



# Red Hat OpenShift Data Foundation 4.9

## IBM Z インフラストラクチャーを使用した OpenShift Data Foundation のデプロイ

IBM Z 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 Red Hat OpenShift  
Data Foundation のデプロイ手順



# Red Hat OpenShift Data Foundation 4.9 IBM Z インフラストラクチャーを使用した OpenShift Data Foundation のデプロイ

---

IBM Z 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 Red Hat OpenShift Data Foundation のデプロイ手順

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Deploying\_OpenShift\_Data\_Foundation\_using\_IBM\_Z\_infrastructure.ent file |.

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

Red Hat OpenShift Data Foundation をインストールして IBM Z 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方法については、本書を参照してください。 While this document refers only to IBM Z, all information in it also applies to LinuxONE.

---

## 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
前書き .....	5
<b>第1章 OPENSIFT DATA FOUNDATION のデプロイの準備 .....</b>	<b>6</b>
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
1.2. VAULT でのキー値のバックエンドパスおよびポリシーの有効化	6
<b>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 .....</b>	<b>8</b>
2.1. RED HAT OPENSIFT DATA FOUNDATION OPERATOR のインストール	8
2.2. ローカルストレージ OPERATOR のインストール	9
2.3. 利用可能なストレージデバイスの検索 (オプション)	10
2.4. IBM Z での OPENSIFT DATA FOUNDATION クラスターの作成	12
<b>第3章 内部接続デバイスモードの OPENSIFT DATA FOUNDATION デプロイメントの確認 .....</b>	<b>18</b>
3.1. POD の状態の確認	18
3.2. OPENSIFT DATA FOUNDATION クラスターの正常性の確認	20
3.3. MULTICLOUD OBJECT GATEWAY が正常であることの確認	20
3.4. OPENSIFT DATA FOUNDATION 固有のストレージクラスが存在することの確認	21
<b>第4章 OPENSIFT DATA FOUNDATION のアンインストール .....</b>	<b>22</b>
4.1. 内部接続デバイスモードの OPENSIFT DATA FOUNDATION のアンインストール	22
4.1.1. 「ローカルストレージ Operator の設定の削除」を参照してください。	30
4.2. OPENSIFT DATA FOUNDATION からのモニタリングスタックの削除	35
4.3. OPENSIFT DATA FOUNDATION からの OPENSIFT CONTAINER PLATFORM レジストリーの削除	38
4.4. OPENSIFT DATA FOUNDATION からのクラスターロギング OPERATOR の削除	40



## 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、[弊社](#)の CTO、Chris Wright の[メッセージ](#)を参照してください。

## RED HAT ドキュメントへのフィードバック (英語のみ)

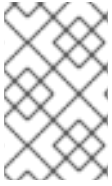
弊社のドキュメントについてのご意見をお聞かせください。ドキュメントの改善点があれば、ぜひお知らせください。フィードバックをお寄せいただくには、以下をご確認ください。

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
  1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上端に **Feedback** ボタンがあることを確認してください。
  2. マウスカーソルで、コメントを追加する部分を強調表示します。
  3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
  4. 表示される手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla の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関するご意見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入してください。
  4. **Submit Bug** をクリックします。



## 前書き

Red Hat OpenShift Data Foundation 4.9 は、接続環境または非接続環境での既存の Red Hat OpenShift Container Platform (RHOCP) IBM Z 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。



### 注記

IBM Z では、OpenShift Data Foundation の内部クラスターのみがサポートされます。デプロイメント要件についての詳細は、『[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、お使いの環境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](#)

## 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。この方法では、ベースサービスを内部でプロビジョニングします。その後、すべてのアプリケーションは追加のストレージクラスにアクセスできます。

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](#)について参照してください。

外部の鍵管理システム (KMS) で、以下を実行します。

- トークンのあるポリシーが存在し、Vault のキー値のバックエンドパスが有効にされていることを確認します。[Vault でのキー値のバックエンドパスおよびポリシーの有効化](#)について参照してください。
- 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。

上記を処理したら、指定した順序で以下の手順を実行します。

1. [Red Hat OpenShift Data Foundation Operator をインストール](#)します。
2. [ローカルストレージ Operator のインストール](#)。
3. [利用可能なストレージデバイスを見つけます](#)。
4. [IBM Z に OpenShift Data Foundation クラスターサービスを作成](#)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 3 つ以上の OpenShift Container Platform ワーカーノードで構成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。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](#)のセクションを参照してください。

- ストレージノードの場合は、FCP ストレージデバイスが必要です。

### 1.2. VAULT でのキー値のバックエンドパスおよびポリシーの有効化

#### 前提条件

- Vault への管理者アクセス。
- 注: 後に変更することはできないため、命名規則に基づいてバックエンド **path** として一意のパス名を選択します。

