



# Red Hat OpenShift Data Foundation 4.16

## ベアメタルインフラストラクチャーを使用した OpenShift Data Foundation のデプロイ

ベアメタルインフラストラクチャーでローカルストレージを使用した OpenShift  
Data Foundation のデプロイ手順



## Red Hat OpenShift Data Foundation 4.16 ベアメタルインフラストラクチャーを使用した OpenShift Data Foundation のデプロイ

---

ベアメタルインフラストラクチャーでローカルストレージを使用した OpenShift Data Foundation のデプロイ手順

## 法律上の通知

Copyright © 2024 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## 概要

ベアメタルインフラストラクチャーでローカルストレージを使用するために Red Hat OpenShift Data Foundation をインストールする方法については、このドキュメントをご覧ください。

---

## 目次

多様性を受け入れるオープンソースの強化 .....	3
RED HAT ドキュメントへのフィードバック (英語のみ) .....	4
はじめに .....	5
第1章 OPENSIFT DATA FOUNDATION のデプロイの準備 .....	6
1.1. ローカルストレージデバイスを使用して OPENSIFT DATA FOUNDATION をインストールするための要件	6
第2章 ローカルストレージデバイスを使用した OPENSIFT DATA FOUNDATION のデプロイ .....	8
2.1. LOCAL STORAGE OPERATOR のインストール	8
2.2. RED HAT OPENSIFT DATA FOUNDATION OPERATOR のインストール	8
2.3. トークン認証方法を使用した KMS を使用したクラスター全体の暗号化の有効化	10
2.4. KUBERNETES 認証方式を使用した KMS でのクラスター全体の暗号化の有効化	11
2.5. ペアメタルでの OPENSIFT DATA FOUNDATION クラスターの作成	13
2.6. OPENSIFT DATA FOUNDATION デプロイメントの確認	19
第3章 スタンドアロンの MULTICLOUD OBJECT GATEWAY のデプロイ .....	25
3.1. LOCAL STORAGE OPERATOR のインストール	25
3.2. RED HAT OPENSIFT DATA FOUNDATION OPERATOR のインストール	26
3.3. スタンドアロンの MULTICLOUD OBJECT GATEWAY の作成	27
第4章 OPENSIFT DATA FOUNDATION トポロジーの表示 .....	32
第5章 OPENSIFT DATA FOUNDATION のアンインストール .....	33
5.1. 内部モードでの OPENSIFT DATA FOUNDATION のアンインストール	33



## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

## RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があれば、ぜひお知らせください。

フィードバックを送信するには、Bugzilla チケットを作成します。

1. [Bugzilla](#) の Web サイトに移動します。
2. **Component** セクションで、**documentation** を選択します。
3. **Description** フィールドに、ドキュメントの改善に向けたご提案を記入してください。ドキュメントの該当部分へのリンクも記載してください。
4. **Submit Bug** をクリックします。



## はじめに

Red Hat OpenShift Data Foundation は、接続環境または非接続環境での既存の Red Hat OpenShift Container Platform (RHOC) ベアメタルクラスターへのデプロイメントをサポートし、プロキシ環境に対する追加設定なしのサポートを提供します。

ベアメタルでは、内部と外部の両方の OpenShift Data Foundation クラスターがサポートされます。デプロイメントの要件の詳細は、[デプロイメントのプランニング](#) および [OpenShift Data Foundation のデプロイの準備](#) を参照してください。

OpenShift Data Foundation をデプロイするには、要件に適したデプロイメントプロセスを実行します。

- 内部モード
  - ローカルストレージデバイスを使用したデプロイ
  - スタンドアロンの Multicloud Object Gateway コンポーネントのデプロイ
- 外部モード

## 第1章 OPENSIFT DATA FOUNDATION のデプロイの準備

ローカルストレージデバイスを使用して OpenShift Data Foundation を OpenShift Container Platform にデプロイすると、内部クラスターリソースを作成できます。このアプローチは、すべてのアプリケーションが追加のストレージクラスにアクセスできるように、基本サービスを内部的にプロビジョニングします。

ローカルストレージを使用して Red Hat OpenShift Data Foundation のデプロイメントを開始する前に、リソース要件を満たしていることを確認してください。[ローカルストレージデバイスを使用して OpenShift Data Foundation をインストールするための要件](#) を参照してください。

- オプション: 外部のキー管理システム (KMS) を使用してクラスター全体の暗号化を有効にする場合は、次の手順に従います。
  - 有効な Red Hat OpenShift Data Foundation Advanced サブスクリプションがあることを確認してください。OpenShift Data Foundation のサブスクリプションがどのように機能するかを知るには、[OpenShift Data Foundation subscriptions に関するナレッジベースの記事](#) を参照してください。
  - 暗号化にトークン認証方式を選択した場合は、[KMS を使用したトークン認証によるクラスター全体の暗号化の有効化](#) を参照してください。
  - 暗号化に Kubernetes 認証方式を選択した場合は、[Kubernetes 認証方式を使用した KMS でのクラスター全体の暗号化の有効化](#) を参照してください。
  - Vault サーバーで署名済みの証明書を使用していることを確認します。

上記に対処した後、次の手順を実行します。

1. [Local Storage Operator](#) をインストールします。
2. [Red Hat OpenShift Data Foundation Operator](#) をインストールします。
3. [ベアメタル上に OpenShift Data Foundation クラスターを作成します](#)。

### 1.1. ローカルストレージデバイスを使用して OPENSIFT DATA FOUNDATION をインストールするための要件

#### ノードの要件

クラスターが少なくとも 3 つの OpenShift Container Platform ワーカーノードまたはインフラストラクチャーノードで構成されており、ノードごとにローカルに接続されたストレージデバイスを備えている必要があります。

- 選択した 3 つのノードのそれぞれで、少なくとも 1 つの raw ブロックデバイスが使用できる。OpenShift Data Foundation は、1 つ以上の使用可能な raw ブロックデバイスを使用します。
- 使用するデバイスが空である。ディスクには物理ボリューム (PV)、ボリュームグループ (VG)、または論理ボリューム (LV) を含めないでください。

詳細は、[プランニングガイド](#) の [リソース要件](#) セクションを参照してください。

#### 障害復旧の要件

Red Hat OpenShift Data Foundation でサポートされる障害復旧機能では、障害復旧ソリューションを正常に実装するために以下の前提条件をすべて満たす必要があります。

- 有効な Red Hat OpenShift Data Foundation Advanced サブスクリプション。
- 有効な Red Hat Advanced Cluster Management (RHACM) for Kubernetes サブスクリプション。

