Red Hat Ceph Storage 3

オペレーションガイド

Red Hat Ceph Storage の操作タスク

Last Updated: 2023-08-02

Red Hat Ceph Storage 3 オペレーションガイド

Red Hat Ceph Storage の操作タスク

法律上の通知

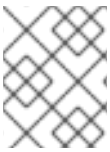
概要

本書では、Red Hat Ceph Storage で動作するタスクを実行する方法について説明します。

目次

第1章 ストレージクラスターのサイズの管理

ストレージ管理者は、ストレージ容量が拡張または縮小する際に Ceph Monitor または OSD を追加または削除することにより、ストレージクラスターのサイズを管理できます。



注記

ストレージクラスターを初めてブートストラップする場合は、Red Hat Ceph Storage 3 Installation Guide でRed Hat Enterprise Linux または Ubuntuを参照してください。

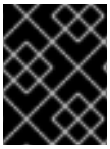
1.1. 前提条件

- 稼働中の Red Hat Ceph Storage クラスター。

1.2. CEPH MONITOR

Ceph モニターは、クラスターマップのマスターコピーを維持する軽量プロセスです。すべての Ceph クライアントは Ceph モニターに問い合わせ、クラスターマップの現在のコピーを取得し、クライアントがプールにバインドし、読み取りと書き込みを可能にします。

Ceph モニターは Paxos プロトコルのバリエーションを使用して、クラスター全体でマップやその他の重要な情報について合意を確立します。Paxos の性質上、Ceph は、クォーラムを確立するためにモニターの大部分を実行する必要があり、合意を確立します。



重要

Red Hat では、実稼働クラスターのサポートを受け取るために、別のホストで少なくとも 3 つのモニターが必要になります。

Red Hat は、奇数のモニターをデプロイすることを推奨します。奇数のモニターは、偶数のモニターよりも障害に対する回復性が高くなっています。たとえば、2 つのモニターのデプロイメントでクォーラムを維持するには、Ceph は障害を許容できません。3 つのモニターでは障害を 1 つ、4 つのモニターでは障害を 1 つ、5 つのモニターでは障害を 2 つ許容します。このため、奇数も推奨されています。要約すると、Ceph は、モニターの大部分 (3 つのうち 2 つ、 4 つのうち 3 つなど) が実行され、相互に通信できるようにする必要があります。

マルチノードの Ceph ストレージクラスターの初回のデプロイには、Red Hat では 3 つのモニターが必要です。3 つ以上のモニターが有効な場合には、一度に数を 2 つ増やします。

モニターは軽量であるため、OpenStack ノードと同じホストで実行できます。ただし、Red Hat は、別のホストでモニターを実行することを推奨します。

重要

Red Hat では、同じノードで Ceph Monitor と OSD を共存させるサポートはありません。これを行うと、ストレージクラスターのパフォーマンスに悪影響を与える可能性があります。

Red Hat は、コンテナー化された環境における Ceph サービスを共存させることのみをサポートしています。

ストレージクラスターからモニターを削除する場合、Ceph モニターは Paxos プロトコルを使用して、マスターストレージクラスターマップに関する合意を確立することを検討してください。クォーラムを確立するには、十分な数のモニターが必要です。

関連情報

- サポートされているすべての Ceph 設定については、ナレッジベースアーティクル **Red Hat Ceph Storage でサポートされる設定** を参照してください。

1.2.1. 新規 Ceph Monitor ノードの準備

新規 Ceph Monitor をストレージクラスターに追加する場合は、それらを別のノードにデプロイします。ノードのハードウェアは、ストレージクラスター内のすべてのノードについて統一する必要があります。

前提条件

- ネットワーク接続
- 新規ノードへの **root** アクセスがあること。
- Installation Guide for Red Hat Enterprise Linux または Ubuntu の **Requirements for Installing Red Hat Ceph Storage**の章を参照してください。

手順

1. 新規ノードをサーバーラックに追加します。
2. 新しいノードをネットワークに接続します。
3. 新規ノードに Red Hat Enterprise Linux 7 または Ubuntu 16.04 のいずれかをインストールします。
4. NTP をインストールし、信頼できるタイムソースを設定します。

```
[root@monitor ~]# yum install ntp
```

5. ファイアウォールを使用している場合は、TCP ポート 6789 を開きます。

Red Hat Enterprise Linux

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6789/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6789/tcp --permanent
```

Ubuntu

```
iptables -I INPUT 1 -i $NIC_NAME -p tcp -s $IP_ADDR/$NETMASK_PREFIX --dport 6789 -j ACCEPT
```

Ubuntu の例

```
[user@monitor ~]$ sudo iptables -I INPUT 1 -i enp6s0 -p tcp -s 192.168.0.11/24 --dport 6789 -j ACCEPT
```

1.2.2. Ansible を使用した Ceph Monitor の追加

Red Hat は、奇数のモニターを維持するために、一度に 2 つのモニターを追加することを推奨します。たとえば、ストレージクラスターに 3 つのモニターがある場合、Red Hat は 5 つのモニターに拡張することを推奨します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター。
- 新規ノードへの **root** アクセスがあること。

手順

1. **[mons]** セクションの下に、新しい Ceph Monitor ノードを **/etc/ansible/hosts** Ansible インベントリーファイルに追加します。

   例

   ```
   [mons]
   monitor01
   monitor02
   monitor03
   $NEW_MONITOR_NODE_NAME
   $NEW_MONITOR_NODE_NAME
   ```

2. Ansible が Ceph ノードと通信できることを確認します。

   ```
   # ansible all -m ping
   ```

3. ディレクトリーを Ansible 設定ディレクトリーに移動します。

   ```
   # cd /usr/share/ceph-ansible
   ```

4. Ansible Playbook の実行:

   ```
   $ ansible-playbook site.yml
   ```

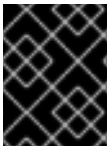
   新規モニターを Ceph のコンテナー化されたデプロイメントに追加する場合は、**site-docker.yml** Playbook を実行します。

   ```
   $ ansible-playbook site-docker.yml
   ```

5. Ansible Playbook が完了すると、新しいモニターノードがストレージクラスターに置かれます。

1.2.3. コマンドラインインターフェイスを使用した Ceph Monitor の追加

Red Hat は、奇数のモニターを維持するために、一度に 2 つのモニターを追加することを推奨します。たとえば、ストレージクラスターに 3 つのモニターがある場合、Red Hat は 5 つのモニターに拡張することを推奨します。



重要

Red Hat は、ノードごとに Ceph モニターデーモンを 1 つだけ実行することを推奨します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター。
- 実行中の Ceph Monitor ノードおよび新規モニターノードへの **root** アクセス権限があること。

手順

1. Red Hat Ceph Storage 3 モニターリポジトリーを追加します。

   Red Hat Enterprise Linux

   ```
   [root@monitor ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-mon-els-rpms
   ```

   Ubuntu

   ```
   [user@monitor ~]$ sudo bash -c 'umask 0077; echo deb https://$CUSTOMER_NAME:$CUSTOMER_PASSWORD@rhcs.download.redhat.com/3-updates/Tools $(lsb_release -sc) main | tee /etc/apt/sources.list.d/Tools.list'
   [user@monitor ~]$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'
   ```

2. **ceph-mon** パッケージを新しい Ceph Monitor ノードにインストールします。

   Red Hat Enterprise Linux

   ```
   [root@monitor ~]# yum install ceph-mon
   ```

   Ubuntu

   ```
   [user@monitor ~]$ sudo apt-get install ceph-mon
   ```

3. ストレージクラスターが起動時または再起動時にモニターを特定するようにするには、モニターの IP アドレスを Ceph 設定ファイルに追加します。
   ストレージクラスターの既存のモニターノードの Ceph 設定ファイルの **[mon]** セクションまたは **[global]** セクションに新規モニターを追加するには、以下を実行します。**mon_host** 設定: DNS で解決できるホスト名または IP アドレスの一覧で、,、;、または で区切ります。オプションとして、新規モニターノード用に Ceph 設定ファイルに特定のセクションを作成することもできます。

   構文

   ```
   [mon]
   mon host = $MONITOR_IP:$PORT $MONITOR_IP:$PORT ... $NEW_MONITOR_IP:$PORT
   ```

   または

```
[mon.$MONITOR_ID]
host = $MONITOR_ID
mon addr = $MONITOR_IP
```

最初のクォーラムグループのモニター部分を作成するには、Ceph 設定ファイルの **[global]** セクションの **mon_initial_members** パラメーターにホスト名を追加する必要もあります。

例

```
[global]
mon initial members = node1 node2 node3 node4 node5
...
[mon]
mon host = 192.168.0.1:6789 192.168.0.2:6789 192.168.0.3:6789 192.168.0.4:6789
192.168.0.5:6789
...
[mon.node4]
host = node4
mon addr = 192.168.0.4

[mon.node5]
host = node5
mon addr = 192.168.0.5
```

重要

本番ストレージクラスターには、高可用性を確保するために、**mon_initial_members**と**mon_host**に少なくとも 3 台のモニターを設定する必要があります。初期モニターが 1 つだけのストレージクラスターにモニターを 2 つ追加しても、それらを**mon_initial_members** および **mon_host** に追加しないと、初期モニターが失敗するとストレージクラスターがロックされます。追加するモニターが **mon_initial_members** および **mon_host** の一部であるモニターを置き換える場合は、新しいモニターを **mon_initial_members** および **mon_host** にも追加する必要があります。

4. 更新された Ceph 設定ファイルをすべての Ceph ノードおよび Ceph クライアントにコピーします。

構文

