



Red Hat OpenShift Container Storage 4.8

IBM Power Systems を使用した OpenShift Container Storage のデプロイ

IBM Power Systems 環境のインストールおよび設定方法

Red Hat OpenShift Container Storage 4.8 IBM Power Systems を使用した OpenShift Container Storage のデプロイ

IBM Power Systems 環境のインストールおよび設定方法

Enter your first name here. Enter your surname here.

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

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Deploying_OpenShift_Container_Storage_using_IBM_Power_Systems.ent file |.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

Red Hat OpenShift Container Storage 4.8 をインストールし、IBM Power Systems インフラストラクチャーでローカルストレージを使用する方法については、本書をお読みください。

目次

多様性を受け入れるオープンソースの強化	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
はじめに	5
第1章 OPENSIFT CONTAINER STORAGE のデプロイの準備	6
1.1. ローカルストレージデバイスを使用して OPENSIFT CONTAINER STORAGE をインストールするための要件	6
1.2. VAULT でのキー値のバックエンドパスおよびポリシーの有効化	7
第2章 ローカルストレージデバイスを使用した OPENSIFT CONTAINER STORAGE のデプロイ	8
2.1. ローカルストレージ OPERATOR のインストール	8
2.2. RED HAT OPENSIFT CONTAINER STORAGE OPERATOR のインストール	8
2.3. 利用可能なストレージデバイスの検索	10
2.4. IBM POWER SYSTEMS での OPENSIFT CONTAINER STORAGE クラスターの作成	11
第3章 内部モードの OPENSIFT CONTAINER STORAGE デプロイメントの確認	16
3.1. POD の状態の確認	16
3.2. OPENSIFT CONTAINER STORAGE クラスターが正常であることの確認	17
3.3. MULTICLOUD OBJECT GATEWAY が正常であることの確認	18
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	18
第4章 OPENSIFT CONTAINER STORAGE のアンインストール	19
4.1. 内部モードでの OPENSIFT CONTAINER STORAGE のアンインストール	19
4.1.1. ローカルストレージ Operator の設定の削除	24
4.2. OPENSIFT CONTAINER STORAGE からのモニターリングスタックの削除	26
4.3. OPENSIFT CONTAINER STORAGE からの OPENSIFT CONTAINER PLATFORM レジストリーの削除	29
4.4. OPENSIFT CONTAINER STORAGE からのクラスターロギング OPERATOR の削除	30

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[弊社の CTO である Chris Wright のメッセージ](#) を参照してください。

RED HAT ドキュメントへのフィードバック (英語のみ)

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

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
 1. ドキュメントの表示が **Multi-page HTML** 形式になっていることを確認してください。ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
 2. マウスカーソルを使用して、コメントを追加するテキストの部分を強調表示します。
 3. 強調表示されたテキストの下に表示される **Add Feedback** ポップアップをクリックします。
 4. 表示される指示に従ってください。
- より詳細なフィードバックをお寄せいただく場合は、Bugzilla のチケットを作成してください。
 1. [Bugzilla](#) の Web サイトに移動します。
 2. Component (コンポーネント) として **Documentation** を使用します。
 3. **Description** フィールドに、ドキュメントの改善に向けたご提案を記入してください。ドキュメントの該当部分へのリンクも追加してください。
 4. **Submit Bug** をクリックします。

はじめに

Red Hat OpenShift Container Storage 4.8 は、接続環境または非接続環境での既存の Red Hat OpenShift Container Platform (RHOC) IBM Power クラスターへのデプロイメントをサポートし、プロキシ環境に対する追加設定なしのサポートを提供します。



注記

IBM Power Systems では、内部の Openshift Container Storage クラスターのみがサポートされます。導入要件の詳細については、[Planning your deployment](#) および [Preparing to deploy OpenShift Container Storage](#) を参照してください。

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

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

第1章 OPENSIFT CONTAINER STORAGE のデプロイの準備

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



注記

IBM Power Systems 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、[Planning your deployment](#) を参照してください。

