



Red Hat OpenShift Data Foundation 4.15

デバイスの置き換え

動作中または故障したデバイスを安全に交換するための手順

動作中または故障したデバイスを安全に交換するための手順

法律上の通知

Copyright © 2024 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

本書では、Red Hat OpenShift Data Foundation のストレージデバイスを安全に置き換える方法について説明します。

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 AWS への OPENSIFT DATA FOUNDATION の動的プロビジョニング	6
1.1. AWS の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で動作するストレージデバイスまたは失敗したストレージデバイスの置き換え	6
1.2. AWS のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え	6
第2章 VMWARE への OPENSIFT DATA FOUNDATION の動的プロビジョニング	7
2.1. VMWARE インフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え	7
第3章 RED HAT VIRTUALIZATION への OPENSIFT DATA FOUNDATION の動的プロビジョニング	13
3.1. RED HAT VIRTUALIZATION のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは障害のあるストレージデバイスの置き換え	13
第4章 MICROSOFT AZURE への OPENSIFT DATA FOUNDATION の動的プロビジョニング	19
4.1. AZURE のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え	19
第5章 GOOGLE CLOUD にデプロイされた動的プロビジョニングされた OPENSIFT DATA FOUNDATION	20
5.1. GOOGLE CLOUD のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え	20
第6章 ローカルストレージデバイスを使用してデプロイされた OPENSIFT DATA FOUNDATION	21
6.1. ローカルストレージデバイスがサポートするクラスターで動作するストレージデバイスまたは障害のあるストレージデバイスの置き換え	21
6.2. IBM POWER で動作するストレージデバイスまたは障害のあるストレージデバイスの置き換え	28
6.3. IBM Z または IBM LINUXONE インフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え	38
第7章 OSD 削除ツール	41
7.1. 自動化ツールを使用したオブジェクトストレージデバイスの削除	41

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、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対する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。

フィードバックを送信するには、Bugzilla チケットを作成します。

1. [Bugzilla](#) の Web サイトに移動します。
2. **Component** セクションで、**documentation** を選択します。
3. **Description** フィールドに、ドキュメントの改善に向けたご提案を記入してください。ドキュメントの該当部分へのリンクも追加してください。
4. **Submit Bug** をクリックします。

はじめに

デプロイメントのタイプに応じて、以下のいずれかの手順を選択してストレージノードを置き換えることができます。

- AWS にデプロイされた動的に作成されたストレージクラスターについては、以下を参照してください。
 - 「AWS の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で動作するストレージデバイスまたは失敗したストレージデバイスの置き換え」
 - 「AWS のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え」
- VMware にデプロイされた動的に作成されたストレージクラスターについては、「VMware インフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え」を参照してください。
- Red Hat Virtualization にデプロイされた動的に作成されたストレージクラスターについては、「Red Hat Virtualization のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは障害のあるストレージデバイスの置き換え」を参照してください。
- Microsoft Azure にデプロイされた動的に作成されたストレージクラスターについては、「Azure のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え」を参照してください。
- ローカルストレージデバイスを使用してデプロイされたストレージクラスターについては、以下を参照してください。
 - 「ローカルストレージデバイスがサポートするクラスターで動作するストレージデバイスまたは障害のあるストレージデバイスの置き換え」
 - 「IBM Power で動作するストレージデバイスまたは障害のあるストレージデバイスの置き換え」
 - 「IBM Z または IBM LinuxONE インフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え」



注記

OpenShift Data Foundation は、異なる OSD サイズをサポートしません。

第1章 AWS への OPENSIFT DATA FOUNDATION の動的プロビジョニング

1.1. AWS の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で動作するストレージデバイスまたは失敗したストレージデバイスの置き換え

AWS の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上の動的に作成されたストレージクラスターのデバイスを置き換える必要がある場合は、ストレージノードを置き換える必要があります。ノードを置き換える方法は、以下を参照してください。

- [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で動作する AWS ノードの置き換え](#)
- [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での失敗した AWS ノードの置き換え](#)

1.2. AWS のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え

AWS のインストーラーでプロビジョニングされるインフラストラクチャーの動的に作成されたストレージクラスターのデバイスを置き換える必要がある場合は、ストレージノードを置き換える必要があります。ノードを置き換える方法は、以下を参照してください。

- [インストーラーでプロビジョニングされるインフラストラクチャーで動作する AWS ノードの置き換え](#)
- [インストーラーでプロビジョニングされるインフラストラクチャーでの失敗した AWS ノードの置き換え](#)

第2章 VMWARE への OPENSIFT DATA FOUNDATION の動的プロビジョニング

2.1. VMWARE インフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え

VMware インフラストラクチャーに動的にデプロイされる OpenShift Data Foundation で1つ以上の仮想マシンディスク (VMDK) を交換する必要がある場合は、新しいボリュームに新しい永続ボリュームグループ (PVC) を作成し、古いオブジェクトストレージデバイス (OSD) を削除します。

前提条件

- データに耐久性があることを確認する。
 - OpenShift Web コンソールで、**Storage** → **Data Foundation** をクリックします。
 - **Storage Systems** タブをクリックし、**ocs-storagecluster-storagesystem** をクリックします。
 - **Block and File** ダッシュボードの **Status card** の **Overview** タブで、**Data Resiliency** に緑色のチェックマークが付いていることを確認します。

手順

1. 置き換える必要がある OSD と、その OSD がスケジュールされている OpenShift Container Platform ノードを特定します。

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

出力例:

```
rook-ceph-osd-0-6d77d6c7c6-m8xj6 0/1 CrashLoopBackOff 0 24h 10.129.0.16
compute-2 <none> <none>
rook-ceph-osd-1-85d99fb95f-2svc7 1/1 Running 0 24h 10.128.2.24 compute-
0 <none> <none>
rook-ceph-osd-2-6c66cdb977-jp542 1/1 Running 0 24h 10.130.0.18 compute-
1 <none> <none>
```

この例では、**rook-ceph-osd-0-6d77d6c7c6-m8xj6** を置き換える必要があり、**compute-2** は OSD がスケジュールされる OpenShift Container platform ノードです。