```
scp /etc/ceph/$CLUSTER_NAME.conf $TARGET_NODE_NAME:/etc/ceph
```

例

```
[root@monitor ~]# scp /etc/ceph/ceph.conf node4:/etc/ceph
```

5. モニターのデータのディレクトリーを新規モニターノードに作成します。

構文

```
mkdir /var/lib/ceph/mon/$CLUSTER_NAME-$MONITOR_ID
```

例

```
[root@monitor ~]# mkdir /var/lib/ceph/mon/ceph-node4
```

6. 実行中のモニターノードおよび新規モニターノードに一時ディレクトリーを作成し、この手順に必要なファイルを保持します。このディレクトリーは、直前の手順で作成したモニターのデフォルトディレクトリーとは異なる必要があり、すべての手順の完了後に削除できます。

   構文

   ```
   mkdir $TEMP_DIRECTORY
   ```

   例

   ```
   [root@monitor ~]# mkdir /tmp/ceph
   ```

7. 実行中のモニターノードから新しいモニターノードに admin キーをコピーし、**ceph** コマンドを実行できるようにします。

   構文

   ```
   scp /etc/ceph/$CLUSTER_NAME.client.admin.keyring $TARGET_NODE_NAME:/etc/ceph
   ```

   例

   ```
   [root@monitor ~]# scp /etc/ceph/ceph.client.admin.keyring node4:/etc/ceph
   ```

8. 実行中のモニターノードから、モニターキーリングを取得します。

   構文

   ```
   ceph auth get mon. -o /$TEMP_DIRECTORY/$KEY_FILE_NAME
   ```

   例

   ```
   [root@monitor ~]# ceph auth get mon. -o /tmp/ceph/ceph_keyring.out
   ```

9. 実行中のモニターノードから、モニターマップを取得します。

   構文

   ```
   ceph mon getmap -o /$TEMP_DIRECTORY/$MONITOR_MAP_FILE
   ```

   例

   ```
   [root@monitor ~]# ceph mon getmap -o /tmp/ceph/ceph_mon_map.out
   ```

10. 収集したモニターデータを新しいモニターノードにコピーします。

    構文

    -

```
scp /tmp/ceph $TARGET_NODE_NAME:/tmp/ceph
```

例

```
[root@monitor ~]# scp /tmp/ceph node4:/tmp/ceph
```

11. 先に収集したデータから、新しいモニターのデータディレクトリーを準備します。モニターからクォーラム情報を取得するため、モニターマップへのパスを**fsid**と共に指定する必要があります。モニターキーリングへのパスも指定する必要があります。

構文

```
ceph-mon -i $MONITOR_ID --mkfs --monmap
/$TEMP_DIRECTORY/$MONITOR_MAP_FILE --keyring
/$TEMP_DIRECTORY/$KEY_FILE_NAME
```

例

```
[root@monitor ~]# ceph-mon -i node4 --mkfs --monmap /tmp/ceph/ceph_mon_map.out --
keyring /tmp/ceph/ceph_keyring.out
```

12. カスタム名を持つストレージクラスターの場合は、以下の行を **/etc/sysconfig/ceph** ファイルに追加します。

Red Hat Enterprise Linux

```
[root@monitor ~]# echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
```

Ubuntu

```
[user@monitor ~]$ sudo echo "CLUSTER=<custom_cluster_name>" >> /etc/default/ceph
```

13. 新規モニターノードで所有者およびグループのパーミッションを更新します。

構文

```
chown -R $OWNER:$GROUP $DIRECTORY_PATH
```

例

```
[root@monitor ~]# chown -R ceph:ceph /var/lib/ceph/mon
[root@monitor ~]# chown -R ceph:ceph /var/log/ceph
[root@monitor ~]# chown -R ceph:ceph /var/run/ceph
[root@monitor ~]# chown -R ceph:ceph /etc/ceph
```

14. 新しい monitor ノードで **ceph-mon** プロセスを有効にして起動します。

構文

```
systemctl enable ceph-mon.target
systemctl enable ceph-mon@$MONITOR_ID
systemctl start ceph-mon@$MONITOR_ID
```

例

```
[root@monitor ~]# systemctl enable ceph-mon.target
[root@monitor ~]# systemctl enable ceph-mon@node4
[root@monitor ~]# systemctl start ceph-mon@node4
```

関連情報

- Installation Guide for Red Hat Enterprise Linux または Ubuntu の**Enabling the Red Hat Ceph Storage Repositories**セクションを参照してください。

1.2.4. Ansible を使用した Ceph Monitor の削除

Ansible で Ceph Monitor を削除するには、Playbook の **shrink-mon.yml** を使用します。

前提条件

- Ansible 管理ノード。
- Ansible によりデプロイされた実行中の Red Hat Ceph Storage クラスター

手順

1. **/usr/share/ceph-ansible/** ディレクトリーに移動します。

```
[user@admin ~]$ cd /usr/share/ceph-ansible
```

2. **infrastructure-playbooks** ディレクトリーから現在のディレクトリーに、**shrink-mon.yml** Playbook をコピーします。

```
[root@admin ceph-ansible]# cp infrastructure-playbooks/shrink-mon.yml .
```

3. Red Hat Ceph Storage の通常デプロイメントまたはコンテナー化されたデプロイメント用に、**shrink-mon.yml** Playbook を実行します。

```
[user@admin ceph-ansible]$ ansible-playbook shrink-mon.yml -e mon_to_kill=<hostname> -u <ansible-user>
```

以下を置き換えます。

- **<hostname>** は、Monitor ノードの短縮ホスト名に置き換えます。複数の Monitor を削除するには、ホスト名をコンマで区切ります。
- **<ansible-user>** は、Ansible ユーザーの名前に置き換えてください。

たとえば、ホスト名 **monitor1** のノードにある Monitor を削除するには、以下を実行します。

```
[user@admin ceph-ansible]$ ansible-playbook shrink-mon.yml -e mon_to_kill=monitor1 -u user
```

4. クラスター内のすべての Ceph 設定ファイルから Monitor エントリーを削除します。

5. Monitor が正常に削除されていることを確認します。

```
[root@monitor ~]# ceph -s
```

関連情報

- Red Hat Ceph Storage のインストールに関する情報は、Installation Guide for Red Hat Enterprise Linux または Ubuntuを参照してください。

1.2.5. コマンドラインインターフェイスを使用した Ceph Monitor の削除

Ceph Monitor を削除するには、ストレージクラスターから **ceph-mon** デーモンを削除し、ストレージクラスターマップを更新します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター。
- モニターノードへの **root** アクセス権限がある。

手順

1. monitor サービスを停止します。

   構文

```
systemctl stop ceph-mon@$MONITOR_ID
```

   例

```
[root@monitor ~]# systemctl stop ceph-mon@node3
```

2. ストレージクラスターからモニターを削除します。

   構文

```
ceph mon remove $MONITOR_ID
```

   例

```
[root@monitor ~]# ceph mon remove node3
```

3. Ceph 設定ファイル (デフォルトでは **/etc/ceph/ceph.conf**) からモニターエントリーを削除します。

4. Ceph 設定ファイルを、ストレージクラスターの残りの全 Ceph ノードに再配布します。

   構文

```
scp /etc/ceph/$CLUSTER_NAME.conf $USER_NAME@$TARGET_NODE_NAME:/etc/ceph/
```

例

```
[root@monitor ~]# scp /etc/ceph/ceph.conf root@$node1:/etc/ceph/
```

5. コンテナーのみ。モニターサービスを無効にします。

注記

コンテナーを使用している場合にのみ、ステップ 5 - 9 を実行します。

構文

```
systemctl disable ceph-mon@$MONITOR_ID
```

例

```
[root@monitor ~]# systemctl disable ceph-mon@node3
```

6. コンテナーのみ。systemd からサービスを削除します。

```
[root@monitor ~]# rm /etc/systemd/system/ceph-mon@.service
```

7. コンテナーのみ。systemd マネージャー設定を再読み込みします。

```
[root@monitor ~]# systemctl daemon-reload
```

8. コンテナーのみ。障害が発生したモニターユニットの状態をリセットします。

```
[root@monitor ~]# systemctl reset-failed
```

9. コンテナーのみ。**ceph-mon** RPM を削除します。

```
[root@monitor ~]# docker exec node3 yum remove ceph-mon
```

10. モニターデータをアーカイブします。

構文

```
mv /var/lib/ceph/mon/$CLUSTER_NAME-$MONITOR_ID /var/lib/ceph/mon/removed-$CLUSTER_NAME-$MONITOR_ID
```

例

```
[root@monitor ~]# mv /var/lib/ceph/mon/ceph-node3 /var/lib/ceph/mon/removed-ceph-node3
```

11. モニターデータを削除します。

構文