## 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 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 です。

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

2. 以下のコマンドを使用して、シークレットでの書き込み操作または削除操作の実行を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
```

## 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

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するオプションが提供されます。このデプロイメント方法に従って、ローカルストレージを使用して OpenShift Container Platform アプリケーションの永続ボリュームをバックアップします。

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 Z インフラストラクチャーに OpenShift Data Foundation をデプロイします。

### 2.1. 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が少なくとも3つある。
- 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、コマンドラインインターフェー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 [How to use dedicated worker nodes for Red Hat OpenShift Data Foundation](#) 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、または **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-4.9** 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。



### 注記

すべてのデフォルト設定を使用することが推奨されます。これを変更すると、予期しない動作が発生する可能性があります。変更後にどうなるのかを認識している場合に限り変更します。

### 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ースに表示されます。このポップアップから **Web コン  
ソールのリフレッシュ** 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、**Operators** に移動し、**OpenShift Data Foundation** が利用可能かどう  
かを確認します。



### 重要

OpenShift Data Foundation Operator のインストール後に console プラグインオプション  
が自動的に有効にされていない場合は、これを有効にする必要があります。

console プラグインを有効にする方法の詳細は、「[Red Hat OpenShift Data Foundation  
console プラグインの有効化](#)」を参照してください。

## 2.2. 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. channel を ana または **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。

### 2.3. 利用可能なストレージデバイスの検索（オプション）

この手順は追加の情報であり、ストレージクラスターの作成時にディスクは自動的に検出されるため、省略できます。以下の手順を使用して、IBM Z 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

1. **OpenShift Data Foundation** ラベルの付いたワーカーノードの名前の一覧を表示し、確認します。

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

出力例:

```

NAME      STATUS  ROLES  AGE   VERSION
bmworker01 Ready   worker 6h45m v1.16.2
bmworker02 Ready   worker 6h45m v1.16.2
bmworker03 Ready   worker 6h45m v1.16.2

```

2.

**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、利用可能な各 **raw** ブロックデバイスの一意的 **by-id** デバイス名を見つけます。

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

出力例:

```

$ oc debug node/bmworker01
Starting pod/bmworker01-debug ...
To use host binaries, run `chroot /host`
Pod IP: 10.0.135.71
If you don't see a command prompt, try pressing enter.
sh-4.2# chroot /host
sh-4.4# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0                7:0  0  500G  0 loop
sda                  8:0  0  120G  0 disk
|-sda1                8:1  0   384M  0 part /boot
`-sda4                8:4  0  119.6G  0 part
`-coreos-luks-root-nocrypt 253:0  0  119.6G  0 dm  /sysroot
sdb                  8:16  0  500G  0 disk

```

この例では、**bmworker01** について利用可能なローカルデバイスは **sdb** です。

3.

手順 2 で選択した各デバイスの一意的 ID を特定します。

```

sh-4.4#ls -l /dev/disk/by-id/ | grep sdb
lrwxrwxrwx. 1 root root 9 Feb  3 16:49 scsi-360050763808104bc2800000000000259 ->
../sdb
lrwxrwxrwx. 1 root root 9 Feb  3 16:49 scsi-SIBM_2145_00e020412f0aXX00 -> ../sdb
lrwxrwxrwx. 1 root root 9 Feb  3 16:49 scsi-0x60050763808104bc2800000000000259 ->
../sdb

```

上記の例で、ローカルデバイス **sdb** の ID は以下になります。