ローカルストレージを使用して Red Hat OpenShift Container Storage のデプロイメントを開始する前に、リソース要件を満たしていることを確認してください。[ローカルストレージデバイスを使用して OpenShift Container Storage をインストールするための要件](#) を参照してください。

- 外部の鍵管理システム (KMS) で、以下を実行します。
 - トークンのあるポリシーが存在し、Vault のキー値のバックエンドパスが有効にされていることを確認します。[Vault のキー値のバックエンドパスおよびポリシーの有効化](#) について参照してください。
 - Vault サーバーで署名済みの証明書を使用していることを確認します。

上記を処理した後に、指定した順序で以下の手順を実行します。

1. [ローカルストレージ Operator のインストール](#)。
2. [Red Hat OpenShift Container Storage Operator をインストールする](#)
3. [利用可能なストレージデバイスの検索](#)
4. [IBM Power Systems での OpenShift Container Storage クラスターの作成](#)

1.1. ローカルストレージデバイスを使用して OPENSIFT CONTAINER STORAGE をインストールするための要件

ノードの要件

- クラスターは、それぞれローカルに接続されたストレージデバイスを持つクラスターの3つ以上の OpenShift Container Platform ワーカーノードで設定される必要があります。
 - 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。
 - 使用するデバイスは空である必要があります。つまり、ディスクには永続ボリューム (PV)、ボリュームグループ (VG)、または論理ボリューム (LV) 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。
 - 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="
```

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。

1.2. VAULT でのキー値のバックエンドパスおよびポリシーの有効化

前提条件

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

手順

1. Vault で Key/Value (KV) バックエンドパスを有効にします。
Vault KV シークレットエンジン API の場合は、バージョン 1 を使用します。

```
$ vault secrets enable -path=ocs kv
```

Vault KV シークレットエンジン API の場合は、バージョン 2 です。

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

2. 以下のコマンドを使用して、シークレットでの書き込み操作または削除操作の実行をユーザーを制限するポリシーを作成します。

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

3. 上記のポリシーに一致するトークンを作成します。

