



Red Hat OpenShift Container Storage 4.8

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

IBM Z 環境のインストールおよび設定方法

Red Hat OpenShift Container Storage 4.8 IBM Z インフラストラクチャー を使用した OpenShift Container Storage のデプロイ

IBM Z 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_Z_infrastructure.ent file |.

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

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

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

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

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

Linux[®] 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 Z インフラストラクチャーでローカルストレージを使用する方法については、本書をお読みください。While this document refers only to IBM Z, all information in it also applies to LinuxONE.

目次

多様性を受け入れるオープンソースの強化	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
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
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	7
2.1. RED HAT OPENSIFT CONTAINER STORAGE OPERATOR のインストール	7
2.2. ローカルストレージ 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 (オプション)	9
2.4. IBM Z での OPENSIFT CONTAINER STORAGE クラスターの作成	10
第3章 内部接続デバイスモードの OPENSIFT CONTAINER STORAGE デプロイメントの確認	14
3.1. POD の状態の確認	14
3.2. OPENSIFT CONTAINER STORAGE クラスターが正常であることの確認	15
3.3. OPENSIFT CONTAINER STORAGE 固有のストレージクラスが存在することの確認	16
第4章 OPENSIFT CONTAINER STORAGE のアンインストール	17
4.1. 内部接続デバイスモードの OPENSIFT CONTAINER STORAGE のアンインストール	17
4.1.1. ローカルストレージ Operator の設定の削除	21
4.2. OPENSIFT CONTAINER STORAGE からのモニターリングスタックの削除	25
4.3. OPENSIFT CONTAINER STORAGE からの OPENSIFT CONTAINER PLATFORM レジストリーの削除	28
4.4. OPENSIFT CONTAINER STORAGE からのクラスターロギング OPERATOR の削除	29

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 (RHOCP) IBM Z クラスターへのデプロイメントをサポートし、プロキシ環境に対する追加設定なしのサポートを提供します。



注記

IBM Z では、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は、[デプロイメントの計画](#) および [OpenShift Container Storage のデプロイメントの準備](#) 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

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。

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。

上記を処理したら、指定した順序で以下の手順を実行します。

1. [Red Hat OpenShift Container Storage Operator をインストールする](#)
2. [ローカルストレージ Operator のインストール](#)
3. [利用可能なストレージデバイスの検索](#)
4. [OpenShift Container Storage cluster service on IBM Z を作成します](#)。

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

ノードの要件

クラスターは、それぞれローカルに接続されたストレージデバイスを持つ 3 つ以上の OpenShift Container Platform ワーカーノードで設定される必要があります。

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

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

- ストレージノードの場合は、FCP ストレージデバイスが必要です。
- Multicloud Object Gateway はサポートされていません。

ノードの最小要件 [テクノロジープレビュー]

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

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

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

ローカルストレージデバイスを使用して OpenShift Container Storage を OpenShift Container Platform にデプロイすると、内部クラスターリソースを作成するオプションが提供されます。このデプロイメント方法に従って、ローカルストレージを使用して OpenShift Container Platform アプリケーションの永続ボリュームをサポートするようにします。

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

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

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

前提条件

- cluster-admin および operator インストールのパーミッションを持つアカウントを使用して OpenShift Container Platform クラスターにアクセスできる。
- Red Hat OpenShift Container Platform クラスターにワーカーノードが少なくとも3つある。
- 必要な追加要件をすべて満たしています。詳細は、[Planning your deployment](#) を参照してください。

注記

- OpenShift Container Storage のクラスタ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、以下のコマンドを使用し、**openshift-storage** namespace の空のノードセレクターを指定できます (この場合、openshift-storage namespace を作成します)。

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

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

手順

1. OpenShift Web コンソールにログインします。
2. **Operators** → **OperatorHub** をクリックします。
3. Operator の一覧から **OpenShift Container Storage** を検索し、これをクリックします。
4. **Install** をクリックします。
5. **Install Operator** ページで、以下のオプションを設定します。

