



Red Hat OpenShift Container Storage 4.5

VMWare vSphere での OpenShift Container Storage のデプロイ

Red Hat OpenShift Container Storage の Red Hat OpenShift Container Platform
VMWare vSphere クラスターへのデプロイ方法

Red Hat OpenShift Container Storage 4.5 VMWare vSphere での OpenShift Container Storage のデプロイ

Red Hat OpenShift Container Storage の Red Hat OpenShift Container Platform VMWare vSphere
クラスターへのデプロイ方法

法律上の通知

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

Red Hat OpenShift Container Platform VMWare vSphere クラスターで 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	14
2.6. 利用可能なストレージデバイスの検索	16
2.7. VMWARE での OPENSIFT CONTAINER STORAGE クラスターの作成	17
第3章 内部モードの OPENSIFT CONTAINER STORAGE デプロイメントの確認	22
3.1. POD の状態の確認	22
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 PLATFORM 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	32
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) vSphere 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

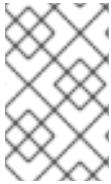
VMware vSphere では、内部と外部の両方の Openshift Container Storage クラスターがサポートされます。デプロイメントの要件についての詳細は、『[デプロイメントのプランニング](#)』を参照してください。

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
 - [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](#)
- [外部モード](#)

第1章 動的ストレージデバイスを使用したデプロイ

VMWare vSphere (ディスク形式: thin) 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

VMware vSphere 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。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。

デプロイメントの以下の手順に従います。

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や出力がない場合は、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 (OCP) クラスターにログインしている必要があります。
- OCP 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

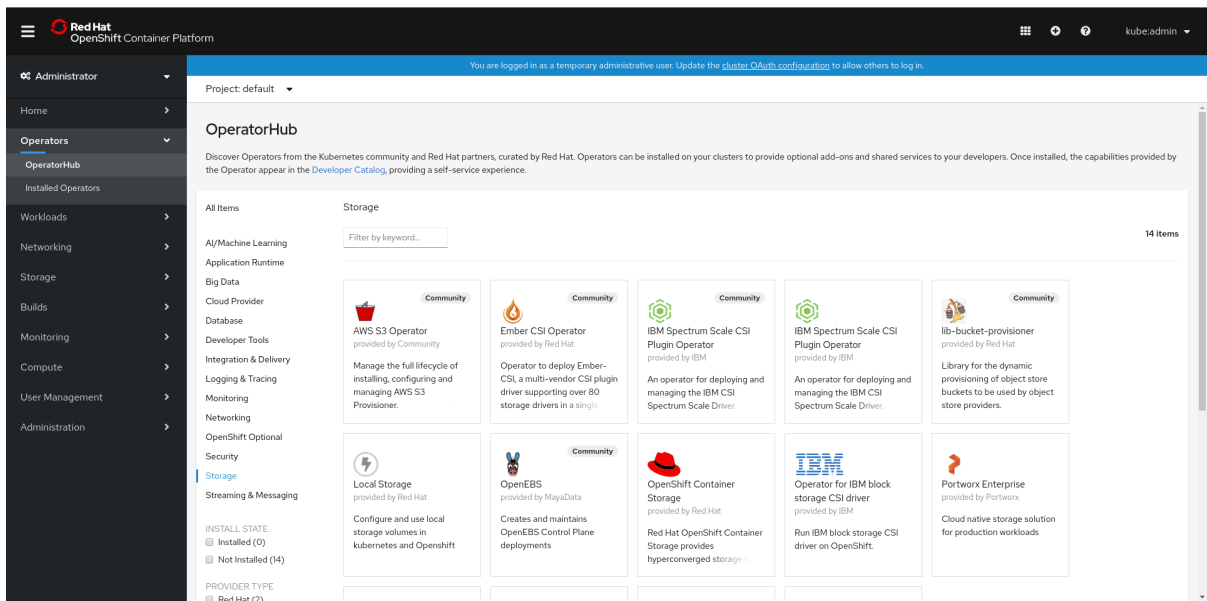
OpenShift Container Storage のクラスター全体でのデフォルトノードセレクターを上書きする必要がある場合は、コマンドラインインターフェースで以下のコマンドを使用し、**openshift-storage** namespace の空のノードセレクターを指定できます。

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

手順

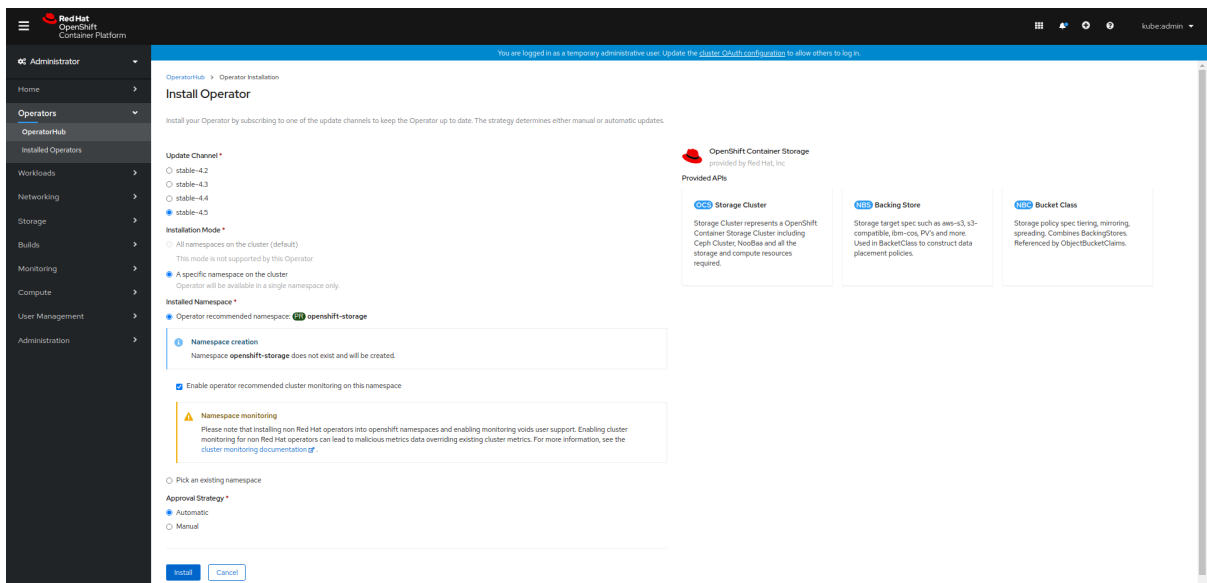
1. OpenShift Web コンソールの左側のペインで、**Operators → OperatorHub** をクリックします。

