



Red Hat OpenShift Data Foundation 4.10

Deploying OpenShift Data Foundation using IBM Power

Instructions on deploying Red Hat OpenShift Data Foundation on IBM Power

Red Hat OpenShift Data Foundation 4.10 Deploying OpenShift Data Foundation using IBM Power

Instructions on deploying Red Hat OpenShift Data Foundation on IBM Power

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 to use local storage on IBM Power.

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 OPENSIFT DATA FOUNDATION USING LOCAL STORAGE DEVICES	8
2.1. INSTALLING LOCAL STORAGE OPERATOR	8
2.2. INSTALLING RED HAT OPENSIFT DATA FOUNDATION OPERATOR	8
2.3. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE TOKEN AUTHENTICATION METHOD	10
2.4. ENABLING CLUSTER-WIDE ENCRYPTION WITH KMS USING THE KUBERNETES AUTHENTICATION METHOD	11
2.5. FINDING AVAILABLE STORAGE DEVICES	13
2.6. CREATING OPENSIFT DATA FOUNDATION CLUSTER ON IBM POWER	15
CHAPTER 3. VERIFYING OPENSIFT DATA FOUNDATION DEPLOYMENT FOR INTERNAL MODE	20
3.1. VERIFYING THE STATE OF THE PODS	20
3.2. VERIFYING THE OPENSIFT DATA FOUNDATION CLUSTER IS HEALTHY	22
3.3. VERIFYING THE MULTICLOUD OBJECT GATEWAY IS HEALTHY	22
3.4. VERIFYING THAT THE OPENSIFT DATA FOUNDATION SPECIFIC STORAGE CLASSES EXIST	22
CHAPTER 4. DEPLOY STANDALONE MULTICLOUD OBJECT GATEWAY	24
4.1. INSTALLING LOCAL STORAGE OPERATOR	24
4.2. INSTALLING RED HAT OPENSIFT DATA FOUNDATION OPERATOR	24
4.3. CREATING STANDALONE MULTICLOUD OBJECT GATEWAY ON IBM POWER	26
CHAPTER 5. UNINSTALLING OPENSIFT DATA FOUNDATION	30
5.1. UNINSTALLING OPENSIFT DATA FOUNDATION IN INTERNAL MODE	30

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) IBM Power clusters in connected or disconnected environments along with out-of-the-box support for proxy environments.

Both internal and external OpenShift Data Foundation clusters are supported on IBM Power. See [Planning your deployment](#) and [Preparing to deploy OpenShift Data Foundation](#) for more information about deployment requirements.

To deploy OpenShift Data Foundation, follow the appropriate deployment process based on your requirement:

- Internal-Attached Devices mode
 - [Deploy using local storage devices](#)
 - [Deploy standalone Multicloud Object Gateway component](#)
- External mode using Red Hat Ceph Storage
 - [External mode](#)

CHAPTER 1. PREPARING TO DEPLOY OPENSIFT DATA FOUNDATION

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

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

- 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.

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

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

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 in the cluster with locally attached storage devices on each of them.
 - Each of the three selected nodes must have at least one raw block device available to be used by OpenShift Data Foundation.
 - The devices to be used must be empty, that is, there should be no persistent volumes (PVs), volume groups (VGs), or local volumes (LVs) remaining on the disks.
- You must have a minimum of three labeled nodes.
 - Each node that has local storage devices to be used by OpenShift Data Foundation must have a specific label to deploy OpenShift Data Foundation pods. To label the nodes, use the following command:
 -

```
❯ $ oc label nodes <NodeNames> cluster.ocs.openshift.io/openshift-storage=""
```

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

CHAPTER 2. DEPLOY OPENSIFT DATA FOUNDATION USING LOCAL STORAGE DEVICES

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

Also, it is possible to deploy only the Multicloud Object Gateway (MCG) component with OpenShift Data Foundation. For more information, see [Deploy standalone Multicloud Object Gateway](#).

