



Red Hat OpenShift Data Foundation 4.12

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
Container Storage のデプロイ手順

Red Hat OpenShift Data Foundation 4.12 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 Container Storage のデプロイ手順

法律上の通知

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.

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. 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. MULTUS ネットワークの作成 [テクノロジープレビュー] | 13 |
| 2.6. ベアメタルでの OPENSIFT DATA FOUNDATION クラスターの作成 | 15 |
| 2.7. OPENSIFT DATA FOUNDATION デプロイメントの確認 | 20 |
| 第3章 スタンドアロンの MULTICLOUD OBJECT GATEWAY のデプロイ | 26 |
| 3.1. ローカルストレージ OPERATOR のインストール | 26 |
| 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 | 28 |
| 第4章 OPENSIFT DATA FOUNDATION のアンインストール | 33 |
| 4.1. 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、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) 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 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法が選択されている場合は、[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 認証方式を選択している場合は、[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. [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. [ベアメタルで 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

クラスターは少なくとも 3 つの OpenShift Container Platform ワーカーノードで設定されており、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。

- 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。

ディザスタリカバリーの要件 テクノロジーレビュー

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。

Arbiter ストレッチクラスタの要件 [テクノロジーレビュー]

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つのゾーンに展開されます。これはテクノロジーレビュー機能であり、現時点では OpenShift Container Platform オンプレミスおよび同じデータセンターで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またがる展開にはお勧めできません。代わりに、Metro-DR を、低遅延ネットワークを備えた複数のデータセンターに展開されたデータ損失のない DR ソリューションの最初のオプションとして検討してください。

詳細な要件と手順については、[ストレッチクラスタ用の OpenShift Data Foundation の設定](#) 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 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 クラスタに追加することができます。Arbiter クラスタでは、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](#) セクションおよび [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. [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. [ベアメタルで OpenShift Data Foundation クラスターを作成します。](#)

2.1. ローカルストレージ OPERATOR のインストール

ローカルストレージデバイスに Red Hat 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. チャンネルを **4.12** または **stable** のいずれかにして更新します。
 - 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** 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。各ノードには1つのディスクが含まれ、3つのディスク(PV)が必要です。ただし、1つのPVはデフォルトで最終的に使用されません。これは想定される動作です。
- その他のリソース要件については、[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、**openshift-storage** namespace の空のノードセクターを指定できます (この場合、**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** 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. Channel を **stable-4.12** 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-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。
 - 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。