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。

図1.2 Install Operator ページ

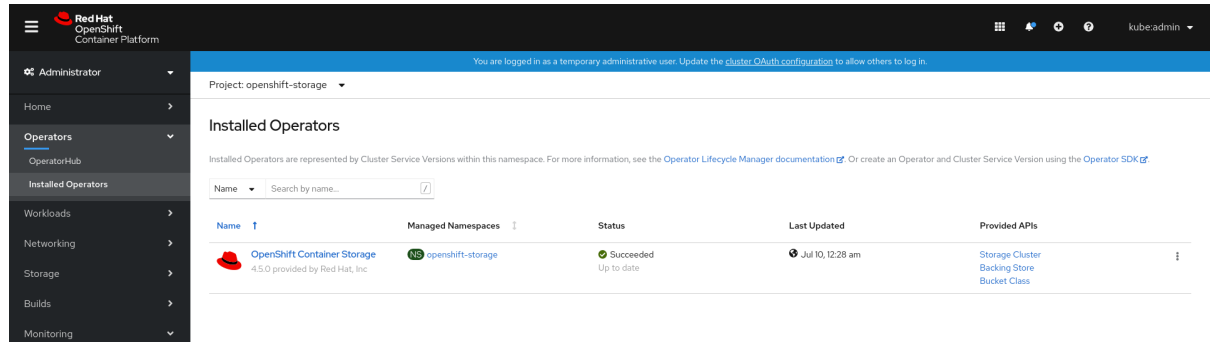


4. Install Operator ページで、以下のオプションが選択されていることを確認します。
 - a. Channel を **stable-4.5**として更新します。
 - b. Installation Mode オプションに **A specific namespace on the cluster**を選択します。
 - c. Installed Namespace に **Operator recommended namespace PR openshift-storage** を選択します。namespace **openshift-storage** が存在しない場合、これは Operator のインストール時に作成されます。
 - d. **Enable operator recommended cluster monitoring on this namespace**が選択されていることを確認します。これはクラスタのモニタリングに必要です。

e. Approval Strategy に **Automatic** を選択します。

5. **Install** をクリックします。

図1.3 Installed Operators ダッシュボード



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

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

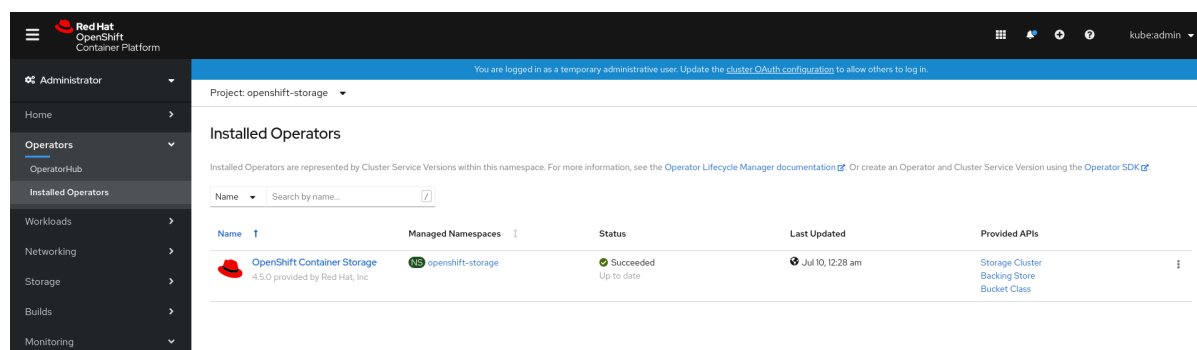
前提条件

- OpenShift Container Storage Operator 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。
- VMware の仮想マシンでは、**disk.EnableUUID** オプションが **TRUE** に設定されていることを確認してください。仮想マシンを設定するには、vCenter アカウントの権限が必要です。詳細は、「[Required vCenter account privileges](#)」を参照してください。**disk.EnableUUID** オプションを設定するには、**Customize hardware** タブで **VM Options** の **Advanced** オプションを使用します。詳細は、「[Creating Red Hat Enterprise Linux CoreOS \(RHCOS\) machines in vSphere](#)」を参照してください。

手順

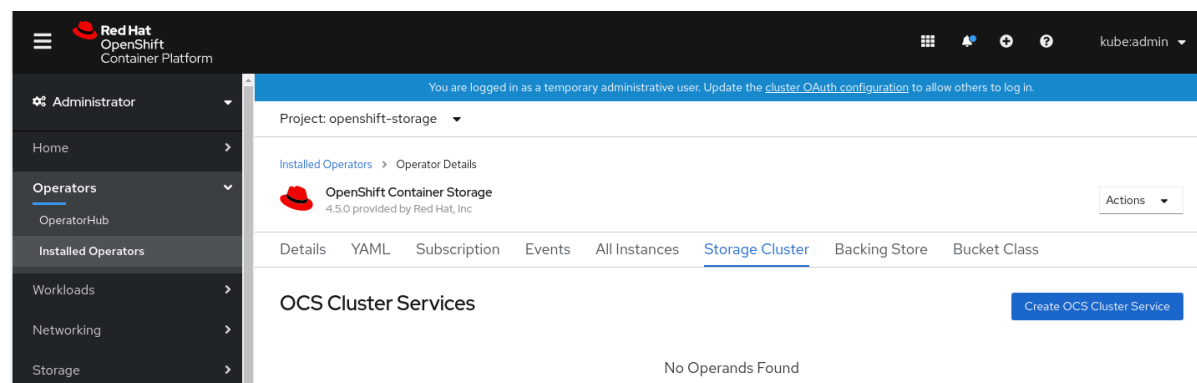
1. OpenShift Web コンソールの左側のペインで **Operators** → **Installed Operators** をクリックし、インストールされた Operator を表示します。

