



Red Hat OpenShift Container Storage 4.5

Amazon Web サービスを使用した OpenShift Container Storage のデプロイ

OpenShift Container Storage の OpenShift Container Platform AWS クラスターへの
インストールおよび設定方法

Red Hat OpenShift Container Storage 4.5 Amazon Web サービスを使用した OpenShift Container Storage のデプロイ

OpenShift Container Storage の OpenShift Container Platform AWS クラスターへのインストール
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_Amazon_Web_Services.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

ローカルまたはクラウドストレージの Amazon Web Services を使用して Red Hat OpenShift Container Storage 4.5 をインストールする方法については、本書をお読みください。

目次

はじめに	3
第1章 動的ストレージデバイスを使用したデプロイ	4
1.1. 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	4
1.2. RED HAT OPENSIFT CONTAINER STORAGE OPERATOR のインストール	5
1.3. 内部モードでの OPENSIFT CONTAINER STORAGE CLUSTER SERVICE の作成	7
第2章 ローカルストレージデバイスを使用したデプロイメント	11
2.1. 内部ローカルストレージを使用したデプロイの概要	11
2.2. ローカルストレージデバイスを使用した OPENSIFT CONTAINER STORAGE のインストール要件	11
2.3. 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	12
2.4. RED HAT OPENSIFT CONTAINER STORAGE OPERATOR のインストール	13
2.5. ローカルストレージ OPERATOR のインストール	15
2.6. 利用可能なストレージデバイスの検索	16
2.7. AMAZON EC2 (ストレージ最適化: I3EN.2XLARGE インスタンスタイプ) での OPENSIFT CONTAINER STORAGE クラスターの作成	18
第3章 内部モードの OPENSIFT CONTAINER STORAGE デプロイメントの確認	23
3.1. POD の状態の確認	23
3.2. OPENSIFT CONTAINER STORAGE クラスターが正常であることの確認	24
3.3. MULTICLOUD OBJECT GATEWAY が正常であることの確認	25
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	26
第4章 OPENSIFT CONTAINER STORAGE のアンインストール	27
4.1. 内部モードでの OPENSIFT CONTAINER STORAGE のアンインストール	27
4.2. OPENSIFT CONTAINER STORAGE からのモニタ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

はじめに

Red Hat OpenShift Container Storage 4.5 は、接続環境または非接続環境での既存の Red Hat OpenShift Container Platform (OCP) AWS 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

AWS では、内部の Openshift Container Storage クラスターのみがサポートされます。デプロイメントの要件についての詳細は、『[デプロイメントのプランニング](#)』を参照してください。

内部モードで 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。

- [動的ストレージデバイスを使用したデプロイ](#)
- [ローカルストレージデバイスを使用したデプロイ](#) [テクノロジープレビュー]

第1章 動的ストレージデバイスを使用したデプロイ

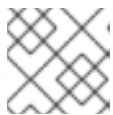
AWS EBS (タイプ: gp2) で提供される動的ストレージデバイスを使用して OpenShift Container Storage を OpenShift Container Platform にデプロイすると、内部クラスターリソースを作成するオプションが提供されます。これにより、ベースサービスの内部プロビジョニングが可能になり、追加のストレージクラスをアプリケーションで使用可能にすることができます。



注記

AWS では、内部の OpenShift Container Storage クラスターのみがサポートされます。デプロイメントの要件についての詳細は、『[デプロイメントのプランニング](#)』を参照してください。

1. 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) の Red Hat Enterprise Linux ベースのホストについては、基礎となるファイルシステムへのコンテナへのアクセスを有効にします。[Red Hat Enterprise Linux ベースのノードでのコンテナのファイルシステムのアクセスを有効にする方法](#) についての手順に従ってください。



注記

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

2. [Red Hat OpenShift Container Storage Operator をインストールします](#)。
3. [OpenShift Container Storage Cluster Service を作成します](#)。

1.1. 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) の Red Hat Enterprise Linux ベースに OpenShift Container Platform をデプロイしても、基礎となる Ceph ファイルシステムへのコンテナアクセスは自動的に提供されません。



注記

このプロセスは、Red Hat Enterprise Linux CoreOS をベースとするホストには不要です。

手順

クラスター内の各ノードで以下の手順を実行します。

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. 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
```

3. 必要なパッケージをインストールします。

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

4. 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
```

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

前提条件

- OpenShift Container Platform クラスターにログインしている必要がある。
- OpenShift Container Platform クラスターにワーカーノードが少なくとも 3 つ必要です。



注記

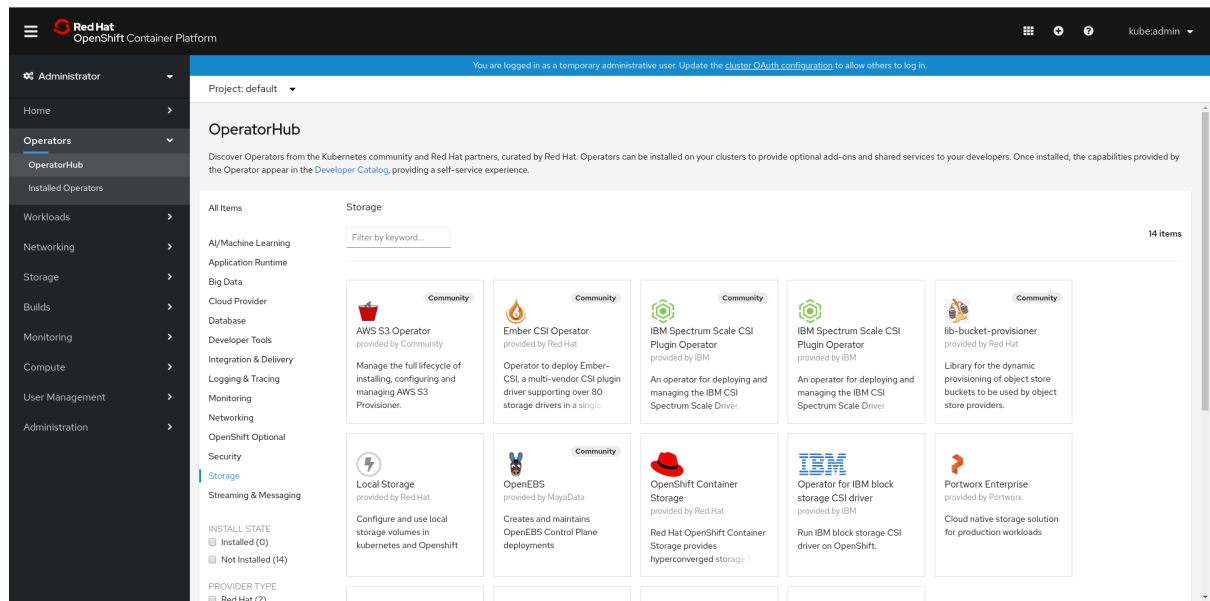
OpenShift Container Storage のクラスター全体でのデフォルトノードセレクターを上書きする必要がある場合は、コマンドラインインターフェースで以下のコマンドを使用し、**openshift-storage** namespace の空のノードセレクターを指定できます。

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

手順

1. OpenShift Web コンソールの左側のペインで、**Operators → OperatorHub** をクリックします。

図1.1 Operator Hub の Operator 一覧



2. OpenShift Container Storage をクリックします。
Filter by keyword テキストボックスまたはフィルター一覧を使用して、Operator の一覧から OpenShift Container Storage を検索できます。
3. OpenShift Container Storage Operator ページで、Install をクリックします。
4. Install Operator ページで、以下のオプションが選択されていることを確認します。
 - a. Channel を **stable-4.5**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 PR 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** として選択している。承認ストラテジーはデフォルトで **Automatic** に設定されます。
 - **Approval Strategy** に **Automatic** を選択します。



