



Red Hat OpenShift Container Storage 4.8

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

ベアメタル環境のインストールおよび設定方法

Red Hat OpenShift Container Storage 4.8 ベアメタルインフラストラクチャーを使用した OpenShift Container Storage のデプロイ

ベアメタル環境のインストールおよび設定方法

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_Container_Storage_using_bare_metal_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[®] 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

Red Hat OpenShift Container Storage 4.8 をインストールし、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方法については、本書をお読みください。

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 CONTAINER STORAGE 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 CONTAINER STORAGE をインストールするための要件	6
1.2. RED HAT ENTERPRISE LINUX ベースのノード上のコンテナでのファイルシステムアクセスの有効化	7
1.3. 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	8
第2章 ローカルストレージデバイスを使用した OPENSIFT CONTAINER STORAGE のデプロイ	9
2.1. ローカルストレージ OPERATOR のインストール	9
2.2. RED HAT OPENSIFT CONTAINER STORAGE OPERATOR のインストール	9
2.3. MULTUS ネットワークの作成	11
2.3.1. ネットワーク接続定義の作成	11
2.4. ベアメタルでの OPENSIFT CONTAINER STORAGE クラスターの作成	13
第3章 OPENSIFT CONTAINER STORAGE デプロイメントの検証	18
3.1. POD の状態の確認	18
3.2. OPENSIFT CONTAINER STORAGE クラスターが正常であることの確認	20
3.3. MULTICLOUD OBJECT GATEWAY が正常であることの確認	20
3.4. OPENSIFT CONTAINER STORAGE 固有のストレージクラスが存在することの確認	20
3.5. MULTUS ネットワークの確認	21
第4章 OPENSIFT CONTAINER STORAGE のアンインストール	24
4.1. 内部モードでの OPENSIFT CONTAINER STORAGE のアンインストール	24
4.1.1. ローカルストレージ Operator の設定の削除	31
4.2. OPENSIFT CONTAINER STORAGE からのモニターリングスタックの削除	33
4.3. OPENSIFT CONTAINER STORAGE からの OPENSIFT CONTAINER PLATFORM レジストリーの削除	36
4.4. OPENSIFT CONTAINER STORAGE からのクラスターロギング OPERATOR の削除	37

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 である 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** 形式になっていることを確認してください。ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
 2. マウスカーソルを使用して、コメントを追加するテキストの部分を強調表示します。
 3. 強調表示された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. 表示される指示に従ってください。
- より詳細なフィードバックをお寄せいただく場合は、Bugzilla のチケットを作成してください。
 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 Container Storage 4.8 は、接続環境または非接続環境での既存の Red Hat OpenShift Container Platform(RHOCP) 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 Container Storage クラスターがサポートされます。導入要件の詳細については、[Planning your deployment](#) および [Preparing to deploy OpenShift Container Storage](#) を参照してください。

OpenShift Container Storage をデプロイするには、お使いの環境に適切なデプロイメントプロセスを実行します。

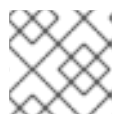
- 内部モード
 - [ローカルストレージデバイスを使用したデプロイ](#)
- [外部モード](#)

第1章 OPENSIFT CONTAINER STORAGE 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 Container Storage 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。これにより、ベースサービスの内部プロビジョニングが可能になり、追加のストレージクラスをアプリケーションで使用可能にすることができます。

ローカルストレージを使用して Red Hat OpenShift Container Storage 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。詳細は、[ローカルストレージデバイスを使用して OpenShift Container Storage をインストールするための要件](#) を参照してください。

- ワーカーノードについて Red Hat Enterprise Linux ベースのホストでのファイルシステムアクセスを有効にします。[Enable file system access for containers on Red Hat Enterprise Linux based nodes](#) を参照してください。



注記

Red Hat Enterprise Linux CoreOS(RHCOS) の場合は、この手順を省略します。

- オプション: 外部鍵管理システム (KMS) を使用してクラスター全体の暗号化を有効にする場合:
 - 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 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。