Perform the following steps to deploy OpenShift Data Foundation:

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

2.1. INSTALLING LOCAL STORAGE OPERATOR

Use this procedure to install the Local Storage Operator from the Operator Hub before creating 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. Approval Strategy as **Automatic**.
5. Click **Install**.

Verification steps

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

2.2. 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.

For information about the hardware and software requirements, see [Planning your deployment](#).

Prerequisites

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



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.

6. 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.

7. Ensure that the **Enable** option is selected for the **Console plugin**.
8. Click **Install**.

Verification steps

- Verify that the **OpenShift Data Foundation** Operator shows a green tick indicating successful installation.
- 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 **Storage** and verify if **Data Foundation** is available.

2.3. 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
```

2.4. 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_CRT"
```

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
```

2.5. FINDING AVAILABLE STORAGE DEVICES

Use this procedure to identify the device names for each of the three or more worker nodes that you have labeled with the OpenShift Data Foundation label **cluster.ocs.openshift.io/openshift-storage=** before creating PVs for IBM Power.

Procedure

1. List and verify the name of the worker nodes with the OpenShift Data Foundation label.

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

Example output:

```
NAME      STATUS  ROLES  AGE   VERSION
worker-0  Ready   worker 2d11h v1.23.3+e419edf
worker-1  Ready   worker 2d11h v1.23.3+e419edf
worker-2  Ready   worker 2d11h v1.23.3+e419edf
```

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

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

Example output:

```
$ oc debug node/worker-0
Starting pod/worker-0-debug ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.0.63
If you don't see a command prompt, try pressing enter.
sh-4.4#
sh-4.4# chroot /host
sh-4.4# lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop1  7:1  0 500G 0 loop
sda    8:0  0 500G 0 disk
sdb    8:16 0 120G 0 disk
|-sdb1 8:17 0   4M 0 part
|-sdb3 8:19 0 384M 0 part
`-sdb4 8:20 0 119.6G 0 part
sdc    8:32 0 500G 0 disk
sdd    8:48 0 120G 0 disk
|-sdd1 8:49 0   4M 0 part
|-sdd3 8:51 0 384M 0 part
`sdd4 8:52 0 119.6G 0 part
sde    8:64 0 500G 0 disk
sdf    8:80 0 120G 0 disk
|-sdf1 8:81 0   4M 0 part
|-sdf3 8:83 0 384M 0 part
`sdf4 8:84 0 119.6G 0 part
sdg    8:96 0 500G 0 disk
sdh    8:112 0 120G 0 disk
|-sdh1 8:113 0   4M 0 part
|-sdh3 8:115 0 384M 0 part
`sdh4 8:116 0 119.6G 0 part
sdi    8:128 0 500G 0 disk
sdj    8:144 0 120G 0 disk
|-sdj1 8:145 0   4M 0 part
|-sdj3 8:147 0 384M 0 part
```

```

`-sdj4 8:148 0 119.6G 0 part
sdk 8:160 0 500G 0 disk
sdl 8:176 0 120G 0 disk
|-sdl1 8:177 0 4M 0 part
|-sdl3 8:179 0 384M 0 part
`-sdl4 8:180 0 119.6G 0 part /sysroot
sdm 8:192 0 500G 0 disk
sdn 8:208 0 120G 0 disk
|-sdn1 8:209 0 4M 0 part
|-sdn3 8:211 0 384M 0 part /boot
`-sdn4 8:212 0 119.6G 0 part
sdo 8:224 0 500G 0 disk
sdp 8:240 0 120G 0 disk
|-sdp1 8:241 0 4M 0 part
|-sdp3 8:243 0 384M 0 part
`sdp4 8:244 0 119.6G 0 part

```

In this example, for worker-0, the available local devices of 500G are **sda, sdc, sde, sdg, sdi, sdk, sdm, sdo**.

3. Repeat the above step for all the other worker nodes that have the storage devices to be used by OpenShift Data Foundation. See this [Knowledge Base article](#) for more details.

2.6. CREATING OPENSIFT DATA FOUNDATION CLUSTER ON IBM POWER

Use this procedure to create an OpenShift Data Foundation cluster after you install the OpenShift Data Foundation operator.

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 (for example, 200 GB SSD) to use local storage devices on IBM Power.
- Verify your OpenShift Container Platform worker nodes are labeled for OpenShift Data Foundation:

```

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

```

To identify storage devices on each node, refer to [Finding available storage devices](#).

Procedure

1. Log into the OpenShift Web Console.
2. In **openshift-local-storage** namespace Click **Operators** → **Installed Operators** to view the installed operators.
3. Click the **Local Storage** installed operator.
4. On the **Operator Details** page, click the **Local Volume** link.

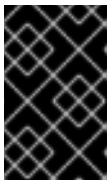
5. Click **Create Local Volume**.
6. Click on **YAML view** for configuring Local Volume.
7. Define a **LocalVolume** custom resource for block PVs using the following YAML.

```

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

```

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



IMPORTANT

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

8. Click **Create**.
9. Confirm whether **diskmaker-manager** pods and **Persistent Volumes** are created.
 - a. For **Pods**
 - i. Click **Workloads → Pods** from the left pane of the OpenShift Web Console.
 - ii. Select **openshift-local-storage** from the **Project** drop-down list.
 - iii. Check if there are **diskmaker-manager** pods for each of the worker node that you used while creating LocalVolume CR.
 - b. For **Persistent Volumes**
 - i. Click **Storage → PersistentVolumes** from the left pane of the OpenShift Web Console.

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



IMPORTANT

- The flexible scaling feature is enabled only when the storage cluster that you created with three or more nodes are spread across fewer than the minimum requirement of three availability zones. For information about flexible scaling, see [Add capacity using YAML](#) section in Scaling Storage guide.

10. In the OpenShift Web Console, click **Operators → Installed Operators** to view all the installed operators.
Ensure that the **Project** selected is **openshift-storage**.
11. Click on the **OpenShift Data Foundation** operator and then click **Create StorageSystem**.
12. In the Backing storage page, perform the following:
 - a. Select **Full Deployment** for the **Deployment type** option.
 - b. Select the **Use an existing StorageClass** option.
 - c. Select the required **Storage Class** that you used while installing LocalVolume. By default, it is set to **none**.
 - d. Click **Next**.
13. 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.
 - b. The **Selected nodes** list shows the nodes based on the storage class.
 - c. Optional: Select the **Taint nodes** checkbox to dedicate the selected nodes for OpenShift Data Foundation.
 - d. Click **Next**.
14. 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. Choose one or both of the following **Encryption level**:
 - **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 **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)** as Multus is not yet supported on OpenShift Data Foundation on IBM Power.

e. Click **Next**.

15. In the **Review and create** page::

- a. Review the configurations details. To modify any configuration settings, click **Back** to go back to the previous configuration page.
- b. 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-storage-system** → **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:
 1. In the OpenShift Web Console, navigate to **Installed Operators** → **OpenShift Data Foundation** → **Storage System** → **ocs-storagecluster-storage-system** → **Resources** → **ocs-storagecluster**.
 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 the components for OpenShift Data Foundation are successfully installed, see [Verifying your OpenShift Data Foundation deployment](#).

Additional resources

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

CHAPTER 3. VERIFYING OPENSIFT DATA FOUNDATION DEPLOYMENT FOR INTERNAL MODE

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

1. [Verify the state of the pods.](#)
2. [Verify that the OpenShift Data Foundation cluster is healthy.](#)
3. [Verify that the Multicloud Object Gateway is healthy.](#)
4. [Verify that the OpenShift Data Foundation specific storage classes exist.](#)

3.1. VERIFYING THE STATE OF THE PODS

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

Procedure

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



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 3.1, "Pods corresponding to OpenShift Data Foundation cluster"](#).

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

Table 3.1. 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)

Component	Corresponding pods
Rook-ceph Operator	rook-ceph-operator-* (1 pod on any worker node)
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	rook-ceph-mon-* (3 pods on each storage node)
MGR	rook-ceph-mgr-* (1 pod on any storage node)
MDS	rook-ceph-mds-ocs-storagecluster-cephfilesystem-* (2 pods distributed across storage node)
RGW	rook-ceph-rgw-ocs-storagecluster-cephobjectstore-* (1 pod on any storage node)
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	rook-ceph-crashcollector-* (1 pod on each storage node)

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

3.2. VERIFYING THE OPENSIFT DATA FOUNDATION CLUSTER IS HEALTHY

Procedure

1. In the OpenShift Web Console, click **Storage** → **Data Foundation**.
2. Click the **Storage Systems** tab and then click on **ocs-storagecluster-storagesystem**.
3. In the **Status card** of Block and File dashboard under Overview tab, verify that both *Storage Cluster* and *Data Resiliency* has a green tick mark.
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](#).

3.3. 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](#).

3.4. 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**

CHAPTER 4. 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. 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

4.1. INSTALLING LOCAL STORAGE OPERATOR

Use this procedure to install the Local Storage Operator from the Operator Hub before creating 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. Approval Strategy as **Automatic**.
5. Click **Install**.

Verification steps

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

4.2. 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.

For information about the hardware and software requirements, see [Planning your deployment](#).

Prerequisites

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



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.
6. 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.
7. Ensure that the **Enable** option is selected for the **Console plugin**.
8. Click **Install**.

Verification steps

- Verify that the **OpenShift Data Foundation** Operator shows a green tick indicating successful installation.
- 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 **Storage** and verify if **Data Foundation** is available.

4.3. CREATING STANDALONE MULTICLOUD OBJECT GATEWAY ON IBM POWER

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.
- (For deploying using local storage devices only) Ensure that Local Storage Operator is installed.

To identify storage devices on each node, refer to [Finding available storage devices](#).

Procedure

1. Log into the OpenShift Web Console.
2. In **openshift-local-storage** namespace, click **Operators** → **Installed Operators** to view the installed operators.
3. Click the **Local Storage** installed operator.
4. On the **Operator Details** page, click the **Local Volume** link.
5. Click **Create Local Volume**.
6. Click on **YAML view** for configuring Local Volume.
7. Define a **LocalVolume** custom resource for filesystem PVs using the following YAML.