注記

Approval Strategy を **Automatic** として選択すると、新規インストール時、または OpenShift Container Storage の最新バージョンへの更新時に承認は必要ありません。

- i. **インストール** をクリックします。
- ii. インストールが開始するまで待機します。これには、最長 20 分の時間がかかる可能性があります。
- iii. **Operators** → **Installed Operators** をクリックします。
- iv. **Project** が **openshift-storage** であることを確認します。デフォルトで、**プロジェクト** は **openshift-storage** です。

- v. OpenShift Container Storage の Status が Succeeded に変更するまで待機します。
- Approval Strategy に Manual を選択します。



注記

Approval Strategy を **Manual** として選択すると、新規インストール時、または OpenShift Container Storage の最新バージョンへの更新時に承認が必要になります。

- i. **Install** をクリックします。
- ii. **Installed Operators** ページで、**ocs-operator** をクリックします。
- iii. **Subscription Details** ページで、**Install Plan** リンクをクリックします。
- iv. **InstallPlan Details** ページで、**Preview Install Plan** をクリックします。
- v. インストール計画を確認し、**Approve** をクリックします。
- vi. **Components** の Status が **Unknown** から **Created** または **Present** のいずれかに変更するまで待機します。
- vii. **Operators** → **Installed Operators** をクリックします。
- viii. **Project** が **openshift-storage** であることを確認します。デフォルトで、**プロジェクト** は **openshift-storage** です。
- ix. OpenShift Container Storage の Status が Succeeded に変更するまで待機します。

検証手順

- OpenShift Container Storage Operator の Status が Installed Operators ダッシュボードで **Succeeded** と表示されることを確認します。

1.3. 内部モードでの OPENSIFT CONTAINER STORAGE CLUSTER SERVICE の作成

以下の手順を使用して、OpenShift Container Storage Operator のインストール後に OpenShift Container Storage Cluster Service を作成します。

前提条件

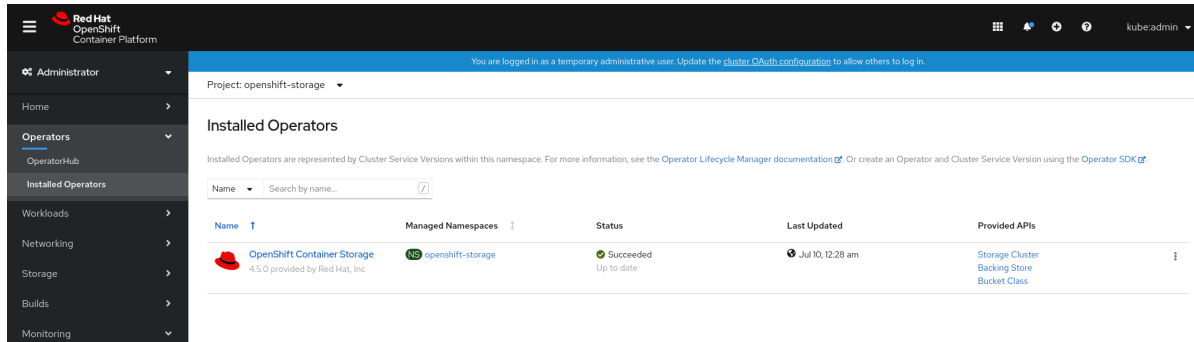
- OpenShift Container Storage は Operator Hub からインストールする必要があります。詳細は、「[Installing OpenShift Container Storage Operator using the Operator Hub](#)」を参照してください。

手順

1. OpenShift Web コンソールから **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。

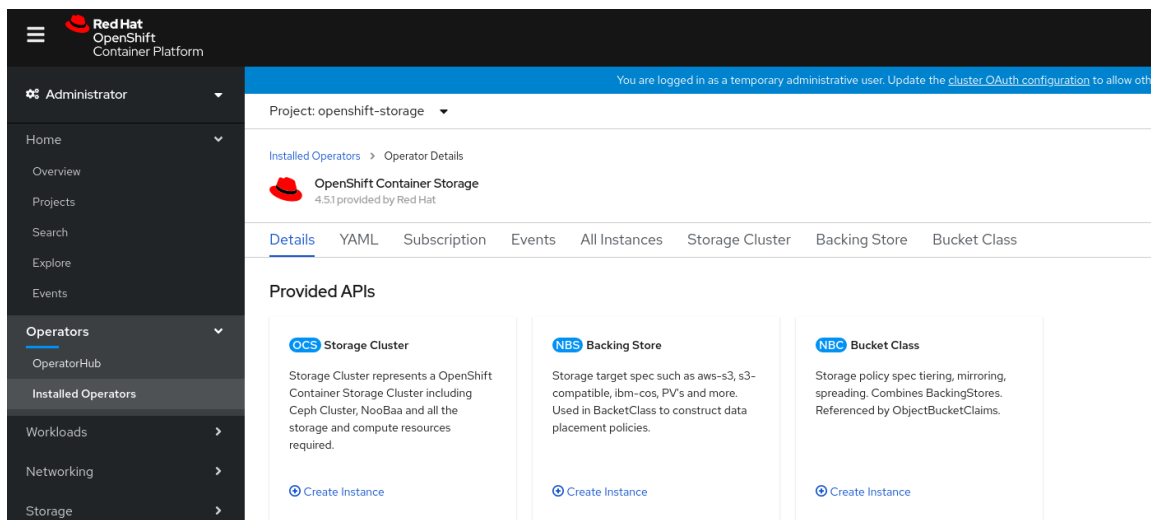
2. Installed Operators ページで、OpenShift Container Storage をクリックします。

図1.2 OpenShift Container Storage Operator ページ



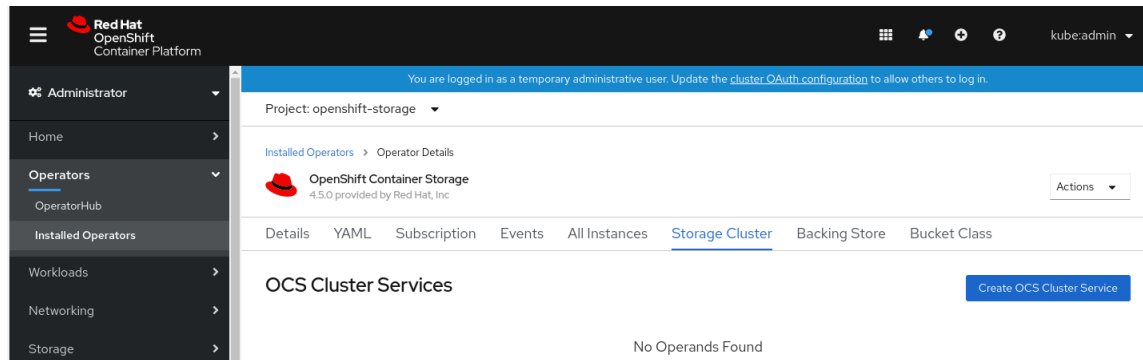
3. Installed Operators → Operator Details ページで、以下のいずれかを実行して Storage Cluster Service を作成します。
 - a. Details タブで Provided APIs → OCS Storage Cluster で、Create Instance をクリックします。

図1.3 Operator Details ページ



- b. または、Storage cluster タブを選択し、Create OCS Cluster Service をクリックします。

図1.4 Storage Cluster タブ



4. Create Storage Cluster ページで、以下のオプションが選択されていることを確認します。

図1.5 Create Storage Cluster ページ

Project: openshift-storage

OpenShift Container Storage > Create OCS Cluster Service

Create Storage Cluster

OCS runs as a cloud-native service for optimal integration with applications in need of storage, and handles the scenes such as provisioning and management.