上記を処理した後に、指定した順序でセクションに従います。

1. [ローカルストレージ Operator のインストール](#)
2. [Red Hat OpenShift Container Storage Operator をインストール](#)
3. [ベアメタルでの OpenShift Container Storage 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 CONTAINER STORAGE 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。

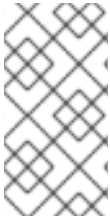
- 選択した 3 つのノードには、OpenShift Container Storage で使用できる raw ブロックデバイスが少なくとも 1 つ必要です。
- 使用するデバイスは空である必要があります。ディスクには物理ボリューム (PV)、ボリュームグループ (VG)、または論理ボリューム (LV) を含めないでください。

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。

Arbiter ストレッチクラスターの要件 [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 オンプレミスでのデプロイメント用とされています。

詳細な要件と手順は、[Configuring OpenShift Container Storage for Metro-DR stretch cluster](#) 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、柔軟なスケーリングと arbiter の両方を有効にすることはできません。Flexible scaling を使用すると、1度に1つのノードを OpenShift Container Storage クラスターに追加することができます。arbiter クラスターでは、2つのデータゾーンごとに1つ以上のノードを追加する必要があります。

compact モードの要件

OpenShift Container Storage 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るようになりました。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。

compact モードで OpenShift Container Platform を設定するには、[3 ノードクラスターの設定](#) および [エッジデプロイメントの3 ノードアーキテクチャーの提供](#) を参照してください。

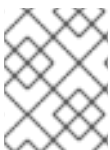
ノードの最小要件 [テクノロジープレビュー]

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

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

1.2. RED HAT ENTERPRISE LINUX ベースのノード上のコンテナでのファイルシステムアクセスの有効化

ユーザーによってプロビジョニングされるインフラストラクチャー (UPI) で Red Hat Enterprise Linux がベースの OpenShift Data Foundation にワーカーノードを含めて OpenShift Container Storage をデプロイしても自動的に、基盤の Ceph ファイルシステムへのコンテナアクセスが提供されるわけではありません。



注記

Red Hat Enterprise Linux CoreOS(RHCOS) をベースとするホストの場合は、このセクションを省略します。

手順

1. Red Hat Enterprise Linux ベースのノードにログインし、ターミナルを開きます。
2. クラスター内の各ノードについて、以下を実行します。
 - a. ノードが `rhel-7-server-extras-rpms` リポジトリにアクセスできることを確認します。

```
# subscription-manager repos --list-enabled | grep rhel-7-server
```

出力に `rhel-7-server-rpms` と `rhel-7-server-extras-rpms` の両方が表示されない場合は、以下のコマンドを実行して各リポジトリを有効にします。

```
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

- b. 必要なパッケージをインストールします。

```
# yum install -y policycoreutils container-selinux
```

- c. SELinux での Ceph ファイルシステムのコンテナの使用を永続的に有効にします。

```
# setsebool -P container_use_cephfs on
```

1.3. 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。
- 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=ocs 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=ocs 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 "ocs/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}
path "sys/mounts" {
  capabilities = ["read"]
}' | vault policy write ocs -
```