```
scsi-0x60050763808104bc2800000000000259
```

4.

上記の手順を繰り返し、**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ノードのデバイスID を特定します。詳細は、[ナレッジベースアールティクル](#) を参照してください。

## 2.4. IBM Z での OPENSIFT DATA FOUNDATION クラスターの作成

以下の手順を使用して、IBM Z 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た OpenShift Data Foundation のインストールの要件](#)についての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 Z または LinuxONE 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 つのワーカーノードが必要です（例：200 GB）。

### 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. Create a new StorageClass using the local storage devices オプションを選択します。
  - b. Advanced を展開し、Deployment type オプションで Full Deployment を選択します。
  - c. 次へをクリックします。



**重要**

インストールされていない場合に、ローカルストレージ Operator をインストールすることを求めるプロンプトが出されます。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。



### 重要

- 柔軟なスケーリング機能は、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。

柔軟なスケーリングについての詳細は、ストレージのスケーリングガイドの [Add capacity using YAML](#) セクションを参照してください。

- 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	デフォルトではブロックが選択されます。
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：セキュリティ および ネットワーク 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。
  - 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.  
**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** はデフォルトで **Vault** に設定されます。
    - ii.  
**Vault Service Name**、Vault サーバーの **Host Address** ('https://<hostname or ip>')、**Port** 番号および **Token** を入力します。

- iii.
  - 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。
    - A. **OpenShift Data Foundation** 専用かつ特有のキー値のシークレットパスを **Backend Path** に入力します。
    - B. オプション : **TLS Server Name** および **Vault Enterprise Namespace** を入力します。
    - C. それぞれの **PEM** でエンコードされた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。
    - D. **保存** をクリックします。
  - d. **Multus** は **IBM Z インフラストラクチャーの OpenShift Data Foundation** でサポートされていないため、**Default (SDN)** を選択します。
  - e. **Next** をクリックします。
7. **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** をクリックして以前の設定ページに戻ります。
  - b. **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** → **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。**

- 柔軟な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、柔軟な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、[Verifying your OpenShift Data Foundation deployment](#) 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 内部接続デバイスモードの 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。

#### 3.1. 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。

各コンポーネントについて予想される Pod 数や、これがノード数によってどのように異なるかの詳細は、[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 タブおよび Completed タブをクリックして、以下の Pod が Running 状態および Completed 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
OpenShift Data Foundation Operator	<ul style="list-style-type: none"> <li>● <b>ocs-operator-*</b> (任意のワーカーノードに 1 Pod)</li> <li>● <b>ocs-metrics-exporter-*</b> (任意のワーカーノードに 1 Pod)</li> <li>● <b>odf-operator-controller-manager-*</b> (任意のワーカーノードに 1 Pod)</li> <li>● <b>odf-console-*</b> (任意のワーカーノードに 1 Pod)</li> </ul>

コンポーネント	対応する Pod
Rook-ceph Operator	<b>rook-ceph-operator-*</b> (任意のワーカーノードに 1 Pod)
Multicloud Object Gateway	<ul style="list-style-type: none"> <li>● <b>noobaa-operator-*</b> (任意のワーカーノードに 1 Pod)</li> <li>● <b>noobaa-core-*</b> (任意のストレージノードに 1 Pod)</li> <li>● <b>noobaa-db-pg-*</b> (任意のストレージノードに 1 Pod)</li> <li>● <b>noobaa-endpoint-*</b> (任意のストレージノードに 1 Pod)</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 1 Pod)
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> (各ワーカーノードに 1 Pod)</li> <li>○ <b>csi-cephfsplugin-provisioner-*</b> (ワーカーノードに分散する 2 Pod)</li> </ul> </li> <li>● <b>rbd</b> <ul style="list-style-type: none"> <li>○ <b>csi-rbdplugin-*</b> (各ワーカーノードに 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>

コンポーネント	対応する Pod
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> <li>● <b>rook-ceph-osd-prepare-ocs-deviceset-*</b> (各デバイス用に 1 Pod)</li> </ul>

### 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 → OpenShift 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** カードで、**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。

Block および File 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。

### 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 → OpenShift 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** カードで、**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。

オブジェクトサービス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。

### 3.4. OPENSIFT DATA FOUNDATION 固有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 OPENSIFT DATA FOUNDATION のアンインストール

### 4.1. 内部接続デバイスモードの OPENSIFT DATA FOUNDATION のアンインストール

このセクションの手順に従って OpenShift Data Foundation をアンインストールします。

#### アノテーションのアンインストール

**Storage Cluster** のアノテーションは、アンインストールプロセスの動作を変更するために使用されます。アンインストールの動作を定義するために、ストレージクラスターに以下の 2 つのアノテーションが導入されました。

- `uninstall.ocs.openshift.io/cleanup-policy: delete`
- `uninstall.ocs.openshift.io/mode: graceful`

以下の表は、これらのアノテーションで使用できる各種値に関する情報を示しています。

表4.1 `uninstall.ocs.openshift.io` でアノテーションの説明をアンインストールする

アノテーション	値	デフォルト	動作
<code>cleanup-policy</code>	削除	はい	Rook は物理ドライブおよび <b>DataDirHostPath</b> をクリーンアップします。
<code>cleanup-policy</code>	Retain (保持)	いいえ	Rook は物理ドライブおよび <b>DataDirHostPath</b> をクリーンアップしません。
<code>mode</code>	<code>graceful</code>	はい	Rook および NooBaa は、管理者/ユーザーが Persistent Volume Claim (PVC) および Object Bucket Claim (OBC) を削除するまで、アンインストールプロセスを一時停止します。

アノテーション	値	デフォルト	動作
mode	forced	いいえ	Rook および NooBaa は、Rook および NooBaa を使用してプロビジョニングされた PVC/OBC がそれぞれ存在している場合でもアンインストールを続行します。

アノテーションの値を編集して、クリーンアップポリシーまたはアンインストールモードを変更します。

```
$ oc -n openshift-storage annotate storagecluster ocs-storagecluster
uninstall.ocs.openshift.io/cleanup-policy="retain" --overwrite
```