Select Mode

Internal
 External

Nodes

Selected nodes will be labeled with `cluster.ocs.openshift.io/openshift-storage=""` to create the OCS Service unless they are already labeled.

A bucket will be created to provide the OCS Service.

Select at least 3 nodes in different failure domains with minimum requirements of 16 CPUs and 64 GiB of RAM per node.
3 selected nodes are used for initial deployment. The remaining selected nodes will be used by OpenShift as scheduling targets for OCS scaling.

Name Search by name...

<input checked="" type="checkbox"/>	Name	Role	Location	CPU	Memory
<input checked="" type="checkbox"/>	ip-10-0-154-195.us-east-2.compute.internal	worker	us-east-2a	16	61.03 GiB
<input checked="" type="checkbox"/>	ip-10-0-184-22.us-east-2.compute.internal	worker	us-east-2b	16	60.37 GiB
<input checked="" type="checkbox"/>	ip-10-0-200-66.us-east-2.compute.internal	worker	us-east-2c	16	60.37 GiB

3 nodes selected

Storage Class: gp2

OCS Service Capacity: 2 TiB

Create Cancel

- デフォルトでは、Select Mode に **Internal** が選択されています。
- Nodes** セクションでは、OpenShift Container Storage サービスを使用するには、利用可能な一覧から3つ以上のワーカーノードを選択します。
複数のアベイラビリティゾーンを持つクラウドプラットフォームの場合は、ノードが異なる場所/アベイラビリティゾーンに分散されていることを確認します。



注記

クラスターで特定のワーカーノードを見つけるには、Name または Label に基づいてノードをフィルターできます。

- Name では、ノード名で検索できます。
- Label では、事前に定義されたラベルを選択して検索できます。

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. **Storage Class** は、AWS ではデフォルトで **gp2** に設定されます。
- d. ドロップダウンリストから **OCS Service Capacity** を選択します。



注記

初期ストレージ容量を選択すると、クラスターの拡張は、選択された使用可能な容量を使用してのみ実行されます (raw ストレージの 3 倍)。

- 5. **Create** をクリックします。



注記

Create ボタンは、最低でも 3 つのワーカーノードを選択した後にのみ有効になります。

デプロイメントに成功すると、3 つのストレージデバイスを持つストレージクラスターが作成されます。これらのデバイスは、選択したノードの 3 つに分散されます。この設定では、3 のレプリケーション係数が使用されます。初期クラスターをスケーリングするには、「[ストレージノードのスケーリング](#)」を参照してください。

検証手順

- 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。

第2章 ローカルストレージデバイスを使用したデプロイメント

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。

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 Amazon EC2 ストレージ最適化 I3 に OpenShift Container Storage をインストールします。



重要

ローカルストレージ Operator を使用した Amazon EC2 ストレージ最適化 I3 インスタンスへの OpenShift Container Storage のインストール機能はテクノロジープレビュー機能です。テクノロジープレビュー機能は Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。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。Red Hat OpenShift Container Storage デプロイメントでは、3つのワーカーノードで実行されるアプリケーションやその他のワークロードを使用せずに、新規クラスターを使用することが想定されます。アプリケーションは追加のワーカーノードで実行されます。

2.1. 内部ローカルストレージを使用したデプロイの概要

ローカルストレージデバイスを使用した OpenShift Container Storage をデプロイするには、以下を実行します。

1. [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確認](#)します。
2. Red Hat Enterprise Linux ベースのホストについては、[Red Hat Enterprise Linux ベースのノードでのコンテナのファイルシステムのアクセスを有効](#)にします。



注記

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

3. [Red Hat OpenShift Container Storage Operator をインストール](#)します。
4. [ローカルストレージ Operator をインストール](#)します。
5. [利用可能なストレージデバイスを見つけ](#)ます。
6. [Amazon EC2 \(ストレージ最適化: i3en.2xlarge インスタンスタイプ\) で OpenShift Container Storage クラスターを作成](#)します。

2.2. ローカルストレージデバイスを使用した OPENSIFT CONTAINER STORAGE のインストール要件

- クラスターに、それぞれローカルで割り当てられたストレージデバイスを持つ OpenShift Container Platform ワーカーノードを 3 つ以上設定する必要があります。

- 3つのノードのそれぞれには、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。
- 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。
- 使用するデバイスは空である必要があります。つまり、ディスクには PV、VG、または LV がない状態でなければなりません。
- 3つ以上のラベルが付けられたノードが必要です。
 - ノードが複数のアベイラビリティゾーンプラットフォームの異なる場所/アベイラビリティゾーンに分散されていることを確認します。
 - OpenShift Container Storage によって使用されるローカルストレージデバイスを持つ各ノードには、OpenShift Container Storage Pod をデプロイするための特定のラベルが必要です。ノードにラベルを付けるには、以下のコマンドを使用します。

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

- Red Hat OpenShift Container Storage のローカルストレージ Operator の使用と競合するストレージノードでローカルにマウントされたストレージを管理するストレージプロバイダーは使用しないでください。
- ローカルストレージ Operator が Red Hat OpenShift Container Storage で完全にサポートされるために、ローカルストレージ Operator のバージョンは Red Hat OpenShift Container Platform バージョンと一致する必要があります。ローカルストレージ Operator は、Red Hat OpenShift Container Platform のアップグレード時にアップグレードされません。

2.3. 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) の Red Hat Enterprise Linux ベースに OpenShift Container Platform をデプロイしても、基礎となる Ceph ファイルシステムへのコンテナアクセスは自動的に提供されません。



注記

このプロセスは、Red Hat Enterprise Linux CoreOS をベースとするホストには不要です。

手順

クラスター内の各ノードで以下の手順を実行します。

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. 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
```

3. 必要なパッケージをインストールします。

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

4. 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
```

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

前提条件

- OpenShift Container Platform クラスターにログインしている必要がある。
- OpenShift Container Platform クラスターにワーカーノードが少なくとも 3 つ必要です。