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=ocs -format json
```

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

ローカルストレージデバイスを使用して OpenShift Container Storage をベアメタルインフラストラクチャーにデプロイするには、以下のセクションを実行します。

1. [ローカルストレージ Operator のインストール](#)
2. [Red Hat OpenShift Container Storage Operator をインストール](#)
3. [ベアメタルでの OpenShift Container Storage 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

Local Storage 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こと。

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** と入力して、オペレータのリストから **Local Storage** オペレーターを検索し、クリックします。
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-4.8** 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。
6. **Install** をクリックします。

検証手順

- ローカルストレージ Operator がステータス **Succeeded** を表示していることを確認します。

2.2. RED HAT OPENSIFT CONTAINER STORAGE OPERATOR のインストール

Red Hat OpenShift Container Storage 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。
- 必要な追加要件をすべて満たしています。詳細は、[Planning your deployment](#) を参照してください。



注記

- OpenShift Container Storage のクラスタ全体でのデフォルトノードセレクターを上書きする必要がある場合は、以下のコマンドを使用し、**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 Container Storage リソースのみがスケジュールされるように、そのノードに **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 Container Storage 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. Operator の一覧から **OpenShift Container Storage** を検索し、これをクリックします。
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-4.8** 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。
 - e. **Install** をクリックします。
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 が新規バージョンに更新されるように更新要求を手動で承認する必要があります。

検証手順

- **OpenShift Container 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. MULTUS ネットワークの作成

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時に、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。それぞれのインターフェイスは、**NetworkAttachmentDefinition** カスタムリソース (CR) を使用して指定できます。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 Container Storage は **macvlan** という 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。

2.3.1. ネットワーク接続定義の作成

Multus を使用するには、適切な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。詳細は、[Recommended network configuration and requirements for a Multus configuration](#) を参照してください。**NetworkAttachmentDefinition** (NAD) は、ストレージクラスターのインストール時に後で選択できるようになりました。これは、ストレージクラスターの前に作成する必要のある理由です。

Planning Guide で説明されているように、作成する Multus ネットワークは、OpenShift Container Storage 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 **NetworkAttachmentDefinitions** 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。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)。

2.4. ベアメタルでの OPENSIFT CONTAINER STORAGE 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た OpenShift Container Storage 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。
- 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、[Multus support](#) および [Creating network attachment definitions](#) 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** → **Installed Operators** をクリックし、インストールされた Operator をすべて表示します。
3. 選択された **Project** が **openshift-storage** であることを確認します。
4. Storage Cluster の **OpenShift Container Storage** → **Create Instance** リンクをクリックします。
5. **Internal-Attached devices** としてモードを選択します。
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。

ストレージボリュームのセットをフィルターすることにより、専用のストレージクラスを作成してストレージを消費できます。

6. 次のいずれかのオプションを使用して、ディスクを検出します。
 - **All nodes**: すべてのノードでディスクを検出します。