```
$ oc -n openshift-storage annotate storagecluster ocs-storagecluster
uninstall.ocs.openshift.io/mode="forced" --overwrite
```

両方のコマンドで予期される出力:

```
storagecluster.ocs.openshift.io/ocs-storagecluster annotated
```

#### 前提条件

- **OpenShift Data Foundation** クラスターの状態が正常であることを確認します。リソースまたはノードの不足により一部の Pod が正常に終了されないと、アンインストールプロセスに失敗する可能性があります。クラスターが状態が正常でない場合は、**OpenShift Data Foundation** をアンインストールする前に Red Hat カスタマーサポートにお問い合わせください。
- アプリケーションが **OpenShift Data Foundation** によって提供されるストレージクラスを使用して永続ボリューム要求 (PVC) またはオブジェクトバケット要求 (OBC) を使用していないことを確認します。
- カスタムリソース（カスタムストレージクラス、cephblockpools など）が管理者によって作成された場合、それらを消費したリソースを削除した後に管理者によって削除される必要が

あります。

## 手順

1. **OpenShift Data Foundation** を使用しているボリュームスナップショットを削除します。

- a. すべての **namespace** からボリュームスナップショットを一覧表示します。

```
$ oc get volumesnapshot --all-namespaces
```

- b. 直前のコマンドの出力から、**OpenShift Data Foundation** を使用しているボリュームスナップショットを特定し、削除します。

```
$ oc delete volumesnapshot <VOLUME-SNAPSHOT-NAME> -n <NAMESPACE>
```

<VOLUME-SNAPSHOT-NAME>

ボリュームスナップショットの名前です。

<NAMESPACE>

プロジェクトの namespace です。

2. **OpenShift Data Foundation** を使用している PVC および OBC を削除します。

デフォルトのアンインストールモード(**graceful**)では、アンインストーラーは **OpenShift Data Foundation** を使用するすべての PVC および OBC が削除されるまで待機します。

PVC を削除せずに **Storage Cluster** を削除する場合は、アンインストールモードのアンインストールを **forced** に設定し、この手順を省略できます。これを行うと、孤立した PVC および OBC がシステムに作成されます。

- a. **OpenShift Data Foundation** を使用して、**OpenShift Container Platform** モニタリングスタック PVC を削除します。

「[OpenShift Data Foundation からのモニタリングスタックの削除](#)」を参照してください。

- b. **OpenShift Data Foundation** を使用して、**OpenShift Container Platform** レジストリー PVC を削除します。

### OpenShift Data Foundation からの OpenShift Container Platform レジストリーの削除

- c. **OpenShift Data Foundation** を使用して、**OpenShift Container Platform** ロギング PVC を削除します。

### OpenShift Data Foundation からのクラスターロギング Operator の削除

- d. **OpenShift Data Foundation** を使用してプロビジョニングした PVC および OBC を削除します。

- 以下に、**OpenShift Data Foundation** を使用してプロビジョニングされる PVC および OBC を特定するサンプルスクリプトを示します。このスクリプトは、**OpenShift Data Foundation** によって内部で使用される PVC を無視します。

```
#!/bin/bash

RBD_PROVISIONER="openshift-storage.rbd.csi.ceph.com"
CEPHFS_PROVISIONER="openshift-storage.cephfs.csi.ceph.com"
NOOBAA_PROVISIONER="openshift-storage.noobaa.io/obc"
RGW_PROVISIONER="openshift-storage.ceph.rook.io/bucket"

NOOBAA_DB_PVC="noobaa-db"
NOOBAA_BACKINGSTORE_PVC="noobaa-default-backing-store-noobaa-pvc"

# Find all the OCS StorageClasses
OCS_STORAGECLASSES=$(oc get storageclasses | grep -e
"$RBD_PROVISIONER" -e "$CEPHFS_PROVISIONER" -e
"$NOOBAA_PROVISIONER" -e "$RGW_PROVISIONER" | awk '{print $1}')

# List PVCs in each of the StorageClasses
for SC in $OCS_STORAGECLASSES
do
    echo
    "=====
=="
    echo "$SC StorageClass PVCs and OBCs"
    echo
    "=====
=="
    oc get pvc --all-namespaces --no-headers 2>/dev/null | grep $SC | grep -v -e
"$NOOBAA_DB_PVC" -e "$NOOBAA_BACKINGSTORE_PVC"
```

```
oc get obc --all-namespaces --no-headers 2>/dev/null | grep $SC
echo
done
```



#### 注記

クラウドプラットフォームの **RGW\_PROVISIONER** を省略します。

- **OBC** を削除します。

```
$ oc delete obc <obc-name> -n <project-name>
```

<obc-name>

**OBC** の名前です。

<project-name>

プロジェクトの名前です。

- **PVC** を削除します。