注記

交換する OSD が正常な場合、Pod のステータスは **Running** になります。

2. 置き換えられる OSD の OSD デプロイメントをスケールダウンします。
OSD を置き換えるたびに、**osd_id_to_remove** パラメーターを OSD ID に更新し、この手順を繰り返します。

```
$ osd_id_to_remove=0
```

```
$ oc scale -n openshift-storage deployment rook-ceph-osd-${osd_id_to_remove} --replicas=0
```

■

ここで、**osd_id_to_remove** は Pod 名の **rook-ceph-osd** 接頭辞の直後にくる整数です。この例では、デプロイメント名は **rook-ceph-osd-0** です。

出力例:

```
deployment.extensions/rook-ceph-osd-0 scaled
```

3. **rook-ceph-osd** Pod が停止していることを確認します。

```
$ oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
```

出力例:

```
No resources found.
```

重要

rook-ceph-osd Pod が **terminating** 状態にある場合は、**force** オプションを使用して Pod を削除します。

```
$ oc delete pod rook-ceph-osd-0-6d77d6c7c6-m8xj6 --force --grace-period=0
```

出力例:

```
warning: Immediate deletion does not wait for confirmation that the running
resource has been terminated. The resource may continue to run on the
cluster indefinitely.
pod "rook-ceph-osd-0-6d77d6c7c6-m8xj6" force deleted
```

4. 新しい OSD を追加できるように、クラスターから古い OSD を削除します。

- a. 古い **ocs-osd-removal** ジョブを削除します。

```
$ oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

- b. **openshift-storage** プロジェクトに移動します。

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

- c. クラスターから以前の OSD を削除します。

```
$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_id_to_remove} FORCE_OSD_REMOVAL=false |oc create -n
openshift-storage -f -
```

3つの OSD しかないクラスター、または OSD が削除された後にデータの3つのレプリカすべてを復元するのに十分なスペースがないクラスターでは、FORCE_OSD_REMOVAL 値を true に変更する必要があります。



警告

この手順により、OSD はクラスターから完全に削除されます。**osd_id_to_remove** の正しい値が指定されていることを確認します。

5. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

Completed のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
$ oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

6. OSD の取り外しが完了したことを確認します。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

出力例:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

重要

ocs-osd-removal-job Pod が失敗し、Pod が期待される **Completed** 状態にない場合は、さらにデバッグするために Pod ログを確認してください。

以下に例を示します。

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

7. 暗号化がインストール時に有効にされている場合は、それぞれの OpenShift Data Foundation ノードから削除された 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。

- a. **ocs-osd-removal-job** Pod のログから、置き換えられた OSD の PVC 名を取得します。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'pvc|deviceset'
```

出力例:

```
2021-05-12 14:31:34.666000 I | cephosd: removing the OSD PVC "ocs-deviceset-xxxx-xxx-xxx-xxx"
```

b. これまでに特定した各ノードに以下を実行します。

i. **debug** 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>
```

<node name>

これはノードの名前です。

```
$ chroot /host
```

ii. 直前の手順で特定された PVC 名に基づいて関連するデバイス名を検索します。

```
$ dmsetup ls | grep <pvc name>
```

<pvc name>

これは PVC の名前です。

出力例:

```
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt (253:0)
```

iii. マップ済みデバイスを削除します。

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt
```

重要

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>
```

<PID>

これはプロセス ID です。

- デバイス名が削除されていることを確認します。

```
$ dmsetup ls
```

8. **ocs-osd-removal** ジョブを削除します。

```
$ oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```



注記

データ暗号化で外部の鍵管理システム (KMS) を使用する場合は、古い OSD 暗号化キーは孤立したキーであるため、Vault サーバーから削除できます。

検証手順

1. 新しい OSD が実行されていることを確認します。

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd
```

出力例:

```
rook-ceph-osd-0-5f7f4747d4-snshw          1/1   Running   0      4m47s
rook-ceph-osd-1-85d99fb95f-2svc7        1/1   Running   0      1d20h
rook-ceph-osd-2-6c66cdb977-jp542       1/1   Running   0      1d20h
```

2. **Bound** 状態の新しい PVC が作成されていることを確認します。

```
$ oc get -n openshift-storage pvc
```

出力例:

NAME	STATUS	VOLUME	CAPACITY	ACCESS
ocs-deviceset-0-0-2s6w4	Bound	pvc-7c9bc7af7-de68-40e1-95f9-0b0d7c0ae2fc	512Gi	
RWO	thin	5m		
ocs-deviceset-1-0-q8fwh	Bound	pvc-9e7e00cb-6b33-402e-9dc5-b8df4fd9010f	512Gi	
RWO	thin	1d20h		
ocs-deviceset-2-0-9v8lq	Bound	pvc-38cdfcee-ea7e-42a5-a6e1-aaa6d4924291	512Gi	
RWO	thin	1d20h		

3. (オプション) クラスタでクラスタ全体の暗号化が有効な場合は、新規 OSD デバイスが暗号化されていることを確認します。

- a. 新規 OSD Pod が実行しているノードを特定します。

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-pod-name>
```

<OSD-pod-name>

これは OSD Pod の名前です。

以下に例を示します。

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

出力例:

```
NODE  
compute-1
```

b. 直前の手順で特定された各ノードに以下を実行します。

i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

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

<node name>

これはノードの名前です。

```
$ chroot /host
```

ii. **ocs-deviceset** 名の横にある **crypt** キーワードを確認します。

```
$ lsblk
```

4. OpenShift Web コンソールにログインし、ストレージダッシュボードを表示します。

第3章 RED HAT VIRTUALIZATION への OPENSIFT DATA FOUNDATION の動的プロビジョニング

3.1. RED HAT VIRTUALIZATION のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは障害のあるストレージデバイスの置き換え

新しいボリュームに新しい永続ボリュームクレーム (PVC) を作成し、古いオブジェクトストレージデバイス (OSD) を削除します。

前提条件

- データに耐久性があることを確認する。
 - OpenShift Web コンソールで、**Storage** → **Data Foundation** をクリックします。
 - **Storage Systems** タブをクリックし、**ocs-storagecluster-storagesystem** をクリックします。
 - **Block and File** ダッシュボードの **Status card** の **Overview** タブで、**Data Resiliency** に緑色のチェックマークが付いていることを確認します。

