 - iii. Upload the respective PEM encoded certificate file to provide the **CA Certificate**, **Client Certificate**, and **Client Private Key**.
 - iv. Click **Save**.
 - d. Click **Next**.
7. In the **Review and create** page, review the configuration details:
To modify any configuration settings, click **Back**.
8. Click **Create StorageSystem**.

Verification steps

Verifying that the OpenShift Data Foundation cluster is healthy

1. In the OpenShift Web Console, click **Storage → Data Foundation**.
2. In the **Status** card of the **Overview** tab, click **Storage System** and then click the storage system link from the pop up that appears.
 - a. In the **Status card** of the **Object** tab, verify that both *Object Service* and *Data Resiliency* have a green tick.
 - b. In the **Details** card, verify that the MCG information is displayed.

Verifying the state of the pods

1. Click **Workloads → Pods** from the OpenShift Web Console.
2. Select **openshift-storage** from the **Project** drop-down list and verify that the following pods are in **Running** state.

**NOTE**

If the **Show default projects** option is disabled, use the toggle button to list all the default projects.

Component	Corresponding pods
OpenShift Data Foundation Operator	<ul style="list-style-type: none"> ● ocs-operator-* (1 pod on any worker node) ● ocs-metrics-exporter-* (1 pod on any worker node) ● odf-operator-controller-manager-* (1 pod on any worker node) ● odf-console-* (1 pod on any worker node) ● csi-addons-controller-manager-* (1 pod on any worker node)
Rook-ceph Operator	<p>rook-ceph-operator-*</p> <p>(1 pod on any worker node)</p>
Multicloud Object Gateway	<ul style="list-style-type: none"> ● noobaa-operator-* (1 pod on any worker node) ● noobaa-core-* (1 pod on any worker node) ● noobaa-db-pg-* (1 pod on any worker node) ● noobaa-endpoint-* (1 pod on any worker node)

CHAPTER 6. UNINSTALLING OPENSIFT DATA FOUNDATION

6.1. UNINSTALLING OPENSIFT DATA FOUNDATION IN INTERNAL MODE

To uninstall OpenShift Data Foundation in Internal mode, refer to the [knowledge base article on Uninstalling OpenShift Data Foundation](#).