```
rm -r /var/lib/ceph/mon/$CLUSTER_NAME-$MONITOR_ID
```

例

```
[root@monitor ~]# rm -r /var/lib/ceph/mon/ceph-node3
```

関連情報

- 詳細は、ナレッジベースソリューションのHow to re-deploy Ceph Monitor in a director deployed Ceph clusterを参照してください。

1.2.6. 異常なストレージクラスターからの Ceph Monitor の削除

この手順では、正常でないストレージクラスターから **ceph-mon** デーモンを削除します。配置グループが **active + clean** にならない、正常でないストレージクラスター。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター。
- モニターノードへの **root** アクセス権限がある。
- Ceph Monitor ノードが少なくとも 1 台実行している。

手順

1. 存続しているモニターを特定し、そのノードにログインします。

```
[root@monitor ~]# ceph mon dump
[root@monitor ~]# ssh $MONITOR_HOST_NAME
```

2. **ceph-mon** デーモンを停止し、**monmap** ファイルのコピーを抽出します。

構文

```
systemctl stop ceph-mon@$MONITOR_ID
ceph-mon -i $MONITOR_ID --extract-monmap $TEMPORARY_PATH
```

例

```
[root@monitor ~]# systemctl stop ceph-mon@node1
[root@monitor ~]# ceph-mon -i node1 --extract-monmap /tmp/monmap
```

3. 存続していないモニターを削除します。

構文

```
monmaptool $TEMPORARY_PATH --rm $MONITOR_ID
```

例

```
[root@monitor ~]# monmaptool /tmp/monmap --rm node2
```

4. 削除されたモニターを含む存続しているモニターマップを、存続しているモニターに挿入します。

構文

```
ceph-mon -i $MONITOR_ID --inject-monmap $TEMPORARY_PATH
```

例

```
[root@monitor ~]# ceph-mon -i node1 --inject-monmap /tmp/monmap
```

1.3. CEPH OSD

Red Hat Ceph Storage クラスターが稼働している場合は、ランタイム時に OSD をストレージクラスターに追加できます。

Ceph OSD は、通常 1 つのストレージドライブおよびノード内の関連付けられたジャーナル用に 1 つの **ceph-osd** デーモンで設定されます。ノードに複数のストレージドライブがある場合は、ドライブごとに 1 つの **ceph-osd** デーモンをマッピングします。

Red Hat は、クラスターの容量を定期的に確認して、ストレージ容量の最後に到達するかどうかを確認することを推奨します。ストレージクラスターが **ほぼ完全** の比率に達すると、1 つ以上の OSD を追加してストレージクラスターの容量を拡張します。

Red Hat Ceph Storage クラスターのサイズを縮小したり、ハードウェアを置き換える場合は、ランタイム時に OSD を削除することも可能です。ノードに複数のストレージドライブがある場合には、そのドライブ用に **ceph-osd** デーモンのいずれかを削除する必要もあります。通常、ストレージクラスターの容量を確認して、容量の上限に達したかどうかを確認することが推奨されます。ストレージクラスターが **ほぼ完全** の比率ではないことを OSD を削除する場合。

重要

OSD を追加する前に、ストレージクラスターが **完全な** 比率を超えないようにします。ストレージクラスターが **ほぼ完全な** 比率に達した後に OSD の障害が発生すると、ストレージクラスターが **完全な** 比率を超過する可能性があります。Ceph は、ストレージ容量の問題を解決するまでデータを保護するための書き込みアクセスをブロックします。**完全な** 比率の影響を考慮せずに OSD を削除しないでください。

1.3.1. Ceph OSD ノードの設定

同様に、Ceph OSD とサポートするハードウェアを、OSD を使用するプールのストレージストラテジーとして設定する必要があります。Ceph は、一貫性のあるパフォーマンスプロファイルを確保するために、プール全体でハードウェアを統一します。最適なパフォーマンスを得るには、同じタイプまたはサイズのドライブのある CRUSH 階層を検討してください。詳細は、Storage Strategies を参照してください。

異なるサイズのドライブを追加する場合は、それに応じて重みを調整しなければならない場合があります。OSD を CRUSH マップに追加する場合は、新規 OSD の重みを考慮してください。ハードドライブの容量は、1 年あたり約 40% 増加するため、新しい OSD ノードはストレージクラスターの古いノードよりも大きなハードドライブを持つ可能性があります。つまり、重みが大きくなる可能性があります。

新規インストールを実行する前に、Installation Guide for Red Hat Enterprise Linux または UbuntuのRequirements for Installing Red Hat Ceph Storageの章を参照してください。

1.3.2. コンテナーの OSD ID のドライブへのマッピング

場合によっては、コンテナー化された OSD が使用しているドライブを特定する必要がある場合があります。たとえば、OSD に問題がある場合には、ドライブのステータスを検証するために使用するドライブを把握しなければならない場合があります。また、コンテナー化されていない OSD の場合は、OSD ID を参照して開始および停止しますが、コンテナー化された OSD を開始および停止するには、使用するドライブを参照する必要があります。

前提条件

- コンテナー化環境で実行中の Red Hat Ceph Storage クラスター
- コンテナーホストへの **root** アクセス権限があること。

手順

1. コンテナー名を見つけます。たとえば、**osd.5** に関連付けられたドライブを特定するには、**osd.5** が実行中のコンテナーノードでターミナルを開き、**docker ps** を実行してすべてのコンテナーを一覧表示します。

   例

   ```
   [root@ceph3 ~]# docker ps
   CONTAINER ID        IMAGE                                                 COMMAND             CREATED
   STATUS              PORTS               NAMES
   3a866f927b74        registry.access.redhat.com/rhceph/rhceph-3-rhel7:latest   "/entrypoint.sh"
   About an hour ago   Up About an hour                        ceph-osd-ceph3-sdd
   91f3d4829079        registry.access.redhat.com/rhceph/rhceph-3-rhel7:latest   "/entrypoint.sh"
   22 hours ago        Up 22 hours                             ceph-osd-ceph3-sdb
   73dfe4021a49        registry.access.redhat.com/rhceph/rhceph-3-rhel7:latest   "/entrypoint.sh"
   7 days ago          Up 7 days                               ceph-osd-ceph3-sdf
   90f6d756af39        registry.access.redhat.com/rhceph/rhceph-3-rhel7:latest   "/entrypoint.sh"
   7 days ago          Up 7 days                               ceph-osd-ceph3-sde
   e66d6e33b306        registry.access.redhat.com/rhceph/rhceph-3-rhel7:latest   "/entrypoint.sh"
   7 days ago          Up 7 days                               ceph-mgr-ceph3
   733f37aafd23        registry.access.redhat.com/rhceph/rhceph-3-rhel7:latest   "/entrypoint.sh"
   7 days ago          Up 7 days                               ceph-mon-ceph3
   ```

2. **docker exec** を使用して、直前の出力から OSD コンテナー名上で **ceph-volume lvm list** を実行します。

   例

   ```
   [root@ceph3 ~]# docker exec ceph-osd-ceph3-sdb ceph-volume lvm list
   ====== osd.5 =======

     [journal]    /dev/journals/journal1

         journal uuid              C65n7d-B1gy-cqX3-vZKY-ZoE0-IEYM-HnIJzs
         osd id                    1
         cluster fsid              ce454d91-d748-4751-a318-ff7f7aa18ffd
         type                      journal
   ```

```
    osd fsid                661b24f8-e062-482b-8110-826ffe7f13fa
    data uuid               SlEgHe-jX1H-QBQk-Sce0-RUls-8KlY-g8HgcZ
    journal device          /dev/journals/journal1
    data device             /dev/test_group/data-lv2
    devices                 /dev/sda

[data]    /dev/test_group/data-lv2

    journal uuid            C65n7d-B1gy-cqX3-vZKY-ZoE0-IEYM-HnIJzs
    osd id                  1
    cluster fsid            ce454d91-d748-4751-a318-ff7f7aa18ffd
    type                    data
    osd fsid                661b24f8-e062-482b-8110-826ffe7f13fa
    data uuid               SlEgHe-jX1H-QBQk-Sce0-RUls-8KlY-g8HgcZ
    journal device          /dev/journals/journal1
    data device             /dev/test_group/data-lv2
    devices                 /dev/sdb
```

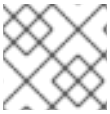
この出力から、**osd.5** が **/dev/sdb** に関連付けられていることがわかります。

関連情報

- 詳細は、障害のある OSD ディスクの置き換え を参照してください。

1.3.3. 同じディスクトポロジーを持つ Ansible を使用した Ceph OSD の追加

同じディスクトポロジーを持つ Ceph OSD の場合には、Ansible は **/usr/share/ceph-ansible/group_vars/osds** ファイルの **devices:** セクションで指定されているのと同じデバイスパスを使用して、他の OSD ノードと同じ数の OSD を追加します。



注記

新しい Ceph OSD ノードは、残りの OSD と同じ設定になります。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター。
- Installation Guide for Red Hat Enterprise Linux または Ubuntu の **Requirements for Installing Red Hat Ceph Storage** の章を参照してください。
- 新規ノードへの **root** アクセスがあること。
- ストレージクラスター内の他の OSD ノードと同じ数の OSD データドライブ。

手順

1. **[osds]** セクションの下に Ceph OSD ノードを **/etc/ansible/hosts** ファイルに追加します。

   例

```
[osds]
...
osd06
```

```
$NEW_OSD_NODE_NAME
```

2. Ansible が Ceph ノードに到達できることを確認します。

```
[user@admin ~]$ ansible all -m ping
```

3. Ansible 設定ディレクトリーに移動します。

```
[user@admin ~]$ cd /usr/share/ceph-ansible
```

4. **add-osd.yml** ファイルを **/usr/share/ceph-ansible/** ディレクトリーにコピーします。