手順

1. 置き換える必要がある OSD と、その OSD がスケジュールされている OpenShift Container Platform ノードを特定します。

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

出力例:

```
rook-ceph-osd-0-6d77d6c7c6-m8xj6 0/1 CrashLoopBackOff 0 24h 10.129.0.16
compute-2 <none> <none>
rook-ceph-osd-1-85d99fb95f-2svc7 1/1 Running 0 24h 10.128.2.24 compute-
0 <none> <none>
rook-ceph-osd-2-6c66cdb977-jp542 1/1 Running 0 24h 10.130.0.18 compute-
1 <none> <none>
```

この例では、**rook-ceph-osd-0-6d77d6c7c6-m8xj6** を置き換える必要があり、**compute-2** は OSD がスケジュールされる OpenShift Container platform ノードです。



注記

置き換える OSD が正常である場合、Pod のステータスは **Running** になります。

2. 置き換えられる OSD の OSD デプロイメントをスケールダウンします。OSD を置き換えるたびに、**osd_id_to_remove** パラメーターを OSD ID に更新し、この手順を繰り返します。

```
$ osd_id_to_remove=0
```

```
$ oc scale -n openshift-storage deployment rook-ceph-osd-${osd_id_to_remove} --replicas=0
```

ここで、**osd_id_to_remove** は Pod 名の **rook-ceph-osd** 接頭辞の直後にくる整数です。この例では、デプロイメント名は **rook-ceph-osd-0** です。

出力例:

```
deployment.extensions/rook-ceph-osd-0 scaled
```

3. **rook-ceph-osd** Pod が停止していることを確認します。

```
$ oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
```

出力例:

```
No resources found.
```

重要

rook-ceph-osd Pod が **terminating** 状態にある場合は、**force** オプションを使用して Pod を削除します。

```
$ oc delete pod rook-ceph-osd-0-6d77d6c7c6-m8xj6 --force --grace-period=0
```

出力例:

```
warning: Immediate deletion does not wait for confirmation that the running
resource has been terminated. The resource may continue to run on the
cluster indefinitely.
pod "rook-ceph-osd-0-6d77d6c7c6-m8xj6" force deleted
```

4. 新しい OSD を追加できるように、クラスターから古い OSD を削除します。

- a. 古い **ocs-osd-removal** ジョブを削除します。

```
$ oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job"
```

- b. **openshift-storage** プロジェクトに移動します。

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

- c. クラスターから以前の OSD を削除します。

```
$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_id_to_remove} FORCE_OSD_REMOVAL=false |oc create -n
openshift-storage -f -
```

3つの OSD しかないクラスター、または OSD が削除された後にデータの3つのレプリカすべてを復元するのに十分なスペースがないクラスターでは、`FORCE_OSD_REMOVAL` 値を `true` に変更する必要があります。



警告

この手順により、OSD はクラスターから完全に削除されます。`osd_id_to_remove` の正しい値が指定されていることを確認します。

5. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

Completed のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
$ oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

6. OSD の取り外しが完了したことを確認します。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

出力例:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

重要

ocs-osd-removal-job が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。

以下に例を示します。

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1'
```

7. 暗号化がインストール時に有効にされている場合は、それぞれの OpenShift Data Foundation ノードから削除された 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。

- a. **ocs-osd-removal-job** Pod のログから、置き換えられた OSD の PVC 名を取得します。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'pvc|deviceset'
```

出力例:

```
2021-05-12 14:31:34.666000 I | cephosd: removing the OSD PVC "ocs-deviceset-xxxx-xxx-xxx-xxx"
```

b. これまでに特定した各ノードに以下を実行します。

i. **debug** 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>
```

<node name>

これはノードの名前です。

```
$ chroot /host
```

ii. 直前の手順で特定された PVC 名に基づいて関連するデバイス名を検索します。

```
$ dmsetup ls | grep <pvc name>
```

<pvc name>

これは PVC の名前です。

出力例:

```
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt (253:0)
```

iii. マップ済みデバイスを削除します。

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt
```

重要

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>
```

<PID>

これはプロセス ID です。

- デバイス名が削除されていることを確認します。

```
$ dmsetup ls
```

8. **ocs-osd-removal** ジョブを削除します。

```
$ oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```



注記

データ暗号化で外部の鍵管理システム (KMS) を使用する場合は、古い OSD 暗号化キーは孤立したキーであるため、Vault サーバーから削除できます。

検証手順

1. 新しい OSD が実行されていることを確認します。

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd
```

出力例:

```
rook-ceph-osd-0-5f7f4747d4-snshw          1/1   Running   0      4m47s
rook-ceph-osd-1-85d99fb95f-2svc7        1/1   Running   0      1d20h
rook-ceph-osd-2-6c66cdb977-jp542        1/1   Running   0      1d20h
```

2. **Bound** 状態の新しい PVC が作成されていることを確認します。

```
$ oc get -n openshift-storage pvc
```

3. (オプション) クラスタでクラスタ全体の暗号化が有効な場合は、新規 OSD デバイスが暗号化されていることを確認します。

- a. 新規 OSD Pod が実行しているノードを特定します。

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-pod-name>
```

<OSD-pod-name>

これは OSD Pod の名前です。
以下に例を示します。

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

出力例:

```
NODE
compute-1
```

- b. これまでに特定した各ノードに以下を実行します。

- i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

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

<node name>

これはノードの名前です。

```
$ chroot /host
```

- ii. **ocs-deviceset** 名の横にある **crypt** キーワードを確認します。

```
$ lsblk
```

4. OpenShift Web コンソールにログインし、ストレージダッシュボードを表示します。

第4章 MICROSOFT AZURE への OPENSIFT DATA FOUNDATION の動的プロビジョニング

4.1. AZURE のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え

Azure のインストーラーでプロビジョニングされるインフラストラクチャーの動的に作成されたストレージクラスターのデバイスを置き換える必要がある場合は、ストレージノードを置き換える必要があります。ノードを置き換える方法は、以下を参照してください。

- [Azure のインストーラーでプロビジョニングされるインフラストラクチャーで動作するノードの置き換え](#)
- [Azure のインストーラーでプロビジョニングされるインフラストラクチャーでの失敗したノードの置き換え](#)