注記

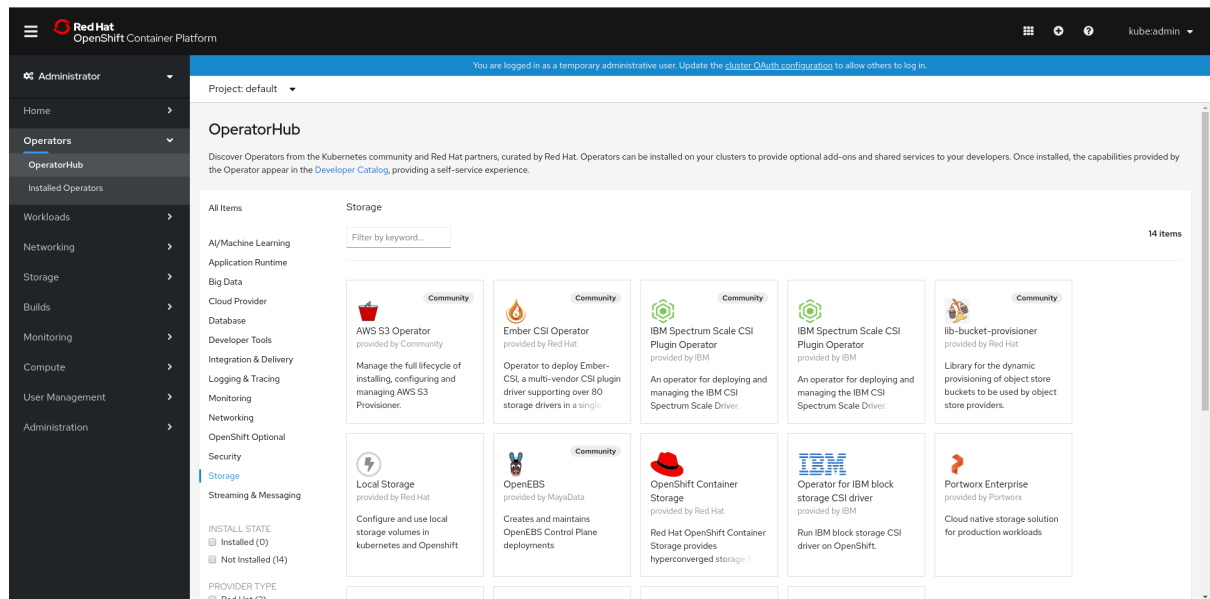
OpenShift Container Storage のクラスター全体でのデフォルトノードセレクターを上書きする必要がある場合は、コマンドラインインターフェースで以下のコマンドを使用し、**openshift-storage** namespace の空のノードセレクターを指定できます。

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

手順

1. OpenShift Web コンソールの左側のペインで、**Operators → OperatorHub** をクリックします。

図2.1 Operator Hub の Operator 一覧



2. OpenShift Container Storage をクリックします。
Filter by keyword テキストボックスまたはフィルター一覧を使用して、Operator の一覧から OpenShift Container Storage を検索できます。
3. OpenShift Container Storage Operator ページで、Install をクリックします。
4. Install Operator ページで、以下のオプションが選択されていることを確認します。
 - a. Channel を **stable-4.5**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 PR 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** として選択している。承認ストラテジーはデフォルトで **Automatic** に設定されます。
 - Approval Strategy に **Automatic** を選択します。



注記

Approval Strategy を **Automatic** として選択すると、新規インストール時、または OpenShift Container Storage の最新バージョンへの更新時に承認は必要ありません。

- i. **インストール** をクリックします。
- ii. インストールが開始するまで待機します。これには、最長 20 分の時間がかかる可能性があります。
- iii. **Operators** → **Installed Operators** をクリックします。
- iv. **Project** が **openshift-storage** であることを確認します。デフォルトで、**プロジェクト** は **openshift-storage** です。

- v. OpenShift Container Storage の Status が Succeeded に変更するまで待機します。
- Approval Strategy に Manual を選択します。



注記

Approval Strategy を **Manual** として選択すると、新規インストール時、または OpenShift Container Storage の最新バージョンへの更新時に承認が必要になります。

- i. **Install** をクリックします。
- ii. **Installed Operators** ページで、**ocs-operator** をクリックします。
- iii. **Subscription Details** ページで、**Install Plan** リンクをクリックします。
- iv. **InstallPlan Details** ページで、**Preview Install Plan** をクリックします。
- v. インストール計画を確認し、**Approve** をクリックします。
- vi. **Components** の Status が **Unknown** から **Created** または **Present** のいずれかに変更するまで待機します。
- vii. **Operators** → **Installed Operators** をクリックします。
- viii. **Project** が **openshift-storage** であることを確認します。デフォルトで、**プロジェクト** は **openshift-storage** です。
- ix. OpenShift Container Storage の Status が Succeeded に変更するまで待機します。

検証手順

- OpenShift Container Storage Operator の Status が Installed Operators ダッシュボードで **Succeeded** と表示されることを確認します。

2.5. ローカルストレージ OPERATOR のインストール

以下の手順を使用して、OpenShift Container Storage クラスターをローカルストレージデバイスに作成する前に Operator Hub からローカルストレージ Operator をインストールします。

前提条件

- 以下のように、**openshift-storage** という namespace を作成します。
 - a. OpenShift Web コンソールの左側のペインで、**Administration** → **Namespaces** をクリックします。
 - b. **Create Namespace** をクリックします。
 - c. Create Namespace ダイアログボックスで、Name に **local-storage** と入力します。
 - d. **Default Network Policy** に **No restrictions** オプションを選択します。
 - e. **Create** をクリックします。

手順

1. OpenShift Web コンソールの左側のペインで、**Operators → OperatorHub** をクリックします。
2. Operator の一覧から **Local Storage Operator** を検索し、これをクリックします。
3. **Install** をクリックします。

図2.2 Install Operator ページ

The screenshot shows the 'Install Operator' page for 'Local Storage' in the OperatorHub. The breadcrumb is 'OperatorHub > Operator Installation'. The title is 'Install Operator'. Below the title is a note: 'Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.'

Update Channel *

- 4.2
- 4.2-s390x
- 4.3
- 4.4
- 4.5

Installation Mode *

- All namespaces on the cluster (default)
This mode is not supported by this Operator
- A specific namespace on the cluster
Operator will be available in a single namespace only.

Installed Namespace *

PR local-storage

Approval Strategy *

- Automatic
- Manual

Buttons: Install, Cancel

Local Storage
provided by Red Hat

Provided APIs

- LV Local Volume**
Manage local storage volumes for OpenShift

4. **Install Operator** ページで、以下のオプションが選択されていることを確認します。
 - a. Channel を **stable-4.5**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 を **local-storage** に選択します。
 - d. Approval Strategy に **Automatic** を選択します。
5. **Install** をクリックします。
6. Local Storage Operator のステータスが **Succeeded** と表示されていることを確認します。

2.6. 利用可能なストレージデバイスの検索

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

手順

1. OpenShift Container Storage ラベルの付いた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
ip-10-0-135-71.us-east-2.compute.internal Ready worker 6h45m v1.16.2
ip-10-0-145-125.us-east-2.compute.internal Ready worker 6h45m v1.16.2
ip-10-0-160-91.us-east-2.compute.internal Ready worker 6h45m v1.16.2
```

2. OpenShift Container Storage リソースに使用される各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の **by-id** デバイス名を見つけます。

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

出力例:

```
$ oc debug node/ip-10-0-135-71.us-east-2.compute.internal
Starting pod/ip-10-0-135-71us-east-2computeinternal-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
xvda                                202:0  0 120G  0 disk
|-xvda1                             202:1  0  384M  0 part /boot
|-xvda2                             202:2  0  127M  0 part /boot/efi
|-xvda3                             202:3  0    1M  0 part
`-xvda4                             202:4  0 119.5G  0 part
   `-coreos-luks-root-nocrypt 253:0  0 119.5G  0 dm  /sysroot
nvme0n1                             259:0  0  2.3T  0 disk
nvme1n1                             259:1  0  2.3T  0 disk
```

この例では、利用可能なローカルデバイスは **nvme0n1** および **nvme1n1** です。

3. 手順 2 で選択した各デバイスの一意の ID を特定します。