OpenShift Data Foundation のサブスクリプションの仕組みを確認するには、[OpenShift Data Foundation subscriptions に関するナレッジベースの記事](#)を参照してください。

障害復旧ソリューションの詳細な要件については、[OpenShift ワークロード用の OpenShift Data Foundation Disaster Recovery の設定 ガイド](#)、および Red Hat Advanced Cluster Management for Kubernetes ドキュメントの [インストールガイド](#) の [要件と推奨事項](#) セクションを参照してください。

## Arbiter ストレッチクラスターの要件

この例では、3 番目のゾーンを Arbiter の場所とした上で、単一クラスターが2つのゾーンにデプロイメントされます。このソリューションは現在、オンプレミスおよび同じデータセンター内の OpenShift Container Platform へのデプロイメントを目的としています。このソリューションは、複数のデータセンターにわたるデプロイメントには推奨できません。代わりに、複数のデータセンターにデプロイされており、ネットワークのレイテンシーが低く、データ損失がない DR ソリューションの1番のオプションとして、Metro-DR を検討してください。

OpenShift Data Foundation のサブスクリプションの仕組みを確認するには、[OpenShift Data Foundation subscriptions に関するナレッジベースの記事](#)を参照してください。



### 注記

スケーリングロジックが競合しているため、フレキシブルスケーリングと Arbiter の両方を同時に有効にすることはできません。フレキシブルスケーリングを使用すると、一度に1つのノードを OpenShift Data Foundation クラスターに追加することができます。Arbiter クラスターでは、2つのデータゾーンごとに1つ以上のノードを追加する必要があります。

## compact モードの要件

OpenShift Data Foundation は、3 ノードの OpenShift のコンパクトなベアメタルクラスターにインストールできます。ここでは、すべてのワークロードが3つの強力なマスターノードで実行されます。ワーカーノードまたはストレージノードは含まれません。

OpenShift Container Platform をコンパクトモードで設定するには、OpenShift Container Platform ドキュメントの [インストールガイド](#) の [3 ノードクラスターの設定](#) セクションと、[エッジデプロイメント用の3ノードアーキテクチャーの提供](#) を参照してください。

## ノードの最小要件

OpenShift Data Foundation クラスターは、標準のデプロイメントリソース要件を満たしていない場合に、最小の設定でデプロイされます。

詳細は、[プランニングガイド](#) の [リソース要件](#) セクションを参照してください。

## 第2章 ローカルストレージデバイスを使用した OPENSIFT DATA FOUNDATION のデプロイ

OpenShift Container Platform がすでにインストールされているベアメタルインフラストラクチャーに OpenShift Data Foundation をデプロイします。

また、OpenShift Data Foundation で Multicloud Object Gateway (MCG) コンポーネントのみをデプロイすることもできます。詳細は、[Deploy standalone Multicloud Object Gateway](#) を参照してください。

OpenShift Data Foundation をデプロイするには、以下の手順を実行します。

1. [Local Storage Operator](#) をインストールします。
2. [Red Hat OpenShift Data Foundation Operator](#) をインストールします。
3. [ベアメタル上に OpenShift Data Foundation クラスタを作成します。](#)

### 2.1. LOCAL STORAGE OPERATOR のインストール

ローカルストレージデバイス上に Red Hat OpenShift Data Foundation クラスタを作成する前に、Operator Hub から Local Storage Operator をインストールします。

#### 手順

1. OpenShift Web コンソールにログインします。
2. **Operators** → **OperatorHub** をクリックします。
3. **Filter by keyword** ボックスに **local storage** を入力し、Operator の一覧から **Local Storage Operator** を見つけ、これをクリックします。
4. **Install Operator** ページで、以下のオプションを設定します。
  - a. Channel を **stable** として更新します。
  - b. インストールモードに **A specific namespace on the cluster** を選択します。
  - c. Installed Namespace に **Operator recommended namespace openshift-local-storage** を選択します。
  - d. 承認を **Automatic** として更新します。
5. **Install** をクリックします。

#### 検証手順

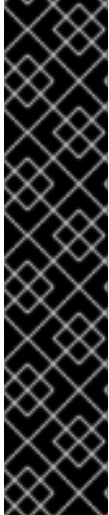
- Local Storage Operator に、インストールが正常に実行されたことを示す緑色のチェックマークが表示されていることを確認します。

### 2.2. RED HAT OPENSIFT DATA FOUNDATION OPERATOR のインストール

Red Hat OpenShift Data Foundation Operator は、Red Hat OpenShift Container Platform Operator Hub を使用してインストールできます。

## 前提条件

- **cluster-admin** 権限および Operator インストール権限を持つアカウントを使用して OpenShift Container Platform クラスターにアクセスできる。
- Red Hat OpenShift Container Platform クラスターにワーカーノードまたはインフラストラクチャーノードが少なくとも3つある。
- その他のリソース要件は、[デプロイメントのプランニング](#) ガイドを参照してください。



### 重要

- OpenShift Data Foundation のクラスター全体でのデフォルトノードセレクターを上書きする必要がある場合は、以下のコマンドを使用して、**openshift-storage** namespace の空のノードセレクターを指定できます (この場合は **openshift-storage** を作成します)。  

```
$ oc annotate namespace openshift-storage openshift.io/node-selector=
```
- ノードに Red Hat OpenShift Data Foundation リソースのみがスケジュールされるように **infra** のテイントを設定します。これにより、サブスクリプションコストを節約できます。詳細は、[ストレージリソースの管理と割り当て](#) ガイドの **Red Hat OpenShift Data Foundation に専用のワーカーノードを使用する方法** セクションを参照してください。

## 手順

1. OpenShift Web コンソールにログインします。
2. **Operators** → **OperatorHub** をクリックします。
3. スクロールするか、**OpenShift Data Foundation** を **Filter by keyword** ボックスに入力し、**OpenShift Data Foundation Operator** を検索します。
4. **Install** をクリックします。
5. **Install Operator** ページで、以下のオプションを設定します。
  - a. Update Channel を **stable-4.16** に設定します。
  - b. Installation Mode オプションに **A specific namespace on the cluster** を選択します。
  - c. Installed Namespace に **Operator recommended namespace openshift-storage** を選択します。namespace **openshift-storage** が存在しない場合は、Operator のインストール時に作成されます。
  - d. 承認ストラテジーを **Automatic** または **Manual** として選択します。  
**Automatic** (自動) 更新を選択すると、Operator Lifecycle Manager (OLM) は介入なしに、Operator の実行中のインスタンスを自動的にアップグレードします。  
  