 - **Select nodes**: 利用可能なノードのサブセットからディスクを検出します。



重要

arbiter モードを使用する場合は、**All nodes** オプションを選択しないでください。代わりに、**Select nodes** オプションを使用して、2つのデータセンターゾーンから、割り当てられたストレージデバイスを持つラベルが付けられたノードを選択します。

選択したノードにティントのマークが付けられており、そのノードがウィザードで検出されない場合は、ローカルストレージ Operator リソースの容認を追加する回避策として [Red Hat ナレッジベースソリューション](#) の手順に従ってください。

選択したノードが集約された 30 CPU および 72 GiB の RAM の OpenShift Container Storage 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 [Resource requirements](#) のセクションを参照してください。

7. **Next** をクリックします。

8. **ストレージクラス** の作成

- a. **Local Volume Set Name** を入力します。
- b. **Storage Class Name** を入力します。デフォルトで、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。名前を変更することもできます。
- c. 直前の手順でディスク検出で選択されたノードは **Filter Disks** セクションに表示されます。以下のいずれかを選択します。
 - **Disks on all nodes** ディスクを検出したすべてのノードを選択します。
 - **Disks on selected nodes** ディスクを検出したノードのサブセットを選択します。高可用性を確保するために、ワーカーノードは 3 つの異なる物理ノード、ラック、障害ドメインに分散します。




重要

柔軟なスケーリング機能は、3 つ以上のノードが最小要件の 3 つ未満の可用性ゾーンに分散されているストレージの作成時に有効にされます。この機能は、OpenShift Container Storage 4.7 クラスターの新規デプロイメントでのみ利用でき、アップグレードされたクラスターをサポートしません。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](#) を参照してください。

d. **Disk Type** の利用可能な一覧から、**SSD/NVMe** を選択します。

e. **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	ディスクタイプを選択します。デフォルトでは、Disk および Part が選択されます。
ディスクサイズ	<p>含める必要のあるデバイスの最小および最大の許容サイズ。</p> <div style="display: flex; align-items: center;">  <div> <p>注記</p> <p>デバイスの最小サイズ 100GB を設定する必要があります。</p> </div> </div>
最大ディスク制限	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。

- f. **Next** をクリックします。新規ストレージクラスの作成を確認するポップアップが表示されます。
 - g. **Yes** をクリックして続行します。
9. **Capacity and nodes** を設定します。
- a. **Storage Class** を選択します。デフォルトで、直前の手順で作成された新規ストレージクラスが選択されます。
 - **選択したノード** には、直前の手順で選択したノードが表示されます。この一覧では、直前の手順で検出されたディスクを反映するのに数分かかります。
 - b. **Next** をクリックします。
10. (オプション) **Security and network** 設定を設定します。
- a. **Enable encryption** チェックボックスを選択して、ブロックおよびファイルストレージを暗号化します。
 - b. **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 (ブロックおよびファイル) を暗号化します。
 - **Storage class encryption**(ストレージクラスの暗号化): 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 サーバーのホスト **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 Container Storage 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て、**CA Certificate**、**Client Certificate**、および **Client Private Key** を指定します。
 - D. **Save** をクリックします。
11. 単一のネットワークを使用する場合は **Default (SDN)** を選択し、複数のネットワークインターフェイスを使用する場合は **Custom (Multus)** ネットワークを選択します。
- a. ドロップダウンメニューから **Public Network Interface** を選択します。
 - b. ドロップダウンメニューから **Cluster Network Interface** を選択します。



注記

1つの追加ネットワークインターフェイスのみを使用する場合は、パブリックネットワークインターフェイス (例: **ocs-public-cluster**) の単一の **NetworkAttachmentDefinition** を選択し、Cluster Network Interface を空白のままにします。

12. **Next** をクリックします。
13. 設定の詳細を確認します。設定設定を変更するには、**Back** をクリックして以前の設定ページに戻ります。
14. **Create** をクリックします。
15. Vault Key/Value (KV) シークレットエンジン API の場合に configmap を編集します。バージョン 2 は、鍵管理システム (KMS) のクラスター全体の暗号化に使用されます。
 - a. OpenShift Web Console で、**Workloads** → **ConfigMaps** をクリックします。
 - b. KMS 接続の詳細を表示するには、**ocs-kms-connection-details** をクリックします。
 - c. configmap を編集します。
 - i. **Action menu**(**!**) → **Edit ConfigMap** をクリックします。
 - ii. **VAULT_BACKEND** パラメーターを **v2** に設定します。

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: ocs-kms-connection-details
  [...]
data:
  KMS_PROVIDER: vault
  KMS_SERVICE_NAME: vault
  [...]
  VAULT_BACKEND: v2
  [...]
```