```
sh-4.4# ls -l /dev/disk/by-id/ | grep Storage
lrwxrwxrwx. 1 root root 13 Mar 17 16:24 nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC -> ../../nvme0n1
lrwxrwxrwx. 1 root root 13 Mar 17 16:24 nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS60382E5D7441494EC -> ../../nvme1n1
```

上記の例では、2つのローカルデバイスの ID は以下になります。

- nvme0n1: nvme-Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC
- nvme1n1: nvme-Amazon_EC2_NVMe_Instance_Storage_AWS60382E5D7441494EC

4. 上記の手順を繰り返し、OpenShift Container Storage 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トを参照してください](#)。

2.7. AMAZON EC2 (ストレージ最適化: I3EN.2XLARGE インスタンスタイプ) での OPENSIFT CONTAINER STORAGE クラスターの作成

以下の手順を使用して、Amazon EC2 (ストレージ最適化: i3en.2xlarge インスタンスタイプ) インフラストラクチャーに OpenShift Container Storage クラスターを作成します。以下が実行されます。

1. **LocalVolume** CR を使用して PV を作成する
2. 新しい **StorageClass** を作成する

Amazon EC2 (ストレージ最適化: i3en.2xlarge インスタンスタイプ) には、2つの NVMe (non-volatile memory express) ディスクが含まれます。この手順の例では、このインスタンスタイプと共に提供される2つのディスクの使用方法について説明します。

AmazonEC2 I3 の一時ストレージを使用する場合は、以下を行います。

- 3つのアベイラビリティゾーンを使用し、すべてのデータを失うリスクを軽減する。
- ec2:StopInstances パーミッションを持つユーザーの数を制限し、インスタンスを誤ってシャットダウンすることを回避する。

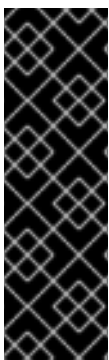


警告

OpenShift Container Storage の永続ストレージに Amazon EC2 I3 の一時ストレージを使用することは推奨されません。3つのノードをすべて停止するとデータ損失が発生する可能性があるためです。

Amazon EC2 I3 の一時ストレージは、以下のようなシナリオでのみ使用することが推奨されます。

- 特定のデータ処理 (data crunching) のためにデータがある場所からコピーされるクラウドバースト (時間に制限がある) が想定される場合。
- 開発環境またはテスト環境が想定される場合。



重要

ローカルストレージ Operator を使用した Amazon EC2 ストレージ最適化 i3en.2xlarge インスタンスの OpenShift Container Storage のインストール機能はテクノロジープレビュー機能です。テクノロジープレビュー機能は Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。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。

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。
- OpenShift Container Platform ワーカーノードに OpenShift Container Storage ラベルを付けられていることを確認します。これは **nodeSelector** として使用されます。

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

出力例:

```
ip-10-0-135-71.us-east-2.compute.internal
ip-10-0-145-125.us-east-2.compute.internal
ip-10-0-160-91.us-east-2.compute.internal
```

手順

1. **LocalVolume** カスタムリソース (CR) を使用してストレージノードにローカル永続ボリューム (PV) を作成します。
OpenShift Storage Container ラベルをノードセレクターおよび **by-id** デバイス識別子として使用する **LocalVolume** CR **local-storage-block.yaml** の例

```
apiVersion: local.storage.openshift.io/v1
kind: LocalVolume
metadata:
  name: local-block
  namespace: local-storage
  labels:
    app: ocs-storagecluster
spec:
  tolerations:
    - key: "node.ocs.openshift.io/storage"
      value: "true"
      effect: NoSchedule
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: cluster.ocs.openshift.io/openshift-storage
            operator: In
            values:
              - ""
  storageClassDevices:
    - storageClassName: localblock
      volumeMode: Block
      devicePaths:
        - /dev/disk/by-id/nvme-
        Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC # <-- modify this line
          - /dev/disk/by-id/nvme-
        Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84FE3E9 # <-- modify this line
          - /dev/disk/by-id/nvme-
        Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE4 # <-- modify this line
          - /dev/disk/by-id/nvme-
        Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441464EP # <-- modify this line
          - /dev/disk/by-id/nvme-
```

```

Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84F43E7 # <-- modify this line
- /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE8 # <-- modify this line

```

各 Amazon EC2 I3 インスタンスには 2 つのディスクがあり、この例ではそれぞれのノードで両方のディスクを使用します。

2. **LocalVolume** CR を作成します。

```
$ oc create -f local-storage-block.yaml
```

出力例:

```
localvolume.local.storage.openshift.io/local-block created
```

3. Pod が作成されているかどうかを確認します。

```
$ oc -n local-storage get pods
```

出力例:

```

NAME                                READY STATUS  RESTARTS  AGE
local-block-local-diskmaker-59rmn  1/1   Running  0         15m
local-block-local-diskmaker-6n7ct  1/1   Running  0         15m
local-block-local-diskmaker-jwtsn  1/1   Running  0         15m
local-block-local-provisioner-6ssxc 1/1   Running  0         15m
local-block-local-provisioner-swwvx 1/1   Running  0         15m
local-block-local-provisioner-zmv5j 1/1   Running  0         15m
local-storage-operator-7848bbd595-686dg 1/1   Running  0         15m

```

4. PV が作成されているかどうかを確認します。

3 つのワーカーノード上の各ローカルストレージデバイスの新規 PV が表示される必要があります。ワークノードごとに利用可能な 2 つのストレージデバイス (各ノードに 2.3 TiB のサイズが設定される) について説明している、[利用可能なストレージデバイスの検索](#)についてのセクションの例を参照してください。

```
$ oc get pv
```

出力例:

```

NAME                                CAPACITY ACCESS MODES RECLAIM POLICY STATUS    CLAIM
STORAGECLASS REASON AGE
local-pv-1a46bc79 2328Gi RWO      Delete    Available localblock 14m
local-pv-429d90ee 2328Gi RWO      Delete    Available localblock
14m
local-pv-4d0a62e3 2328Gi RWO      Delete    Available localblock
14m
local-pv-55c05d76 2328Gi RWO      Delete    Available localblock 14m
local-pv-5c7b0990 2328Gi RWO      Delete    Available localblock 14m
local-pv-a6b283b 2328Gi RWO      Delete    Available localblock 14m

```

5. **LocalVolume** CR の作成時に表示される新規の **StorageClass** の有無を確認します。この **StorageClass** は、以下の手順で **StorageCluster** PVC を提供するために使用されます。

-


```
$ oc get sc | grep localblock
```

出力例:

```
NAME          PROVISIONER          RECLAIMPOLICY
VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE
localblock    kubernetes.io/no-provisioner Delete
WaitForFirstConsumer false                15m
```

6. ローカルストレージ Operator で作成される PV を消費するために **localblock** StorageClass を使用する **StorageCluster** CR を作成します。
monDataDirHostPath および **localblock** StorageClass を使用する **StorageCluster** CR **ocs-cluster-service.yaml** の例。

```
apiVersion: ocs.openshift.io/v1
kind: StorageCluster
metadata:
  name: ocs-storagecluster
  namespace: openshift-storage
spec:
  manageNodes: false
  resources:
    mds:
      limits:
        cpu: 3
        memory: 8Gi
      requests:
        cpu: 1
        memory: 8Gi
  monDataDirHostPath: /var/lib/rook
  storageDeviceSets:
  - count: 2
    dataPVCTemplate:
      spec:
        accessModes:
        - ReadWriteOnce
        resources:
          requests:
            storage: 2328Gi
        storageClassName: localblock
        volumeMode: Block
    name: ocs-deviceset
    placement: {}
    portable: false
    replica: 3
    resources:
      limits:
        cpu: 2
        memory: 5Gi
      requests:
        cpu: 1
        memory: 5Gi
```



重要

OSD でノード全体に Guaranteed サイズが保証されるようにするには、**storageDeviceSets** のストレージサイズを、ノードで作成される PV のサイズ以下に指定する必要があります。

7. **StorageCluster** CR を作成します。