```
$ vault token create -policy=ocs -format json
```

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

このセクションを使用して、OpenShift Container Platform がすでにインストールされている IBM Power Systems インフラストラクチャーに OpenShift Container Storage をデプロイします。

指定された順序で以下の手順を実行します。

1. [ローカルストレージ Operator のインストール](#)。
2. [Red Hat OpenShift Container Storage Operator をインストールする](#)
3. [利用可能なストレージデバイスの検索](#)
4. [IBM Power Systems での OpenShift Container Storage クラスターの作成](#)

2.1. ローカルストレージ OPERATOR のインストール

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

手順

1. OpenShift Web コンソールにログインします。
2. **Operators** → **OperatorHub** をクリックします。
3. **Filter by keyword...** ボックスに **local storage** と入力して、オペレータのリストから **Local Storage** オペレーターを検索し、クリックします。
4. **Install** をクリックします。
5. **Install Operator** ページで、以下のオプションを設定します。
 - a. Channel を **stable-4.8** として更新します。
 - b. Installation Mode オプションに **A specific namespace on the cluster** を選択します。
 - c. Installed Namespace に **Operator recommended namespace openshift-local-storage** を選択します。
 - d. Approval Strategy に **Automatic** を選択します。
6. **Install** をクリックします。
7. ローカルストレージ Operator がステータス **Succeeded** を表示していることを確認します。

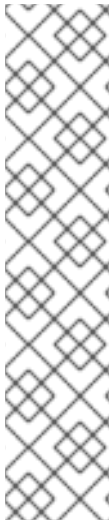
2.2. RED HAT OPENSIFT CONTAINER STORAGE OPERATOR のインストール

Red Hat OpenShift Container Storage は、Red Hat OpenShift Container Platform Operator Hub を使用してインストールできます。

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

- cluster-admin および Operator インストールのパーミッションを持つアカウントを使用して OpenShift Container Platform クラスターにアクセスできること。
- RHOCP 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イスで以下のコマンドを使用し、**openshift-storage** namespace の空のノードセレクターを指定できます (この場合、openshift-storage namespace を作成します)。

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

- ノードに Red Hat OpenShift Container Storage リソースのみがスケジュールされるように、そのノードに **infra** のテイントを設定します。これにより、サブスクリプションコストを節約できます。詳細は、ストレージリソースの管理および割り当てガイドの [Red Hat OpenShift Container Storage に専用のワーカーノードを使用する方法](#) の章を参照してください。

手順

1. OpenShift Web コンソールの左側のペインに移動し、**Operators → OperatorHub** をクリックします。
2. スクロールするか、またはキーワードを Filter by keyword ボックスに入力し、OpenShift Container Storage Operator を検索します。
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、以下の必須オプションがデフォルトで選択されます。
 - a. Channel を **stable-4.8** として更新します。
 - b. Installation Mode オプションに **A specific namespace on the cluster** を選択します。
 - c. Installed Namespace に **Operator recommended namespace openshift-storage** を選択します。namespace **openshift-storage** が存在しない場合、これは Operator のインストール時に作成されます。
 - d. **承認ストラテジー** を **Automatic** または **Manual** として選択します。
 - e. **Install** をクリックします。
Automatic (自動) 更新を選択している場合、Operator Lifecycle Manager (OLM) は介入なしに、Operator の実行中のインスタンスを自動的にアップグレードします。

Manual (手動) 更新を選択している場合、OLM は更新要求を作成します。クラスター管理者は、Operator が新規バージョンに更新されるように更新要求を手動で承認する必要があります。

検証手順

OpenShift Container Storage Operator に、インストールが正常に実行されたことを示す緑色のチェックマークが表示されていることを確認します。

2.3. 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、IBM Power Systems 用に 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 つ以上のワーカーノードのそれぞれのデバイス名を特定します。

手順

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

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

出力例:

```
NAME      STATUS  ROLES  AGE  VERSION
worker-0  Ready   worker 2d11h v1.21.1+f36aa36
worker-1  Ready   worker 2d11h v1.21.1+f36aa36
worker-2  Ready   worker 2d11h v1.21.1+f36aa36
```

2. OpenShift Container Storage リソースに使用される各ワーカーノードにログインし、OpenShift Container Platform をデプロイする際にアタッチした追加ディスクの名前を見つめます。

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

出力例:

```
$ oc debug node/worker-0
Starting pod/worker-0-debug ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.0.63
If you don't see a command prompt, try pressing enter.
sh-4.4#
sh-4.4# chroot /host
sh-4.4# lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop1  7:1  0  500G  0 loop
sda    8:0  0  500G  0 disk
sdb    8:16 0  120G  0 disk
|-sdb1 8:17 0   4M  0 part
|-sdb3 8:19 0  384M  0 part
`-sdb4 8:20 0 119.6G 0 part
sdc    8:32 0  500G  0 disk
sdd    8:48 0  120G  0 disk
|-sdd1 8:49 0   4M  0 part
|-sdd3 8:51 0  384M  0 part
`sdd4 8:52 0 119.6G 0 part
sde    8:64 0  500G  0 disk
sdf    8:80 0  120G  0 disk
|-sdf1 8:81 0   4M  0 part
|-sdf3 8:83 0  384M  0 part
`sdf4 8:84 0 119.6G 0 part
sdg    8:96 0  500G  0 disk
```