```
[user@admin ceph-ansible]$ sudo cp infrastructure-playbooks/add-osd.yml .
```

5. Ceph の通常のデプロイメントまたはコンテナー化されたデプロイメント向けに Ansible Playbook を実行します。

```
[user@admin ceph-ansible]$ ansible-playbook add-osd.yml
```

注記

OSD を追加する際に、**PGs were not reported as active+clean** で Playbook が失敗する場合は、**all.yml** ファイルに以下の変数を設定し、再試行と遅延を調整します。

```
# OSD handler checks
handler_health_osd_check_retries: 50
handler_health_osd_check_delay: 30
```

1.3.4. 異なるディスクトポロジーが設定された Ansible を使用した Ceph OSD の追加

異なるディスクトポロジーを持つ Ceph OSD については、新しい OSD ノードを既存のストレージクラスターに追加する 2 つの方法があります。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター。
- Installation Guide for Red Hat Enterprise Linux または Ubuntu の **Requirements for Installing Red Hat Ceph Storage** の章を参照してください。
- 新規ノードへの **root** アクセスがあること。

手順

1. **最初の操作**

   a. **[osds]** セクションの下に、新しい Ceph OSD ノードを **/etc/ansible/hosts** ファイルに追加します。

   例

```
[osds]
...
osd06
$NEW_OSD_NODE_NAME
```

b. ストレージクラスターに追加される新しい Ceph OSD ノードを **/etc/ansible/host_vars/** ディレクトリーに作成します。

構文

```
touch /etc/ansible/host_vars/$NEW_OSD_NODE_NAME
```

例

```
[root@admin ~]# touch /etc/ansible/host_vars/osd07
```

c. 新しいファイルを編集し、**devices:** および **dedicated_devices:** セクションをファイルに追加します。以下の各セクションに、**-** およびスペースを追加してから、この OSD ノードのブロックデバイス名への完全パスを追加します。

例

```
devices:
  - /dev/sdc
  - /dev/sdd
  - /dev/sde
  - /dev/sdf

dedicated_devices:
  - /dev/sda
  - /dev/sda
  - /dev/sdb
  - /dev/sdb
```

d. Ansible がすべての Ceph ノードに到達できることを確認します。

```
[user@admin ~]$ ansible all -m ping
```

e. ディレクトリーを Ansible 設定ディレクトリーに移動します。

```
[user@admin ~]$ cd /usr/share/ceph-ansible
```

f. **add-osd.yml** ファイルを **/usr/share/ceph-ansible/** ディレクトリーにコピーします。

```
[user@admin ceph-ansible]$ sudo cp infrastructure-playbooks/add-osd.yml .
```

g. Ansible Playbook の実行:

```
[user@admin ceph-ansible]$ ansible-playbook add-osd.yml
```

2. **2 つ目の方法**

a. 新しい OSD ノード名を **/etc/ansible/hosts** ファイルに追加し、**devices** オプションおよび **dedicated_devices** オプションを使用して、異なるディスクトポロジーを指定します。

例

```
[osds]
...
osd07 devices="['/dev/sdc', '/dev/sdd', '/dev/sde', '/dev/sdf']" dedicated_devices="
['/dev/sda', '/dev/sda', '/dev/sdb', '/dev/sdb']"
```

b. Ansible がすべての Ceph ノードに到達できることを確認します。

```
[user@admin ~]$ ansible all -m ping
```

c. ディレクトリーを Ansible 設定ディレクトリーに移動します。

```
[user@admin ~]$ cd /usr/share/ceph-ansible
```

d. **add-osd.yml** ファイルを **/usr/share/ceph-ansible/** ディレクトリーにコピーします。

```
[user@admin ceph-ansible]$ sudo cp infrastructure-playbooks/add-osd.yml .
```

e. Ansible Playbook の実行:

```
[user@admin ceph-ansible]$ ansible-playbook add-osd.yml
```

1.3.5. コマンドラインインターフェイスを使用した Ceph OSD の追加

OSD を Red Hat Ceph Storage に手動で追加するハイレベルのワークフローを以下に示します。

1. **ceph-osd** パッケージをインストールして、新規 OSD インスタンスを作成します。
2. OSD データおよびジャーナルドライブを準備してマウントします。
3. 新規 OSD ノードを CRUSH マップに追加します。
4. 所有者およびグループパーミッションを更新します。
5. **ceph-osd** デーモンを有効にして起動します。

重要

ceph-disk コマンドは非推奨となりました。**ceph-volume** コマンドは、コマンドラインインターフェイスから OSD をデプロイするのに推奨される方法です。現在、**ceph-volume** コマンドは **lvm** プラグインのみをサポートしています。Red Hat は、本ガイドで両方のコマンドを参照として使用している例を提供します。これにより、ストレージ管理者は **ceph-disk** に依存するカスタムスクリプトを **ceph-volume** に変換できます。

ceph-volume コマンドの使用方法は、Red Hat Ceph Storage Administration Guide を参照してください。

注記

カスタムストレージクラスター名の場合は、**ceph** コマンドおよび **ceph-osd** コマンドで **--cluster $CLUSTER_NAME** オプションを使用します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター。
- Installation Guide for Red Hat Enterprise Linux または Ubuntu の **Requirements for Installing Red Hat Ceph Storage** の章を参照してください。
- 新規ノードへの **root** アクセスがあること。

手順

1. Red Hat Ceph Storage 3 OSD ソフトウェアリポジトリーを有効にします。

   Red Hat Enterprise Linux

   ```
   [root@osd ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-osd-els-rpms
   ```

   Ubuntu

   ```
   [user@osd ~]$ sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/Tools
   $(lsb_release -sc) main | tee /etc/apt/sources.list.d/Tools.list'
   [user@osd ~]$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-
   key add -'
   ```

2. **/etc/ceph/** ディレクトリーを作成します。

   ```
   # mkdir /etc/ceph
   ```

3. 新しい OSD ノードで、Ceph 管理キーリングと設定ファイルを Ceph Monitor ノードの 1 つからコピーします。

   構文

   ```
   scp
   $USER_NAME@$MONITOR_HOST_NAME:/etc/ceph/$CLUSTER_NAME.client.admin.keyrin
   g /etc/ceph
   scp $USER_NAME@$MONITOR_HOST_NAME:/etc/ceph/$CLUSTER_NAME.conf
   /etc/ceph
   ```

   例

   ```
   [root@osd ~]# scp root@node1:/etc/ceph/ceph.client.admin.keyring /etc/ceph/
   [root@osd ~]# scp root@node1:/etc/ceph/ceph.conf /etc/ceph/
   ```

4. **ceph-osd** パッケージを新しい Ceph OSD ノードにインストールします。

   Red Hat Enterprise Linux

```
[root@osd ~]# yum install ceph-osd
```

Ubuntu

```
[user@osd ~]$ sudo apt-get install ceph-osd
```

5. 新規 OSD について、ジャーナルを共存させるか、または専用のジャーナルを使用するかどうかを決定します。

注記

--filestore オプションが必要です。

a. ジャーナルを共存させる OSD の場合:

構文

```
[root@osd ~]# ceph-disk --setuser ceph --setgroup ceph prepare  --filestore
/dev/$DEVICE_NAME
```

例

```
[root@osd ~]# ceph-disk --setuser ceph --setgroup ceph prepare  --filestore /dev/sda
```

b. 専用のジャーナルを持つ OSD の場合:

構文

```
[root@osd ~]# ceph-disk --setuser ceph --setgroup ceph prepare  --filestore
/dev/$DEVICE_NAME /dev/$JOURNAL_DEVICE_NAME
```

または

```
[root@osd ~]# ceph-volume lvm prepare  --filestore --data /dev/$DEVICE_NAME --
journal /dev/$JOURNAL_DEVICE_NAME
```

例

```
[root@osd ~]# ceph-disk --setuser ceph --setgroup ceph prepare  --filestore /dev/sda
/dev/sdb
```

```
[root@osd ~]# ceph-volume lvm prepare  --filestore --data /dev/vg00/lvol1 --journal
/dev/sdb
```

6. **noup** オプションを設定します。

```
[root@osd ~]# ceph osd set noup
```

7. 新しい OSD をアクティベートします。

構文

構文

```
[root@osd ~]# ceph-disk activate /dev/$DEVICE_NAME
```

または

```
[root@osd ~]# ceph-volume lvm activate --filestore $OSD_ID $OSD_FSID
```

例

```
[root@osd ~]# ceph-disk activate /dev/sda
```

```
[root@osd ~]# ceph-volume lvm activate --filestore 0 6cc43680-4f6e-4feb-92ff-9c7ba204120e
```

8. OSD を CRUSH マップに追加します。

構文

```
ceph osd crush add $OSD_ID $WEIGHT [$BUCKET_TYPE=$BUCKET_NAME ...]
```

例

```
[root@osd ~]# ceph osd crush add 4 1 host=node4
```

注記

複数のバケットを指定する場合、コマンドは OSD を指定したバケットから最も具体的なバケットに配置、**および** 指定した他のバケットに従ってバケットを移動します。

注記

CRUSH マップを手動で編集することもできます。Red Hat Ceph Storage 3 の Storage Strategies ガイドの Editing a CRUSH map セクションを参照してください。

重要

ルートバケットのみを指定する場合、OSD はルートに直接アタッチしますが、CRUSH ルールは OSD がホストバケット内に置かれることを想定します。

9. **noup** オプションの設定を解除します。