```
$ oc delete pvc <pvc-name> -n <project-name>
```

<pvc-name>

**PVC** の名前です。

<project-name>

プロジェクトの名前です。



#### 注記

クラスターに作成されているカスタムバックングストア、バケットクラスなどを削除していることを確認します。

3. **Storage System** オブジェクトを削除し、関連付けられたリソースが削除されるのを待機します。

```
$ oc delete -n openshift-storage storagesystem --all --wait=true
```

4.

**uninstall.ocs.openshift.io/cleanup-policy** が **delete (default)** に設定されている場合にクリーンアップ Pod の有無を確認し、それらのステータスが **Completed** していることを確認します。

```
$ oc get pods -n openshift-storage | grep -i cleanup
```

出力例:

NAME	READY	STATUS	RESTARTS	AGE
cluster-cleanup-job-<xx>	0/1	Completed	0	8m35s
cluster-cleanup-job-<yy>	0/1	Completed	0	8m35s
cluster-cleanup-job-<zz>	0/1	Completed	0	8m35s

5.

**/var/lib/rook** ディレクトリーが空であることを確認します。このディレクトリーは空になるのは、**uninstall.ocs.openshift.io/cleanup-policy** アノテーションが **delete (デフォルト)** に設定されている場合に限られます。

```
$ for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o jsonpath='{.items[*].metadata.name}'); do oc debug node/${i} -- chroot /host ls -l /var/lib/rook; done
```

6.

暗号化がインストール時に有効にされている場合は、すべての **OpenShift Data Foundation** ノードの **OSD** デバイスから **dm-crypt** で管理される **device-mapper** マッピングを削除します。

a.

デバッグ Pod を作成し、ストレージノードのホストに対して **chroot** を作成します。

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

```
$ chroot /host
```

```
<node-name>
```

ノードの名前です。

b.

デバイス名を取得し、**OpenShift Data Foundation** デバイスについてメモします。

```
$ dmsetup ls
```

出力例:

```
ocs-deviceset-0-data-0-57snx-block-dmccrypt (253:1)
```

c.

マップ済みデバイスを削除します。

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-0-data-0-57snx-block-dmccrypt
```

### 重要

権限が十分でないため、コマンドがスタックした場合には、以下のコマンドを実行します。

- CTRL+Z を押して上記のコマンドを終了します。

- スタックしたプロセスの PID を検索します。

```
$ ps -ef | grep crypt
```

- kill コマンドを使用してプロセスを終了します。

```
$ kill -9 <PID>
```

<PID>

プロセス ID です。

- デバイス名が削除されていることを確認します。

```
$ dmsetup ls
```

7.

namespace を削除し、削除が完了するまで待機します。openshift-storage がアクティブなプロジェクトである場合は、別のプロジェクトに切り替える必要があります。



以下は例になります。

```
$ oc project default
```

```
$ oc delete project openshift-storage --wait=true --timeout=5m
```

以下のコマンドが **NotFound** エラーを返すと、プロジェクトが削除されます。

```
$ oc get project openshift-storage
```



#### 注記

OpenShift Data Foundation のアンインストール時に、namespace が完全に削除されず、Terminating 状態のままである場合は、「[Troubleshooting and deleting remaining resources during Uninstall](#)」の手順を実行して namespace の終了をブロックしているオブジェクトを特定します。

8. ローカルストレージデバイスを使用して OpenShift Data Foundation をデプロイした場合は、ローカルストレージ Operator 設定を削除します。「[ローカルストレージ Operator の設定の削除](#)」を参照してください。

9. ストレージノードのラベルを解除します。

```
$ oc label nodes --all cluster.ocs.openshift.io/openshift-storage-
```

```
$ oc label nodes --all topology.rook.io/rack-
```

10. ノードにテイントのマークが付けられている場合に OpenShift Data Foundation テイントを削除します。

```
$ oc adm taint nodes --all node.ocs.openshift.io/storage-
```

11. OpenShift Data Foundation を使用してプロビジョニングした永続ボリューム (PV) がすべて削除されていることを確認します。Released 状態のままの PV がある場合は、これを削除します。

```
$ oc get pv
```

```
$ oc delete pv <pv-name>
```

<pv-name>

Pod の名前です。

12.

**CustomResourceDefinitions** を削除します。

```
$ oc delete crd backingstores.noobaa.io bucketclasses.noobaa.io
cephblockpools.ceph.rook.io cephclusters.ceph.rook.io cephfilesystems.ceph.rook.io
cephnfses.ceph.rook.io cephobjectstores.ceph.rook.io cephobjectstoreusers.ceph.rook.io
noobaas.noobaa.io ocsinitializations.ocs.openshift.io storageclusters.ocs.openshift.io
cephclients.ceph.rook.io cephobjectrealms.ceph.rook.io cephobjectzonegroups.ceph.rook.io
cephobjectzones.ceph.rook.io cephrbdmirrors.ceph.rook.io storagesystems.odf.openshift.io --
wait=true --timeout=5m
```

13.

