



Red Hat OpenShift Data Foundation 4.15

IBM Power を使用した OpenShift Data Foundation のデプロイ

IBM Power を使用した Red Hat OpenShift Data Foundation のデプロイ手順

Red Hat OpenShift Data Foundation 4.15 IBM Power を使用した OpenShift Data Foundation のデプロイ

IBM Power を使用した Red Hat 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[®] 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.

概要

IBM Power 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. ローカルストレージ 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. 利用可能なストレージデバイスの検索	13
2.6. IBM POWER での OPENSIFT DATA FOUNDATION クラスターの作成	15
第3章 内部モードの OPENSIFT DATA FOUNDATION デプロイメントの確認	22
3.1. POD の状態の確認	22
3.2. OPENSIFT DATA FOUNDATION クラスターの正常性の確認	24
3.3. MULTICLOUD OBJECT GATEWAY が正常であることの確認	24
3.4. 特定のストレージクラスが存在することの確認	25
第4章 スタンドアロンの MULTICLOUD OBJECT GATEWAY のデプロイ	26
4.1. ローカルストレージ OPERATOR のインストール	26
4.2. RED HAT OPENSIFT DATA FOUNDATION OPERATOR のインストール	26
4.3. IBM POWER でのスタンドアロン MULTICLOUD OBJECT GATEWAY の作成	28
第5章 OPENSIFT DATA FOUNDATION トポロジーの表示	33
第6章 OPENSIFT DATA FOUNDATION のアンインストール	34
6.1. 内部モードでの OPENSIFT DATA FOUNDATION のアンインストール	34

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、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。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追加してください。
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) IBM Power 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。

IBM Power 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。

- 内部接続デバイスモード
 - [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](#)
- Red Hat Ceph Storage を使用した外部モード
 - [外部モード](#)

第1章 OPENSIFT DATA FOUNDATION のデプロイの準備

IBM Power が提供するローカルストレージデバイスを使用して OpenShift Container Platform に OpenShift Data Foundation をデプロイすると、内部クラスターリソースを作成できます。このアプローチでは、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 Key Management System (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。
 - 暗号化にトークン認証方法が選択されている場合は、[Enabling cluster-wide encryption with the Token authentication using KMS](#) を参照してください。
 - 暗号化に Kubernetes 認証方式が選択されている場合は、[KMS を使用した Kubernetes 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。
 - 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. [ローカルストレージ Operator のインストール](#)
2. [Red Hat OpenShift Data Foundation Operator のインストール](#)
3. [利用可能なストレージデバイスの検索](#)
4. [IBM Power で OpenShift Data Foundation クラスターの作成](#)

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

- クラスターは、それぞれローカルに接続されたストレージデバイスを持つクラスターの3つ以上の OpenShift Container Platform ワーカーノードで設定される必要があります。
 - 選択した3つのノードには、OpenShift Data Foundation で使用できる raw ブロックデバイスが少なくとも1つ必要です。
 - 使用するデバイスは空である必要があります。つまり、ディスクには永続ボリューム (PV)、ボリュームグループ (VG)、または論理ボリューム (LV) がない状態であればなりません。
- 3つ以上のラベルが付けられたワーカーノードが必要です。

- OpenShift Data Foundation によって使用されるローカルストレージデバイスを持つ各ノードには、OpenShift Data Foundation Pod をデプロイするための特定のラベルが必要です。ノードにラベルを付けるには、以下のコマンドを使用します。

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

詳細は、プランニングガイドの [リソース要件](#) の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。

詳細な障害復旧の要件は、[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](#) の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

このセクションを使用して、OpenShift Container Platform がすでにインストールされている IBM Power 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. [ローカルストレージ Operator のインストール](#)
2. [Red Hat OpenShift Data Foundation Operator のインストール](#)
3. [利用可能なストレージデバイスの検索](#)
4. [IBM Power 上での OpenShift Data Foundation クラスターの作成](#)

2.1. ローカルストレージ OPERATOR のインストール

以下の手順を使用して、ローカルストレージデバイスに OpenShift Data Foundation クラスターを作成する前に、Operator Hub からローカルストレージ 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. 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-local-storage** を選択します。
 - d. Approval Strategy に **Automatic** を選択します。
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。

ハードウェアおよびソフトウェアの要件に関する詳細は、[デプロイメントのプランニング](#) を参照してください。

前提条件

- **cluster-admin** および Operator インストールのパーミッション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が少なくとも3つある。