```

sdh 8:112 0 120G 0 disk
|-sdh1 8:113 0 4M 0 part
|-sdh3 8:115 0 384M 0 part
`-sdh4 8:116 0 119.6G 0 part
sdi 8:128 0 500G 0 disk
sdj 8:144 0 120G 0 disk
|-sdj1 8:145 0 4M 0 part
|-sdj3 8:147 0 384M 0 part
`-sdj4 8:148 0 119.6G 0 part
sdk 8:160 0 500G 0 disk
sdl 8:176 0 120G 0 disk
|-sdl1 8:177 0 4M 0 part
|-sdl3 8:179 0 384M 0 part
`sdl4 8:180 0 119.6G 0 part /sysroot
sdm 8:192 0 500G 0 disk
sdn 8:208 0 120G 0 disk
|-sdn1 8:209 0 4M 0 part
|-sdn3 8:211 0 384M 0 part /boot
`sdn4 8:212 0 119.6G 0 part
sdo 8:224 0 500G 0 disk
sdp 8:240 0 120G 0 disk
|-sdp1 8:241 0 4M 0 part
|-sdp3 8:243 0 384M 0 part
`sdp4 8:244 0 119.6G 0 part

```

この例では、worker-0 の場合、利用可能な 500G のローカルデバイスは **sda**、**sdc**、**sde**、**sdg**、**sdi**、**sdk**、**sdm**、**sdo** です。

3. 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ワーカーノードについてこの手順を繰り返します。詳細は、[ナレッジベースアトキクル](#) を参照してください。

2.4. IBM POWER SYSTEMS での OPENSIFT CONTAINER STORAGE クラスターの作成

前提条件

- [ローカルストレージデバイスを使用した OpenShift Container Storage のインストールの要件](#) についてのセクションにあるすべての要件を満たしていることを確認します。
- IBM Power Systems でローカルストレージデバイスを使用するために、同じストレージタイプおよびサイズが各ノードに接続された 3 つ以上のワーカーノードが必要です (例: 200 GB SSD)。
- 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

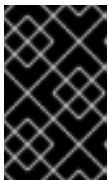
1. OpenShift Web コンソールにログインします。
2. **openshift-local-storage** namespace で、**Operators** → **Installed Operators** をクリックし、インストールされた Operator を表示します。
3. **Local Storage** のインストールされた Operator をクリックします。
4. **Operator Details** ページで、**Local Volume** リンクをクリックします。
5. **Create Local Volume** をクリックします。
6. ローカルボリュームを設定するには、**YAML view** をクリックします。
7. 以下の YAML を使用して、ブロック PV の **LocalVolume** カスタムリソースを定義します。