- a. Channel を **stable-4.8** として更新します。
- b. Installation Mode オプションに **A specific namespace on the cluster** を選択します。
- c. Installed Namespace に **Operator recommended namespace openshift-storage** を選択します。namespace **openshift-storage** が存在しない場合、これは Operator のインストール時に作成されます。
- d. 承認ストラテジー を **Automatic** または **Manual** として選択します。
- e. **Install** をクリックします。
Automatic (自動) 更新を選択した場合、Operator Lifecycle Manager (OLM) は介入なしに、Operator の実行中のインスタンスを自動的にアップグレードします。

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

検証手順

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

2.2. ローカルストレージ OPERATOR のインストール

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

前提条件

- cluster-admin および Operator インストールのパーミッションを持つアカウントを使用して OpenShift Container Platform クラスターにアクセスできること。

手順

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

検証手順

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

2.3. 利用可能なストレージデバイスの検索 (オプション)

このステップは追加の情報であり、ストレージクラスターの作成時にディスクは自動的に検出されるので、省略することができます。以下の手順を使用して、IBM Z 用に PV を作成する前に、OpenShift 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

- 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
bmworker01   Ready  worker 6h45m v1.16.2
bmworker02   Ready  worker 6h45m v1.16.2
bmworker03   Ready  worker 6h45m v1.16.2
```

- 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、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/<node name>
```

出力例:

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

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

- 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 sdb
lrwxrwxrwx. 1 root root 9 Feb 3 16:49 scsi-360050763808104bc2800000000000259 ->
../../sdb
```

```
lrwxrwxrwx. 1 root root 9 Feb 3 16:49 scsi-SIBM_2145_00e020412f0aXX00 -> ../../sdb
lrwxrwxrwx. 1 root root 9 Feb 3 16:49 scsi-0x60050763808104bc280000000000259 ->
../../sdb
```

上記の例で、ローカルデバイス **sdb** の ID は以下になります。