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、コマンドラインインターフェイスで以下のコマンドを使用し、**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、openshift-storage namespace 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および割り当てガイドの [Red Hat OpenShift Data Foundation に専用のワーカーノードを使用する方法](#) の章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. チャンネルを **stable-4.15** に更新します。
 - 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、これは Operator のインストール時に作成されます。
6. **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) 更新を選択した場合、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 更新を選択した場合、OLM は更新要求を作成します。クラスタ管理者は、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。

7. **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。
8. **Install** をクリックします。

検証手順

- **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。
- 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 コンソールで、**Storage** に移動し、**Data Foundation** 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 -
```

- 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 は Operator Hub からインストールしておく。
- 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

- 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
```

- 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
```

- 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. 利用可能なストレージデバイスの検索

以下の手順を使用して、IBM Power 用に PV を作成する前に、OpenShift Data Foundation ラベル **cluster.ocs.openshift.io/openshift-storage=** でラベルを付けた 3 つ以上のワーカーノードのそれぞれのデバイス名を特定します。

手順

- OpenShift Data Foundation ラベルの付いたワーカーノードの名前のリストを表示し、確認します。

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

出力例:

```
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
```

- OpenShift Data Foundation リソースに使用される各ワーカーノードにログインし、OpenShift Container Platform をデプロイする際にアタッチした追加ディスクの名前を見つけます。

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

出力例:

```
$ 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
```

この例では、worker-0 の場合、利用可能な 500G のローカルデバイスは **sda**、**sd**c、**sde**、**sdg**、**sdi**、**sd**k、**sd**m、**sd**o です。

3. OpenShift Data Foundation で使用されるストレージデバイスを持つその他のすべてのワーカーノードについてこの手順を繰り返します。詳細は、[ナレッジベースア](#)ーティクルを参照してください。

2.6. IBM POWER での OPENSIFT DATA FOUNDATION クラスターの作成

この手順を使用して、OpenShift Data Foundation Operator のインストール後に OpenShift Data Foundation クラスターを作成します。

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。
- IBM Power でローカルストレージデバイスを使用するために、同じストレージタイプおよびサイズが各ノードに接続された 3 つ以上のワーカーノードを用意する (例: 200 GB SSD)。
- OpenShift Data Foundation ワーカーノードに OpenShift Container Storage ラベルが付けられている。

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

各ノードのストレージデバイスを特定するには、[利用可能なストレージデバイスの検索](#) 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. **openshift-local-storage** namespace 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。
3. **Local Storage** のインストールされた Operator をクリックします。
4. **Operator Details** ページで、**Local Volume** リンクをクリックします。
5. **Create Local Volume** をクリックします。
6. ローカルボリュームを設定するには、**YAML view** をクリックします。
7. 以下の YAML を使用して、ブロック PV の **LocalVolume** カスタムリソースを定義します。

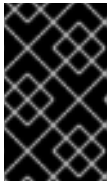
```
| apiVersion: local.storage.openshift.io/v1
| kind: LocalVolume
| metadata:
|   name: localblock
```