```

apiVersion: local.storage.openshift.io/v1
kind: LocalVolume
metadata:
  name: localblock
  namespace: openshift-local-storage
spec:
  logLevel: Normal
  managementState: Managed
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - worker-0
              - worker-1
              - worker-2
  storageClassDevices:
    - devicePaths:
        - /dev/sda
      storageClassName: localblock
      volumeMode: Block

```

上記の定義は、**worker-0**、**worker-1**、および **worker-2** ノードから **sda** ローカルデバイスを選択します。**localblock** ストレージクラスが作成され、永続ボリュームが **sda** からプロビジョニングされます。



重要

環境に応じて nodeSelector の適切な値を指定します。デバイス名はすべてのワーカーノードで同一である必要があります。複数の devicePaths を指定することもできます。

8. **Create** をクリックします。
9. **diskmaker-manager** Pod および **Persistent Volumes** が作成されているかどうかを確認します。
 - a. Pod の場合

OpenShift Web コンソールの右側のペインから **Workloads** → **Pods** をクリックしま

- i. OpenShift Web コンソールの左側のヘイブから **workloads** → **Pods** をクリックします。
 - ii. **Project** ドロップダウンリストから **openshift-local-storage** を選択します。
 - iii. LocalVolume CR の作成に使用した各ワーカーノードについて、**diskmaker-manager** Pod があるかどうかを確認します。
- b. 永続ボリュームの場合
- i. OpenShift Web コンソールの左側のペインから **Storage** → **PersistentVolumes** をクリックします。
 - ii. **local-pv**-* 名で永続ボリュームを確認します。永続ボリュームの数は、localVolume CR の作成中にプロビジョニングされたワーカーノードの数とストレージデバイスの数と同じです。



重要

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

10. 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。
11. **Project** ドロップダウンリストから **openshift-storage** を選択します。
12. **OpenShift Container Storage** インストール Operator をクリックします。
13. **Operator Details** ページで、**Storage Cluster** リンクをクリックします。
14. **Create Storage Cluster** をクリックします。
 - a. **Select Mode** に **Internal-Attached devices** を選択します。
 - b. **Storage and Nodes** をクリックします。
 - c. 必要なストレージクラスを選択します。
 - d. ストレージクラスに対応するノードは、ドロップダウンで選択したストレージクラスに基づいて表示されます。
 - e. **Next** をクリックします。
 - f. (オプション) **Security and network** 設定を設定します。
 - i. **Enable encryption** チェックボックスを選択して、ブロックおよびファイルストレージを暗号化します。
 - ii. 1つまたは両方の **Encryption level** を選択します。
 - **クラスター全体の暗号化** クラスター全体 (ブロックおよびファイル) を暗号化します。

- **Storage class encryption**(ストレージクラスの暗号化): 暗号化対応のストレージクラスを使用して暗号化された永続ボリューム (ブロックのみ) を作成します。
- iii. **Connect to an external key management service** チェックボックスを選択します。これはクラスター全体の暗号化の場合はオプションになります。
 - A. **Key Management Service Provider** はデフォルトで **Vault** に設定されます。
 - B. **Vault Service Name**、Vault サーバーのホスト **Address**('https://<hostname または ip>')、**Port number** および **Token** を入力します。
 - iv. **Advanced Settings** を拡張して、追加の設定および証明書の詳細を入力します。
 - A. OpenShift Container Storage 専用かつ特有のキー値のシークレットパスを **Backend Path** に入力します。
 - B. **TLS Server Name** および **Vault Enterprise Namespace** を入力します。
 - C. それぞれの PEM でエンコードされた証明書ファイルをアップロードして、**CA Certificate**、**Client Certificate** および **Client Private Key** を指定します。
 - D. **Save** をクリックします。
 - g. **Next** をクリックします。
 - h. 設定の詳細を確認します。設定を変更するには、**Back** をクリックして直前の設定ページに戻ります。
 - i. **Create** をクリックします。

検証手順

- インストールされたストレージクラスターの最後の **Status** が緑色のチェックマークと共に **Phase: Ready** と表示されていることを確認します。
 - **Operators** → **Installed Operators** → **Storage Cluster** のリンクをクリックして、ストレージクラスターのインストールのステータスを表示します。
 - または、**Operator Details** タブで、**Storage Cluster** タブをクリックすると、ステータスを表示できます。
- 柔軟なスケーリングがストレージクラスターで有効にされているかどうかを確認するには、以下の手順を実行します。
 1. **Storage Cluster** タブで **ocs-storagecluster** をクリックします。
 2. **YAML** タブで、**spec** セクションのキー **flexibleScaling** と **status** セクションの **flexibleScaling** を検索します。 **flexible scaling** が **true** であり、 **failureDomain** が **host** に設定されている場合、柔軟なスケーリング機能が有効になります。