**Manual** 更新を選択すると、OLM は更新要求を作成します。クラスター管理者は、Operator を新しいバージョンに更新できるように更新要求を手動で承認する必要があります。
  - e. **Console プラグイン** に **Enable** オプションが選択されていることを確認します。

- f. **Install** をクリックします。

### 検証手順

- Operator が正常にインストールされると、**Web console update is available** メッセージを含むポップアップがユーザーインターフェイスに表示されます。このポップアップから **Refresh web console** をクリックして、反映するコンソールを変更します。
- Web コンソールに移動します。
  - Installed Operators に移動し、**OpenShift Data Foundation Operator** に、インストールが正常に実行されたことを示す緑色のチェックマークが表示されていることを確認します。
  - **Storage** に移動し、**Data Foundation** ダッシュボードが使用可能かどうかを確認します。

## 2.3. トークン認証方法を使用した KMS を使用したクラスター全体の暗号化の有効化

トークン認証のために、Vault でキーと値のバックエンドパスおよびポリシーを有効にできます。

### 前提条件

- Vault への管理者アクセス。
- 有効な Red Hat OpenShift Data Foundation Advanced サブスクリプション。詳細は、[OpenShift Data Foundation サブスクリプションに関するナレッジベースの記事](#) を参照してください。
- 後で変更できないため、命名規則に従って一意のパス名をバックエンド **path** として慎重に選択してください。

### 手順

1. Vault で Key/Value (KV) バックエンドパスを有効にします。  
Vault KV シークレットエンジン API の場合は、バージョン 1 です。

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

Vault KV シークレットエンジン API の場合は、バージョン 2 を使用します。

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

2. シークレットに対して書き込み操作または削除操作を実行するようにユーザーを制限するポリシーを作成します。

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

3. 上記のポリシーに一致するトークンを作成します。

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

## 2.4. KUBERNETES 認証方式を使用した KMS でのクラスター全体の暗号化の有効化

キー管理システム (KMS) を使用して、クラスター全体の暗号化に対して Kubernetes 認証方式を有効にできます。

### 前提条件

- Vault への管理者アクセス。
- 有効な Red Hat OpenShift Data Foundation Advanced サブスクリプション。詳細は、[OpenShift Data Foundation サブスクリプションに関するナレッジベースの記事](#) を参照してください。
- OpenShift Data Foundation Operator が Operator Hub からインストールされている。
- バックエンド **path** として一意のパス名を選択する。これは命名規則に厳密に準拠する必要があります。このパス名は後で変更できません。

### 手順

1. サービスアカウントを作成します。

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

ここで、**<serviceaccount\_name>** はサービスアカウントの名前を指定します。

以下に例を示します。

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

2. **clusterrolebindings** と **clusterroles** を作成します。

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

以下に例を示します。

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

3. **serviceaccount** トークンおよび CA 証明書のシークレットを作成します。

```
$ 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
```

ここで、<serviceaccount\_name> は、前の手順で作成したサービスアカウントです。

- シークレットからトークンと CA 証明書を取得します。

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

- OCF クラスターエンドポイントを取得します。

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

- サービスアカウントの発行者を取得します。

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

- 前の手順で収集した情報を使用して、Vault で Kubernetes 認証方法を設定します。

```
$ 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"
```

### 重要

発行者が空の場合は Vault で Kubernetes 認証方法を設定します。

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

- Vault で Key/Value (KV) バックエンドパスを有効にします。  
Vault KV シークレットエンジン API の場合は、バージョン 1 を使用します。