```
$ oc create -f ocs-cluster-service.yaml
```

出力例

```
storagecluster.ocs.openshift.io/ocs-cluster-service created
```

検証手順

[OpenShift Container Storage インストールの検証](#) について参照してください。

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

このセクションを使用して、OpenShift Container Storage が正常に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

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

手順

1. OpenShift Web コンソールの左側のペインから **Workloads** → **Pods** をクリックします。
2. **Project** ドロップダウンリストから **openshift-storage** を選択します。
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。
3. **Running** および **Completed** タブをクリックして、以下の Pod が実行中および完了状態にあることを確認します。

表3.1 OpenShift 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	ocs-operator-* (任意のワーカーノードに 1Pod)
Rook-ceph Operator	rook-ceph-operator-* (任意のワーカーノードに 1Pod)
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) ● nooba-db-* (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)

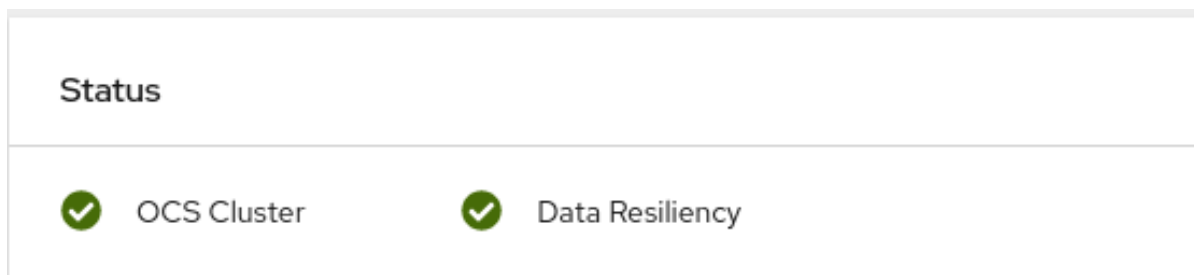
コンポーネント	対応する Pod
MGR	rook-ceph-mgr-* (任意のストレージノードに 1Pod)
MDS	rook-ceph-mds-ocs-storagecluster-cephfilesystem-* (ストレージノードに分散する 2 Pod)
CSI	<ul style="list-style-type: none"> ● cephfs <ul style="list-style-type: none"> ○ csi-cephfsplugin-* (各ワーカーノードに 1Pod) ○ csi-cephfsplugin-provisioner-* (ストレージノードに分散する 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-drain-canary	rook-ceph-drain-canary-* (各ストレージノードに 1Pod)
rook-ceph-crashcollector	rook-ceph-crashcollector-* (各ストレージノードに 1Pod)
OSD	<ul style="list-style-type: none"> ● rook-ceph-osd-* (各デバイス用に 1Pod) ● rook-ceph-osd-prepare-ocs-deviceset-* (各デバイス用に 1Pod)

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

永続ストレージダッシュボードを使用して OpenShift Container Storage クラスタの正常性を確認できます。詳細は、『[OpenShift Container Storage のモニタリング](#)』を参照してください。

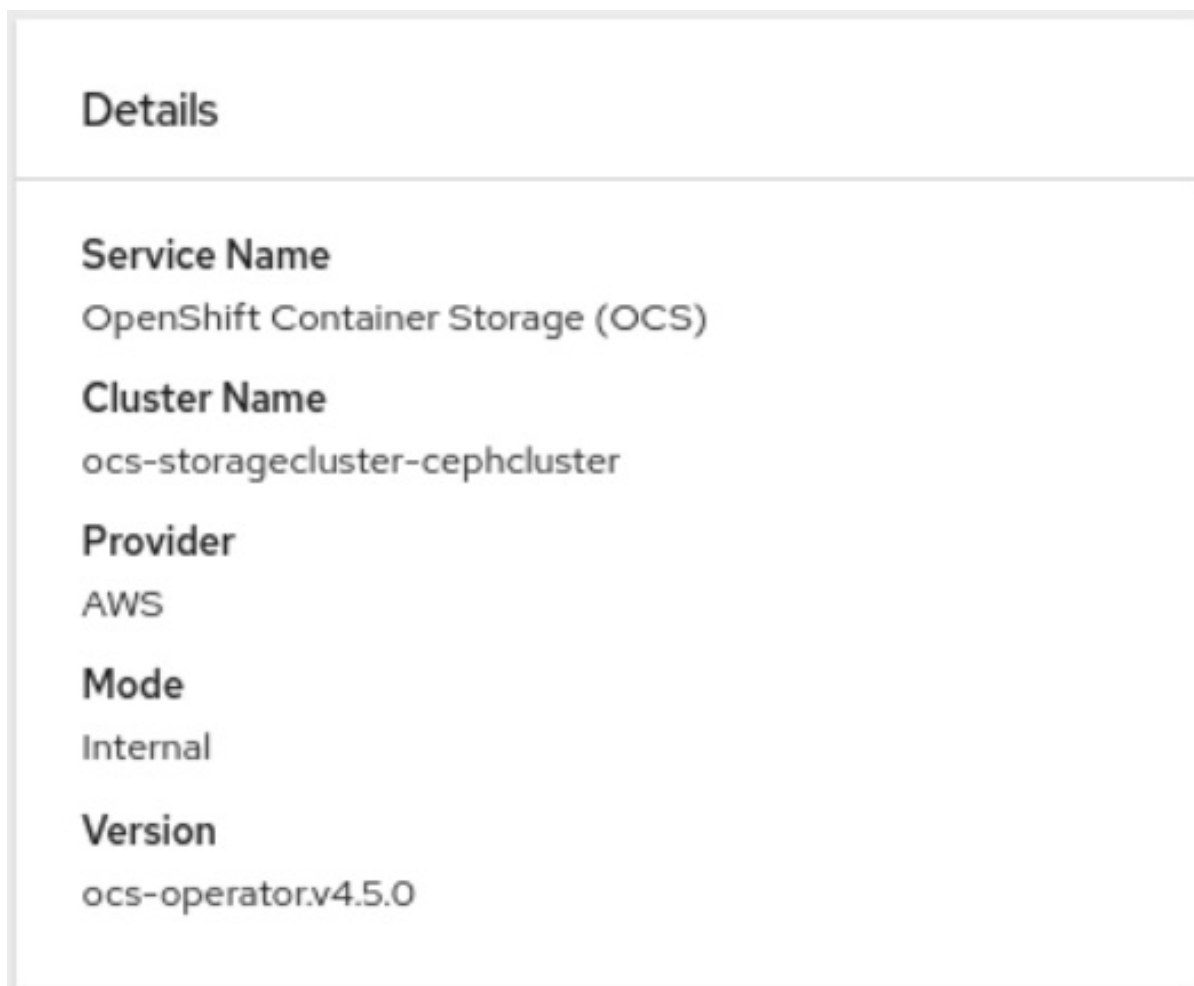
- OpenShift Web コンソールの左側のペインから **Home** → **Overview** をクリックし、**Persistent Storage** タブをクリックします。
- **Status** カードで、以下の画像のように **OCS Cluster** に緑色のチェックマークが表示されていることを確認します。

図3.1 Persistent Storage Overview ダッシュボードの Health status カード



- Details カード で、以下のようにクラスター情報が適切に表示されていることを確認します。

図3.2 Persistent Storage Overview ダッシュボードの Details カード

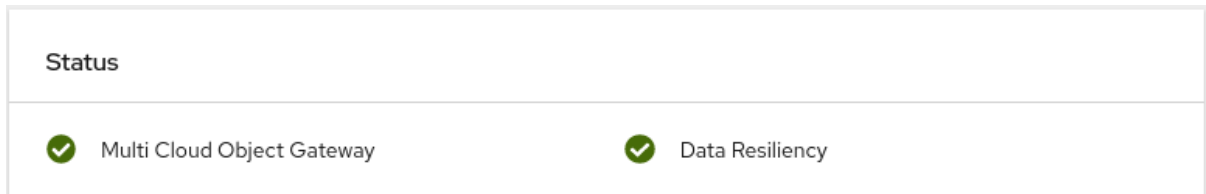


3.3. MULTICLOUD OBJECT GATEWAY が正常であることの確認

オブジェクトサービスダッシュボードを使用して、OpenShift Container Storage クラスターの正常性を確認できます。詳細は、『[OpenShift Container Storage のモニタリング](#)』を参照してください。

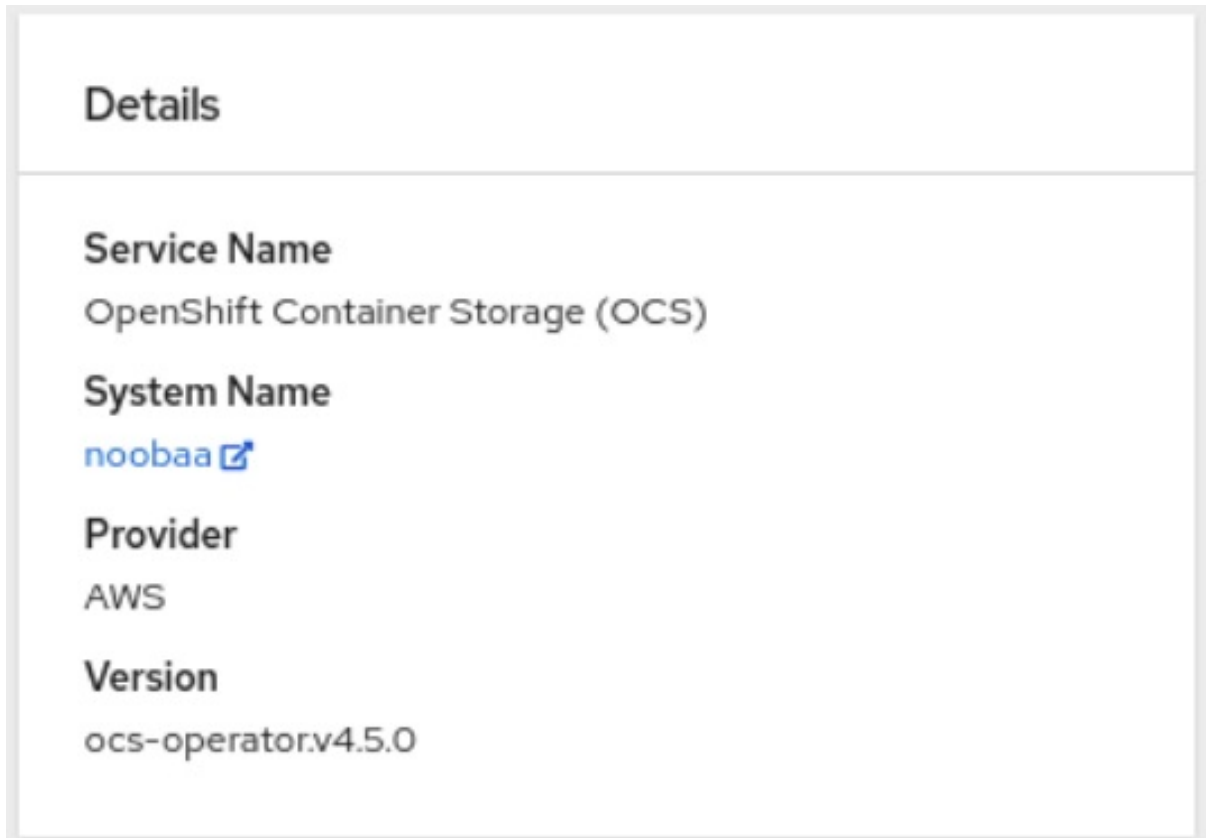
- OpenShift Web コンソールの左側のペインから **Home** → **Overview** をクリックし、**Object Service** タブをクリックします。
- **Status** カード で、以下のように Multicloud Object Gateway (MCG) ストレージに緑色のチェックマークが表示されていることを確認します。

図3.3 Object Service Overview ダッシュボードの Health status カード



- Details カード で、MCG 情報が以下のように適切に表示されることを確認します。

図3.4 Object Service Overview ダッシュボードの Details カード



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、以下を実行します。

- 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。
- 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**

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

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を使用して、ユーザーインターフェースから Uninstall オプションを使用せずに OpenShift Container Storage をアンインストールします。

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。一部の Pod がリソースまたはノードの不足により正常に終了しないと、削除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 Persistent Volume Claim（永続ボリューム要求、PVC）または Object Bucket Claim（オブジェクトバケット要求）を使用していないことを確認します。PVC および OBC はアンインストールプロセスで削除されます。

手順

1. OpenShift Container Storage ベースのストレージクラスプロビジョナーを使用する PVC および OBC をクエリーします。
以下は例になります。