図1.4 OpenShift Container Storage Operator ページ



2. Installed Operators ページで、インストールされた OpenShift Container Storage Operator の Provided APIs 列で **Storage Cluster** リンクをクリックします。

図1.5 Storage Cluster タブ



3. Create OCS Cluster Service をクリックします。

図1.6 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"/>	compute-0	worker	-	16	61.81 GiB
<input checked="" type="checkbox"/>	compute-1	worker	-	16	61.81 GiB
<input checked="" type="checkbox"/>	compute-2	worker	-	16	61.81 GiB

3 nodes selected

Storage Class: thin

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ために、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。
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することが推奨されます。



注記

- クラスタ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。
- vCenter の非アフィニティーを使用して OpenShift Container Storage のラックラベルをデータセンターの物理ノードおよびラックラベルに合わせて調整し、同じ物理シャーシに 2 つのワーカーノードがスケジュールされないようにします。

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。

6. **Storage Class** は、VMware ではデフォルトで **thin** に設定されます。
7. 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)。

8. **Create** をクリックします。
Create ボタンは、3 つのノードを選択した後にのみ有効になります。選択したノードごとに、3 つのストレージデバイスを持つ新しいストレージクラスターが作成されます。デフォルト設定では、レプリケーション係数 3 を使用します。

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 VMware インフラストラクチャーに OpenShift Container Storage をインストールします。

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. [VMware での OpenShift Container Storage クラスターサービスの作成](#)。

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。
 - 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することが推奨されます。

- 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や出力がない場合は、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 polycoreutils 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 (OCP) クラスターにログインする必要があります。
- OCP 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

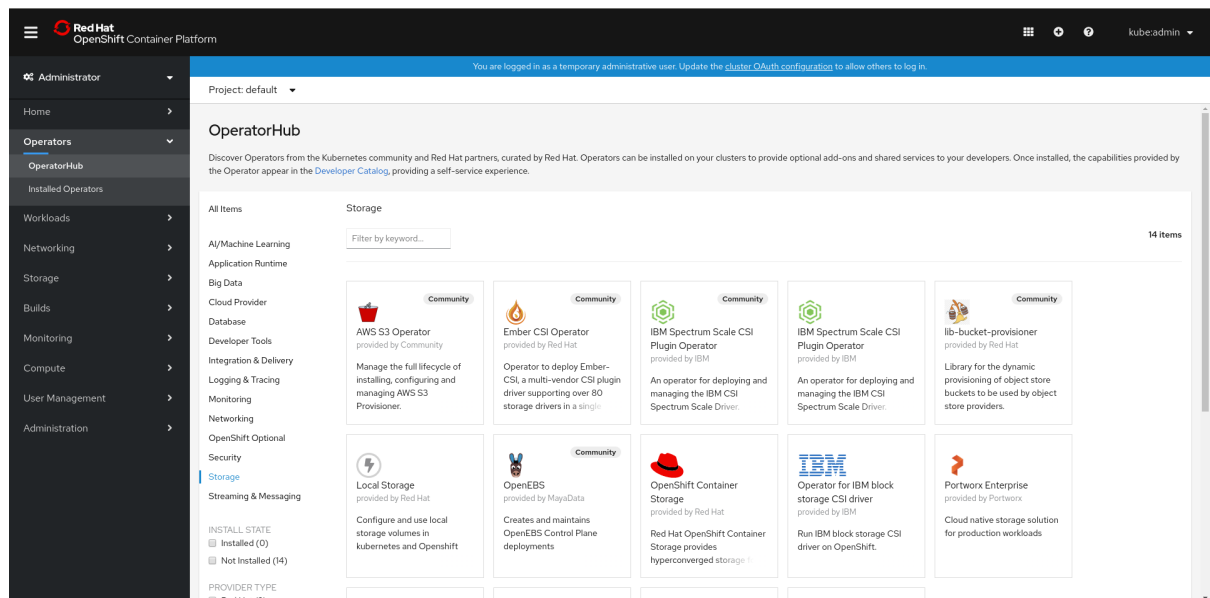
OpenShift Container Storage のクラスター全体でのデフォルトノードセクターを上書きする必要がある場合は、コマンドラインインターフェースで以下のコマンドを使用し、**openshift-storage** namespace の空のノードセクターを指定できます。

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

手順

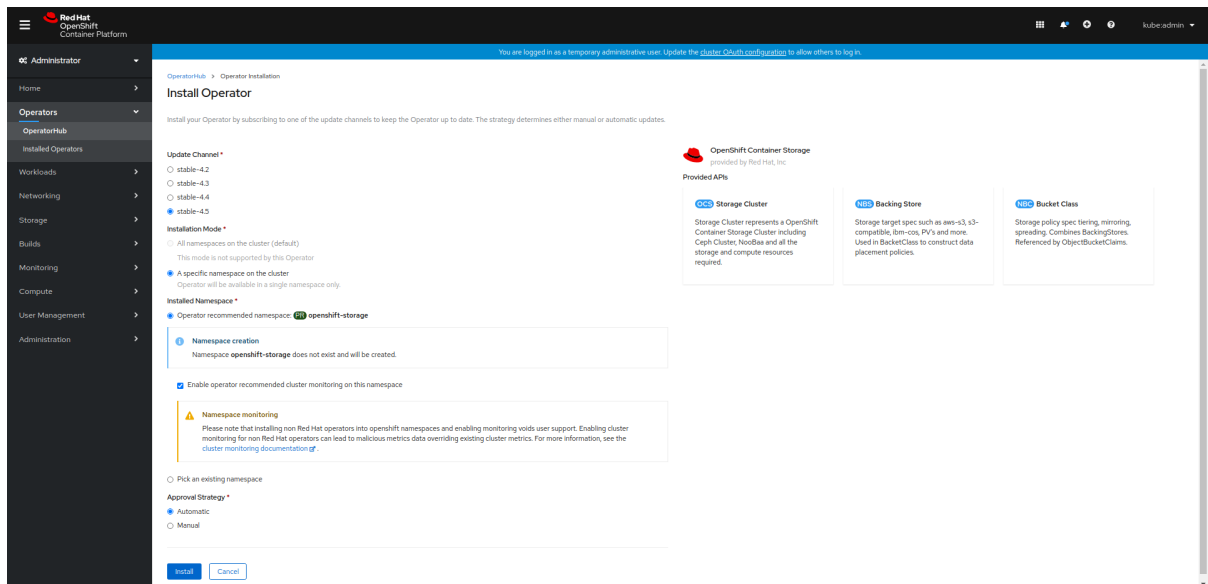
1. OpenShift Web コンソールの左側のペインで、**Operators → OperatorHub** をクリックします。