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

Vault KV シークレットエンジン API の場合は、バージョン 2 を使用します。



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

- シークレットに対して **write** または **delete** 操作を実行するようにユーザーを制限するポリシーを作成します。

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

- ロールを作成します。

```
$ 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
```

ロール **odf-rook-ceph-op** は、後でストレージシステムの作成中に KMS 接続の詳細を設定するときに使用されます。

```
$ 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. ベアメタルでの OPENSIFT DATA FOUNDATION クラスターの作成

### 前提条件

- ローカルストレージデバイスを使用して OpenShift Data Foundation をインストールするための要件 セクションにあるすべての要件を満たしていることを確認します。
- マルチネットワークプラグイン (Multus) を使用する場合は、デプロイメント前に、後でクラスターに接続されるネットワークアタッチメント定義 (NAD) を作成しておく。詳細は、[マルチネットワークプラグイン \(Multus\) のサポート](#) および [ネットワークアタッチメント定義の作成](#) を参照してください。

### 手順

- OpenShift Web コンソールで、**Operators** → **Installed Operators** をクリックし、インストールされた Operator を表示します。  
選択された **Project** が **openshift-storage** であることを確認します。
- OpenShift Data Foundation Operator** をクリックした後、**Create StorageSystem** をクリックします。
- Backing storage** ページで、以下を実行します。

- a. **Deployment type** オプションで **Full Deployment** を選択します。
- b. **Create a new StorageClass using the local storage devices** オプションを選択します。
- c. オプション: **Use Ceph RBD as the default StorageClass** を選択します。これにより、StorageClass に手動でアノテーションを付ける必要がなくなります。
- d. オプション: 外部 PostgreSQL を使用するには、**Use external PostgreSQL** チェックボックスを選択します [テクノロジープレビュー]。  
これにより、PostgreSQL Pod が単一障害点となるマルチクラウドオブジェクトゲートウェイの高可用性ソリューションが提供されます。
  - i. 以下の接続の詳細を指定します。
    - ユーザー名
    - パスワード
    - サーバー名とポート
    - データベース名
  - ii. **Enable TLS/SSL** チェックボックスを選択して、Postgres サーバーの暗号化を有効にします。
- e. **Next** をクリックします。



### 重要

Local Storage Operator がまだインストールされていない場合は、インストールするように求められます。**Install** をクリックし、[Local Storage Operator のインストール](#) で説明されている手順に従います。

4. **Create local volume set** ページで、以下の情報を提供します。
  - a. **LocalVolumeSet** および **StorageClass** の名前を入力します。  
ローカルボリュームセット名は、ストレージクラス名のデフォルト値として表示されず。名前を変更できます。
  - b. 以下のいずれかを選択します。
    - **Disks on all nodes**  
すべてのノードにある選択したフィルターに一致する利用可能なディスクを使用します。
    - **Disks on selected nodes**  
選択したノードにある選択したフィルターにのみ一致する利用可能なディスクを使用します。



## 重要

- フレキシブルスケール機能は、3つ以上のノードで作成したストレージクラスターが3つ以上のアベイラビリティゾーンの最低要件未満に分散されている場合にのみ有効になります。  
フレキシブルスケールの詳細は、[フレキシブルスケールが有効な場合に YAML を使用した OpenShift Data Foundation クラスターのスケールに関するナレッジベースの記事](#)を参照してください。
- フレキシブルスケール機能はデプロイ時に有効になり、後で有効または無効にすることはできません。
- 選択したノードが集約された 30 CPU および 72 GiB の RAM の OpenShift Data Foundation クラスターの要件と一致しない場合は、最小クラスターがデプロイされます。  
ノードの最小要件については、[プランニングガイドのリソース要件](#)セクションを参照してください。

c. **Disk Type** の利用可能なリストから、**SSD/NVMe** を選択します。

d. **Advanced** セクションを拡張し、以下のオプションを設定します。

ボリュームモード	Block がデフォルト値として選択されます。
デバイスタイプ	ドロップダウンリストから1つ以上のデバイスタイプを選択します。
ディスクサイズ	デバイスの最小サイズ 100GB と、含める必要のあるデバイスの最大サイズを設定します。
ディスクの最大数の制限	これは、ノード上に作成できる永続ボリューム (PV) の最大数を示します。このフィールドが空のままの場合、PV は一致するノードで利用可能なすべてのディスクに作成されます。

e. **Next** をクリックします。

LocalVolumeSet の作成を確認するポップアップが表示されます。

f. **Yes** をクリックして続行します。

5. **Capacity and nodes** ページで、以下を設定します。

a. **Available raw capacity** には、ストレージクラスに関連付けられた割り当てられたすべてのディスクに基づいて容量の値が設定されます。これには少し時間がかかります。**Selected nodes** リストには、ストレージクラスに基づくノードが表示されます。

b. **Configure performance** セクションで、以下のパフォーマンスプロファイルのいずれかを選択します。

- Lean

これは、最小リソースが推奨値よりも少ない、リソースに制約のある環境で使用します。このプロファイルでは、割り当てられる CPU とメモリーの数が少なくなり、リソースの消費が最小限に抑えられます。

- **balanced (デフォルト)**  
推奨リソースが利用可能な場合にこれを使用します。このプロファイルは、さまざまなワークロードのリソース消費とパフォーマンスのバランスを提供します。
- **パフォーマンス**  
最高のパフォーマンスを得るために十分なリソースがある環境でこれを使用してください。このプロファイルは、負荷の高いワークロードを最適に実行できるように十分なメモリとCPUを割り当てることで、高いパフォーマンスを実現するように調整されています。



### 注記

**StorageSystems** タブのオプションメニューから **Configure performance** オプションを使用して、デプロイメント後にパフォーマンスプロファイルを設定するオプションがあります。



### 重要

リソースプロファイルを選択する前に、クラスター内のリソースの現在の可用性を必ず確認してください。リソースが不十分なクラスターでより高いリソースプロファイルを選択すると、インストールが失敗する可能性があります。

リソース要件の詳細は、[パフォーマンスプロファイルのリソース要件](#) を参照してください。

- オプション: 選択したノードを OpenShift Data Foundation 専用にする場合は、**Taint nodes** チェックボックスを選択します。
  - Next** をクリックします。
- オプション: **Security and network** ページで、要件に応じて以下を設定します。
    - 暗号化を有効にするには、**Enable data encryption for block and file storage** を選択します。
    - 以下の **Encryption level** のいずれかまたは両方を選択します。
      - **クラスター全体の暗号化**  
クラスター全体を暗号化します (ブロックおよびファイル)。
      - **StorageClass の暗号化**  
暗号化対応のストレージクラスを使用して、暗号化された永続ボリューム (ブロックのみ) を作成します。
    - オプション: **Connect to an external key management service** チェックボックスを選択します。これはクラスター全体の暗号化の場合はオプションになります。
      - Key Management Service Provider** ドロップダウンリストから、次のいずれかのプロバイダーを選択し、必要な詳細情報を入力します。
        - **Vault**
          - Authentication Method** を選択します。
            - **トークン認証方式の使用**

- Vault ('https://<hostname or ip>') サーバーの一意的 **Connection Name**、ホストの **Address**、**Port** 番号および **Token** を入力します。