```
apiVersion: local.storage.openshift.io/v1
kind: LocalVolume
metadata:
  name: localblock
  namespace: openshift-local-storage
spec:
  logLevel: Normal
  managementState: Managed
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - worker-0
              - worker-1
```

```

- worker-2
storageClassDevices:
- devicePaths:
- /dev/sda
storageClassName: localblock
volumeMode: Filesystem

```

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



IMPORTANT

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

8. Click **Create**.
9. In the OpenShift Web Console, click **Operators** → **Installed Operators** to view all the installed operators.
Ensure that the **Project** selected is **openshift-storage**.
10. Click **OpenShift Data Foundation** operator and then click **Create StorageSystem**.
11. In the **Backing storage** page, select **Multicloud Object Gateway** for **Deployment type**.
12. Select the **Use an existing StorageClass** option for **Backing storage type**.
 - a. Select the **Storage Class** that you used while installing LocalVolume.
13. Click **Next**.
14. 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**.
15. In the **Review and create** page, review the configuration details:
To modify any configuration settings, click **Back**.

16. 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. Click the **Storage Systems** tab and then click on **ocs-storagecluster-storagesystem**.
 - 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>

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 worker node)● noobaa-db-pg-* (1 pod on any worker node)● noobaa-endpoint-* (1 pod on any worker node)● noobaa-default-backing-store-noobaa-pod-* (1 pod on any worker node)

CHAPTER 5. UNINSTALLING OPENSIFT DATA FOUNDATION

5.1. 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](#).