```

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

```

上記の定義は、**worker-0**、**worker-1**、および **worker-2** ノードから **sda** ローカルデバイスを選択します。**localblock** ストレージクラスが作成され、永続ボリュームが **sda** からプロビジョニングされます。



重要

環境に応じて `nodeSelector` の適切な値を指定します。デバイス名はすべてのワーカーノードで同一である必要があります。複数の `devicePaths` を指定することもできます。

8. **Create** をクリックします。
9. **diskmaker-manager** Pod および **Persistent Volumes** が作成されているかどうかを確認します。
 - a. **Pod** の場合
 - i. OpenShift Web コンソールの左側のペインから **Workloads** → **Pods** をクリックします。
 - ii. **Project** ドロップダウンリストから **openshift-local-storage** を選択します。
 - iii. LocalVolume CR の作成に使用した各ワーカーノードについて、**diskmaker-manager** Pod があるかどうかを確認します。
 - b. **永続ボリューム** の場合
 - i. OpenShift Web コンソールの左側のペインから **Storage** → **PersistentVolumes** をクリックします。
 - ii. **local-pv-*** 名で永続ボリュームを確認します。永続ボリュームの数は、localVolume CR の作成中にプロビジョニングされたワーカーノードの数とストレージデバイスの数と同じです。



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。

- 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。
 - Deployment type** オプションで **Full Deployment** を選択します。
 - Use an existing StorageClass** オプションを選択します。
 - LocalVolume のインストール時に使用した必要な **Storage Class** を選択します。
デフォルトでは **none** に設定されます。
 - 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。
 - 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。
 - 以下の接続の詳細を指定します。
 - ユーザー名
 - Password
 - サーバー名とポート
 - データベース名
 - 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。
 - Next** をクリックします。
- Capacity and nodes** ページで、以下を設定します。
 - 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ーと 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。

- c. 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。
- d. **Next** をクリックします。
14. 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。
- a. 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。
- i. 暗号化レベル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。
- ii. 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. **認証方法** 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 証明書**、**クライアント証明書**、および **クライアントの秘密鍵** 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 証明書**、**クライアント証明書**、および **クライアントの秘密鍵** 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的 **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. **クライアント証明書、CA 証明書、およびクライアント秘密鍵** 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。
- D. **Network** を選択します。
- I. Multus は IBM Power の OpenShift Data Foundation でサポートされていないため、**Default (OVN)** ネットワークを選択します。
- E. **Next** をクリックします。
- b. 転送中の暗号化を有効にするには、**In-transit encryption** を選択します。
 - i. **Network** を選択します。
 - ii. **Next** をクリックします。
15. **Data Protection** ページで、OpenShift Data Foundation のリージョナル 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。
16. **Review and create** ページで、以下を実行します。
- a. 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。
 - b. **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 モニターの低いアラート数の解決](#) 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** → **ocs-storagecluster-storagesystem** → **Resources** の順に移動します。
 - b. **StorageCluster** の **Status** が **Ready** になっており、その横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。

- 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。
 1. OpenShift Web 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。
 2. YAML タブで、**spec** セクションのキー **flexibleScaling** と **status** セクションの **flexibleScaling** を検索します。**flexible scaling** が true であり、**failureDomain** が host 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 デプロイメントの確認](#) 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。

第3章 内部モードの OPENSIFT DATA FOUNDATION デプロイメントの確認

このセクションを使用して、OpenShift Data Foundation が正しくデプロイされていることを確認します。

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します。

3.1. POD の状態の確認

OpenShift Data Foundation が正常にデプロイされているかどうかを判別するために、Pod の状態が **Running** であることを確認できます。

手順

1. OpenShift Web コンソールの左側のペインから **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。

コンポーネントごとに想定される Pod 数や、ノード数に合わせてこの数値がどのように変化するかなどの詳細は、[表3.1「OpenShift Data Foundation クラスターに対応する Pod」](#) を参照してください。

3. **Running** および **Completed** タブをクリックして、以下の Pod が実行中および完了状態にあることを確認します。

表3.1 OpenShift Data Foundation クラスターに対応する Pod

コンポーネント	対応する Pod
---------	----------

コンポーネント	対応する Pod
OpenShift Data Foundation Operator	<ul style="list-style-type: none"> ● ocs-operator-* (任意のストレージノードに 1Pod) ● ocs-metrics-exporter-* (任意のストレージノードに 1Pod) ● odf-operator-controller-manager-* (任意のストレージノードに 1Pod) ● odf-console-* (任意のストレージノードに 1Pod) ● csi-addons-controller-manager-* (任意のストレージノードに 1Pod)
Rook-ceph Operator	rook-ceph-operator-* (任意のストレージノードに 1Pod)
UX バックエンド	ux-backend-server-* (任意のストレージノード上の 1Pod)
Multicloud Object Gateway	<ul style="list-style-type: none"> ● noobaa-operator-* (任意のストレージノードに 1Pod) ● noobaa-core-* (任意のストレージノードに 1Pod) ● noobaa-db-pg-* (任意のストレージノードに 1Pod) ● noobaa-endpoint-* (任意のストレージノードに 1Pod)
MON	rook-ceph-mon-* (各ストレージノードに 3 Pod)
MGR	rook-ceph-mgr-* (任意のストレージノードに 1 Pod)
MDS	rook-ceph-mds-ocs-storagecluster-cephfilesystem-* (ストレージノードに分散する 2 Pod)
RGW	rook-ceph-rgw-ocs-storagecluster-cephobjectstore-* (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
CSI	<ul style="list-style-type: none"> ● cephfs <ul style="list-style-type: none"> ○ csi-cephfsplugin-* (各ストレージノードに 1 Pod) ○ csi-cephfsplugin-provisioner-* (ストレージノードに分散する 2 Pod) ● rbd <ul style="list-style-type: none"> ○ csi-rbdplugin-* (各ストレージノードに 1 Pod) ○ csi-rbdplugin-provisioner-* (ストレージノードに分散する 2 Pod)
rook-ceph-crashcollector	rook-ceph-crashcollector-* (各ストレージノードに 1 Pod)
OSD	<ul style="list-style-type: none"> ● rook-ceph-osd-* (各デバイス用に 1 Pod) ● rook-ceph-osd-prepare-* (各デバイス用に 1 Pod)

3.2. OPENSIFT DATA FOUNDATION クラスターの正常性の確認

手順

1. OpenShift Web コンソールで、**Storage → Data Foundation** をクリックします。
2. **Storage Systems** タブをクリックし、**ocs-storagecluster-storagesystem** をクリックします。
3. Overview タブの Block および File ダッシュボードの **Status card** で、**Storage Cluster** 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。
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、[OpenShift Data Foundation の監視](#) を参照してください。

3.3. MULTICLOUD OBJECT GATEWAY が正常であることの確認

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。

3.4. 特定のストレージクラスが存在することの確認

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**

第4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。

- ローカルストレージ 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

4.1. ローカルストレージ OPERATOR のインストール

以下の手順を使用して、ローカルストレージデバイスに OpenShift Data Foundation クラスターを作成する前に、Operator Hub からローカルストレージ 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. 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-local-storage** を選択します。
 - d. Approval Strategy に **Automatic** を選択します。
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。

4.2. 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。

ハードウェアおよびソフトウェアの要件に関する詳細は、[デプロイメントのプランニング](#) を参照してください。

前提条件

- **cluster-admin** および Operator インストールのパーミッション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が少なくとも3つある。



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、コマンドラインインターフェイスで以下のコマンドを使用し、**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、openshift-storage namespace 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および割り当てガイドの [Red Hat OpenShift Data Foundation に専用のワーカーノードを使用する方法](#) の章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. チャンネルを **stable-4.15** に更新します。
 - 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、これは Operator のインストール時に作成されます。
6. **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) 更新を選択した場合、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 更新を選択した場合、OLM は更新要求を作成します。クラスタ管理者は、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。
7. **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。
8. **Install** をクリックします。

検証手順

- **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。
- 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 コンソールで、**Storage** に移動し、**Data Foundation** が使用可能かどうかを確認します。

4.3. IBM POWER でのスタンドアロン 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。
- (ローカルストレージデバイスを使用したデプロイのみ) Local Storage Operator がインストールされている。

各ノードのストレージデバイスを特定するには、[利用可能なストレージデバイスの検索](#) 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. **openshift-local-storage** namespace 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。
3. **Local Storage** のインストールされた Operator をクリックします。
4. **Operator Details** ページで、**Local Volume** リンクをクリックします。
5. **Create Local Volume** をクリックします。
6. ローカルボリュームを設定するには、**YAML view** をクリックします。
7. 次の YAML を使用して、ファイルシステム PV の **LocalVolume** カスタムリソースを定義します。