- **Advanced Settings** をデプロイメントして、**Vault** 設定に基づいて追加の設定および証明書の詳細を入力します。
  - OpenShift Data Foundation 専用かつ特有のキーと値のシークレットパスを **Backend Path** に入力します。
- オプション: **TLS Server Name** および **Vault Enterprise Namespace** を入力します。
- PEM でエンコードされた、該当の証明書ファイルをアップロードし、**CA Certificate**、**Client Certificate**、および **Client Private Key** を指定します。
- **Save** をクリックします。
- **Kubernetes 認証方式の使用**
  - Vault ('https://<hostname or ip>') サーバーの一意的 **Connection Name**、ホストの **Address**、**Port** 番号、および **Role** 名を入力します。
  - **Advanced Settings** をデプロイメントして、**Vault** 設定に基づいて追加の設定および証明書の詳細を入力します。
    - OpenShift Data Foundation 専用かつ特有のキーと値のシークレットパスを **Backend Path** に入力します。
    - 該当する場合は、**TLS Server Name** および **Authentication Path** を入力します。
    - PEM でエンコードされた、該当の証明書ファイルをアップロードし、**CA Certificate**、**Client Certificate**、および **Client Private Key** を指定します。
  - **Save** をクリックします。
- **Thales CipherTrust Manager (KMIP を使用)**
  - A. プロジェクト内のキー管理サービスの一意的 **Connection Name** を入力します。
  - B. **Address** および **Port** セクションで、Thales CipherTrust Manager の IP と、KMIP インターフェイスが有効になっているポートを入力します。以下に例を示します。
    - **Address:** 123.34.3.2
    - **Port:** 5696
  - C. **Client Certificate**、**CA certificate**、および **Client Private Key** をアップロードします。
  - D. StorageClass 暗号化が有効になっている場合は、上記で生成された暗号化および復号化に使用する一意の識別子を入力します。

E. **TLS Server** フィールドはオプションであり、KMIP エンドポイントの DNS エントリーがない場合に使用します。たとえば、`kmip_all_<port>.ciphertrustmanager.local` などです。

- ii. **Network** を選択します。
- d. 以下のいずれかを選択します。
- **Default (SDN)**  
単一のネットワークを使用している場合。
  - **Custom (Multus)**  
複数のネットワークインターフェイスを使用している場合。
    - i. ドロップダウンメニューから **Public Network Interface** を選択します。
    - ii. ドロップダウンメニューから **Cluster Network Interface** を選択します。



### 注記

追加のネットワークインターフェイスを1つだけ使用している場合は、単一の**NetworkAttachmentDefinition**(Public Network Interface には**ocs-public-cluster**)を選択し、Cluster Network Interface は空白のままにします。

- e. **Next** をクリックします。
7. **Data Protection** ページで、OpenShift Data Foundation の Regional DR ソリューションを設定している場合は、**Prepare cluster for disaster recovery(Regional-DR only)**チェックボックスを選択し、それ以外の場合は **Next** をクリックします。
  8. **Review and create** ページで、設定の詳細を確認します。  
設定を変更するには、**Back** をクリックして前の設定ページに戻ります。
  9. **Create StorageSystem** をクリックします。



### 注記

デプロイメントに5つ以上のノード、ラック、またはルームがあり、デプロイメント内に5つ以上の障害ドメインが存在する場合、ラックまたはゾーンの数に基づいて Ceph モニター数を設定できます。OpenShift Web コンソールの通知パネルまたはアラートセンターにアラートが表示され、Ceph モニター数を増やすオプションが示されます。アラートで **Configure** オプションを使用して、Ceph モニター数を設定できます。詳細は、[Ceph モニター数が少ないというアラートの解決](#) を参照してください。

### 検証手順

- インストールされたストレージクラスターの最終ステータスを確認するには、以下を実行します。
  - a. OpenShift Web コンソールで、**Installed Operators** → **OpenShift Data Foundation** → **Storage System** に移動します。
  - b. `ocs-storagecluster-storagesystem` → **Resources** をクリックします。

- c. **StorageCluster** の **Status** が **Ready** 完了で、横に緑色のチェックマークが付いていることを確認します。
- フレキシブルスケーリングがストレージクラスターで有効にされているかどうかを確認するには、以下の手順を実行します (arbiter モードの場合、柔軟なスケーリングが無効になります)。
  1. OpenShift Web コンソールで、**Installed Operators** → **OpenShift Data Foundation** → **Storage System** に移動します。
  2. **ocs-storagecluster-storagesystem** → **Resources** → **ocs-storagecluster** をクリックします。
  3. YAML タブで、**spec** セクションでキー **flexibleScaling** を検索し、**status** セクションで **failureDomain** を検索します。 **flexible scaling** が true で、 **failureDomain** がホストに設定されている場合、フレキシブルスケーリング機能が有効になります。

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

- OpenShift Data Foundation のすべてのコンポーネントが正常にインストールされていることを確認するには、[OpenShift Data Foundation インストールの確認](#)を参照してください。
- マルチネットワーク (Multus) を確認するには、[Verifying the Multus networking](#)を参照してください。

## 関連情報

- 初期クラスターの容量を拡張するには、[ストレージのスケーリング ガイド](#)を参照してください。

## 2.6. OPENSIFT DATA FOUNDATION デプロイメントの確認

OpenShift Data Foundation が正常にデプロイされていることを確認するには、以下を実行します。

1. Pod の状態を確認します。
2. [OpenShift Data Foundation クラスターが正常であることを確認](#)します。
3. [Multicloud Object Gateway が正常であることを確認](#)
4. [OpenShift Data Foundation 固有のストレージクラスが存在](#)することを確認
5. [Multus ネットワークを確認](#)します。

### 2.6.1. Pod の状態の確認

#### 手順

1. OpenShift Web コンソールから **Workloads** → **Pods** をクリックします。
2. **Project** ドロップダウンリストから **openshift-storage** を選択します。



## 注記

**Show default projects** オプションが無効になっている場合は、切り替えボタンを使用して、すべてのデフォルトプロジェクトをリスト表示します。

各コンポーネントの予想される Pod 数と、それがノード数によってどのように変化するかについては、次の表を参照してください。

1. 実行中および完了した Pod のフィルターを設定して、次の Pod が **Running** および **Completed** 状態であることを確認します。

コンポーネント	対応する Pod
OpenShift Data Foundation Operator	<ul style="list-style-type: none"> <li>● <b>ocs-operator-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>ocs-metrics-exporter-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>odf-operator-controller-manager-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>odf-console-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>csi-addons-controller-manager-*</b> (任意のストレージノードに 1Pod)</li> </ul>