```
spec:
  flexibleScaling: true
  [...]
status:
  failureDomain: host
```

- OpenShift Container Storage のすべてのコンポーネントが正常にインストールされていることを確認するには、[OpenShift Container Storage インストールの確認](#) を参照してください。

追加リソース

- 初期クラスターの容量を拡張するには、[ストレージのスケーリングガイド](#) を参照してください。

第3章 内部モードの 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。
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	<ul style="list-style-type: none"> ● ocs-operator-* (任意のワーカーノードに 1 Pod) ● ocs-metrics-exporter-*
Rook-ceph Operator	rook-ceph-operator-* (任意のワーカーノードに 1 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) ● noobaa-db-pg-* (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	rook-ceph-mon-* (各ストレージノードに 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1 Pod)
MDS	rook-ceph-mds-ocs-storagecluster-cephfilesystem-* (ストレージノードに分散する 2 Pod)
RGW	rook-ceph-rgw-ocs-storagecluster-cephobjectstore-* (任意のストレージノードに1 Pod)
CSI	<ul style="list-style-type: none"> ● cephfs <ul style="list-style-type: none"> ○ csi-cephfsplugin-* (各ワーカーノードに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-crashcollector	rook-ceph-crashcollector-* (各ストレージノードに1 Pod)
OSD	<ul style="list-style-type: none"> ● rook-ceph-osd-* (各デバイス用に1 Pod) ● rook-ceph-osd-prepare-ocs-deviceset-* (各デバイス用に1 Pod)

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

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

手順

1. **Storage → Overview** をクリックし、**Block and File** タブをクリックします。
2. **Status** カードで、**Storage Cluster** および **Data Resiliency** に緑色のチェックマークが表示されていることを確認します。
3. **Details** カードで、クラスター情報が表示されていることを確認します。

ブロックおよびファイルダッシュボードを使用した OpenShift Container Storage クラスターの正常性については、[OpenShift Container Storage のモニターリング](#) を参照してください。

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

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

手順

1. OpenShift Web コンソールから **Storage → Overview** をクリックし、**Object** タブをクリックします。
2. **Status card** で、**Object Service** と **Data Resiliency** の両方が **Ready** 状態 (緑のチェックマーク) にあることを確認します。
3. **Details カード** で、Multicloud Object Gateway 情報が表示されることを確認します。

オブジェクトサービスダッシュボードを使用した OpenShift Container Storage クラスターの正常性については、[OpenShift Container Storage のモニターリング](#) を参照してください。

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

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

手順

1. OpenShift Web コンソールから **Storage → Storage Classes** をクリックします。
2. 以下のストレージクラスが OpenShift Container Storage クラスターの作成時に作成されることを確認します。
 - **ocs-storagecluster-ceph-rbd**
 - **ocs-storagecluster-cephfs**
 - **openshift-storage.noobaa.io**
 - **ocs-storagecluster-ceph-rgw**

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に従って OpenShift Container Storage をアンインストールします。

アノテーションのアンインストール

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

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

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

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

Annotation	値	デフォルト	動作
<code>cleanup-policy</code>	<code>delete</code>	はい	Rook は物理ドライブおよび DataDirHostPath をクリーンアップします。
<code>cleanup-policy</code>	<code>Retain</code>	いいえ	Rook は物理ドライブおよび DataDirHostPath をクリーンアップし ません 。
<code>mode</code>	<code>graceful</code>	はい	Rook および NooBaa は PVC および OBC が管理者/ユーザーによって削除されるまでアンインストールプロセスを一時停止します。
<code>mode</code>	<code>forced</code>	いいえ	Rook および NooBaa は、Rook および NooBaa を使用してプロビジョニングされた PVC/OBC がそれぞれ存在している場合でもアンインストールを続行します。

以下のコマンドを使用してアノテーションの値を編集し、クリーンアップポリシーまたはアンインストールモードを変更できます。

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

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

前提条件

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

手順

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

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

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

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

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

2. OpenShift Container Storage を使用している PVC および OBC を削除します。
デフォルトのアンインストールモード (graceful) では、アンインストーラーは OpenShift Container Storage を使用するすべての PVC および OBC が削除されるまで待機します。

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

- a. OpenShift Container Storage を使用して、OpenShift Container Platform モニターリングスタック PVC を削除します。
詳細は、[OpenShift Container Storage からのモニターリングスタックの削除](#) を参照してください。
- b. OpenShift Container Storage を使用して、OpenShift Container Platform レジストリー PVC を削除します。
詳細は、[Removing OpenShift Container Platform registry from OpenShift Container Storage](#) を参照してください。

- c. OpenShift Container Storage を使用して、OpenShift Container Platform ロギング PVC を削除します。
詳細は、[OpenShift Container Storage からのクラスターロギング Operator の削除](#) を参照してください。
- d. OpenShift Container Storage を使用してプロビジョニングした PVC および OBC を削除します。
- 以下に、OpenShift Container Storage を使用してプロビジョニングされる PVC および OBC を特定するサンプルスクリプトを示します。このスクリプトは、OpenShift Container Storage によって内部で使用される PVC を無視します。