```
scsi-0x60050763808104bc280000000000259
```

- 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、[ナレッジベースアティクル](#) を参照してください。

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

以下の手順を使用して、IBM Z にストレージクラスターを作成します。

前提条件

- ローカルストレージデバイスを使用した OpenShift 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 Z または LinuxONE でローカルストレージデバイスを使用するために、同じストレージタイプおよびサイズが各ノードに接続された 3 つのワーカーノードが必要です (例: 200 GB)。

手順

- OpenShift Web コンソールにログインします。
- Operators** → **Installed Operators** をクリックし、インストールされた Operator をすべて表示します。
選択された **Project** が **openshift-storage** であることを確認します。
- OpenShift Container Storage** をクリックします。
- Storage Cluster の **Create Instance** リンクをクリックします。
- Select Mode** に **Internal-Attached devices** を選択します。デフォルトでは、Internal が選択されます。
- ディスク検出、ストレージクラスターの作成、およびストレージクラスターの作成を含むウィザードを使用してストレージクラスターを作成します。
インストールされていない場合に、ローカルストレージ Operator をインストールすることを求めるプロンプトが出されます。**Install** をクリックし、[ローカルストレージ Operator](#) で説明されているように Operator をインストールします。

ディスクの検出

選択したノードで使用可能になる可能性のあるディスクの一覧を検出することができます。使用されていないブロックディスクおよびパーティション、永続ボリューム (PV) のプロビジョニングに利用できるブロックディスクおよびパーティションが検出されます。

- 以下のいずれかを選択します。
 - All nodes:** すべてのノードでディスクを検出します。
 - Select nodes:** 一覧表示されるノードのサブセットからディスクを検出します。

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 を使用するとノード名で検索でき、Label を使用すると事前に定義されたラベルを選択して検索できます。

選択したノードが集約された 30 CPU および 72 GiB の RAM の OpenShift Container Storage クラスターの要件と一致しない場合は、最小クラスターがデプロイされます。ノードの最小要件については、プランニングガイドの [リソース要件](#) セクションを参照してください。



注記

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

- b. **Next** をクリックします。

ストレージクラスの作成

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

- a. **Volume Set Name** を入力します。
- b. **Storage Class Name** を入力します。デフォルトで、ボリュームセット名がストレージクラス名について表示されます。
- c. 先の手順でディスク検出で選択されたノードは **Filter Disks** セクションに表示されます。以下のいずれかを選択します。

- **All nodes:** ディスクを検出したすべてのノードを選択します。
- **Select nodes:** ディスクを検出したノードのサブセットを選択します。
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 を使用するとノード名で検索でき、Label を使用すると事前に定義されたラベルを選択して検索できます。

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


OpenShift Container Storage のラックラベルがデータセンターの物理ラックに合わせて調整されていることを確認し、障害ドメインのレベルで二重ノードに障害が発生しないようにします。

- d. 必要な **Disk Type** を選択します。以下のオプションが利用可能になります。

All	ノードに存在するすべての種類のディスクを選択します。デフォルトでは、このオプションは選択されます。
SSD/NVME E	SSD NVME タイプのディスクのみを選択します。

HDD	HDD タイプのディスクのみを選択します。
-----	-----------------------

e. **Advanced** セクションでは、以下を設定できます。

ボリュームモード	デフォルトでは Block (ブロック) が選択されます。
ディスクサイズ	<p>含める必要のあるデバイスの最小および最大の許容サイズ。</p>  <p>注記 デバイスの最小サイズ 100GB を設定する必要があります。</p>
最大ディスク制限	これは、ノードで作成可能な PV の最大数を示します。このフィールドが空のままの場合、PV は一致するノードで利用可能なすべてのディスクに作成されます。

f. (オプション) **Select Capacity** チャートを使用して選択されたノードでディスクの選択した容量を表示できます。
このチャートは、直前の手順で検出されたディスクを反映するのに数分かかる場合があります。

チャートで **Nodes** および **Disks** リンクをクリックして、ノードおよびディスクの一覧を表示し、詳細情報を表示します。

g. **Next** をクリックします。

h. メッセージアラートで **Yes** をクリックし、ストレージクラスの実成を確認します。
ローカルボリュームセットおよびストレージクラスの実成後に、この手順に戻ることはできません。

ストレージクラスの実成

a. 必要なストレージクラスを選択します。
選択したストレージクラスに対応するストレージノードが追加されるまでに数分待機する必要があります。ストレージクラスに対応するノードは、ドロップダウン一覧で選択したストレージクラスに基づいて表示されます。

b. **Next** をクリックします。

c. (任意) Encryption セクションで、トグルを **Enabled** に設定して、クラスターでデータ暗号化を有効にします。セキュリティ設定

d. **Save** をクリックします。

e. **Next** をクリックしてストレージクラスを実成確認します。

f. **Create** をクリックします。

Create ボタンは、最低でも 3 つのノードを選択した場合にのみ有効になります。各ノードに接続されたデバイスの合計数と同等のボリューム数を持つ新規ストレージクラスが作成されます。デフォルト設定では、レプリケーション係数 1 を使用します。

初期クラスターの容量を拡張するには、[Scaling Storage](#) ガイドを参照してください。

検証手順

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

第3章 内部接続デバイスモードの OPENSIFT 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 Containers Storage の Pod が実行状態にあることを確認するには、以下の手順に従います。

手順

1. OpenShift Web コンソールにログインします。
2. OpenShift Web コンソールの左側のペインから **Workloads** → **Pods** をクリックします。
3. **Project** ドロップダウンリストから **openshift-storage** を選択します。
各コンポーネントについて予想される Pod 数や、これがノード数によってどのように異なるかについての詳細は、[表3.1「OpenShift Container Storage クラスタに対応する Pod」](#) を参照してください。
4. **Running** タブと **Completed** タブをクリックして、Pod が実行中で完了状態になっていることを確認します。

表3.1 OpenShift Container Storage クラスタに対応する 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-* (任意のワーカーノードに 1 Pod)
Rook-ceph Operator	<ul style="list-style-type: none"> ● 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) ● 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)

コンポーネント	対応する Pod
MON	rook-ceph-mon-* (ストレージノードに分散する 3 Pod)
MGR	rook-ceph-mgr-* (任意のストレージノードに 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 クラスタが正常であることの確認

OpenShift Container Storage のクラスタ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. 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. 内部接続デバイスモードの 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ノテーションを 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。
[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。
[OpenShift Container Storage からの OpenShift Container Platform レジストリーの削除](#)
- c. OpenShift Container Storage を使用して、OpenShift Container Platform ロギング PVC を削除します。
[OpenShift Container Storage からのクラスターロギング Operator の削除](#)

- 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-0-data-0-57snx-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-0-data-0-57snx-block-dmccrypt
```