第5章 GOOGLE CLOUD にデプロイされた動的プロビジョニングされた OPENSIFT DATA FOUNDATION

5.1. GOOGLE CLOUD のインストーラーでプロビジョニングされるインフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え

Google Cloud のインストーラーでプロビジョニングされるインフラストラクチャーの動的に作成されたストレージクラスターのデバイスを置き換える必要がある場合は、ストレージノードを置き換える必要があります。ノードを置き換える方法は、以下を参照してください。

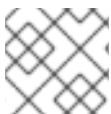
- [Replacing operational nodes on Google Cloud installer-provisioned infrastructure](#)
- [Google Cloud のインストーラーでプロビジョニングされるインフラストラクチャーでの失敗したノードの置き換え](#)

第6章 ローカルストレージデバイスを使用してデプロイされた OPENSIFT DATA FOUNDATION

6.1. ローカルストレージデバイスがサポートするクラスターで動作するストレージデバイスまたは障害のあるストレージデバイスの置き換え

以下のインフラストラクチャーでローカルストレージデバイスを使用してデプロイされた OpenShift Data Foundation のオブジェクトストレージデバイス (OSD) を置き換えることができます。

- ベアメタル
- VMware
- Red Hat Virtualization



注記

1つ以上の基盤となるストレージデバイスを交換する必要がある場合があります。

前提条件

- Red Hat は、交換用デバイスを、交換するデバイスと同様のインフラストラクチャーおよびリソースで設定することを推奨しています。
- データに耐久性があることを確認する。
 - OpenShift Web コンソールで、**Storage** → **Data Foundation** をクリックします。
 - **Storage Systems** タブをクリックし、**ocs-storagecluster-storagesystem** をクリックします。
 - **Block and File** ダッシュボードの **Status card** の **Overview** タブで、**Data Resiliency** に緑色のチェックマークが付いていることを確認します。

手順

1. 関連するワーカーノードから基礎となるストレージデバイスを削除します。
2. 関連する OSD Pod が CrashLoopBackOff 状態になったことを確認します。置き換える必要がある OSD と、その OSD がスケジュールされている OpenShift Container Platform ノードを特定します。

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

出力例:

```
rook-ceph-osd-0-6d77d6c7c6-m8xj6 0/1 CrashLoopBackOff 0 24h 10.129.0.16
compute-2 <none> <none>
rook-ceph-osd-1-85d99fb95f-2svc7 1/1 Running 0 24h 10.128.2.24 compute-
0 <none> <none>
rook-ceph-osd-2-6c66cdb977-jp542 1/1 Running 0 24h 10.130.0.18 compute-
1 <none> <none>
```

この例では、**rook-ceph-osd-0-6d77d6c7c6-m8xj6** を置き換える必要があり、**compute-2** は OSD がスケジュールされる OpenShift Container platform ノードです。

- 置き換えられる OSD の OSD デプロイメントをスケールダウンします。

```
$ oc scale -n openshift-storage deployment rook-ceph-osd-${osd_id_to_remove} --replicas=0
```

```
$ oc scale -n openshift-storage deployment rook-ceph-osd-${osd_id_to_remove} --replicas=0
```

ここで、**osd_id_to_remove** は Pod 名の **rook-ceph-osd** 接頭辞の直後にくる整数です。この例では、デプロイメント名は **rook-ceph-osd-0** です。

出力例:

```
deployment.extensions/rook-ceph-osd-0 scaled
```

- rook-ceph-osd** Pod が停止していることを確認します。

```
$ oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
```

出力例:

```
No resources found in openshift-storage namespace.
```

重要

rook-ceph-osd Pod が数分以上 **terminating** 状態である場合は、**force** オプションを使用して Pod を削除します。

```
$ oc delete -n openshift-storage pod rook-ceph-osd-0-6d77d6c7c6-m8xj6 --grace-period=0 --force
```

出力例:

```
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
pod "rook-ceph-osd-0-6d77d6c7c6-m8xj6" force deleted
```

- 新しい OSD を追加できるように、クラスターから古い OSD を削除します。

- 古い **ocs-osd-removal** ジョブを削除します。

```
$ oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

- openshift-storage** プロジェクトに移動します。

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

- c. クラスタから以前の OSD を削除します。

```
$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_id_to_remove} FORCE_OSD_REMOVAL=false |oc create -n
openshift-storage -f -
```

3つの OSD しかないクラスタ、または OSD が削除された後にデータの3つのレプリカすべてを復元するのに十分なスペースがないクラスタでは、`FORCE_OSD_REMOVAL` 値を `true` に変更する必要があります。



警告

この手順により、OSD はクラスタから完全に削除されます。`osd_id_to_remove` の正しい値が指定されていることを確認します。

6. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

Completed のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
$ oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

7. OSD の取り外しが完了したことを確認します。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed
removal'
```

出力例:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

重要

ocs-osd-removal-job が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。

以下に例を示します。

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

8. 暗号化がインストール時に有効にされている場合は、それぞれの OpenShift Data Foundation ノードから削除された 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。

- a. **ocs-osd-removal-job** Pod のログから、置き換えられた OSD の永続ボリューム要求 (PVC) 名を取得します。

(i) PVC) 上で実行します。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'pvc|deviceset'
```

出力例:

```
2021-05-12 14:31:34.666000 I | cephosd: removing the OSD PVC "ocs-deviceset-xxxx-xxx-xxx-xxx"
```

b. これまでに特定した各ノードに以下を実行します。

i. **debug** 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>
```

<node name>

これはノードの名前です。

```
$ chroot /host
```

ii. 直前の手順で特定された PVC 名に基づいて関連するデバイス名を検索します。

```
$ dmsetup ls| grep <pvc name>
```

<pvc name>

これは PVC の名前です。

出力例:

```
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt (253:0)
```

iii. マップ済みデバイスを削除します。

```
$ cryptsetup luksClose --debug --verbose <ocs-deviceset-name>
```

<ocs-deviceset-name>

前の手順で特定した PVC 名に基づいた、関連するデバイスの名前です。

重要

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>
```

<PID>

これはプロセス ID です。

- デバイス名が削除されていることを確認します。

```
$ dmsetup ls
```

9. 削除する必要がある永続ボリューム (PV) を検索します。

```
$ oc get pv -L kubernetes.io/hostname | grep <storageclass-name> | grep Released
```

出力例:

```
local-pv-d6bf175b      1490Gi    RWO      Delete    Released    openshift-
storage/ocs-devic
eset-0-data-0-6c5pw  localblock 2d22h    compute-1
```

10. PV を削除します。

```
$ oc delete pv <pv_name>
```

11. 物理的に新規デバイスをノードに追加します。

12. **deviceInclusionSpec** に一致するデバイスの PV のプロビジョニングを追跡します。PV をプロビジョニングするのに数分かかる場合があります。

```
$ oc -n openshift-local-storage describe localvolumeset <lvs-name>
```

出力例:

```
[...]
Status:
Conditions:
  Last Transition Time:      2020-11-17T05:03:32Z
  Message:                  DiskMaker: Available, LocalProvisioner: Available
  Status:                   True
  Type:                     DaemonSetsAvailable
  Last Transition Time:      2020-11-17T05:03:34Z
```

```

Message:          Operator reconciled successfully.
Status:           True
Type:             Available
Observed Generation: 1
Total Provisioned Device Count: 4
Events:
Type Reason Age From Message
---- -
Normal Discovered 2m30s (x4 localvolumeset- node.example.com -
NewDevice over 2m30s) symlink-controller found possible
matching disk,
waiting 1m to claim

Normal FoundMatch 89s (x4 localvolumeset- node.example.com -
ingDisk over 89s) symlink-controller symlinking matching
disk

```

PV がプロビジョニングされると、新規 OSD Pod が PV 用に自動的に作成されます。

13. **ocs-osd-removal** ジョブを削除します。

```
$ oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```



注記

データ暗号化で外部の鍵管理システム (KMS) を使用する場合は、古い OSD 暗号化キーは孤立したキーであるため、Vault サーバーから削除できます。

検証手順

1. 新しい OSD が実行されていることを確認します。

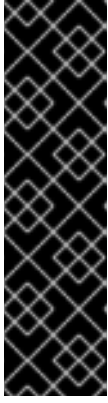
```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd
```

出力例:

```

rook-ceph-osd-0-5f7f4747d4-snshw 1/1 Running 0 4m47s
rook-ceph-osd-1-85d99fb95f-2svc7 1/1 Running 0 1d20h
rook-ceph-osd-2-6c66cdb977-jp542 1/1 Running 0 1d20h

```



重要

数分経過しても新規 OSD が **Running** と表示されない場合は、**rook-ceph-operator** Pod を再起動して強制的に調整を行います。

```
$ oc delete pod -n openshift-storage -l app=rook-ceph-operator
```

出力例:

```
pod "rook-ceph-operator-6f74fb5bff-2d982" deleted
```

2. 新規 PVC が作成されていることを確認します。

```
$ oc get -n openshift-storage pvc | grep <lvs-name>
```

出力例:

```
ocs-deviceset-0-0-c2mqb Bound local-pv-b481410 1490Gi RWO localblock
5m
ocs-deviceset-1-0-959rp Bound local-pv-414755e0 1490Gi RWO localblock
1d20h
ocs-deviceset-2-0-79j94 Bound local-pv-3e8964d3 1490Gi RWO localblock
1d20h
```

3. (オプション) クラスタでクラスタ全体の暗号化が有効な場合は、新規 OSD デバイスが暗号化されていることを確認します。
 - a. 新規 OSD Pod が実行しているノードを特定します。

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-pod-name>
```

<OSD-pod-name>

これは OSD Pod の名前です。
以下に例を示します。

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

出力例:

```
NODE
compute-1
```

- b. 直前の手順で特定された各ノードに以下を実行します。
 - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

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

<node name>

これはノードの名前です。

```
$ chroot /host
```

- ii. **ocs-deviceset** 名の横にある **crypt** キーワードを確認します。

```
$ lsblk
```

4. OpenShift Web コンソールにログインし、ストレージダッシュボードで OSD のステータスを確認します。



注記

データの完全復旧には、復元されるデータ量により、時間がかかる場合があります。

6.2. IBM POWER で動作するストレージデバイスまたは障害のあるストレージデバイスの置き換え

IBM Power でローカルストレージデバイスを使用してデプロイされた OpenShift Data Foundation のオブジェクトストレージデバイス (OSD) を置き換えることができます。



注記

1つ以上の基盤となるストレージデバイスを交換する必要がある場合があります。

前提条件

- Red Hat は、交換用デバイスを、交換するデバイスと同様のインフラストラクチャーおよびリソースで設定することを推奨しています。
- データに耐久性があることを確認する。
 - OpenShift Web コンソールで、**Storage** → **Data Foundation** をクリックします。
 - **Storage Systems** タブをクリックし、**ocs-storagecluster-storagesystem** をクリックします。
 - **Block and File** ダッシュボードの **Status card** の **Overview** タブで、**Data Resiliency** に緑色のチェックマークが付いていることを確認します。

手順

1. 置き換える必要がある OSD と、その OSD がスケジュールされている OpenShift Container Platform ノードを特定します。

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

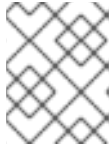
出力例:

```
rook-ceph-osd-0-86bf8cdc8-4nb5t 0/1 crashLoopBackOff 0 24h 10.129.2.26
worker-0 <none> <none>
rook-ceph-osd-1-7c99657cfb-jdzvz 1/1 Running 0 24h 10.128.2.46 worker-1
```



```
<none> <none>
rook-ceph-osd-2-5f9f6dfb5b-2mnw9 1/1 Running 0 24h 10.131.0.33 worker-2
<none> <none>
```

この例では、**rook-ceph-osd-0-86bf8cdc8-4nb5t** を置き換える必要があり、**worker-0** は OSD がスケジュールされる RHOCP ノードです。



注記

置き換える必要のある OSD が正常である場合、Pod のステータスは **Running** になります。

- 置き換えられる OSD の OSD デプロイメントをスケールダウンします。

```
$ osd_id_to_remove=0
```

```
$ oc scale -n openshift-storage deployment rook-ceph-osd-${osd_id_to_remove} --replicas=0
```

ここで、**osd_id_to_remove** は Pod 名の **rook-ceph-osd** 接頭辞の直後にくる整数です。この例では、デプロイメント名は **rook-ceph-osd-0** です。

出力例:

```
deployment.extensions/rook-ceph-osd-0 scaled
```

- rook-ceph-osd** Pod が停止していることを確認します。

```
$ oc get -n openshift-storage pods -l ceph-osd-id=${osd_id_to_remove}
```

出力例:

```
No resources found in openshift-storage namespace.
```



重要

rook-ceph-osd Pod が数分以上 **terminating** 状態である場合は、**force** オプションを使用して Pod を削除します。

```
$ oc delete -n openshift-storage pod rook-ceph-osd-0-86bf8cdc8-4nb5t --
  grace-period=0 --force
```

出力例:

```
warning: Immediate deletion does not wait for confirmation that the running
resource has been terminated. The resource may continue to run on the
cluster indefinitely.
pod "rook-ceph-osd-0-86bf8cdc8-4nb5t" force deleted
```

- 新しい OSD を追加できるように、クラスターから古い OSD を削除します。

- 置き換える OSD に関連付けられた **DeviceSet** を特定します。

-

```
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd- $\{osd\_id\_to\_remove\}$  |
grep ceph.rook.io/pvc
```

出力例:

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-64xjl
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-64xjl
```

この例では、永続ボリューム要求 (PVC) 名は **ocs-deviceset-localblock-0-data-0-64xjl** です。

- b. PVC に関連付けられた 永続ボリューム (PV) を特定します。

```
$ oc get -n openshift-storage pvc ocs-deviceset- $\langle x \rangle$ - $\langle y \rangle$ - $\langle pvc\text{-suffix} \rangle$ 
```

ここで、**x**、**y**、および **pvc-suffix** は、直前の手順で特定された **DeviceSet** の値です。

出力例:

```
NAME                               STATUS    VOLUME          CAPACITY  ACCESS MODES
STORAGECLASS  AGE
ocs-deviceset-localblock-0-data-0-64xjl  Bound    local-pv-8137c873  256Gi    RWO
localblock    24h
```

この例では、関連付けられた PV は **local-pv-8137c873** です。

- c. 置き換えるデバイスの名前を特定します。

```
$ oc get pv local-pv- $\langle pv\text{-suffix} \rangle$  -o yaml | grep path
```

ここで、**pv-suffix** は、前のステップで特定された PV 名の値です。

出力例:

```
path: /mnt/local-storage/localblock/vdc
```

この例では、デバイス名は **vdc** です。

- d. 置き換える OSD に関連付けられた **prepare-pod** を特定します。

```
$ oc describe -n openshift-storage pvc ocs-deviceset- $\langle x \rangle$ - $\langle y \rangle$ - $\langle pvc\text{-suffix} \rangle$  | grep Used
```

ここで、**x**、**y**、および **pvc-suffix** は、直前の手順で特定された **DeviceSet** の値です。

出力例:

```
Used By: rook-ceph-osd-prepare-ocs-deviceset-localblock-0-data-0-64knzkc
```

この例では、**prepare-pod** の名前は **rook-ceph-osd-prepare-ocs-deviceset-localblock-0-data-0-64knzkc** です。

- e. 古い **ocs-osd-removal** ジョブを削除します。

```
$ oc delete -n openshift-storage job ocs-osd-removal-job
```

■

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

- f. **openshift-storage** プロジェクトを変更します。

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

- g. クラスタから以前の OSD を削除します。

```
$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_id_to_remove} FORCE_OSD_REMOVAL=false |oc create -n
openshift-storage -f -
```

3つの OSD しかないクラスタ、または OSD が削除された後にデータの3つのレプリカすべてを復元するのに十分なスペースがないクラスタでは、FORCE_OSD_REMOVAL 値を true に変更する必要があります。



警告

この手順により、OSD はクラスタから完全に削除されます。**osd_id_to_remove** の正しい値が指定されていることを確認します。

5. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

Completed のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
$ oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

6. OSD の取り外しが完了したことを確認します。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed
removal'
```

出力例:

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```



重要

ocs-osd-removal-job が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。

以下に例を示します。

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

7. 暗号化がインストール時に有効にされている場合は、それぞれの OpenShift Data Foundation ノードから削除された 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。

- a. **ocs-osd-removal-job** Pod のログから、置き換えられた OSD の PVC 名を取得します。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 |egrep -i 'pvc|deviceset'
```

出力例:

```
2021-05-12 14:31:34.666000 I | cephosd: removing the OSD PVC "ocs-deviceset-xxxx-xxx-xxx-xxx"
```

- b. これまでに特定した各ノードに以下を実行します。

- i. **debug** 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>
```

<node name>

これはノードの名前です。

```
$ chroot /host
```

- ii. 直前の手順で特定された PVC 名に基づいて関連するデバイス名を検索します。

```
$ dmsetup ls| grep <pvc name>
```

<pvc name>

これは PVC の名前です。

出力例:

```
ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt (253:0)
```

- iii. マップ済みデバイスを削除します。

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-xxx-xxx-xxx-xxx-block-dmccrypt
```

重要

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>
```

<PID>

これはプロセス ID です。

- デバイス名が削除されていることを確認します。

```
$ dmsetup ls
```

8. 削除する必要のある PV を検索します。

```
$ oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

出力例:

```
local-pv-d6bf175b      1490Gi    RWO      Delete    Released    openshift-
storage/ocs-deviceset-0-data-0-6c5pw  localblock  2d22h    compute-1
```

9. PV を削除します。

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

<pv-name>

PV の名前です。

10. 古いデバイスを置き換え、新規デバイスを使用して新規の OpenShift Container Platform PV を作成します。

- a. 置き換えるデバイスで OpenShift Container Platform ノードにログインします。この例では、OpenShift Container Platform ノードは **worker-0** です。

```
$ oc debug node/worker-0
```

出力例:

```
Starting pod/worker-0-debug ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.88.21
If you don't see a command prompt, try pressing enter.
# chroot /host
```

-
- b. 先に特定したデバイス名 **vdc** を使用して置き換える **/dev/disk** の内容を記録します。

```
# ls -alh /mnt/local-storage/localblock
```