```
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

```

上記の定義は、**worker-0**、**worker-1**、および **worker-2** ノードから **sda** ローカルデバイスを選択します。**localblock** ストレージクラスが作成され、永続ボリュームが **sda** からプロビジョニングされます。



重要

環境に応じて nodeSelector の適切な値を指定します。デバイス名はすべてのワーカーノードで同一である必要があります。複数の devicePaths を指定することもできます。

8. **Create** をクリックします。
9. 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。
10. **OpenShift Data Foundation Operator** をクリックした後、**Create StorageSystem** をクリックします。
11. **Backing storage** ページで、**Deployment type** に **Multicloud Object Gateway** を選択します。
12. **Backing storage type** の **Use an existing StorageClass** オプションを選択します。
 - a. **LocalVolume** のインストール時に使用した **Storage Class** を選択します。
13. **Next** をクリックします。
14. オプション: **Security** ページで、**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. **認証方法** 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 証明書**、**クライアント証明書**、および **クライアントの秘密鍵** 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 専用で固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 証明書**、**クライアント証明書**、および **クライアントの秘密鍵** を提供します。
 - **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的 **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. **クライアント証明書**、**CA 証明書**、および **クライアント秘密鍵** 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。
15. **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。

16. **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. **Storage Systems** タブをクリックし、**ocs-storagecluster-storagesystem** 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"> ● ocs-operator-* (任意のストレージノードに 1Pod) ● ocs-metrics-exporter-* (任意のストレージノードに 1Pod) ● odf-operator-controller-manager-* (任意のストレージノードに 1Pod) ● odf-console-* (任意のストレージノードに 1Pod) ● csi-addons-controller-manager-* (任意のストレージノードに 1Pod)
Rook-ceph Operator	<p>rook-ceph-operator-*</p> <p>(任意のストレージノードに 1Pod)</p>

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

第5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6章 OPENSIFT DATA FOUNDATION のアンインストール

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

OpenShift Data Foundation を内部モードでアンインストールするには、[Uninstalling OpenShift Data Foundation](#) のナレッジベース記事を参照してください。