**OpenShift Container Platform Web** コンソールで、**OpenShift Data Foundation** が完全にアンインストールされていることを確認するには、以下を実行します。

a.

ストレージ をクリックします。

b.

**OpenShift Data Foundation** が **Storage** に表示されていないことを確認します。

#### 4.1.1. 「ローカルストレージ Operator の設定の削除」を参照してください。

ローカルストレージデバイスを使用して **OpenShift Data Foundation** をデプロイした場合にのみこのセクションの説明を使用します。



#### 注記

**OpenShift Data Foundation** デプロイメントで **localvolume** リソースのみを使用する場合は、直接、手順 8 に移動します。

#### 手順

1.

**LocalVolumeSet** および **OpenShift Data Foundation** で使用される対応する **StorageClassName** を特定します。

```
$ oc get localvolumesets.local.storage.openshift.io -n openshift-local-storage
```

2.

**LocalVolumeSet を提供する StorageClass に変数 SC を設定します。**

```
$ export SC="<StorageClassName>"
```

3.

後にクリーンアップするデバイスを一覧表示し、これをメモします。ディスクのデバイス ID を一覧表示するには、ここで説明されている手順に従います。[利用可能なストレージデバイスの検索](#)について参照してください。

出力例:

```
/dev/disk/by-id/scsi-360050763808104bc28000000000000eb
/dev/disk/by-id/scsi-360050763808104bc28000000000000ef
/dev/disk/by-id/scsi-360050763808104bc28000000000000f3
```

4.

**LocalVolumeSet を削除します。**

```
$ oc delete localvolumesets.local.storage.openshift.io <name-of-volumeset> -n openshift-
local-storage
```

5.

**指定された StorageClassName のローカルストレージ PV を削除します。**

```
$ oc get pv | grep $SC | awk '{print $1}' | xargs oc delete pv
```

6.

**StorageClassName を削除します。**

```
$ oc delete sc $SC
```

7.

**LocalVolumeSet によって作成されるシンボリックリンクを削除します。**

```
[[ ! -z $SC ]] && for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o
jsonpath='{.items[*].metadata.name}'); do oc debug node/${i} -- chroot /host rm -rfv
/mnt/local-storage/${SC}/; done
```

8.

**LocalVolumeDiscovery を削除します。**

```
$ oc delete localvolumediscovery.local.storage.openshift.io/auto-discover-devices -n
openshift-local-storage
```

9.

**LocalVolume** リソースを削除します (ある場合)。

以下の手順を使用して、現行または直前の **OpenShift Data Foundation** バージョンで **PV** のプロビジョニングに使用した **LocalVolume** リソースを削除します。また、これらのリソースがクラスターの他のテナントで使用されていないことを確認します。

ローカルボリュームごとに、以下を実行します。

a.

**LocalVolume** および **OpenShift Data Foundation** で使用される対応する **StorageClassName** を特定します。

```
$ oc get localvolume.local.storage.openshift.io -n openshift-local-storage
```

b.

変数 **LV** を **LocalVolume** の名前に設定し、変数 **SC** を **StorageClass** の名前に設定します。

以下は例になります。

```
$ LV=local-block  
$ SC=localblock
```

c.

後にクリーンアップするデバイスを一覧表示し、これをメモします。

```
$ oc get localvolume -n openshift-local-storage $LV -o jsonpath='{  
.spec.storageClassDevices[].devicePaths[]}'{"\n"}
```

出力例:

```
/dev/sdb  
/dev/sdc  
/dev/sdd  
/dev/sde
```

d.

ローカルボリュームリソースを削除します。

```
$ oc delete localvolume -n openshift-local-storage --wait=true $LV
```

e.

残りの PV および StorageClass が存在する場合はこれらを削除します。

```
$ oc delete pv -l storage.openshift.com/local-volume-owner-name=${LV} --wait --
timeout=5m
$ oc delete storageclass $SC --wait --timeout=5m
```

f.

そのリソースのストレージノードからアーティファクトをクリーンアップします。

```
$ [[ ! -z $SC ]] && for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o
jsonpath='{.items[*].metadata.name}'); do oc debug node/${i} -- chroot /host rm -rfv
/mnt/local-storage/${SC}/; done
```

出力例:

```
Starting pod/node-xxx-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
Starting pod/node-yyy-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
Starting pod/node-zzz-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
```

10.

手順 1 と 8 に一覧表示されている各ローカルボリュームセットまたはローカルボリュームのディスクを消去して、それらを再利用できるようにします。

a.