- 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 -
```

- 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. MULTUS ネットワークの作成 [テクノロジープレビュー]

OpenShift Container Platform は、Multus CNI プラグインを使用して CNI プラグインのチェーンを許可します。クラスターのインストール中にデフォルトの Pod ネットワークを設定できます。デフォルトのネットワークは、クラスターのすべての通常のネットワークトラフィックを処理します。

利用可能な CNI プラグインに基づいて追加のネットワークを定義し、1つまたは複数のネットワークを Pod に割り当てることができます。追加のネットワークを Pod に割り当てするには、インターフェイスの割り当て方法を定義する設定を作成する必要があります。

NetworkAttachmentDefinition (NAD) カスタムリソース (CR) を使用して、各インターフェイスを指定します。それぞれの NetworkAttachmentDefinition 内の CNI 設定は、インターフェイスの作成方法を定義します。

OpenShift Data Foundation は、macvlan と呼ばれる CNI プラグインを使用します。macvlan ベースの追加ネットワークを作成することで、ホスト上の Pod が物理ネットワークインターフェイスを使用して他のホストやそれらのホストの Pod と通信できます。macvlan ベースの追加ネットワークに割り当てられる各 Pod には固有の MAC アドレスが割り当てられます。



重要

Multus のサポートはテクノロジープレビュー機能としてのみサポートされ、ベアメタルおよび VMWare デプロイメントでテストされます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

2.5.1. ネットワーク接続定義の作成

Multus を使用するには、正しいネットワーク設定ですでに機能するクラスターが必要です。[Recommended network configuration and requirements for a Multus configuration](#) を参照してください。新規に作成された NetworkAttachmentDefinition (NAD) は、Storage Cluster のインストール時に選択できます。これは、Storage Cluster の前に作成する必要のある理由です。

Storage Cluster のインストール中に、新しく作成された **NetworkAttachmentDefinition** (NAD) を選択できます。これが、Storage Cluster を作成する前に NAD を作成する必要がある理由です。

プランニングガイドで説明されているように、作成する Multus ネットワークは、OpenShift Data Foundation トラフィックで利用可能なネットワークインターフェイスの数によって異なります。すべてのストレージトラフィックを2つのインターフェイス (デフォルトの OpenShift SDN に使用されるインターフェイス1つ) に分割するか、ストレージストレージトラフィック (パブリック) およびストレージレプリケーショントラフィック (プライベートまたはクラスター) にさらに分割することもできます。

以下は、同じインターフェイス上のすべてのストレージトラフィック (パブリックおよびクラスター) の **NetworkAttachmentDefinition** の例です。すべてのスケジュール可能なノードに1つの追加インターフェイスが必要になります (OpenShift のデフォルト: 個別のネットワークインターフェイス上の OpenShift のデフォルト)。

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



注記

すべてのネットワークインターフェイス名は、Multus ネットワークに接続されているすべてのノードで同じである必要があります (例: **ocs-public-cluster** の場合は **ens2**)。

以下は、個別の Multus ネットワーク上のストレージトラフィックの **NetworkAttachmentDefinition** の例になります。これは、クライアントストレージトラフィックのパブリックおよびレプリケーショントラフィック用のクラスターです。Object Storage Device (OSD) Pod をホストする OpenShift ノードに 2 つの追加インターフェイスと、他のスケジュール可能なすべてのノードで 1 つの追加インターフェイス (OpenShift デフォルト SDN) が必要です。

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

NetworkAttachmentDefinition の例:

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



注記

すべてのネットワークインターフェイス名は、Multus ネットワークにアタッチされたすべてのノードで同じである必要があります (つまり、**ocs-public** の場合は **ens2**、**ocs-cluster** の場合は **ens3** です)。

2.6. 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。

- multus サポートのテクノロジープレビュー機能を使用する必要がある場合には、デプロイメントの前に、後でクラスターにアタッチされるネットワーク接続定義 (NAD) を作成する必要があります。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 [ネットワーク接続定義の作成](#) を参照してください。

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. **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、[ローカルストレージ Operator のインストール](#) で説明されているように手順に従います。

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ません。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** の利用可能な一覧から、**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。

- 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** 一覧には、ストレージクラスに基づくノードが表示されます。
- 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。

6. 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。

b. 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。

c. 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。

- i. **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。
- ii. **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** を展開して、**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** を展開して、**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。
- iii. **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。
- A. 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。
 - 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. **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。
 - 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 エントリがない場合に使用します。たとえば、**kmip_all_<port>.ciphertrustmanager.local** などです。
- iv. **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. **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。
8. **Create StorageSystem** をクリックします。

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。

2.7. 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。

2.7.1. 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一覧表示します。

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2.1「OpenShift Data Foundation クラスターに対応する Pod」](#) を参照してください。

3. 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。

表2.1 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 |
|------------------------------------|--|
| 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> |
| 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 | <p>rook-ceph-mon-*</p> <p>(ストレージノードに分散する 3 Pod)</p> |

| コンポーネント | 対応する 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 1 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-ocs-deviceset-* (各デバイス用に 1 Pod) |

2.7.2. OpenShift 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. **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。

2.7.3. 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は、[OpenShift Data Foundation の監視](#) を参照してください。

2.7.4. 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**

2.7.5. 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。

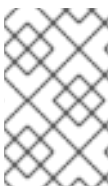
```
$ oc get storagecluster ocs-storagecluster \
-n openshift-storage \
-o=jsonpath='{.spec.network}'
```

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\.cnf\.io/network-status}'
```

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\.cnf\.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。

- 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

3.1. ローカルストレージ OPERATOR のインストール

ローカルストレージデバイスに Red Hat 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. チャンネルを **4.12** または **stable** のいずれかにして更新します。
 - 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** 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。各ノードには1つのディスクが含まれ、3つのディスク(PV)が必要です。ただし、1つのPVはデフォルトで最終的に使用されません。これは想定される動作です。
- その他のリソース要件については、[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、**openshift-storage** namespace の空のノードセクターを指定できます (この場合、**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** 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. Channel を **stable-4.12** 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-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。
 - 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。
 - 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、[ローカルストレージ Operator のインストール](#)で説明されているように手順に従います。

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せん。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** の利用可能な一覧から、**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。 Volume Mode でファイルシステム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** 一覧には、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. **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** を展開して、**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** を展開して、**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. **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。
7. **Review and create** ページで、設定の詳細を確認します。

設定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一覧表示します。

| 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) |

第4章 OPENSIFT DATA FOUNDATION のアンインストール

4.1. 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。