```
[root@osd ~]# ceph osd unset noup
```

10. 新規作成されたディレクトリーの所有者とグループのパーミッションを更新します。

構文

```
chown -R $OWNER:$GROUP $PATH_TO_DIRECTORY
```

例

```
[root@osd ~]# chown -R ceph:ceph /var/lib/ceph/osd
[root@osd ~]# chown -R ceph:ceph /var/log/ceph
[root@osd ~]# chown -R ceph:ceph /var/run/ceph
[root@osd ~]# chown -R ceph:ceph /etc/ceph
```

11. カスタム名のクラスターを使用する場合は、以下の行を適切なファイルに追加します。

Red Hat Enterprise Linux

```
[root@osd ~]# echo "CLUSTER=$CLUSTER_NAME" >> /etc/sysconfig/ceph
```

Ubuntu

```
[user@osd ~]$ sudo echo "CLUSTER=$CLUSTER_NAME" >> /etc/default/ceph
```

$CLUSTER_NAME は、カスタムクラスター名に置き換えます。

12. 新規 OSD が **起動** し、データを受信する準備ができていることを確認するには、OSD サービスを有効にして起動します。

構文

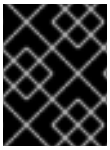
```
systemctl enable ceph-osd@$OSD_ID
systemctl start ceph-osd@$OSD_ID
```

例

```
[root@osd ~]# systemctl enable ceph-osd@4
[root@osd ~]# systemctl start ceph-osd@4
```

1.3.6. Ansible を使用した Ceph OSD の削除

Red Hat Ceph Storage クラスターの容量をスケールダウンしないといけない場合があります。Ansible を使用して Red Hat Ceph Storage クラスターから OSD を削除するには、使用する OSD のシナリオに応じて、Playbook **shrink-osd.yml** または **shrink-osd-ceph-disk.yml** を実行します。**osd_scenario** が **collocated** または **non-collocated** に設定されている場合には、Playbook **shrink-osd-ceph-disk.yml** を使用します。**osd_scenario** を **lvm** に設定した場合は、Playbook **shrink-osd.yml** を使用します。



重要

ストレージクラスターから OSD を削除すると、その OSD に含まれるすべてのデータが破棄されます。

前提条件

- Ansible によりデプロイされた実行中の Red Hat Ceph Storage
- 実行中の Ansible 管理ノード
- Ansible 管理ノードへの root レベルのアクセス。

手順

1. **/usr/share/ceph-ansible/** ディレクトリーに移動します。

```
[user@admin ~]$ cd /usr/share/ceph-ansible
```

2. Ceph Monitor ノードの **/etc/ceph/** から、削除する OSD が含まれるノードに管理キーリングをコピーします。

3. **infrastructure-playbooks** ディレクトリーから現在のディレクトリーに、適切な Playbook をコピーします。

```
[root@admin ceph-ansible]# cp infrastructure-playbooks/shrink-osd.yml .
```

または

```
[root@admin ceph-ansible]# cp infrastructure-playbooks/shrink-osd-ceph-disk.yml .
```

4. **ベアメタル** または **コンテナー** のデプロイメントの場合は、適切な Ansible Playbook を実行します。

構文

```
ansible-playbook shrink-osd.yml -e osd_to_kill=$ID -u $ANSIBLE_USER
```

または

```
ansible-playbook shrink-osd-ceph-disk.yml -e osd_to_kill=$ID -u $ANSIBLE_USER
```

以下を置き換えます。

- **$id** は、OSD の ID に置き換えます。複数の OSD を削除するには、OSD ID をコンマで区切ります。
- **$ANSIBLE_USER** は、Ansible ユーザーの名前に置き換えてください。

例

```
[user@admin ceph-ansible]$ ansible-playbook shrink-osd.yml -e osd_to_kill=1 -u user
```

または

```
[user@admin ceph-ansible]$ ansible-playbook shrink-osd-ceph-disk.yml -e osd_to_kill=1 -u user
```

5. OSD が正常に削除されていることを確認します。

```
[root@mon ~]# ceph osd tree
```

関連情報

- 詳細は、Red Hat Enterprise Linux または UbuntuのRed Hat Ceph Storage Installation Guideを参照してください。

1.3.7. コマンドラインインターフェイスを使用した Ceph OSD の削除

ストレージクラスターから OSD を削除するには、クラスターマップの更新、その認証キーの削除、OSD マップからの OSD の削除、および **ceph.conf** ファイルからの OSD の削除を行う必要があります。ノードに複数のドライブがある場合は、この手順を繰り返して、それぞれのドライブについて OSD を削除する必要がある場合があります。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- 利用可能な OSD が十分になるようにして、ストレージクラスターが **ほぼ完全** な比率にならないようにしてください。
- OSD ノードへの **root** アクセス権限があること。

手順

1. OSD サービスを無効にし、停止します。

   構文

   ```
   systemctl disable ceph-osd@$OSD_ID
   systemctl stop ceph-osd@$OSD_ID
   ```

   例

   ```
   [root@osd ~]# systemctl disable ceph-osd@4
   [root@osd ~]# systemctl stop ceph-osd@4
   ```

   OSD が停止したら、**停止** します。

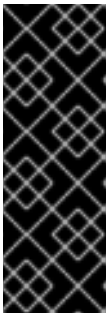
2. ストレージクラスターから OSD を削除します。

   構文

   ```
   ceph osd out $OSD_ID
   ```

   例

   ```
   [root@osd ~]# ceph osd out 4
   ```



重要

OSD が削除されると、Ceph は再バランス調整を開始し、データをストレージクラスター内の他の OSD にコピーします。Red Hat は、次の手順に進む前に、ストレージクラスターが **active+clean** になるまで待つことを推奨します。データの移行を確認するには、以下のコマンドを実行します。

```
[root@monitor ~]# ceph -w
```

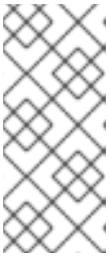
3. CRUSH マップから OSD を削除して、データを受信しないようにします。

   構文

   ```
   ceph osd crush remove $OSD_NAME
   ```

   例

   ```
   [root@osd ~]# ceph osd crush remove osd.4
   ```



注記

CRUSH マップをコンパイルし、デバイス一覧から OSD を削除して、ホストバケットの項目としてデバイスを削除するか、またはホストバケットを削除することもできます。CRUSH マップにあり、ホストを削除するには、マップを再コンパイルしてからこれを設定します。詳細は、Storage Strategies Guide を参照してください。

4. OSD 認証キーを削除します。

   構文

   ```
   ceph auth del osd.$OSD_ID
   ```

   例

   ```
   [root@osd ~]# ceph auth del osd.4
   ```

5. OSD を削除します。

   構文

   ```
   ceph osd rm $OSD_ID
   ```

   例

   ```
   [root@osd ~]# ceph osd rm 4
   ```

6. ストレージクラスターの設定ファイル (デフォルトでは **/etc/ceph.conf**) を編集して、OSD エントリーが存在する場合は削除します。

   例

```
[osd.4]
host = $HOST_NAME
```

7. OSD を手動で追加している場合は、**/etc/fstab** ファイルで OSD への参照を削除します。
8. 更新された設定ファイルを、ストレージクラスター内の他のすべてのノードの **/etc/ceph/** ディレクトリーにコピーします。

構文

```
scp /etc/ceph/$CLUSTER_NAME.conf $USER_NAME@$HOST_NAME:/etc/ceph/
```

例

```
[root@osd ~]# scp /etc/ceph/ceph.conf root@node4:/etc/ceph/
```

1.3.8. コマンドラインインターフェイスを使用したジャーナルの置き換え

ジャーナルとデータデバイスが同じ物理デバイスにある場合 (**osd_scenario: collocated** を使用する場合など) にジャーナルを置き換えるには、OSD 全体を置き換える必要があります。ただし、ジャーナルがデータデバイスとは別の物理デバイスにある OSD では (**osd_scenario: non-collocated** を使用する場合)、ジャーナルデバイスのみを置き換えることができます。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター
- 新しいパーティションまたはストレージデバイス。

手順

1. クラスターを **noout** に設定してバックフィルを防ぎます。

```
[root@osd1 ~]# ceph osd set noout
```

2. ジャーナルが変更された OSD を停止します。

```
[root@osd1 ~]# systemctl stop ceph-osd@$OSD_ID
```

3. OSD のジャーナルをフラッシュします。

```
[root@osd1 ~]# ceph-osd -i $OSD_ID --flush-journal
```

4. 古いジャーナルパーティションを削除して、パーティションの UUID が新しいパーティションと競合しないようにします。

```
sgdisk --delete=$OLD_PART_NUM -- $OLD_DEV_PATH
```

置き換え

- **$OLD_PART_NUM** は、古いジャーナルデバイスのパーティション番号に置き換えます。

- **$OLD_DEV_PATH** は、古いジャーナルデバイスへのパスに置き換えます。

例

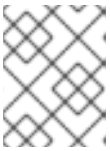
```
[root@osd1 ~]# sgdisk --delete=1 -- /dev/sda
```

5. 新しいデバイスに新しいジャーナルパーティションを作成します。この **sgdisk** コマンドは、次に利用可能なパーティション番号を自動的に使用します。