ストレージノードを一覧表示します。

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

出力例:

```
NAME          STATUS  ROLES  AGE  VERSION
```

```
node-xxx Ready worker 4h45m v1.18.3+6c42de8
node-yyy Ready worker 4h46m v1.18.3+6c42de8
node-zzz Ready worker 4h45m v1.18.3+6c42de8
```

b.

プロンプトが表示されたらノードコンソールを取得し、**chroot /host** コマンドを実行します。

```
$ oc debug node/node-xxx
Starting pod/node-xxx-debug ...
To use host binaries, run `chroot /host`
Pod IP: w.x.y.z
If you don't see a command prompt, try pressing enter.
sh-4.2# chroot /host
```

c.

ディスクパスを引用符内の **DISKS** 変数に保存します。ディスクパスの一覧は、ローカルボリュームセットおよびローカルボリュームセットおよびローカルボリュームのステップ 3 および手順 8.c を参照してください。

出力例:

```
sh-4.4# DISKS="/dev/disk/by-id/scsi-360050763808104bc28000000000000eb
/dev/disk/by-id/scsi-360050763808104bc28000000000000ef /dev/disk/by-id/scsi-
360050763808104bc28000000000000f3 "
or
sh-4.2# DISKS="/dev/sdb /dev/sdc /dev/sdd /dev/sde "
```

d.

すべてのディスクで **sgdisk --zap-all** を実行します。

```
sh-4.4# for disk in $DISKS; do sgdisk --zap-all $disk;done
```

出力例:

```
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
```

- e. シェルを終了し、他のノードに対して手順を繰り返します。

```
sh-4.4# exit
exit
sh-4.2# exit
exit
Removing debug pod ...
```

11. **openshift-local-storage namespace** を削除し、削除が完了するまで待機します。 **openshift-local-storage namespace** がアクティブなプロジェクトである場合、別のプロジェクトに切り換える必要があります。

以下は例になります。

```
$ oc project default
$ oc delete project openshift-local-storage --wait=true --timeout=5m
```

以下のコマンドが **NotFound** エラーを返すと、プロジェクトが削除されます。

```
$ oc get project openshift-local-storage
```

## 4.2. OPENSIFT DATA FOUNDATION からのモニタリングスタックの削除

このセクションでは、モニタリングスタックを **OpenShift Data Foundation** からクリーンアップします。

モニタリングスタックの設定の一部として作成される **Persistent Volume Claims (PVC)** は **openshift-monitoring namespace** に置かれます。

### 前提条件

- **PVC** は **OpenShift Container Platform** モニタリングスタックを使用できるように設定されます。

詳細は、[モニタリングスタックの設定](#)を参照してください。

### 手順

- 1.

**openshift-monitoring namespace** で現在実行されている Pod および PVC を一覧表示します。

```
$ oc get pod,pvc -n openshift-monitoring
```

出力例:

NAME	READY	STATUS	RESTARTS	AGE
pod/alertmanager-main-0	3/3	Running	0	8d
pod/alertmanager-main-1	3/3	Running	0	8d
pod/alertmanager-main-2	3/3	Running	0	8d
pod/cluster-monitoring-operator-84457656d-pkrxm	1/1	Running	0	8d
pod/grafana-79ccf6689f-2ll28	2/2	Running	0	8d
pod/kube-state-metrics-7d86fb966-rvd9w	3/3	Running	0	8d
pod/node-exporter-25894	2/2	Running	0	8d
pod/node-exporter-4dsd7	2/2	Running	0	8d
pod/node-exporter-6p4zc	2/2	Running	0	8d
pod/node-exporter-jbjvg	2/2	Running	0	8d
pod/node-exporter-jj4t5	2/2	Running	0	6d18h
pod/node-exporter-k856s	2/2	Running	0	6d18h
pod/node-exporter-rf8gn	2/2	Running	0	8d
pod/node-exporter-rmb5m	2/2	Running	0	6d18h
pod/node-exporter-zj7kx	2/2	Running	0	8d
pod/openshift-state-metrics-59dbd4f654-4clng	3/3	Running	0	8d
pod/prometheus-adapter-5df5865596-k8dzn	1/1	Running	0	7d23h
pod/prometheus-adapter-5df5865596-n2gj9	1/1	Running	0	7d23h
pod/prometheus-k8s-0	6/6	Running	1	8d
pod/prometheus-k8s-1	6/6	Running	1	8d
pod/prometheus-operator-55cfb858c9-c4zd9	1/1	Running	0	6d21h
pod/telemeter-client-78fc8fc97d-2rgfp	3/3	Running	0	8d

NAME	CAPACITY	ACCESS MODES	STATUS	VOLUME	AGE
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-0	40Gi	RWO	Bound	pvc-0d519c4f-15a5-11ea-baa0-026d231574aa	8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-1	40Gi	RWO	Bound	pvc-0d5a9825-15a5-11ea-baa0-026d231574aa	8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-2	40Gi	RWO	Bound	pvc-0d6413dc-15a5-11ea-baa0-026d231574aa	8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-0	40Gi	RWO	Bound	pvc-0b7c19b0-15a5-11ea-baa0-026d231574aa	8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-1	40Gi	RWO	Bound	pvc-0b8aed3f-15a5-11ea-baa0-026d231574aa	8d



2.