```
#!/bin/bash

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

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

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

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



注記

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

- OBC を削除します。

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

- PVC を削除します。

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



注記

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

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

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

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

```
$ oc get pods -n openshift-storage | grep -i cleanup
NAME                                READY STATUS RESTARTS AGE
cluster-cleanup-job-<xx>            0/1   Completed 0       8m35s
cluster-cleanup-job-<yy>            0/1   Completed 0       8m35s
cluster-cleanup-job-<zz>            0/1   Completed 0       8m35s
```

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

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

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

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

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

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

```
$ dmsetup ls
ocs-deviceset-localblock-0-data-1x4r7g-block-dmccrypt (253:1)
```

- マップ済みデバイスを削除します。

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-localblock-0-data-1x4r7g-block-dmccrypt
```



注記

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

- **CTRL+Z** を押して上記のコマンドを終了します。

- スタックしたプロセスの PID を検索します。

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

- **kill** コマンドを使用してプロセスを終了します。

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

- デバイス名が削除されていることを確認します。

```
$ dmsetup ls
```

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

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

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

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



注記

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

8. ローカルストレージデバイスを使用して OpenShift Container Storage をデプロイしている場合は、ローカルストレージ Operator 設定を削除します。[ローカルストレージ Operator の設定の削除](#) を参照してください。
9. ストレージノードのラベルを解除します。

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

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

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

11. OpenShift Container Storage を使用してプロビジョニングした PV がすべて削除されていることを確認します。**Released** 状態のままの PV がある場合は、これを削除します。

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

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

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

13. **CustomResourceDefinitions** を削除します。

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

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

- a. **ストレージ** をクリックします。
- b. **Overview** が Storage に表示されていないことを確認します。

4.1.1. ローカルストレージ Operator の設定の削除

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



注記

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

手順

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

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

3. **LocalVolumeSet** を削除します。

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

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

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

5. **StorageClassName** を削除します。

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

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

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

7. **LocalVolumeDiscovery** を削除します。

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

8. **LocalVolume** リソースを削除します (ある場合)。

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

ローカルボリュームごとに、以下を実行します。

- LocalVolume** および OpenShift Container Storage で使用される対応する **StorageClassName** を特定します。
- 変数 LV を LocalVolume の名前に設定し、変数 SC を StorageClass の名前に設定します。以下に例を示します。

```
$ LV=localblock
$ SC=localblock
```

- ローカルボリュームリソースを削除します。

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

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

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

- そのリソースのストレージノードからアーティファクトをクリーンアップします。

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

出力例:

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

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

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

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

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

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

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

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

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

前提条件

- PVC は OpenShift Container Platform モニタ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、[モニターリングスタックの設定](#) を参照してください。

手順

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

```
$ oc get pod,pvc -n openshift-monitoring
NAME                                READY STATUS  RESTARTS  AGE
pod/alertmanager-main-0             3/3   Running   0         8d
pod/alertmanager-main-1             3/3   Running   0         8d
pod/alertmanager-main-2             3/3   Running   0         8d
pod/cluster-monitoring-
operator-84457656d-pkrxm            1/1   Running   0         8d
```

```

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

```

NAME	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、これを保存します。

編集前

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

編集後


```

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

```

この例では、**alertmanagerMain** および **prometheusK8s** モニターリングコンポーネントは OpenShift Container Storage PVC を使用しています。

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

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

4.3. OPENSIFT CONTAINER STORAGE からの OPENSIFT CONTAINER PLATFORM レジストリーの削除

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

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

前提条件

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

手順

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

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

編集前

```

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

```

編集後

```

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

```

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

2. PVC を削除します。

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

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

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

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

前提条件

- クラスターロギングインスタンスは、OpenShift Container Storage PVC を使用するよう設定する必要があります。

手順

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

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

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

2. PVC を削除します。

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