```
sgdisk --new=0:0:$JOURNAL_SIZE -- $NEW_DEV_PATH
```

置き換え

- **$JOURNAL_SIZE** は、環境に適したジャーナルサイズに置き換えます (例: **10240M**)。
- **NEW_DEV_PATH** は、新規ジャーナルに使用するデバイスへのパスに置き換えます。



注記

ジャーナルの最小デフォルトサイズは 5 GB です。通常、10 GB を超える値は必要ありません。詳細は、Red Hat サポート にお問い合わせください。

例

```
[root@osd1 ~]# sgdisk --new=0:0:10240M -- /dev/sda
```

6. 新しいパーティションに適切なパラメーターを設定します。

```
sgdisk --change-name=0:"ceph journal" --partition-guid=0:$OLD_PART_UUID --
typecode=0:45b0969e-9b03-4f30-b4c6-b4b80ceff106 --mbrtogpt -- $NEW_DEV_PATH
```

置き換え

- **$OLD_PART_UUID** は、関連する OSD の **journal_uuid** ファイルの UUID に置き換えます。たとえば、OSD が **0** の場合は、**/var/lib/ceph/osd/ceph-0/journal_uuid** の UUID を使用します。
- **NEW_DEV_PATH** は、新規ジャーナルに使用するデバイスへのパスに置き換えます。

例

```
[root@osd1 ~]# sgdisk --change-name=0:"ceph journal" --partition-guid=0:a1279726-a32d-
4101-880d-e8573bb11c16 --typecode=0:097c058d-0758-4199-a787-ce9bacb13f48 --
mbrtogpt -- /dev/sda
```

上記の **sgdisk** コマンドを実行すると、新しいジャーナルパーティションが Ceph 用に準備され、ジャーナルを作成できます。

重要

sgdisk の制限により、パーティションが正常に作成されないため、このコマンドをパーティション作成コマンドと組み合わせることはできません。

7. 新しいジャーナルを作成します。

```
[root@osd1 ~]# ceph-osd -i $OSD_ID --mkjournal
```

8. OSD を起動します。

```
[root@osd1 ~]# systemctl start ceph-osd@$OSD_ID
```

1. OSD で **noout** フラグを削除します。

```
[root@osd1 ~]# ceph osd unset noout
```

2. ジャーナルが正しいデバイスに関連付けられていることを確認します。

```
[root@osd1 ~]# ceph-disk list
```

1.3.9. データ移行の監視

OSD を CRUSH マップに追加または削除すると、Ceph は配置グループを新規または既存の OSD に移行してデータのリバランスを開始します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- 最近 OSD を追加または削除した。

手順

1. データの移行を確認するには、以下を実行します。

```
[root@monitor ~]# ceph -w
```

2. 配置グループのステータスが **active+clean** から **active, some degraded objects** し、最後に移行の完了時に **active+clean** に変わるのを確認します。

3. ユーティリティーを終了するには、**Ctrl + C** を押します。

1.4. 配置グループの再計算

配置グループ (PG) は、利用可能な OSD にまたがるプールデータの分散を定義します。配置グループは、使用する冗長性アルゴリズムに基づいて構築されます。3 方向のレプリケーションでは、冗長性は 3 つの異なる OSD を使用するように設定されます。イレイジャーコーディングされたプールの場合、使用する OSD の数はチャンクの数で定義されます。

プールを定義する場合、配置グループの数によって、使用可能なすべての OSD にデータが分散される粒度のグレードが定義されます。数値が大きいほど、容量負荷の均等化が向上します。ただし、データ

を再構築する場合は配置グループの処理も重要であるため、事前に慎重に選択する必要があります。計算をサポートするには、アジャイル環境の生成に利用できるツールを使用できます。

ストレージクラスターの有効期間中、プールが最初に予想される制限を上回る可能性があります。ドライブの数が増えると、再計算が推奨されます。OSD ごとの配置グループの数は約 100 である必要があります。ストレージクラスターに OSD を追加すると、OSD あたりの PG の数は時間の経過とともに減少します。ストレージクラスターで最初に 120 ドライブを使用し、プールの **pg_num** を 4000 に設定すると、レプリケーション係数が 3 の場合に、OSD ごとに 100 PG に設定されます。時間の経過とともに、OSD の数が 10 倍になると、OSD あたりの PG の数は 10 になります。OSD ごとの PG の数が少ないと、容量が不均一に分散される傾向があるため、プールごとの PG を調整することを検討してください。

配置グループ数の調整は、オンラインで実行できます。再計算は PG 番号の再計算だけでなく、データの再配置が関係し、長いプロセスになります。ただし、データの可用性はいつでも維持されます。

OSD ごとの非常に高い PG は、失敗した OSD 上でのすべての PG を再構築すると同時に起動されるため、回避する必要があります。時間内に再構築を実行するには、多くの IOPS が必要ですが、これは利用できない可能性があります。これにより、I/O キューが深くなり、待ち時間が長くなり、ストレージクラスターが使用できなくなったり、修復時間が長くなったりします。

関連情報

- 特定のユースケースで値を算出する場合は PG calculator を参照してください。
- 詳細は、Red Hat Ceph Storage Strategies Guide のErasure Code Poolsの章を参照してください。

1.5. CEPH MANAGER バランサーモジュールの使用

バランサーは、Ceph Manager のモジュールで、OSD 全体の配置グループ (PG) の配置を最適化することで、自動または監視された方法でバランスの取れた分散を実現します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

バランサーの起動

1. balancer モジュールが有効になっていることを確認します。

```
[root@mon ~]# ceph mgr module enable balancer
```

2. balancer モジュールをオンにします。

```
[root@mon ~]# ceph balancer on
```

モード

現在、サポートされるバランサーモードが 2 つあります。

- **crush-compat**: CRUSH compat モードは、Ceph Luminous で導入された compat の **weight-set** 機能を使用して、CRUSH 階層のデバイスの別の重みセットを管理します。通常の重みは、デバイスに保存する目的のデータ量を反映するために、デバイスのサイズに設定したままにする必要があります。その後バランサーは、可能な限り目的のディストリビューションに一致す

るディストリビューションを達成するために、**weight-set** の値を少しずつ増減させ、値を最適化します。PG の配置は擬似ランダムプロセスであるため、配置には自然なばらつきが伴います。重みを最適化することで、バランサーはこの自然なばらつきに対応します。
このモードは、古いクライアントと完全に後方互換性があります。OSDMap および CRUSH マップが古いクライアントと共有されると、バランサーは最適化された重みを実際の重みとして提示します。

このモードの主な制限は、階層のサブツリーが OSD を共有する場合に、バランサーが配置ルールの異なる複数の CRUSH 階層を処理できないことです。この設定では、共有 OSD での領域の使用を管理するのが困難になるため、一般的には推奨されません。そのため、通常、この制限は問題にはなりません。

- **upmap**: Luminous 以降、OSDMap は、通常の CRUSH 配置計算への例外として、個々の OSD の明示的なマッピングを保存できます。これらの **upmap** エントリーにより、PG マッピングを細かく制御できます。この CRUSH モードは、バランスの取れた分散を実現するために、個々の PG の配置を最適化します。多くの場合、この分散は各 OSD の PG 数 +/-1 PG で完璧です。これは割り切れない場合があるためです。

重要

upmap を使用するには、すべてのクライアントが Red Hat Ceph Storage 3.x 以降および Red Hat Enterprise Linux 7.5 以降を実行している必要があります。

この機能を使用できるようにするには、以下のコマンドにより、luminous クライアントまたはそれ以降のクライアントしかサポートする必要がないことをクラスターに指示する必要があります。

```
[root@admin ~]# ceph osd set-require-min-compat-client luminous
```

このコマンドは、luminous 以前のクライアントまたはデーモンがモニターに接続されていると失敗します。

既知の問題により、カーネル CephFS クライアントは自身を jewel クライアントとして報告します。この問題を回避するには、**--yes-i-really-mean-it** フラグを使用します。

```
[root@admin ~]# ceph osd set-require-min-compat-client luminous --yes-i-really-mean-it
```

使用しているクライアントのバージョンは、以下で確認できます。

```
[root@admin ~]# ceph features
```

警告

Red Hat Ceph Storage 3.x では、upmap 機能は、クラスターが使用される際の PG のバランスのために Ceph Manager バランサーモジュールによって使用される場合にのみサポートされます。Red Hat Ceph Storage 3.x では、upmap 機能を使用した PG の手動リバランスはサポートされません。

デフォルトのモードは **crush-compat** です。モードは以下のように変更できます。

```
[root@mon ~]# ceph balancer mode upmap
```

または

```
[root@mon ~]# ceph balancer mode crush-compat
```

ステータス

バランサーの現在のステータスは、以下を実行していつでも確認できます。

```
[root@mon ~]# ceph balancer status
```

自動バランシング

デフォルトでは、バランサーモジュールをオンにする場合、自動分散が使用されます。