Rook-ceph Operator	<b>rook-ceph-operator-*</b> (任意のストレージノードに 1Pod)
Multicloud Object Gateway	<ul style="list-style-type: none"> <li>● <b>noobaa-operator-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>noobaa-core-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>noobaa-db-pg-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>noobaa-endpoint-*</b> (任意のストレージノードに 1Pod)</li> </ul>
MON	<b>rook-ceph-mon-*</b> (ストレージノードに分散する 3 Pod)
MGR	<b>rook-ceph-mgr-*</b> (任意のストレージノードに 1Pod)



MDS	<b>rook-ceph-mds-ocs-storagecluster-cephfilesystem-*</b>  (ストレージノードに分散する 2 Pod)
RGW	<b>rook-ceph-rgw-ocs-storagecluster-cephobjectstore-*</b> (任意のストレージノードに 1 Pod)
CSI	<ul style="list-style-type: none"> <li>● <b>cephfs</b> <ul style="list-style-type: none"> <li>○ <b>csi-cephfsplugin-*</b> (各ストレージノードに 1 Pod)</li> <li>○ <b>csi-cephfsplugin-provisioner-*</b> (ストレージノードに分散する 2 Pod)</li> </ul> </li> <li>● <b>rbd</b> <ul style="list-style-type: none"> <li>○ <b>csi-rbdplugin-*</b> (各ストレージノードに 1 Pod)</li> <li>○ <b>csi-rbdplugin-provisioner-*</b> (ストレージノードに分散する 2 Pod)</li> </ul> </li> </ul>
rook-ceph-crashcollector	<b>rook-ceph-crashcollector-*</b>  (各ストレージノードに 1 Pod)
OSD	<ul style="list-style-type: none"> <li>● <b>rook-ceph-osd-*</b> (各デバイス用に 1 Pod)</li> </ul>

## 2.6.2. OpenShift Data Foundation クラスターが正常であることを確認

### 手順

1. OpenShift Web コンソールで、**Storage → Data Foundation** をクリックします。
2. **Overview** タブの **Status** カードで **Storage System** をクリックし、表示されたポップアップからストレージシステムリンクをクリックします。
3. **Block and File** タブの **Status** カードで、**Storage Cluster** に緑色のチェックマークが表示されていることを確認します。
4. **Details** カードで、クラスター情報が表示されていることを確認します。

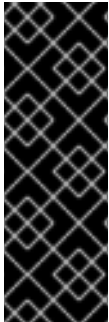
ブロックおよびファイルダッシュボードを使用した OpenShift Data Foundation クラスターの正常性については、[Monitoring OpenShift Data Foundation](#) を参照してください。

## 2.6.3. Multicloud Object Gateway が正常であることを確認

### 手順

1. OpenShift Web コンソールで、**Storage → Data Foundation** をクリックします。
2. **Overview** タブの **Status** カードで **Storage System** をクリックし、表示されたポップアップからストレージシステムリンクをクリックします。
  - a. **Object** タブの **Status card** で、**Object Service** と **Data Resiliency** の両方に緑色のチェックマークが表示されていることを確認します。
  - b. **Details** カードで、MCG 情報が表示されることを確認します。

ブロックおよびファイルダッシュボードを使用した OpenShift Data Foundation クラスターの正常性については、[OpenShift Data Foundation の監視](#) を参照してください。



### 重要

Multicloud Object Gateway には、データベースのコピー (NooBaa DB) が1つだけあります。つまり、NooBaa DB PVC が破損し、回復できない場合は、Multicloud Object Gateway に存在するアプリケーションデータが完全に失われる可能性があります。このため、Red Hat では NooBaa DB PVC のバックアップを定期的にとることを推奨しています。NooBaa DB に障害が発生して回復できない場合は、最新のバックアップバージョンに戻すことができます。NooBaa DB をバックアップする手順は、[こちらのナレッジベースの記事](#) の手順に従ってください。

## 2.6.4. 特定のストレージクラスが存在することを確認

### 手順

1. OpenShift Web コンソールの左側のペインから **Storage → Storage Classes** をクリックします。
2. 以下のストレージクラスが OpenShift Data Foundation クラスターの作成時に作成されることを確認します。
  - **ocs-storagecluster-ceph-rbd**
  - **ocs-storagecluster-cephfs**
  - **openshift-storage.noobaa.io**
  - **ocs-storagecluster-ceph-rgw**

## 2.6.5. Multus ネットワークの確認

Multus がクラスターで機能しているかどうかを判別するには、Multus ネットワークを確認します。

### 手順

ネットワーク設定の選択に応じて、OpenShift Data Foundation Operator は以下の1つを行います。

- 単一の NetworkAttachmentDefinition (例:**ocs-public-cluster**) のみが Public Network Interface に対して選択される場合、アプリケーション Pod と OpenShift Data Foundation クラスター間のトラフィックはこのネットワークで生じます。さらに、クラスターは、このネットワークを OSD 間のレプリケーションに使用し、OSD 間のトラフィックを再バランスするように自己設定します。
- NetworkAttachmentDefinitions (例:**ocs-public** および **ocs-cluster**) が Public Network Interface

にそれぞれ選択されており、Storage Cluster のインストール時に Cluster Network Interface にそれぞれ選択される場合、クライアントストレージトラフィックは OSD 間でのレプリケーションおよびクラスターネットワークについてパブリックネットワークおよびクラスターネットワークに置かれます。

ネットワーク設定が正しいことを確認するには、以下の手順を実施します。

OpenShift コンソールで、**Installed Operators** → **OpenShift Data Foundation** → **Storage System** → **ocs-storagecluster-storagesystem** → **Resources** → **ocs-storagecluster** の順に移動します。

YAML タブで、**spec** セクションで **network** を検索し、設定がネットワークインターフェイスの選択に適したことを確認します。この例では、クライアントストレージトラフィックをストレージレプリケーショントラフィックから分離するためのものです。

出力サンプル

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

コマンドラインインターフェイスを使用してネットワーク設定が正しいことを確認するには、以下のコマンドを実行します。

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

出力サンプル

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

### OSD Pod が正しいネットワークを使用していることの確認

**openshift-storage** namespace は OSD Pod の1つを使用して、Pod が正しいネットワークに接続されていることを確認します。この例では、クライアントストレージトラフィックをストレージレプリケーショントラフィックから分離するためのものです。



#### 注記

両方が作成されると、OSD Pod のみが Multus パブリックおよびクラスターネットワークの両方に接続します。他のすべての OCS Pod は Multus パブリックネットワークに接続されます。

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

## 出力サンプル