```
$ oc get pvc -o=jsonpath='{range .items[?(@.spec.storageClassName=="ocs-storagecluster-ceph-rbd")]}{"Name: "}{@.metadata.name}{ " Namespace: "}{@.metadata.namespace}{ " Labels: "}{@.metadata.labels}{ "\n"}{end}' --all-namespaces|awk '! ( /Namespace: openshift-storage/ && /app:noobaa/ ) | grep -v noobaa-default-backing-store-noobaa-pvc
```

```
$ oc get pvc -o=jsonpath='{range .items[?(@.spec.storageClassName=="ocs-storagecluster-cephfs")]}{"Name: "}{@.metadata.name}{ " Namespace: "}{@.metadata.namespace}{ "\n"}{end}' --all-namespaces
```

```
$ oc get obc -o=jsonpath='{range .items[?(@.spec.storageClassName=="openshift-storage.noobaa.io")]}{"Name: "}{@.metadata.name}{ " Namespace: "}{@.metadata.namespace}{ "\n"}{end}' --all-namespaces
```

2. 以下の手順に従って、直前の手順に記載されている PVC および OBC が削除されていることを確認します。
モニタリングスタック、クラスターロギング Operator、またはイメージレジストリーの設定の一部として PVC を作成した場合は、必要に応じて以下のセクションで説明されているクリーンアップ手順を実行する必要があります。
 - [「OpenShift Container Storage からのモニタリングスタックの削除」](#)
 - [「OpenShift Container Storage からの OpenShift Container Platform レジストリーの削除」](#)
 - [「OpenShift Container Storage からのクラスターロギング Operator の削除」](#)
残りの PVC または OBC のそれぞれに、以下の手順を実行します。

- a. PVC または OBC を使用する Pod を判別します。
- b. **Deployment**、**StatefulSet**、**DaemonSet**、**Job**、またはカスタムコントローラーなどの制御する側の API オブジェクトを特定します。
各 API オブジェクトには、**OwnerReference** として知られるメタデータフィールドがあります。これは、関連付けられたオブジェクトの一覧です。**controller** フィールドが true に設定された **OwnerReference** は、**ReplicaSet**、**StatefulSet**、**DaemonSet** などの制御するオブジェクトを参照します。
- c. API オブジェクトが OpenShift Container Storage によって提供される PVC または OBC を使用していないことを確認します。オブジェクトを削除するか、ストレージを置き換える必要があります。プロジェクトオーナーに、オブジェクトを安全に削除または変更できることを確認するよう依頼します。



注記

noobaa Pod は無視できます。

- d. OBC を削除します。

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

- e. 作成したカスタムバケットクラスを削除します。