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。

図2.2 Install Operator ページ

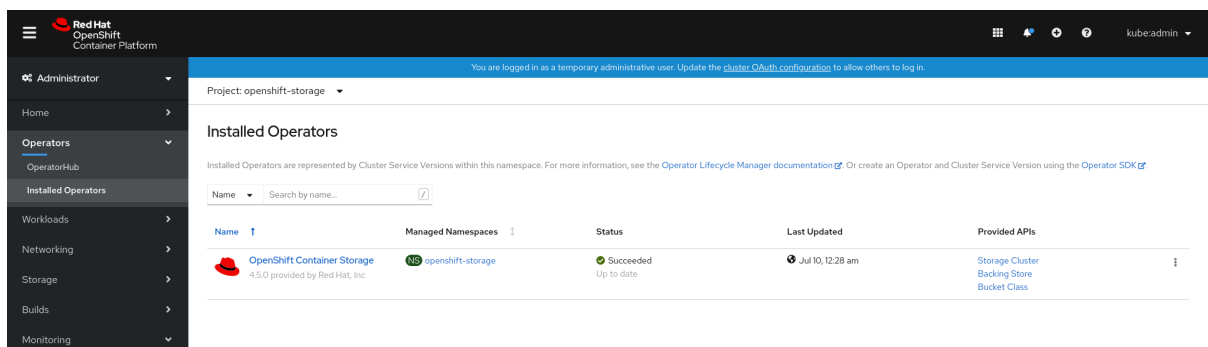


4. Install Operator ページで、以下のオプションが選択されていることを確認します。

- Channel を **stable-4.5**として更新します。
- Installation Mode オプションに **A specific namespace on the cluster**を選択します。
- 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。
- Enable operator recommended cluster monitoring on this namespace**が選択されていることを確認します。これはクラスタのモニタリングに必要です。
- Approval Strategy に **Automatic** を選択します。

5. Install をクリックします。

図2.3 Installed Operators ダッシュボード



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 **local-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2.4 Install Operator ページ

The screenshot shows the 'Install Operator' page for the 'Local Storage Operator'. The breadcrumb is 'OperatorHub > Operator Installation'. The title is 'Install Operator'. Below the title, there is a description: '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.'

The configuration options are as follows:

- Update Channel ***: Radio buttons for 4.2, 4.2-s390x, 4.3, 4.4, and 4.5. The 4.5 option is selected.
- Installation Mode ***: Radio buttons for 'All namespaces on the cluster (default)' and 'A specific namespace on the cluster'. The 'A specific namespace on the cluster' option is selected. A note below it says 'Operator will be available in a single namespace only.'
- Installed Namespace ***: A dropdown menu with 'local-storage' selected.
- Approval Strategy ***: Radio buttons for 'Automatic' and 'Manual'. The 'Automatic' option is selected.

On the right side, there is a section for 'Local Storage' provided by Red Hat. Below it, under 'Provided APIs', there is a box for 'Local Volume' with the description 'Manage local storage volumes for OpenShift'.

At the bottom left, there are two buttons: 'Install' (in blue) and 'Cancel' (in light blue).

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. 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。

2.6. 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
compute-0 Ready   worker 106m v1.18.3+2cf11e2
compute-1 Ready   worker 106m v1.18.3+2cf11e2
compute-2 Ready   worker 106m v1.18.3+2cf11e2
```

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覧の **by-id** デバイス名を見つけます。

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

出力例:

```
$ oc debug node/compute-0
Starting pod/compute-0-debug ...
To use host binaries, run `chroot /host`
Pod IP: 10.1.50.36
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
sda                                  8:0  0 120G  0 disk
|-sda1                               8:1  0  384M  0 part /boot
|-sda2                               8:2  0 127M  0 part /boot/efi
|-sda3                               8:3  0   1M  0 part
`-sda4                               8:4  0 119.5G  0 part
`-coreos-luks-root-nocrypt 253:0  0 119.5G  0 dm  /sysroot
nvme0n1                             259:0  0  1.5T  0 disk
```

この例では、**compute-0** の利用可能なローカルデバイスは **nvme0n1** です。

- 手順 2 で選択した各デバイスの一意的 ID を特定します。

```
sh-4.4#
ls -l /dev/disk/by-id/ | grep nvme0n1
lrwxrwxrwx. 1 root root 13 Aug 19 06:41 nvme-
Dell_Express_Flash_NVMe_P4610_1.6TB_SFF_PHLN951601QF1P6AGN -> ../../nvme0n1
lrwxrwxrwx. 1 root root 13 Aug 19 06:41 nvme-eui.01000000010000005cd2e4895e0e5251 -
> ../../nvme0n1
```

上記の例では、ローカルデバイス 'nvme0n1' の ID は以下になります。