```

[
  {
    "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": {}
  }
]

```

コマンドラインインターフェイスを使用して OSD Pod が正しいネットワークを使用していることを確認するには、以下のコマンドを実行します (jq ユーティリティが必要です)。

```

$ 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'

```

## 出力サンプル

```

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

```

## 第3章 スタンドアロンの MULTICLOUD OBJECT GATEWAY のデプロイ

OpenShift Data Foundation で Multicloud Object Gateway コンポーネントのみをデプロイすると、デプロイメントで柔軟性が高まり、リソース消費を減らすことができます。このセクションには、次の手順が含まれており、スタンドアロンの Multicloud Object Gateway コンポーネントのみをデプロイする場合に使用します。

- Local Storage Operator のインストール
- Red Hat OpenShift Data Foundation Operator のインストール
- スタンドアロンの Multicloud Object Gateway の作成



### 重要

Multicloud Object Gateway には、データベースのコピー (NooBaa DB) が1つだけあります。つまり、NooBaa DB PVC が破損し、回復できない場合は、Multicloud Object Gateway に存在するアプリケーションデータが完全に失われる可能性があります。このため、Red Hat では NooBaa DB PVC のバックアップを定期的にとることを推奨しています。NooBaa DB に障害が発生して回復できない場合は、最新のバックアップバージョンに戻すことができます。NooBaa DB をバックアップする手順は、[こちらのナレッジベースの記事](#) の手順に従ってください。

### 3.1. LOCAL STORAGE OPERATOR のインストール

ローカルストレージデバイス上に Red Hat OpenShift Data Foundation クラスタを作成する前に、Operator Hub から Local Storage Operator をインストールします。

#### 手順

1. OpenShift Web コンソールにログインします。
2. **Operators** → **OperatorHub** をクリックします。
3. **Filter by keyword** ボックスに **local storage** を入力し、Operator の一覧から **Local Storage Operator** を見つけ、これをクリックします。
4. **Install Operator** ページで、以下のオプションを設定します。
  - a. Channel を **stable** として更新します。
  - b. インストールモードに **A specific namespace on the cluster** を選択します。
  - c. Installed Namespace に **Operator recommended namespace openshift-local-storage** を選択します。
  - d. 承認を **Automatic** として更新します。
5. **Install** をクリックします。

#### 検証手順

- Local Storage Operator に、インストールが正常に実行されたことを示す緑色のチェックマークが表示されていることを確認します。

## 3.2. RED HAT OPENSIFT DATA FOUNDATION OPERATOR のインストール

Red Hat OpenShift Data Foundation Operator は、Red Hat OpenShift Container Platform Operator Hub を使用してインストールできます。

### 前提条件

- **cluster-admin** 権限および Operator インストール権限を持つアカウントを使用して OpenShift Container Platform クラスタにアクセスできる。
- Red Hat OpenShift Container Platform クラスタにワーカーノードまたはインフラストラクチャーノードが少なくとも 3 つある。
- その他のリソース要件は、[デプロイメントのプランニング](#) ガイドを参照してください。

### 重要

- OpenShift Data Foundation のクラスタ全体でのデフォルトノードセレクターを上書きする必要がある場合は、以下のコマンドを使用して、**openshift-storage** namespace の空のノードセレクターを指定できます (この場合は **openshift-storage** を作成します)。

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

- ノードに Red Hat OpenShift Data Foundation リソースのみがスケジュールされるように **infra** のテイントを設定します。これにより、サブスクリプションコストを節約できます。詳細は、[ストレージリソースの管理と割り当て](#) ガイドの **Red Hat OpenShift Data Foundation に専用のワーカーノードを使用する方法** セクションを参照してください。

### 手順

1. OpenShift Web コンソールにログインします。
2. **Operators** → **OperatorHub** をクリックします。
3. スクロールするか、**OpenShift Data Foundation** を **Filter by keyword** ボックスに入力し、**OpenShift Data Foundation Operator** を検索します。
4. **Install** をクリックします。
5. **Install Operator** ページで、以下のオプションを設定します。
  - a. Update Channel を **stable-4.16** に設定します。
  - b. Installation Mode オプションに **A specific namespace on the cluster** を選択します。
  - c. Installed Namespace に **Operator recommended namespace openshift-storage** を選択します。namespace **openshift-storage** が存在しない場合は、Operator のインストール時に作成されます。
  - d. 承認ストラテジーを **Automatic** または **Manual** として選択します。  
**Automatic** (自動) 更新を選択すると、Operator Lifecycle Manager (OLM) は介入なしに、Operator の実行中のインスタンスを自動的にアップグレードします。

**Manual** 更新を選択すると、OLM は更新要求を作成します。クラスター管理者は、Operator を新しいバージョンに更新できるように更新要求を手動で承認する必要があります。

- e. **Console プラグイン** に **Enable** オプションが選択されていることを確認します。
- f. **Install** をクリックします。

#### 検証手順

- Operator が正常にインストールされると、**Web console update is available** メッセージを含むポップアップがユーザーインターフェイスに表示されます。このポップアップから **Refresh web console** をクリックして、反映するコンソールを変更します。
- Web コンソールに移動します。
  - **Installed Operators** に移動し、**OpenShift Data Foundation Operator** に、インストールが正常に実行されたことを示す緑色のチェックマークが表示されていることを確認します。
  - **Storage** に移動し、**Data Foundation** ダッシュボードが使用可能かどうかを確認します。

### 3.3. スタンドアロンの MULTICLOUD OBJECT GATEWAY の作成

OpenShift Data Foundation のデプロイ中には、スタンドアロンの Multicloud Object Gateway コンポーネントのみを作成できます。

#### 前提条件

- OpenShift Data Foundation Operator がインストールされている。

#### 手順

1. OpenShift Web コンソールで、**Operators** → **Installed Operators** をクリックし、インストールされた Operator を表示します。  
選択された **Project** が **openshift-storage** であることを確認します。
2. **OpenShift Data Foundation Operator** をクリックした後、**Create StorageSystem** をクリックします。
3. **Backing storage** ページで、以下を選択します。
  - a. **Deployment type** の **Multicloud Object Gateway** を選択します。
  - b. **Create a new StorageClass using the local storage devices** オプションを選択します。
  - c. **Next** をクリックします。



#### 注記

Local Storage Operator がまだインストールされていない場合は、インストールするように求められます。**Install** をクリックし、[Local Storage Operator のインストール](#) で説明されている手順に従います。

4. **Create local volume set** ページで、以下の情報を提供します。

- a. **LocalVolumeSet** および **StorageClass** の名前を入力します。  
デフォルトで、ローカルボリュームセット名がストレージクラス名について表示されま  
す。名前を変更できます。
- b. 以下のいずれかを選択します。
  - **Disks on all nodes**  
すべてのノードにある選択したフィルターに一致する利用可能なディスクを使用しま  
す。
  - **Disks on selected nodes**  
選択したノードにある選択したフィルターにのみ一致する利用可能なディスクを使用し  
ます。
- c. **Disk Type** の利用可能なリストから、**SSD/NVMe** を選択します。
- d. **Advanced** セクションを拡張し、以下のオプションを設定します。

ボリューム モード	デフォルトではファイルシステムが選択されています。 <b>Volume Mode</b> で Filesystem が選択されていることを常に確認してください。
デバイスタ イプ	ドロップダウンリストから1つ以上のデバイスタイプを選択します。