```
[root@mon ~]# ceph balancer on
```

以下を使用して、バランサーを再度オフにできます。

```
[root@mon ~]# ceph balancer off
```

これには、古いクライアントと後方互換性があり、時間の経過とともにデータディストリビューションに小さな変更を加えて、OSD を同等に利用されるようにする **crush-compat** モードを使用します。

スロットリング

たとえば、OSD が失敗し、システムがまだ修復していない場合などに、クラスターのパフォーマンスが低下する場合は、PG ディストリビューションには調整は行われません。

クラスターが正常な場合、バランサーは変更を調整して、置き間違えた、または移動する必要のある PG の割合がデフォルトで 5% のしきい値を下回るようにします。このパーセンテージは、**max_misplaced** 設定を使用して調整できます。たとえば、しきい値を 7% に増やすには、次のコマンドを実行します。

```
[root@mon ~]# ceph config-key set mgr/balancer/max_misplaced .07
```

監視付き最適化

balancer 操作はいくつかの異なるフェーズに分類されます。

1. **プラン** の構築
2. 現在の PG 分散または **プラン** 実行後に得られる PG 分散に対するデータ分散の品質の評価
3. **プラン** の実行
   - 現在のディストリビューションを評価し、スコアを付けます。

     ```
     [root@mon ~]# ceph balancer eval
     ```

   - 単一プールのディストリビューションを評価するには、以下を実行します。

```
[root@mon ~]# ceph balancer eval <pool-name>
```

- 評価の詳細を表示するには、以下を実行します。

```
[root@mon ~]# ceph balancer eval-verbose ...
```

- 現在設定されているモードを使用してプランを生成するには、以下を実行します。

```
[root@mon ~]# ceph balancer optimize <plan-name>
```

<plan-name> は、カスタムプラン名に置き換えます。

- プランの内容を表示するには、以下を実行します。

```
[root@mon ~]# ceph balancer show <plan-name>
```

- 古いプランを破棄するには、以下を実行します。

```
[root@mon ~]# ceph balancer rm <plan-name>
```

- 現在記録されているプランを表示するには、status コマンドを使用します。

```
[root@mon ~]# ceph balancer status
```

- プラン実行後に生じるディストリビューションの品質を計算するには、以下を実行します。

```
[root@mon ~]# ceph balancer eval <plan-name>
```

- プランを実行するには、以下を実行します。

```
[root@mon ~]# ceph balancer execute <plan-name>
```

[注記]:ディストリビューションの改善が想定される場合にのみ、プランを実行します。実行後、プランは破棄されます。

1.6. 関連情報

- 詳細は、Red Hat Ceph Storage Strategies Guide のPlacement Groups (PGs)の章を参照してください。

第2章 ディスク障害の処理

ストレージ管理者は、ストレージクラスターのライフサイクル時に、特定の時点でディスク障害に対応する必要があります。実際の障害が発生する前にディスク障害をテストおよびシミュレーションすることで、実際に発生したときに備えて準備が整います。

障害の発生したディスクを置き換えるための高度なワークフローを以下に示します。

1. 障害のある OSD を検索します。
2. OSD を取得します。
3. ノード上の OSD デーモンを停止します。
4. Ceph のステータスを確認します。
5. CRUSH マップから OSD を削除します。
6. OSD 認証を削除します。
7. ストレージクラスターから OSD を削除します。
8. ノードのファイルシステムのマウントを解除します。
9. 障害が発生したドライブを置き換えます。
10. OSD をストレージクラスターに追加します。
11. Ceph のステータスを確認します。

2.1. 前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- 障害の発生したディスク。

2.2. ディスクの失敗

Ceph は耐障害性を確保できるように設計されているため、Ceph はデータを損失せずに動作が **degraded** の状態になっています。Ceph は、データストレージドライブに障害が発生してもも引き続き動作します。**degraded** 状態は、他の OSD に保存されるデータの追加コピーがストレージクラスター内の他の OSD に自動的にバックフィルされることを意味します。OSD が **down** としてマークされる場合は、ドライブに障害が発生したことを意味します。

ドライブに障害が発生すると、最初に OSD ステータスは **down** になりますが、ストレージクラスター内 (**in**) に残ります。ネットワークの問題は、実際に起動 (**up**) していても OSD をダウン (**down**) としてマークすることもできます。まず、環境内のネットワークの問題を確認します。ネットワークが問題がないことを確認する場合は、OSD ドライブが失敗した可能性があります。

最新のサーバーは通常、ホットスワップ可能なドライブでデプロイして、障害が発生したドライブをプルし、ノードを停止せずに新しいドライブに置き換えます。ただし、Ceph では、OSD のソフトウェア定義部分を削除する必要もあります。

2.2.1. 障害のある OSD ディスクの置き換え

OSD を置き換える一般的な手順には、OSD をストレージクラスターから削除し、ドライブを置き換えてから OSD を再作成する必要があります。

重要

BlueStore OSD のデータベースパーティションが含まれる BlueStore の **block.db** ディスクを置き換える場合、Red Hat は Ansible を使用したすべての OSD の再デプロイのみをサポートします。破損した **block.db** ファイルは、その **block.db** ファイルに含まれるすべての OSD に影響を与えます。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- 障害の発生したディスク。

手順

1. ストレージクラスターの正常性を確認します。

```
# ceph health
```

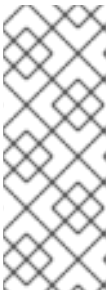
2. CRUSH 階層で OSD の場所を特定します。

```
# ceph osd tree | grep -i down
```

3. OSD ノードで、OSD の起動を試行します。

```
# systemctl start ceph-osd@$OSD_ID
```

コマンドが OSD がすでに実行されていることを示す場合、ハートビートまたはネットワークの問題がある可能性があります。OSD を再起動できない場合は、ドライブが失敗する可能性があります。



注記

OSD が **down** すると、OSD は最終的に **out** とマークされます。Ceph Storage では、これは通常の動作です。OSD が **out** とマークすると、失敗した OSD のデータのコピーが含まれる他の OSD がバックフィルを開始し、必要な数のコピーがストレージクラスター内に存在していることを確認します。ストレージクラスターがバックフィル状態である間、クラスターの状態は **degraded** になります。

4. Ceph のコンテナー化されたデプロイメントでは、OSD に関連付けられたドライブを参照し、OSD コンテナーを起動します。

```
# systemctl start ceph-osd@$OSD_DRIVE
```

コマンドが OSD がすでに実行されていることを示す場合、ハートビートまたはネットワークの問題がある可能性があります。OSD を再起動できない場合は、ドライブが失敗する可能性があります。

注記

OSD に関連付けられたドライブは、コンテナー OSD ID をドライブにマッピングして判断できます。

5. 失敗した OSD のマウントポイントを確認します。

注記

Ceph のコンテナー化されたデプロイメントでは、OSD がダウンし、OSD ドライブのマウントが解除されるため、**df** を実行してマウントポイントを確認することはできません。別の方法を使用して、OSD ドライブが失敗したかどうかを判別します。たとえば、コンテナーノードからドライブで **smartctl** を実行します。

```
# df -h
```

OSD を再起動できない場合は、マウントポイントを確認できます。マウントポイントが表示されない場合は、OSD ドライブを再マウントして OSD を再起動することができます。マウントポイントを復元できない場合は、OSD ドライブが失敗している可能性があります。

smartctl ユーティリティー cab を使用して、ドライブが正常かどうかを確認します。以下に例を示します。

```
# yum install smartmontools
# smartctl -H /dev/$DRIVE
```

ドライブに障害が発生した場合は、それを置き換えなければならない場合があります。

6. OSD プロセスを停止します。

```
# systemctl stop ceph-osd@$OSD_ID
```

a. **FileStore** を使用している場合は、ジャーナルをディスクにフラッシュします。

```
# ceph osd -i $$OSD_ID --flush-journal
```

7. Ceph のコンテナー化されたデプロイメントの場合には、OSD に関連付けられたドライブを参照して、OSD コンテナーを停止します。

```
# systemctl stop ceph-osd@$OSD_DRIVE
```

8. ストレージクラスターから OSD を削除します。

```
# ceph osd out $OSD_ID
```

9. 失敗した OSD がバックフィルされていることを確認します。

```
# ceph -w
```

10. CRUSH マップから OSD を削除します。

```
# ceph osd crush remove osd.$OSD_ID
```

注記

この手順は、OSD を永続的に削除し、再デプロイしない場合にのみ必要になります。

11. OSD の認証キーを削除します。

```
# ceph auth del osd.$OSD_ID
```

12. OSD のキーが一覧表示されていないことを確認します。

```
# ceph auth list
```

13. ストレージクラスターから OSD を削除します。

```
# ceph osd rm osd.$OSD_ID
```

14. 障害が発生したドライブパスのマウントを解除します。

注記

Ceph のコンテナー化されたデプロイメントでは、OSD がコンテナーがダウンし、OSD ドライブのマウントが解除されます。この場合、マウント解除するものがないため、この手順はスキップできます。

```
# umount /var/lib/ceph/osd/$CLUSTER_NAME-$OSD_ID
```

15. 物理ドライブを置き換えます。ノードのハードウェアベンダーのドキュメントを参照してください。ドライブのホットスワップが可能である場合は、障害が発生したドライブを新しいドライブに置き換えます。ドライブがホットスワップできず、ノードに複数の OSD が含まれている場合は、物理ドライブを交換するためにノードを停止する必要がある場合があります。ノードを一時的に停止する必要がある場合は、バックフィルを防ぐためにクラスターを **noout** に設定できます。