```
nvme-eui.01000000010000005cd2e4895e0e5251
```

- 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. VMWARE での OPENSIFT CONTAINER STORAGE クラスターの作成

以下の手順を使用して、VMware インフラストラクチャーにストレージクラスターを作成します。

VMware は、以下の 3 つのタイプのローカルストレージをサポートします。

- 仮想マシンディスク (VMDK)
- raw デバイスマッピング (RDM)
- VMDirectPath I/O

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。
- VMware でローカルストレージデバイスを使用するために、各ノードに同じストレージタイプおよびサイズが割り当てられた 3 つのワーカーノードが必要です。
- VMware の仮想マシンでは、**disk.EnableUUID** オプションが **TRUE** に設定されていることを確認してください。仮想マシンを設定するには、vCenter アカウントの権限が必要です。詳細は、「[Required vCenter account privileges](#)」を参照してください。**disk.EnableUUID** オプションを設定するには、**Customize hardware** タブの **VM Options** の **Advanced** オプションを使用します。詳細は、「[Creating Red Hat Enterprise Linux CoreOS \(RHCOS\) machines in vSphere](#)」を参照してください。
- 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。

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

各ノードのストレージデバイスを特定するには、[利用可能なストレージデバイスの検索](#)について参照してください。

手順

1. ブロック PV の LocalVolume CR を作成します。
OpenShift Container Storage ラベルをノードセレクターとして使用する **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:
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: cluster.ocs.openshift.io/openshift-storage
            operator: In
            values:
              - ""
  storageClassDevices:
    - storageClassName: localblock
      volumeMode: Block
      devicePaths:
        - /dev/disk/by-id/nvme-eui.01000000010000005cd2e4895e0e5251 # <-- modify this
line
        - /dev/disk/by-id/nvme-eui.01000000010000005cd2e4ea2f0f5251 # <-- modify this line
        - /dev/disk/by-id/nvme-eui.01000000010000005cd2e4de2f0f5251 # <-- modify this line

```

2. ブロック PV の **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-5brzv	1/1	Running	0	31s
local-block-local-diskmaker-8sxcs	1/1	Running	0	31s
local-block-local-diskmaker-s7s9p	1/1	Running	0	31s
local-block-local-provisioner-9cbw8	1/1	Running	0	31s
local-block-local-provisioner-cpddv	1/1	Running	0	31s
local-block-local-provisioner-f6h7h	1/1	Running	0	31s
local-storage-operator-75b9776b75-vwdzh	1/1	Running	0	12m

4. 新規 **localblock** StorageClass を確認します。

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

出力例:

```
localblock    kubernetes.io/no-provisioner Delete    WaitForFirstConsumer false
96s
```

5. **Available** のステータスで作成されている PV を確認します。

```
$ oc get pv
```

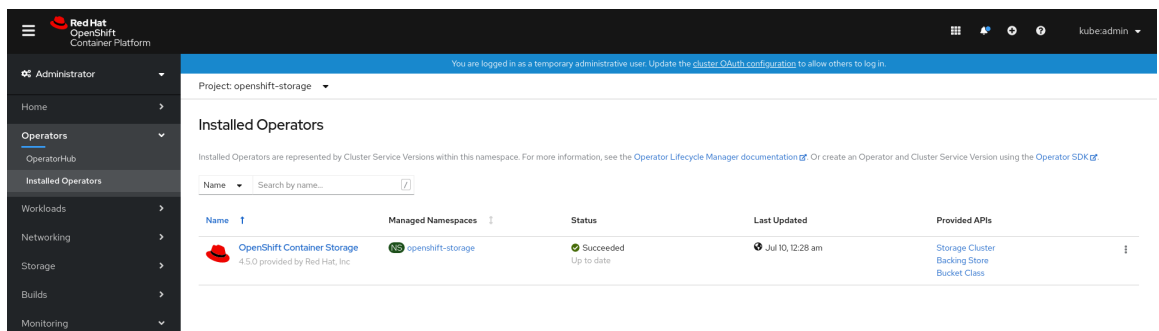
出力例:

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM
STORAGECLASS	REASON	AGE			
local-pv-264b0256	1490Gi	RWO	Delete	Available	localblock
108s					
local-pv-8b0e9b53	1490Gi	RWO	Delete	Available	localblock
99s					
local-pv-8dcc8c60	1490Gi	RWO	Delete	Available	localblock
98s					

6. **localblock** Storage Class ストレージクラスを使用する OpenShift Container Storage Cluster Service を作成します。

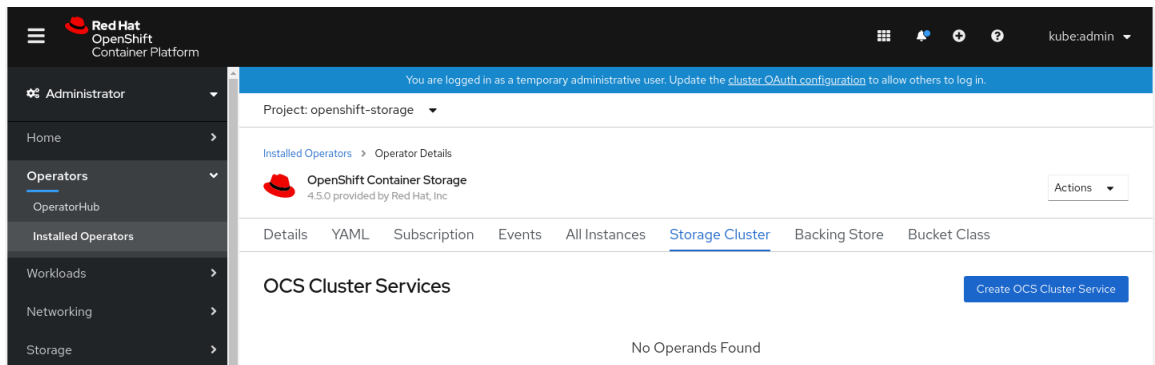
- OpenShift Web コンソールにログインします。
- OpenShift Web コンソールの左側のペインで **Operators** → **Installed Operators** をクリックし、インストールされた Operator を表示します。