- iii. **Save** 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後の **Status** が緑色のチェックマークと共に **Phase: Ready** と表示されていることを確認します。
 - **Operators** → **Installed Operators** → **Storage Cluster** のリンクをクリックして、ストレージクラスターのインストールのステータスを表示します。
 - または、Operator **Details** タブで、**Storage Cluster** タブをクリックすると、ステータスを表示できます。
- 柔軟なスケール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、柔軟なスケールリングが無効になります)。
 1. **Storage Cluster** タブで **ocs-storagecluster** をクリックします。

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は、柔軟なスケーリング機能が有効になります。

```
spec:  
  flexibleScaling: true  
  [...]br/>status:  
  failureDomain: host
```

- OpenShift Container Storage 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 Container Storage installation](#) を参照してください。
- OpenShift Container Storage 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 [Container Storage インストールの確認](#) 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 OPENSIFT CONTAINER STORAGE デプロイメントの検証

内部モードの OpenShift Container Storage が正常にデプロイされていることを確認するには、以下のセクションを参照してください。

1. Pod の状態の確認
2. OpenShift Container Storage クラスターが正常であることの確認。
3. Multicloud Object Gateway が正常であることの確認。
4. OpenShift Container Storage 固有のストレージクラスが存在することの確認。

3.1. POD の状態の確認

OpenShift Containers Storage の Pod が実行状態にあることを確認するには、以下の手順に従います。

手順

1. OpenShift Web コンソールにログインします。
2. 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。
3. **Project** ドロップダウンリストから **openshift-storage** を選択します。
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についての詳細は、表3.1「OpenShift Container Storage クラスターに対応する Pod」を参照してください。
4. **Running** タブと **Completed** タブをクリックして、Pod が実行中で完了状態になっていることを確認します。

表3.1 OpenShift Container Storage クラスターに対応する Pod

コンポーネント	対応する Pod
OpenShift Container Storage Operator	<ul style="list-style-type: none"> ● ocs-operator-* (任意のワーカーノードに 1 Pod) ● ocs-metrics-exporter-*
Rook-ceph Operator	<p>rook-ceph-operator-*</p> <p>(任意のワーカーノードに 1 Pod)</p>

コンポーネント	対応する Pod
Multicloud Object Gateway	<ul style="list-style-type: none"> ● noobaa-operator-* (任意のワーカーノードに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-* (ストレージノードに分散する 3 Pod)
MGR	rook-ceph-mgr-* (任意のストレージノードに1Pod)
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)
CSI	<ul style="list-style-type: none"> ● cephfs <ul style="list-style-type: none"> ○ csi-cephfsplugin-* (各ワーカーノードに1Pod) ○ csi-cephfsplugin-provisioner-* (ワーカーノードに分散する 2 Pod) ● rbd <ul style="list-style-type: none"> ○ csi-rbdplugin-* (各ワーカーノードに1Pod) ○ 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1Pod)

コンポーネント	対応する 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)

3.2. OPENSIFT CONTAINER STORAGE クラスターが正常であることの確認

OpenShift Container Storage のクラスターが正常であることを確認するには、手順の手順に従います。

手順

1. **Storage → Overview** をクリックし、**Block and File** タブをクリックします。
2. **Status** カードで、**Storage Cluster** および **Data Resiliency** 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。
3. **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 Container Storage クラスターの正常性については、[OpenShift Container Storage のモニターリング](#) 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

OpenShift Container Storage Multicloud Object Gateway が正常であることを確認するには、手順のステップに従います。

手順

1. OpenShift Web コンソールから **Storage → Overview** をクリックし、**Object** タブをクリックします。
2. **Status card** で、**Object Service** と **Data Resiliency** の両方が **Ready** 状態 (緑のチェックマーク) にあることを確認します。
3. **Details** カードで、Multicloud Object Gateway 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 Container Storage クラスターの正常性については、[OpenShift Container Storage のモニターリング](#) を参照してください。

3.4. OPENSIFT CONTAINER STORAGE 固有のストレージクラスが存在することの確認

ストレージクラスがクラスターに存在することを確認するには、手順のステップに従います。

手順

1. OpenShift Web コンソールから **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 Container Storage 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**

3.5. 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

1. ネットワーク設定の選択に応じて、OpenShift Container Storage 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は、アプリケーション Pod と OpenShift Container Storage 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パブリックネットワーク上にあり、クラスターネットワークはレプリケーションおよび OSD 間のトラフィックのリバランス用です。
2. ネットワーク設定が正しいことを確認するには、次の手順に従います。
 - a. OpenShift コンソールで、**Installed Operators** → **Storage Cluster** → **ocs-storagecluster** をクリックします。
 - b. 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:
  provider: multus
  selectors:
    cluster: openshift-storage/ocs-cluster
    public: openshift-storage/ocs-public
  [..]
```

3. 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

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

4. OSD Pod が正しいネットワークを使用していることを確認します。
 - a. **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

両方が作成されている場合は、OSD Pod のみが Multus パブリックネットワークと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します。