```
$ oc get bucketclass -A | grep -v noobaa-default-bucket-class
```

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

- f. カスタム Multi Cloud Gateway バックイングストアを作成している場合は、それらを削除します。

- バックイングストアの一覧を表示し、これらをメモします。

```
for bs in $(oc get backingstore -o name -n openshift-storage | grep -v noobaa-  
default-backing-store); do echo "Found backingstore $bs"; echo "Its has the  
following pods running :"; echo "$(oc get pods -o name -n openshift-storage |  
grep $(echo ${bs} | cut -f2 -d/))"; done
```

- 上記の各バックイングストアを削除し、依存するリソースも削除されていることを確認します。

```
for bs in $(oc get backingstore -o name -n openshift-storage | grep -v noobaa-  
default-backing-store); do echo "Deleting Backingstore $bs"; oc delete -n  
openshift-storage $bs; done
```

- 上上記のバックイングストアのいずれかが pv-pool をベースとする場合、対応する Pod および PVC も削除してください。

```
$ oc get pods -n openshift-storage | grep noobaa-pod | grep -v noobaa-default-  
backing-store-noobaa-pod
```

```
$ oc get pvc -n openshift-storage --no-headers | grep -v noobaa-db | grep  
noobaa-pvc | grep -v noobaa-default-backing-store-noobaa-pvc
```


- g. 手順 1 に記載されている残りの PVC を削除します。

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

3. バッキングローカルボリュームオブジェクトを一覧表示し、これをメモします。結果がない場合は、手順 7 と 8 に進みます。

```
$ for sc in $(oc get storageclass|grep 'kubernetes.io/no-provisioner' |grep -E $(oc get
storagecluster -n openshift-storage -o jsonpath='{
.items[*].spec.storageDeviceSets[*].dataPVCTemplate.spec.storageClassName}' | sed 's/
/\/g')| awk '{ print $1 }');
do
  echo -n "StorageClass: $sc ";
  oc get storageclass $sc -o jsonpath="{ 'LocalVolume: ' }{
.metadata.labels['local\.storage\.openshift\.io/owner-name'] } { '\n' }";
done
```

出力例:

```
StorageClass: localblock LocalVolume: local-block
```

4. **StorageCluster** 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
```

5. 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、別のプロジェクトに切り替える必要があります。

- a. openshift-storage がアクティブな namespace の場合に別の namespace に切り替えます。以下は例になります。

```
$ oc project default
```

- b. openshift-storage namespace を削除します。

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

- c. 約 5 分間待機し、プロジェクトが正常に削除されたかどうかを確認します。

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

出力:

```
Error from server (NotFound): namespaces "openshift-storage" not found
```



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。

6. 各ノードでストレージ Operator アーティファクトをクリーンアップします。

```
$ 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 rm -rfv /var/lib/rook; done
```

削除されたディレクトリ `/var/lib/rook` が出力に表示されることを確認します。

ディレクトリが存在しないことを確認します。

```
$ 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
```

7. デプロイメント時に作成されたローカルボリュームを削除し、手順 3 に記載されている各ローカルボリュームについてこれを繰り返します。
ローカルボリュームごとに、以下を実行します。

- a. 変数 **LV** を LocalVolume の名前に設定し、変数 **SC** を手順 3 に一覧表示されている StorageClass の名前に設定します。
以下に例を示します。

```
$ LV=local-block
```

```
$ SC=localblock
```

- b. 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 local-storage $LV -o jsonpath='{.spec.storageClassDevices[*].devicePaths[*]}'
```

出力例:

```
/dev/disk/by-id/nvme-xxxxxx  
/dev/disk/by-id/nvme-yyyyyy  
/dev/disk/by-id/nvme-zzzzzz
```

- c. 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
```

- d. 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
```

- e. 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 ...
```

8. 手順3に一覧表示されている各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. 手順7(ii)で収集されたディスクパスを引用符内の **DISKS** 変数に保存します。

```
sh-4.2# DISKS="/dev/disk/by-id/nvme-xxxxxx
/dev/disk/by-id/nvme-yyyyyy /dev/disk/by-id/nvme-zzzzzz"
```

- 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:

```
Problem opening /dev/disk/by-id/nvme-xxxxxx for reading! Error is 2.
The specified file does not exist!
Problem opening " for writing! Program will now terminate.
Warning! MBR not overwritten! Error is 2!
Problem opening /dev/disk/by-id/nvme-yyyyyy for reading! Error is 2.
The specified file does not exist!
Problem opening " for writing! Program will now terminate.
Warning! MBR not overwritten! Error is 2!
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
NOTE
Ignore file-not-found warnings as they refer to disks that are on other machines.
```

- e. シェルを終了し、他のノードについて手順を繰り返します。

```
sh-4.4# exit
exit
sh-4.2# exit
exit

Removing debug pod ...
```

9. **openshift-storage.noobaa.io** ストレージクラスを削除します。

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

10. 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-
```



注記

label <label> not found のようなラベルが解除されているノードについて表示される警告は無視できます。

11. すべての PV が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 | egrep 'ocs-storagecluster-ceph-rbd|ocs-storagecluster-cephfs'
```

```
# oc delete pv <pv name>
```

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
storageclusterinitializations.ocs.openshift.io storageclusters.ocs.openshift.io
cephclients.ceph.rook.io --wait=true --timeout=5m
```



注記

AWS で OpenShift Container Storage クラスターをアンインストールすると、ターゲットバケットに保存されているすべての OpenShift Container Storage データが削除されますが、ユーザーによって作成されたターゲットバケットや、OpenShift Container Storage のインストール時に自動的に作成されたターゲットバケットは削除されず、OpenShift Container Storage に属していないデータはこれらのターゲットバケットに残ります。

13. 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. **Home** → **Overview** をクリックし、ダッシュボードにアクセスします。
 - b. **Persistent Storage** および **Object Service** タブが **Cluster** タブの横に表示されないことを確認します。

4.2. OPENSIFT CONTAINER STORAGE 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 Container Storage 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 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せん。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
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	ocs-storagecluster-ceph-rbd	8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-1	40Gi	RWO	Bound	pvc-0d5a9825-15a5-11ea-baa0-026d231574aa	ocs-storagecluster-ceph-rbd	8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-2	40Gi	RWO	Bound	pvc-0d6413dc-15a5-11ea-baa0-026d231574aa	ocs-storagecluster-ceph-rbd	8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-0	40Gi	RWO	Bound	pvc-0b7c19b0-15a5-11ea-baa0-026d231574aa	ocs-storagecluster-ceph-rbd	8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-1	40Gi	RWO	Bound	pvc-0b8aed3f-15a5-11ea-baa0-026d231574aa	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は、「[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

<pre> . . . storage: pvc: claim: registry-cephfs-rwx-pvc . . . </pre>
編集後
<pre> . . . storage: . . . </pre>

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

このセクションでは、クラスターロギング Operator を OpenShift Container Storage 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 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
```