図2.5 OpenShift Container Storage Operator ページ

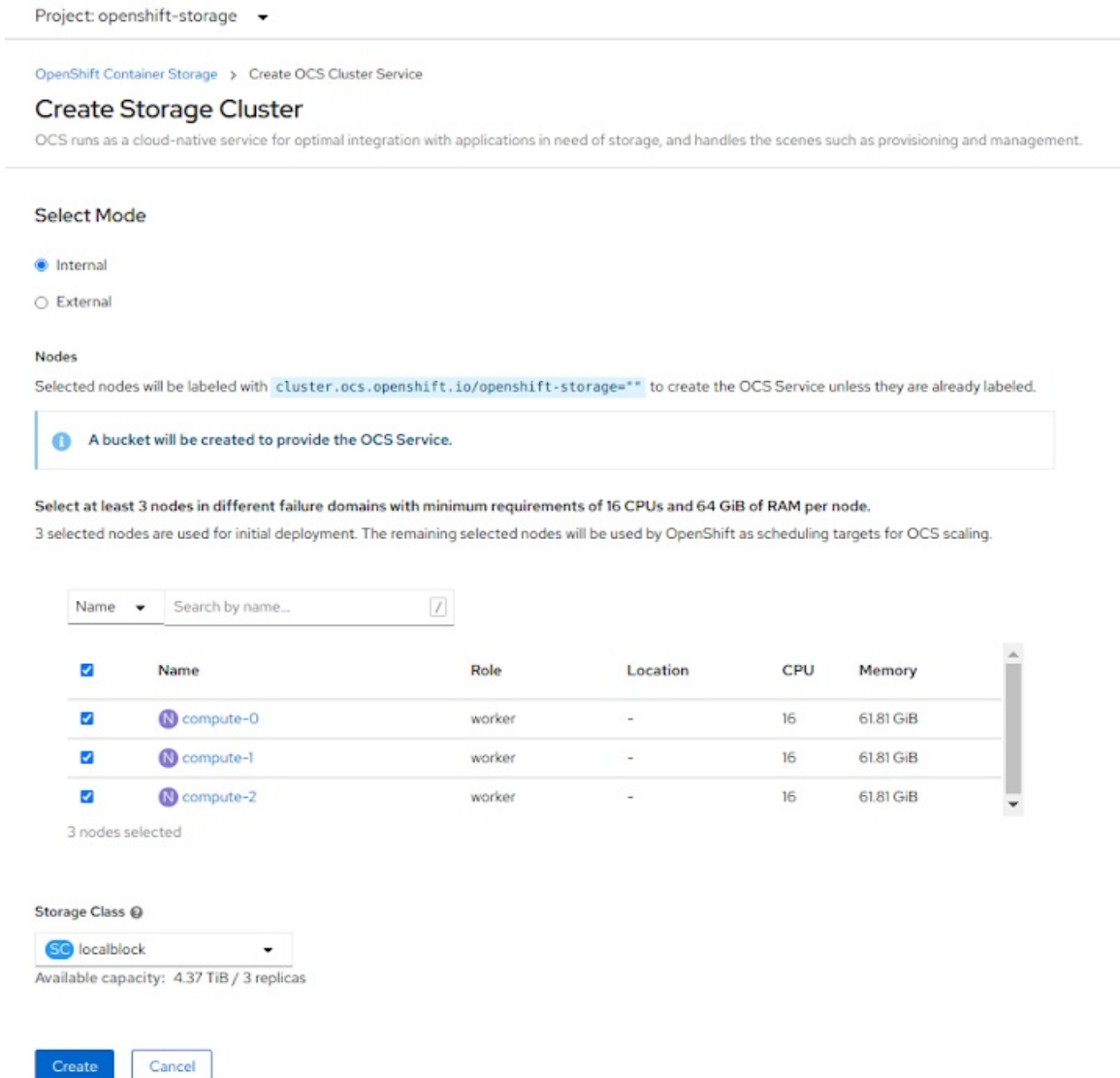


- OpenShift Container Storage インストール Operator をクリックします。
- Operator Details ページで、Storage Cluster リンクをクリックします。

図2.6 Storage Cluster タブ



- e. **Create OCS Cluster Service** をクリックします。



- f. **Select Mode** を **Internal** のままにします。

- g. **Nodes** セクションでは、OpenShift Container Storage サービスを使用するために、利用可能なノードの一覧から3つ以上のワーカーノードを選択します。
 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することが推奨されます。



注記

- クラスタ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。
- vCenter の非アフィニティーを使用して OpenShift Container Storage のラックラベルをデータセンターの物理ノードおよびラックラベルに合わせて調整し、同じ物理シャーシに2つのワーカーノードがスケジュールされないようにします。

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。

h. **Storage Class** ドロップダウンリストから **localblock** を選択します。

i. **Create** をクリックします。

Create ボタンは、3つのノードを選択した後にのみ有効になります。3つのボリュームからなる新規ストレージクラスターは、1ワーカーノードごとに1つのボリュームを設定して作成されます。デフォルト設定では、レプリケーション係数3を使用します。

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 クラスタに対応する Pod」](#) 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手順を実行できます。

1. **openshift-storage** namespace の空のノードセクターを指定します。

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

2. **DaemonSets** によって生成される元の Pod を削除します。

```
oc delete pod -l app=csi-cephfsplugin -n openshift-storage
oc delete pod -l app=csi-rbdplugin -n openshift-storage
```

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)

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 1 Pod) ● noobaa-core-* (任意のストレージノードに 1 Pod) ● nooba-db-* (任意のストレージノードに 1 Pod) ● noobaa-endpoint-* (任意のストレージノードに 1 Pod)
MON	<p>rook-ceph-mon-*</p> <p>(ストレージノード全体に分散する 3 Pod)</p>
MGR	<p>rook-ceph-mgr-*</p> <p>(任意のストレージノードに 1 Pod)</p>
MDS	<p>rook-ceph-mds-ocs-storagecluster-cephfilesystem-*</p> <p>(ストレージノードに分散する 2 Pod)</p>
RGW	<p>rook-ceph-rgw-ocs-storagecluster-cephobjectstore-* (ストレージノードに分散する 2 Pod)</p>
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 1 Pod) ○ csi-cephfsplugin-provisioner-* (ストレージノードに分散する 2 Pod) ● rbd <ul style="list-style-type: none"> ○ csi-rbdplugin-* (各ワーカーノードに 1 Pod) ○ csi-rbdplugin-provisioner-* (ストレージノードに分散する 2 Pod)
rook-ceph-drain-canary	<p>rook-ceph-drain-canary-*</p> <p>(各ストレージノードに 1 Pod)</p>