注記

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

- **CTRL+Z** を押して上記のコマンドを終了します。
- スタックしたプロセスの PID を検索します。

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

- **kill** コマンドを使用してプロセスを終了します。

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

- デバイス名が削除されていることを確認します。

```
$ dmsetup ls
```

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

```
$ oc project default
$ oc delete project openshift-storage
```

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 をデプロイした場合には、ローカルのストレージ 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-
```


- 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>
```

- Multicloud Object Gateway storageclass を削除します。

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

- 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
```

- 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。
 - ストレージ** をクリックします。
 - 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

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

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

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

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

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

出力例:

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

4. **LocalVolumeSet** を削除します。

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

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

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

6. **StorageClassName** を削除します。

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

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

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

8. **LocalVolumeDiscovery** を削除します。

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

9. **LocalVolume** リソースを削除します (ある場合)。

以下の手順を使用して、現行または直前の OpenShift 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の他のテナントで使用されていないことを確認します。

ローカルボリュームごとに、以下を実行します。

- a. **LocalVolume** および OpenShift Container Storage で使用される対応する **StorageClassName** を特定します。

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

- b. 変数 LV を LocalVolume の名前に設定し、変数 SC を StorageClass の名前に設定します。以下に例を示します。

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

- c. 後にクリーンアップするデバイスを一覧表示し、これをメモします。

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

出力例:

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

- d. ローカルボリュームリソースを削除します。

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

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

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

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

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

出力例:

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

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

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

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

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

- a. ストレージノードを一覧表示します。

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

出力例:

```
NAME      STATUS  ROLES  AGE   VERSION
node-xxx  Ready  worker  4h45m v1.18.3+6c42de8
```

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

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

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

- c. ディスクパスを引用符内の **DISKS** 変数に保存します。ディスクパスの一覧は、ローカルボリュームおよびローカルボリュームセットおよびボリュームのそれぞれについてステップ 3 およびステップ 8.c を参照してください。

出力例:

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

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

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

出力例:

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

- e. シェルを終了し、他のノードに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 ...
```

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

以下に例を示します。

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

手順

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

```
$ oc get pod,pvc -n openshift-monitoring
NAME                                READY STATUS RESTARTS AGE
pod/alertmanager-main-0             3/3 Running 0      8d
pod/alertmanager-main-1             3/3 Running 0      8d
pod/alertmanager-main-2             3/3 Running 0      8d
pod/cluster-monitoring-
operator-84457656d-pkrxm            1/1 Running 0      8d
pod/grafana-79ccf6689f-2ll28        2/2 Running 0      8d
pod/kube-state-metrics-
7d86fb966-rvd9w                     3/3 Running 0      8d
pod/node-exporter-25894              2/2 Running 0      8d
pod/node-exporter-4dsd7              2/2 Running 0      8d
pod/node-exporter-6p4zc              2/2 Running 0      8d
pod/node-exporter-jbjvg              2/2 Running 0      8d
pod/node-exporter-jj4t5              2/2 Running 0     6d18h
pod/node-exporter-k856s              2/2 Running 0     6d18h
pod/node-exporter-rf8gn              2/2 Running 0      8d
pod/node-exporter-rmb5m              2/2 Running 0     6d18h
pod/node-exporter-zj7kx              2/2 Running 0      8d
pod/openshift-state-metrics-
59dbd4f654-4clng                    3/3 Running 0      8d
pod/prometheus-adapter-
5df5865596-k8dzn                    1/1 Running 0     7d23h
pod/prometheus-adapter-
5df5865596-n2gj9                    1/1 Running 0     7d23h
pod/prometheus-k8s-0                 6/6 Running 1      8d
```

```

pod/prometheus-k8s-1      6/6   Running 1      8d
pod/prometheus-operator-
55cfb858c9-c4zd9        1/1   Running 0      6d21h
pod/telemeter-client-
78fc8fc97d-2rgfp        3/3   Running 0      8d

```

```

NAME                                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

<pre> . . . managementState: Managed storage: pvc: claim: registry-cephfs-rwx-pvc . . . </pre>
編集後
<pre> . . . managementState: Removed storage: emptyDir: {} . . . </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

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 を使用するように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
```