出力例:

```
total 0
drwxr-xr-x. 2 root root 17 Nov 18 15:23 .
drwxr-xr-x. 3 root root 24 Nov 18 15:23 ..
lrwxrwxrwx. 1 root root 8 Nov 18 15:23 vdc -> /dev/vdc
```

- c. **LocalVolume** CR の名前を見つけ、置き換えるデバイス **/dev/disk** を削除またはコメントアウトします。

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

出力例:

```
NAME      AGE
localblock 25h
```

```
# oc edit -n openshift-local-storage localvolume localblock
```

出力例:

```
[...]
storageClassDevices:
- devicePaths:
# - /dev/vdc
storageClassName: localblock
volumeMode: Block
[...]
```

CR の編集後に変更を保存するようにしてください。

11. 置き換えるデバイスで OpenShift Container Platform ノードにログインし、古い **symlink** を削除します。

```
$ oc debug node/worker-0
```

出力例:

```
Starting pod/worker-0-debug ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.88.21
If you don't see a command prompt, try pressing enter.
# chroot /host
```

- a. 置き換えるデバイス名の古い **symlink** を特定します。この例では、デバイス名は **vdc** です。

```
# ls -alh /mnt/local-storage/localblock
```

出力例:

```
total 0
drwxr-xr-x. 2 root root 17 Nov 18 15:23 .
drwxr-xr-x. 3 root root 24 Nov 18 15:23 ..
lrwxrwxrwx. 1 root root 8 Nov 18 15:23 vdc -> /dev/vdc
```

b. **symlink** を削除します。

```
# rm /mnt/local-storage/localblock/vdc
```

c. **symlink** が削除されていることを確認します。

```
# ls -alh /mnt/local-storage/localblock
```

出力例:

```
total 0
drwxr-xr-x. 2 root root 6 Nov 18 17:11 .
drwxr-xr-x. 3 root root 24 Nov 18 15:23 ..
```

12. 古いデバイスを新しいデバイスに置き換えます。

13. 正しい OpenShift Container Platform ノードにログインし、新規ドライブのデバイス名を特定します。同じデバイスを使用しない限り、デバイス名は変更する必要があります。

```
# lsblk
```

出力例:

```
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda                                  252:0  0  40G  0 disk
|-vda1                               252:1  0   4M  0 part
|-vda2                               252:2  0 384M  0 part /boot
`-vda4                               252:4  0 39.6G  0 part
   `-coreos-luks-root-nocrypt 253:0  0 39.6G  0 dm  /sysroot
vdb                                  252:16  0 512B  1 disk
vdd                                  252:32  0 256G  0 disk
```

この例では、新しいデバイス名は **vdd** です。

14. 新しい **/dev/disk** が使用可能になったら、Local Volume CR に新しいディスクエントリーを追加できます。

a. LocalVolume CR を編集し、新規の **/dev/disk** を追加します。
この例では、新しいデバイスは **/dev/vdd** です。

```
# oc edit -n openshift-local-storage localvolume localblock
```

出力例:

```
[...]
storageClassDevices:
```

```

- devicePaths:
  # - /dev/vdc
  - /dev/vdd
  storageClassName: localblock
  volumeMode: Block
[...]

```

CR の編集後に変更を保存するようにしてください。

- 新規 PV が **Available** 状態にあり、正しいサイズであることを確認します。

```
$ oc get pv | grep 256Gi
```

出力例:

```

local-pv-1e31f771 256Gi RWO Delete Bound openshift-storage/ocs-deviceset-
localblock-2-data-0-6xhkf localblock 24h
local-pv-ec7f2b80 256Gi RWO Delete Bound openshift-storage/ocs-deviceset-
localblock-1-data-0-hr2fx localblock 24h
local-pv-8137c873 256Gi RWO Delete Available
localblock 32m

```

- 新規デバイスの OSD を作成します。
新しい OSD をデプロイします。Operator の調整を強制的に実行するには、**rook-ceph-operator** を再起動する必要があります。

- rook-ceph-operator** の名前を特定します。

```
$ oc get -n openshift-storage pod -l app=rook-ceph-operator
```

出力例:

```

NAME                                READY STATUS RESTARTS AGE
rook-ceph-operator-85f6494db4-sg62v 1/1   Running 0      1d20h

```

- rook-ceph-operator** を削除します。

```
$ oc delete -n openshift-storage pod rook-ceph-operator-85f6494db4-sg62v
```

出力例:

```
pod "rook-ceph-operator-85f6494db4-sg62v" deleted
```

この例では、rook-ceph-operator Pod 名は **rook-ceph-operator-85f6494db4-sg62v** です。

- rook-ceph-operator** Pod が再起動していることを確認します。

```
$ oc get -n openshift-storage pod -l app=rook-ceph-operator
```

出力例:


```

NAME                                READY STATUS RESTARTS AGE
rook-ceph-operator-85f6494db4-wx9xx 1/1   Running 0      50s

```

新規 OSD の作成には、Operator が再起動するまでに数分かかる場合があります。

17. **ocs-osd-removal** ジョブを削除します。

```
$ oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```



注記

データ暗号化で外部の鍵管理システム (KMS) を使用する場合は、古い OSD 暗号化キーは孤立したキーであるため、Vault サーバーから削除できます。

検証手順

1. 新しい OSD が実行されていることを確認します。

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd
```

出力例:

```

rook-ceph-osd-0-76d8fb97f9-mn8qz 1/1   Running 0      23m
rook-ceph-osd-1-7c99657cfb-jdzvz 1/1   Running 1      25h
rook-ceph-osd-2-5f9f6dfb5b-2mnw9 1/1   Running 0      25h

```

2. 新規 PVC が作成されていることを確認します。

```
$ oc get -n openshift-storage pvc | grep localblock
```

出力例:

```

ocs-deviceset-localblock-0-data-0-q4q6b Bound local-pv-8137c873    256Gi  RWO
localblock    10m
ocs-deviceset-localblock-1-data-0-hr2fx Bound local-pv-ec7f2b80    256Gi  RWO
localblock    1d20h
ocs-deviceset-localblock-2-data-0-6xhkf Bound local-pv-1e31f771    256Gi  RWO
localblock    1d20h