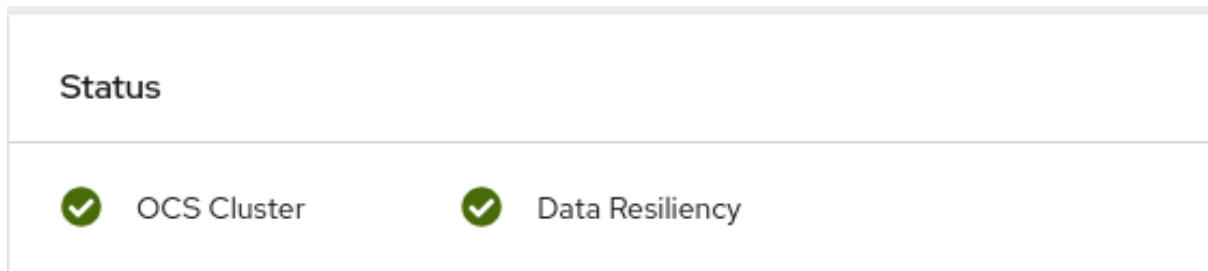
コンポーネント	対応する Pod
rook-ceph-crashcollector	rook-ceph-crashcollector-* (各ストレージノードに1Pod)
OSD	<ul style="list-style-type: none"> ● rook-ceph-osd-* (各デバイスに1Pod) ● rook-ceph-osd-prepare-ocs-deviceset-* (各デバイスに1Pod)

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

Details	
Service Name	OpenShift Container Storage (OCS)
Cluster Name	ocs-storagecluster-cephcluster
Provider	VSphere
Mode	Internal
Version	ocs-operator.v4.5.0

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。

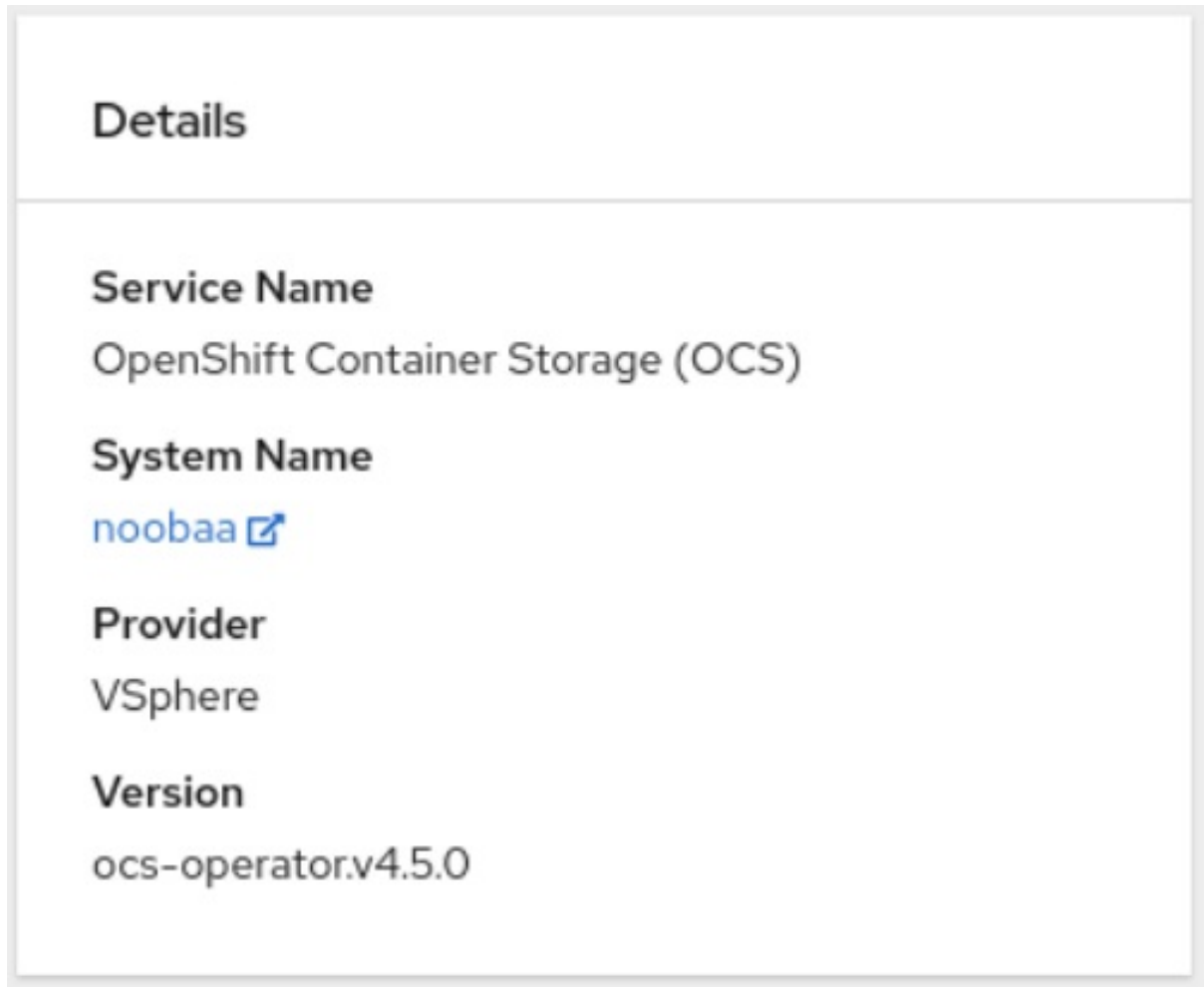
- OpenShift Web コンソールの左側のペインから **Home** → **Overview** をクリックし、**Object Service** タブをクリックします。
- **Status** カードで、以下のように Multicloud Object Gateway (MCG) ストレージに緑色のチェックマークが表示されていることを確認します。

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

Status	
 Multi Cloud Object Gateway	 Data Resiliency

- **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**
 - **ocs-storagecluster-ceph-rgw**