モニタリング configmap を編集します。

```
$ oc -n openshift-monitoring edit configmap cluster-monitoring-config
```

以下の例が示すように、OpenShift Data Foundation ストレージクラスを参照する config セクションを削除し、これを保存します。

#### 編集前

```
.  
. .  
apiVersion: v1  
data:  
  config.yaml: |  
    alertmanagerMain:  
      volumeClaimTemplate:  
        metadata:  
          name: my-alertmanager-claim  
        spec:  
          resources:  
            requests:  
              storage: 40Gi  
          storageClassName: ocs-storagecluster-ceph-rbd  
    prometheusK8s:  
      volumeClaimTemplate:  
        metadata:  
          name: my-prometheus-claim  
        spec:  
          resources:  
            requests:  
              storage: 40Gi  
          storageClassName: ocs-storagecluster-ceph-rbd  
kind: ConfigMap  
metadata:  
  creationTimestamp: "2019-12-02T07:47:29Z"  
  name: cluster-monitoring-config  
  namespace: openshift-monitoring  
  resourceVersion: "22110"  
  selfLink: /api/v1/namespaces/openshift-monitoring/configmaps/cluster-monitoring-config  
  uid: fd6d988b-14d7-11ea-84ff-066035b9efa8  
. . .
```

## 編集後

```

.
.
.
apiVersion: v1
data:
  config.yaml: |
kind: ConfigMap
metadata:
  creationTimestamp: "2019-11-21T13:07:05Z"
  name: cluster-monitoring-config
  namespace: openshift-monitoring
  resourceVersion: "404352"
  selfLink: /api/v1/namespaces/openshift-monitoring/configmaps/cluster-monitoring-config
  uid: d12c796a-0c5f-11ea-9832-063cd735b81c
.
.
.

```

この例では、`alertmanagerMain` および `prometheusK8s` モニタリングコンポーネントは `OpenShift Data Foundation PVC` を使用しています。

3. 関連する PVC を削除します。ストレージクラスを使用するすべての PVC を削除してください。

```
$ oc delete -n openshift-monitoring pvc <pvc-name> --wait=true --timeout=5m
```

<pvc-name>

PVC の名前です。

#### 4.3. OPENSIFT DATA FOUNDATION からの OPENSIFT CONTAINER PLATFORM レジストリーの削除

このセクションを使用して、`OpenShift Data Foundation` から `OpenShift Container Platform` レジストリーをクリーンアップします。代替ストレージを設定する必要がある場合は、[Image registry](#) を参照してください。

`OpenShift Container Platform` レジストリーの設定の一部として作成される `Persistent Volume Claims (PVC)` は `openshift-image-registry namespace` に置かれます。

## 前提条件

- イメージレジストリーは OpenShift Data Foundation PVC を使用するように設定されている必要があります。

## 手順

1. `configs.imageregistry.operator.openshift.io` オブジェクトを編集し、`storage` セクションのコンテンツを削除します。

```
$ oc edit configs.imageregistry.operator.openshift.io
```

### 編集前

```
.  
.  
.  
storage:  
  pvc:  
    claim: registry-cephfs-rwx-pvc  
.  
.  
.
```

### 編集後

```
.  
.  
.  
storage:  
  emptyDir: {}  
.  
.  
.
```

この例では、PVC は `registry-cephfs-rwx-pvc` と呼ばれ、これは安全に削除できます。

2. PVC を削除します。

```
$ oc delete pvc <pvc-name> -n openshift-image-registry --wait=true --timeout=5m
```

<pvc-name>

PVC の名前です。

#### 4.4. OPENSIFT DATA FOUNDATION からのクラスターロギング OPERATOR の削除

このセクションでは、クラスターロギング Operator を OpenShift Data Foundation からクリーンアップします。

クラスターロギング Operator の設定の一部として作成される Persistent Volume Claims (PVC) は openshift-logging namespace にあります。

##### 前提条件

- クラスターロギングインスタンスは、OpenShift Data Foundation PVC を使用するように設定されている必要があります。

##### 手順

1. namespace の ClusterLogging インスタンスを削除します。

```
$ oc delete clusterlogging instance -n openshift-logging --wait=true --timeout=5m
```

openshift-logging namespace の PVC は安全に削除できます。

2. PVC を削除します。

```
$ oc delete pvc <pvc-name> -n openshift-logging --wait=true --timeout=5m
```

<pvc-name>

PVC の名前です。