```
$ 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\.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": {}
}
```

5. 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 ユーティリティーが必要です)。

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

出力サンプル

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

第4章 OPENSIFT CONTAINER STORAGE のアンインストール

OpenShift Container Storage をアンインストールし、削除するには、以下のセクションを実行します。

1. [内部モードでの OpenShift Container Storage のアンインストール](#)
2. [OpenShift Container Storage からのモニタリングスタックの削除。](#)
3. [OpenShift Container Storage からの OpenShift Container Platform レジストリーの削除](#)
4. [クラスターロギング Operator を openShift Container Storage から削除](#)

4.1. 内部モードでの OPENSIFT CONTAINER STORAGE のアンインストール

このセクションの手順に従って OpenShift Container Storage 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

Annotation	値	デフォルト	動作
<code>cleanup-policy</code>	<code>delete</code>	はい	Rook は物理ドライブおよび DataDirHostPath をクリーンアップします。
<code>cleanup-policy</code>	<code>Retain</code>	いいえ	Rook は物理ドライブおよび DataDirHostPath 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 PVC および OBC が管理者/ユーザーによって削除されるまでアンインストールプロセスを一時停止します。

Annotation	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。

以下のコマンドを使用してアノテーションの値を編集し、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できます。

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

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

前提条件

- OpenShift Container Storage 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 Container Storage 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 Container Storage 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 Container Storage 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 Container Storage 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>
```

2. OpenShift Container Storage 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 Container Storage 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 を事前に削除せずに 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、システム内に孤立した PVC と OBC が発生します。

- a. OpenShift Container Storage を使用して、OpenShift Container Platform モニターリングスタック PVC を削除します。
詳細は、「[OpenShift Container Storage からのモニターリングスタックの削除](#)」を参照してください。
- b. OpenShift Container Storage を使用して、OpenShift Container Platform レジストリー PVC を削除します。
詳細は、「[OpenShift Container Storage からの OpenShift Container Platform レジストリーの削除](#)」を参照してください。
- c. OpenShift Container Storage を使用して、OpenShift Container Platform ロギング PVC を削除します。
詳細は、「[OpenShift Container Storage からのクラスターロギング Operator の削除](#)」を参照してください。
- d. OpenShift Container Storage 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。
 - 次のスクリプトは、OpenShift Container Storage を使用してプロビジョニングされた PVC および OBC を識別するためのサンプルスクリプトです。このスクリプトは、OpenShift Container Storage 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>
```

- PVC を削除します。

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



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。

- Storage Cluster 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 storagecluster --all --wait=true
```

- 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
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
```

- /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
```

- 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 Container Storage 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。

- 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
```

- b. デバイス名を取得し、OpenShift Container Storage デバイスについてメモします。

```
$ dmsetup ls
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
```



注記

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>
```

- 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。以下に例を示します。

```
$ 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 Container Storage 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](#) の記事に記載の手順を実行して 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 Container Storage をデプロイしている場合は、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 Container Storage テイントを削除します。

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

11. OpenShift Container Storage を使用してプロビジョニングした 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>
```

12. Multicloud Object Gateway storageclass を削除します。

```
$ oc delete storageclass openshift-storage.noobaa.io --wait=true --timeout=5m
```

13. **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 --wait=true --timeout=5m
```

14. オプション: vault キーが完全に削除されるようにするには、vault キーに関連付けられているメタデータを手動で削除する必要があります。



注記

このステップは、Vault Key/Value (KV) シークレットエンジン API バージョン 2 が、OpenShift Container Storage のアンインストール中に Vault キーが削除済みとしてマークされ、完全に削除されないため、Key Management System (KMS) によるクラスター全体の暗号化に使用される場合にのみ実行してください。必要に応じて、後でいつでも復元できます。

- a. Vault 内のキーを一覧表示します。

```
$ vault kv list <backend_path>
```

<backend_path>

暗号化キーが保存されている Vault 内のパスです。
以下に例を示します。

```
$ vault kv list kv-v2
```

出力例:

```
Keys
-----
```

```
NOOBAA_ROOT_SECRET_PATH/
rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
rook-ceph-osd-encryption-key-ocs-deviceset-thin-1-data-0sq227
rook-ceph-osd-encryption-key-ocs-deviceset-thin-2-data-0xzszb
```

- b. Vault キーに関連付けられているメタデータを一覧表示します。

```
$ vault kv get kv-v2/<key>
```

Multicloud Object Gateway (MCG) キーの場合:

```
$ vault kv get kv-v2/NOOBAA_ROOT_SECRET_PATH/<key>
```

<key>

暗号化キーです。
以下に例を示します。

```
$ vault kv get kv-v2/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
```

出力例:

```
==== Metadata =====
Key          Value
---          -
created_time 2021-06-23T10:06:30.650103555Z
deletion_time 2021-06-23T11:46:35.045328495Z
destroyed    false
version      1
```

- c. メタデータを削除します。

```
$ vault kv metadata delete kv-v2/<key>
```

MCG キーの場合:

```
$ vault kv metadata delete kv-v2/NOOBAA_ROOT_SECRET_PATH/<key>
```

<key>

暗号化キーです。
以下に例を示します。

```
$ vault kv metadata delete kv-v2/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-
data-0m27q8
```

出力例:

```
Success! Data deleted (if it existed) at: kv-v2/metadata/rook-ceph-osd-encryption-key-
ocs-deviceset-thin-0-data-0m27q8
```

- d. これらの手順を繰り返して、すべての Vault キーに関連付けられているメタデータを削除します。
15. OpenShift Container Platform Web コンソールで、OpenShift Container Storage 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. **Overview** 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

ローカルストレージデバイスを使用して OpenShift Container Storage をデプロイした場合にのみ、本セクションの手順を使用します。



注記

localvolume リソースのみを使用する OpenShift Container Storage デプロイメントについては、ステップ 8 を参照してください。

手順

1. **LocalVolumeSet** および OpenShift Container Storage で使用される対応する **StorageClassName** を特定します。
2. **LocalVolumeSet** を提供する **StorageClass** に変数 **SC** を設定します。

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

3. **LocalVolumeSet** を削除します。

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

4. 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
```

5. **StorageClassName** を削除します。

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

6. **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
```

7. **LocalVolumeDiscovery** を削除します。

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

8. 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 Container Storage 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、次の手順を実行します。

- LocalVolume** および OpenShift Container Storage で使用される対応する **StorageClassName** を特定します。
- 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。以下に例を示します。

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

- ローカルボリュームリソースを削除します。

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

- 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
```

- 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 ...
```

4.2. OPENSIFT CONTAINER STORAGE からのモニターリングスタックの削除

このセクションでは、モニターリングスタックを OpenShift Container Storage からクリーンアップします。

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

前提条件

- PVC は OpenShift Container Platform モニター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、[モニター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
```

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 STORAGECLASS	AGE
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-0	40Gi	RWO	Bound	pvc-0d519c4f-15a5-11ea-baa0-026d231574aa	8d
				ocs-storagecluster-ceph-rbd	

```
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-1 Bound pvc-0d5a9825-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-2 Bound pvc-0d6413dc-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-0 Bound pvc-0b7c19b0-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-1 Bound pvc-0b8aed3f-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
```

2. モニタリング **configmap** を編集します。

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

3. 以下の例が示すように、OpenShift Container Storage 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** モニターリングコンポーネントは OpenShift Container Storage PVC を使用しています。

4. 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
```

4.3. OPENSIFT CONTAINER STORAGE 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 Container Storage 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、[イメージレジストリー](#) 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 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 Container Storage 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
```

4.4. OPENSIFT CONTAINER STORAGE からのクラスターロギング OPERATOR の削除

OpenShift Container Storage からクラスターロギングオペレーターを削除するには、手順のステップに従います。

クラスターロギング Operator の設定の一部として作成される 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 Container Storage PVC を使用するよう設定する必要があります。

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
```