第4章 OPENSIFT CONTAINER PLATFORM 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
```

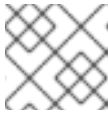
```
$ oc get obc -o=jsonpath='{range .items[?(@.spec.storageClassName=="ocs-storagecluster-ceph-rgw")]}{"Name: "}{@.metadata.name}{ " Namespace: "}{@.metadata.namespace}{ "\n"}{end}' --all-namespaces
```

```
$ 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 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コントロール側の API オブジェクトを特定します。
各オブジェクトには、**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 に設定される **OwnerReference** が、**ReplicaSet**、**StatefulSet**、**DaemonSet** などのコントロール側のオブジェクトを参照します。
 - c. 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。

- i. 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
```

- ii. 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
```

- iii. 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. バックアップローカルボリュームオブジェクトを一覧表示します。結果がない場合は、手順 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. クリーンアップポリシーを設定するために StorageCluster にラベルを付けます。

```
$ oc label -n openshift-storage storagecluster --all cleanup.ocs.openshift.io=yes-really-destroy-data
```

出力:

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

上記のコマンドは、StorageCluster を削除する際に以下を実行します。

- dataDirHostPath でクラスターの namespace をクリーンアップします。
 - dataDirHostPath ですべての ceph monitor ディレクトリーを削除します。
 - 各ノードでデバイスをクリーンアップします。
5. **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
```

6. 生成されるクリーンアップジョブが完了し、すべてのノードがクリーンアップされていることを確認します。
クリーンアップ Pod のステータスを確認します。次の手順に進む前に、**Completed** のステータスが表示されていることを確認します。

```
$ oc get pods -n openshift-storage | grep cleanup
cluster-cleanup-job-compute-0    0/1    Completed    0    21s
cluster-cleanup-job-compute-1    0/1    Completed    0    23s
cluster-cleanup-job-compute-2    0/1    Completed    0    25s
```

cluster-cleanup-job-xxx Pod が作成されなかった場合や、クリーンアップ Pod が **Completed** 状態にない場合は、[Red Hat サポート](#) にお問い合わせください。

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、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。

8. 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 StorageClass の名前に設定します。以下は例になります。

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

```
$ SC=localblock
```

- b. 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
```

- c. 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
```

- d. 該当する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/compute-xx ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'

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

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

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
```

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
す。
詳細は、「[モニタリングスタックの設定](#)」を参照してください。

手順

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

```
$ oc get pod,pvc -n openshift-monitoring
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                                STATUS VOLUME
CAPACITY ACCESS MODES STORAGECLASS      AGE
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-0 Bound pvc-0d519c4f-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
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. PVC を使用する Pod を一覧表示します。

この例では、PVC を使用していた **alertmanagerMain** および **prometheusK8s** Pod は **Terminating** 状態にあります。これらの Pod が OpenShift Container Storage PVC を使用しなくなった後に PVC を削除できます。

```

$ oc get pod,pvc -n openshift-monitoring
NAME                                READY STATUS   RESTARTS AGE
pod/alertmanager-main-0             3/3 Terminating 0    10h
pod/alertmanager-main-1             3/3 Terminating 0    10h
pod/alertmanager-main-2             3/3 Terminating 0    10h
pod/cluster-monitoring-operator-84cd9df668-zhjfn 1/1 Running    0    18h
pod/grafana-5db6fd97f8-pmtbf        2/2 Running    0    10h
pod/kube-state-metrics-895899678-z2r9q 3/3 Running    0    10h
pod/node-exporter-4njxv             2/2 Running    0    18h
pod/node-exporter-b8ckz             2/2 Running    0    11h
pod/node-exporter-c2vp5             2/2 Running    0    18h
pod/node-exporter-cq65n             2/2 Running    0    18h
pod/node-exporter-f5sm7             2/2 Running    0    11h
pod/node-exporter-f852c             2/2 Running    0    18h
pod/node-exporter-l9zn7             2/2 Running    0    11h
pod/node-exporter-ngbs8             2/2 Running    0    18h
pod/node-exporter-rv4v9             2/2 Running    0    18h
pod/openshift-state-metrics-77d5f699d8-69q5x 3/3 Running    0    10h
pod/prometheus-adapter-765465b56-4tbxx 1/1 Running    0    10h
pod/prometheus-adapter-765465b56-s2qg2 1/1 Running    0    10h
pod/prometheus-k8s-0                6/6 Terminating 1    9m47s
pod/prometheus-k8s-1                6/6 Terminating 1    9m47s
pod/prometheus-operator-cbfd89f9-ldnwc 1/1 Running    0    43m
pod/telemeter-client-7b5ddb4489-2xfpz 3/3 Running    0    10h

```

```

NAME                                STATUS VOLUME

```

```

CAPACITY ACCESS MODES STORAGECLASS          AGE
persistentvolumeclaim/ocs-alertmanager-claim-alertmanager-main-0 Bound pvc-
2eb79797-1fed-11ea-93e1-0a88476a6a64 40Gi RWO      ocs-storagecluster-ceph-
rbd 19h
persistentvolumeclaim/ocs-alertmanager-claim-alertmanager-main-1 Bound pvc-
2ebeee54-1fed-11ea-93e1-0a88476a6a64 40Gi RWO      ocs-storagecluster-ceph-
rbd 19h
persistentvolumeclaim/ocs-alertmanager-claim-alertmanager-main-2 Bound pvc-2ec6a9cf-
1fed-11ea-93e1-0a88476a6a64 40Gi RWO      ocs-storagecluster-ceph-rbd 19h
persistentvolumeclaim/ocs-prometheus-claim-prometheus-k8s-0 Bound pvc-3162a80c-
1fed-11ea-93e1-0a88476a6a64 40Gi RWO      ocs-storagecluster-ceph-rbd 19h
persistentvolumeclaim/ocs-prometheus-claim-prometheus-k8s-1 Bound pvc-
316e99e2-1fed-11ea-93e1-0a88476a6a64 40Gi RWO      ocs-storagecluster-ceph-
rbd 19h

```

5. 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

このセクションでは、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: emptyDir: {} . . . </pre>

この例で、PVC は **registry-cephfs-rwx-pvc** と呼ばれ、これは安全に削除することができます。

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 を使用するように設定されている必要があります。

手順

1. namespace にある **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
```