ディスクサ イズ	デバイスの最小サイズ 100GB と、含める必要のあるデバイスの最大サイズを設 定します。
ディスクの 最大数の制 限	これは、ノードで作成できる PV の最大数を示します。このフィールドが空のま まの場合、PV は一致するノードで利用可能なすべてのディスクに作成されま す。

- e. **Next** をクリックします。  
LocalVolumeSet の作成を確認するポップアップが表示されます。
  - f. **Yes** をクリックして続行します。
5. **Capacity and nodes** ページで、以下を設定します。
    - a. **Available raw capacity** には、ストレージクラスに関連付けられた割り当てられたすべての  
ディスクに基づいて容量の値が設定されます。これには少し時間がかかります。**Selected  
nodes** リストには、ストレージクラスに基づくノードが表示されます。
    - b. **Next** をクリックします。
  6. オプション: **Connect to an external key management service** チェックボックスを選択しま  
す。これはクラスター全体の暗号化の場合はオプションになります。
    - a. **Key Management Service Provider** ドロップダウンリストから、**Vault** または **Thales  
CipherTrust Manager (using KMIP)** を選択します。**Vault** を選択した場合は、次の手順に  
進みます。**Thales CipherTrust Manager (using KMIP)** を選択した場合は、手順 iii に進み  
ます。
    - b. **Authentication Method** を選択します。  
  
**トークン認証方式の使用**



- Vault ('https://<hostname or ip>') サーバーの一意の **Connection Name**、ホストの **Address**、**Port** 番号および **Token** を入力します。
- **Advanced Settings** をデプロイメントして、**Vault** 設定に基づいて追加の設定および証明書の詳細を入力します。
  - OpenShift Data Foundation 専用かつ特有のキーと値のシークレットパスを **Backend Path** に入力します。
  - オプション: **TLS Server Name** および **Vault Enterprise Namespace** を入力します。
  - PEM でエンコードされた、該当の証明書ファイルをアップロードし、**CA Certificate**、**Client Certificate**、および **Client Private Key** を指定します。
  - **Save** をクリックして、手順 iv に進みます。

### Kubernetes 認証方式の使用

- Vault ('https://<hostname or ip>') サーバーの一意の **Connection Name**、ホストの **Address**、**Port** 番号、および **Role** 名を入力します。
  - **Advanced Settings** をデプロイメントして、**Vault** 設定に基づいて追加の設定および証明書の詳細を入力します。
    - OpenShift Data Foundation 専用かつ特有のキーと値のシークレットパスを **Backend Path** に入力します。
    - 該当する場合は、**TLS Server Name** および **Authentication Path** を入力します。
    - PEM でエンコードされた、該当の証明書ファイルをアップロードし、**CA Certificate**、**Client Certificate**、および **Client Private Key** を指定します。
    - **Save** をクリックして、手順 iv に進みます。
- c. **Thales CipherTrust Manager (using KMIP)** を KMS プロバイダーとして使用するには、次の手順に従います。
- i. プロジェクト内のキー管理サービスの一意の **Connection Name** を入力します。
  - ii. **Address** および **Port** セクションで、Thales CipherTrust Manager の IP と、KMIP インターフェイスが有効になっているポートを入力します。以下に例を示します。
    - **Address:** 123.34.3.2
    - **Port:** 5696
  - iii. **Client Certificate**、**CA certificate**、および **Client Private Key** をアップロードします。
  - iv. StorageClass 暗号化が有効になっている場合は、上記で生成された暗号化および復号化に使用する一意の識別子を入力します。
  - v. **TLS Server** フィールドはオプションであり、KMIP エンドポイントの DNS エントリがない場合に使用します。たとえば、**kmip\_all\_<port>.ciphertrustmanager.local** などです。

- d. **Network** を選択します。
  - e. **Next** をクリックします。
7. **Review and create** ページで、設定の詳細を確認します。  
設定を変更するには、**Back** をクリックします。
  8. **Create StorageSystem** をクリックします。

## 検証手順

### OpenShift Data Foundation クラスタが正常であることの確認

1. OpenShift Web コンソールで、**Storage** → **Data Foundation** をクリックします。
2. **Overview** タブの **Status** カードで **Storage System** をクリックし、表示されたポップアップからストレージシステムリンクをクリックします。
  - a. **Object** タブの **Status card** で、**Object Service** と **Data Resiliency** の両方に緑色のチェックマークが表示されていることを確認します。
  - b. **Details** カードで、MCG 情報が表示されることを確認します。

### Pod の状態の確認

1. OpenShift Web コンソールから **Workloads** → **Pods** をクリックします。
2. **Project** ドロップダウンリストから **openshift-storage** を選択し、以下の Pod が **Running** 状態にあることを確認します。



#### 注記

**Show default projects** オプションが無効になっている場合は、切り替えボタンを使用して、すべてのデフォルトプロジェクトをリスト表示します。

コンポーネント	対応する Pod
OpenShift Data Foundation Operator	<ul style="list-style-type: none"> <li>● <b>ocs-operator-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>ocs-metrics-exporter-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>odf-operator-controller-manager-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>odf-console-*</b> (任意のストレージノードに 1Pod)</li> <li>● <b>csi-addons-controller-manager-*</b> (任意のストレージノードに 1Pod)</li> </ul>
Rook-ceph Operator	<b>rook-ceph-operator-*</b> (任意のストレージノードに 1Pod)

コンポーネント	対応する Pod
Multicloud Object Gateway	<ul style="list-style-type: none"><li>● <b>noobaa-operator-*</b> (任意のストレージノードに 1Pod)</li><li>● <b>noobaa-core-*</b> (任意のストレージノードに 1Pod)</li><li>● <b>noobaa-db-pg-*</b> (任意のストレージノードに 1Pod)</li><li>● <b>noobaa-endpoint-*</b> (任意のストレージノードに 1Pod)</li></ul>

## 第4章 OPENSIFT DATA FOUNDATION トポロジーの表示

トポロジーは、OpenShift Data Foundation ストレージクラスターをマップしたた視覚情報をさまざまな抽象化レベルで示し、このような階層の操作も可能にします。このビューでは、ストレージクラスターがさまざまな要素でどのように構成されているかがわかります。

### 手順

1. OpenShift Web コンソールで、**Storage** → **Data Foundation** → **Topology** に移動します。  
このビューには、ストレージクラスターとその内部のゾーンが表示されます。ノードがゾーン内(点線で示されている)にある円形のエンティティで表示されていることがわかります。各アイテムまたはリソースのラベルには、ステータスやヘルス、アラートの状態などの基本情報が含まれています。
2. ノードを選択すると、右側のパネルにノードの詳細が表示されます。検索/プレビューデコレーターアイコンをクリックして、ノード内のリソースまたはデプロイメントにアクセスすることもできます。
3. デプロイメントの詳細を表示します。
  - a. ノード上のプレビューデコレーターをクリックします。ノードの上にモーダルウィンドウが表示され、そのノードに関連付けられているすべてのデプロイメントとそのステータスが表示されます。
  - b. モデルの左上隅にある **Back to main view** ボタンをクリックしてモデルを閉じ、前のビューに戻ります。
  - c. 特定のデプロイメントを選択すると、そのデプロイメントに関する詳細が表示されます。関連するデータがすべてサイドパネルに表示されます。
4. **Resources** タブをクリックして Pod 情報を表示します。このタブを使用すると、問題の理解を深めることができるだけでなく、複数の詳細レベルが提供されるので適切にトラブルシューティングができるようになります。
5. Pod のリンクをクリックして、OpenShift Container Platform の Pod 情報ページを表示します。リンクは新しいウィンドウで開きます。

## 第5章 OPENSIFT DATA FOUNDATION のアンインストール

### 5.1. 内部モードでの OPENSIFT DATA FOUNDATION のアンインストール

内部モードで OpenShift Data Foundation をアンインストールするには、[OpenShift Data Foundation のアンインストールに関するナレッジベース記事](#)を参照してください。