```

3. (オプション) クラスタでクラスタ全体の暗号化が有効な場合は、新規 OSD デバイスが暗号化されていることを確認します。

- a. 新規 OSD Pod が実行しているノードを特定します。

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/<OSD-
pod-name>
```

```
<OSD-pod-name>
```

これは OSD Pod の名前です。
以下に例を示します。

```
$ oc get -n openshift-storage -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

出力例:

```
NODE
compute-1
```

b. これまでに特定した各ノードに以下を実行します。

i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

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

<node name>

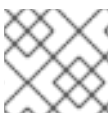
これはノードの名前です。

```
$ chroot /host
```

ii. **ocs-deviceset** 名の横にある **crypt** キーワードを確認します。

```
$ lsblk
```

4. OpenShift Web コンソールにログインし、Storage セクションの OpenShift Data Foundation ダッシュボードでステータスカードをチェックします。



注記

データの完全復旧には、復元されるデータ量により、時間がかかる場合があります。

6.3. IBM Z または IBM LINUXONE インフラストラクチャーで動作するストレージデバイスまたは失敗したストレージデバイスの置き換え

IBM Z または IBM® Linux ONE インフラストラクチャー上の操作可能または故障したストレージデバイスを、新しい Small Computer System Interface (SCSI) ディスクと交換できます。

IBM Z または IBM® LinuxONE は、外部ディスクストレージからの永続ストレージとして SCSI FCP ディスク論理ユニット (SCSI ディスク) に対応します。SCSI ディスクは、FCP デバイス番号、2 つのターゲットワールドワイドポート名 (WWPN1 と WWPN2)、および論理ユニット番号 (LUN) を使用して識別できます。詳細は、https://www.ibm.com/support/knowledgecenter/SSB27U_6.4.0/com.ibm.zvm.v640.hcpa5/scsiover.h を参照してください。

前提条件

- データに耐久性があることを確認する。
 - OpenShift Web コンソールで、**Storage** → **Data Foundation** をクリックします。

- **Storage Systems** タブをクリックし、**ocs-storagecluster-storagesystem** をクリックします。
- **Block and File** ダッシュボードの **Status card** の **Overview** タブで、**Data Resiliency** に緑色のチェックマークが付いていることを確認します。

手順

1. すべてのディスクをリスト表示します。

```
$ lszdev
```

出力例:

```
TYPE      ID
zfcplib  0.0.8204                yes yes
zfcplib  0.0.8204:0x102107630b1b5060:0x4001402900000000 yes no  sda sg0
zfcplib  0.0.8204:0x500407630c0b50a4:0x3002b03000000000 yes yes sdb sg1
qeth     0.0.bdd0:0.0.bdd1:0.0.bdd2                yes no  encbdd0
generic-ccw 0.0.0009                yes no
```

SCSI ディスクは、**ID** セクションの **<device-id>:<wwpn>:<lun-id>** 構造で **zfcplib** として表されます。最初のディスクはオペレーティングシステムに使用されます。1つのストレージデバイスに障害が発生した場合は、それを新しいディスクに置き換えることができます。

2. ディスクを削除します。
ディスクで以下のコマンドを実行し、**scsi-id** を、置き換えるディスクの SCSI ディスク識別子に置き換えます。

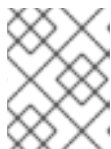
```
$ chzdev -d scsi-id
```

たとえば、以下のコマンドはデバイス ID **0.0.8204**、WWPN **0x500507630a0b50a4**、および LUN **0x4002403000000000** のディスクを1つ削除します。

```
$ chzdev -d 0.0.8204:0x500407630c0b50a4:0x3002b03000000000
```

3. 新しい SCSI ディスクを追加します。

```
$ chzdev -e 0.0.8204:0x500507630b1b50a4:0x4001302a00000000
```



注記

新規ディスクのデバイス ID は、置き換えるディスクと同じである必要があります。新規ディスクは、WWPN および LUN ID で識別されます。

4. すべての FCP デバイスをリスト表示して、新規ディスクが設定されていることを確認します。

```
$ lszdev zfcplib
```

出力例:

TYPE	ID	ON	PERS	NAMES
zfcplun	0.0.8204:0x102107630b1b5060:0x4001402900000000	yes	no	sda sg0
zfcplun	0.0.8204:0x500507630b1b50a4:0x4001302a00000000	yes	yes	sdb sg1

第7章 OSD 削除ツール

7.1. 自動化ツールを使用したオブジェクトストレージデバイスの削除

OpenShift Data Foundation は、オブジェクトストレージデバイス (OSD) の削除プロセスを自動化するツールを提供します。このツールは、OSD の削除中に発生する可能性のあるデータ損失を回避するのに役立ちます。

前提条件

- OSD 削除ツールをローカルマシンにダウンロードする。

```
$ oc image extract --registry-config <pull_secret_path> registry.redhat.io/odf4/odf4-odf-cli-rhel9:v4.15 --confirm --path <architecture_path>:<local_dir>
```

ローカルマシンのアーキテクチャーに応じて **<architecture_path>** を置き換える。

Linux:

- /usr/share/odf/linux/odf-amd64
- /usr/share/odf/linux/odf-ppc64le
- /usr/share/odf/linux/odf-s390x
- /usr/share/odf/linux/odf-arm64

MAC:

- /usr/share/odf/linux/odf-amd64/macosx/odf

Windows:

- /usr/share/odf/linux/odf-amd64/windows/odf

手順

1. 削除する必要がある OSD を特定します。削除が必要な OSD は **CrashLoopBackOff** または **Error** の状態です。以下の例では、OSD **0** を削除します。

```
$ oc get -n openshift-storage pods -l app=rook-ceph-osd -o wide
```

出力例:

```
rook-ceph-osd-0-6d77d6c7c6-m8xj6 0/1 CrashLoopBackOff 0 24h 10.129.0.16
compute-2 <none> <none>
```

2. 以下のコマンドを実行して OSD **0** を削除します。

```
$ odf purge-osd 0
```

OSD の削除が配置グループ (PG) のステータスに影響を与える場合は、**yes-force-destroy-osd** を入力するように求められます。

3. コマンド出力の最後の行に **cephosd: completed removal of OSD 0** が含まれていることを確認します。

4. **oc** デプロイメント **rook-ceph-osd-0** を使用して、対応するデプロイメントが削除されていることを確認します。