```
# ceph osd set noout
```

ドライブを置き換えて、ノードとその OSD をオンラインに戻したら、**noout** 設定を削除します。

```
# ceph osd unset noout
```

新しいドライブを **/dev/** ディレクトリーの下に表示されるように、ドライブパスを書き留めて作業を続行します。

16. OSD ドライブを特定し、ディスクをフォーマットします。

17. OSD を再作成します。

    a. Ansible の使用

b. コマンドラインインターフェイスの使用

18. CRUSH 階層をチェックして、これが正確であることを確認します。

```
# ceph osd tree
```

CRUSH 階層の OSD の場所が適切でない場合は、**move** コマンドを使用して移動できます。

```
# ceph osd crush move $BUCKET_TO_MOVE $BUCKET_TYPE=$PARENT_BUCKET
```

19. OSD がオンラインであることを確認します。

2.2.2. OSD ID の保持中に OSD ドライブの置き換え

障害のある OSD ドライブを置き換える場合は、元の OSD ID および CRUSH マップエントリーを保持できます。

注記

ceph-volume lvm コマンドのデフォルトは、OSD 用の BlueStore です。FileStore OSD を使用するには、**--filestore**、**--data**、および **--journal** オプションを使用します。

詳細は、OSD データおよびジャーナルドライブの準備 セクションを参照してください。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- 障害の発生したディスク。

手順

1. OSD を破棄します。

```
ceph osd destroy $OSD_ID --yes-i-really-mean-it
```

例

```
$ ceph osd destroy 1 --yes-i-really-mean-it
```

2. 必要に応じて、交換ディスクを以前使用していた場合は、ディスクを **ザッピングする** 必要があります。

```
ceph-volume lvm zap $DEVICE
```

例

```
$ ceph-volume lvm zap /dev/sdb
```

3. 既存の OSD ID で新規 OSD を作成します。

```
ceph-volume lvm create --osd-id $OSD_ID --data $DEVICE
```

例

```
$ ceph-volume lvm create --osd-id 1 --data /dev/sdb
```

2.3. ディスク障害のシミュレーション

ハードとソフト の 2 つのディスク障害シナリオがあります。ハードな障害が発生すると、ディスクが置き換えられます。ソフト障害は、デバイスドライバーまたはその他のソフトウェアコンポーネントに問題がある可能性があります。

ソフトエラーが発生した場合、ディスクの置き換えは不要になる可能性があります。ディスクを置き換える場合、ステップの後に障害の発生したディスクを削除し、交換ディスクを Ceph に追加する必要があります。ソフトディスク障害をシミュレートするために、デバイスを削除するのが最適です。デバイスを選択し、システムからデバイスを削除します。

```
echo 1 > /sys/block/$DEVICE/device/delete
```

例

```
[root@ceph1 ~]# echo 1 > /sys/block/sdb/device/delete
```

Ceph OSD のログでは、OSD ノードで Ceph が障害を検出し、復縁プロセスを自動的に開始します。

例

```
[root@ceph1 ~]# tail -50 /var/log/ceph/ceph-osd.1.log
2017-02-02 12:15:27.490889 7f3e1fa3d800 -1 ^[[0;31m ** ERROR: unable to open OSD superblock
on /var/lib/ceph/osd/ceph-1: (5) Input/output error^[[0m
2017-02-02 12:34:17.777898 7fb7df1e7800  0 set uid:gid to 167:167 (ceph:ceph)
2017-02-02 12:34:17.777933 7fb7df1e7800  0 ceph version 10.2.3-17.el7cp
(ca9d57c0b140eb5cea9de7f7133260271e57490e), process ceph-osd, pid 1752
2017-02-02 12:34:17.788885 7fb7df1e7800  0 pidfile_write: ignore empty --pid-file
2017-02-02 12:34:17.870322 7fb7df1e7800  0 filestore(/var/lib/ceph/osd/ceph-1) backend xfs (magic
0x58465342)
2017-02-02 12:34:17.871028 7fb7df1e7800  0 genericfilestorebackend(/var/lib/ceph/osd/ceph-1)
detect_features: FIEMAP ioctl is disabled via 'filestore fiemap' config option
2017-02-02 12:34:17.871035 7fb7df1e7800  0 genericfilestorebackend(/var/lib/ceph/osd/ceph-1)
detect_features: SEEK_DATA/SEEK_HOLE is disabled via 'filestore seek data hole' config option
2017-02-02 12:34:17.871059 7fb7df1e7800  0 genericfilestorebackend(/var/lib/ceph/osd/ceph-1)
detect_features: splice is supported
2017-02-02 12:34:17.897839 7fb7df1e7800  0 genericfilestorebackend(/var/lib/ceph/osd/ceph-1)
detect_features: syncfs(2) syscall fully supported (by glibc and kernel)
2017-02-02 12:34:17.897985 7fb7df1e7800  0 xfsfilestorebackend(/var/lib/ceph/osd/ceph-1)
detect_feature: extsize is disabled by conf
2017-02-02 12:34:17.921162 7fb7df1e7800  1 leveldb: Recovering log #22
2017-02-02 12:34:17.947335 7fb7df1e7800  1 leveldb: Level-0 table #24: started
2017-02-02 12:34:18.001952 7fb7df1e7800  1 leveldb: Level-0 table #24: 810464 bytes OK
2017-02-02 12:34:18.044554 7fb7df1e7800  1 leveldb: Delete type=0 #22
2017-02-02 12:34:18.045383 7fb7df1e7800  1 leveldb: Delete type=3 #20
2017-02-02 12:34:18.058061 7fb7df1e7800  0 filestore(/var/lib/ceph/osd/ceph-1) mount: enabling
WRITEAHEAD journal mode: checkpoint is not enabled
2017-02-02 12:34:18.105482 7fb7df1e7800  1 journal _open /var/lib/ceph/osd/ceph-1/journal fd 18:
1073741824 bytes, block size 4096 bytes, directio = 1, aio = 1
```

```
2017-02-02 12:34:18.130293 7fb7df1e7800  1 journal _open /var/lib/ceph/osd/ceph-1/journal fd 18:
1073741824 bytes, block size 4096 bytes, directio = 1, aio = 1
2017-02-02 12:34:18.130992 7fb7df1e7800  1 filestore(/var/lib/ceph/osd/ceph-1) upgrade
2017-02-02 12:34:18.136547 7fb7df1e7800  0 <cls> cls/cephfs/cls_cephfs.cc:202: loading
cephfs_size_scan
2017-02-02 12:34:18.142863 7fb7df1e7800  0 <cls> cls/hello/cls_hello.cc:305: loading cls_hello
2017-02-02 12:34:18.255019 7fb7df1e7800  0 osd.1 51 crush map has features 2200130813952,
adjusting msgr requires for clients
2017-02-02 12:34:18.255041 7fb7df1e7800  0 osd.1 51 crush map has features 2200130813952 was
8705, adjusting msgr requires for mons
2017-02-02 12:34:18.255048 7fb7df1e7800  0 osd.1 51 crush map has features 2200130813952,
adjusting msgr requires for osds
2017-02-02 12:34:18.296256 7fb7df1e7800  0 osd.1 51 load_pgs
2017-02-02 12:34:18.561604 7fb7df1e7800  0 osd.1 51 load_pgs opened 152 pgs
2017-02-02 12:34:18.561648 7fb7df1e7800  0 osd.1 51 using 0 op queue with priority op cut off at 64.
2017-02-02 12:34:18.562603 7fb7df1e7800 -1 osd.1 51 log_to_monitors {default=true}
2017-02-02 12:34:18.650204 7fb7df1e7800  0 osd.1 51 done with init, starting boot process
2017-02-02 12:34:19.274937 7fb7b78ba700  0 -- 192.168.122.83:6801/1752 >>
192.168.122.81:6801/2620 pipe(0x7fb7ec4d1400 sd=127 :6801 s=0 pgs=0 cs=0 l=0
c=0x7fb7ec42e480).accept connect_seq 0 vs existing 0 state connecting
```

osd ディスクツリーを見ると、ディスクがオフラインであると認識されます。

```
[root@ceph1 ~]# ceph osd tree
ID WEIGHT  TYPE NAME      UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 0.28976 root default
-2 0.09659     host ceph3
 1 0.09659         osd.1       down 1.00000          1.00000
-3 0.09659     host ceph1
 2 0.09659         osd.2       up  1.00000          1.00000
-4 0.09659     host ceph2
 0 0.09659         osd.0       up  1.00000          1.00000
```

第3章 ノードの障害の処理

ストレージクラスター内でノード全体に障害が発生する可能性があります。ストレージ管理者が行うノード障害の処理は、ディスク障害の処理と同様です。ノードの障害として Ceph が 1 つのディスクに対してのみ PG(配置グループ) を復元する代わりに、そのノード内のディスクのすべての PG を復元する必要があります。Ceph は OSD がすべてダウンしていることを検出し、自己修復として知られる復元プロセスを自動的に開始します。

ノードの障害シナリオは 3 つあります。ノードを置き換える際の各シナリオにおけるハイレベルのワークフローを以下に示します。

- ノードの置き換えには、失敗したノードから root ディスクおよび Ceph OSD ディスクを使用します。
  1. バックフィルを無効にします。
  2. ノードを置き換え、古いノードからディスクを取得し、それらを新規ノードに追加します。
  3. バックフィルを有効にします。
- ノードを置き換え、オペレーティングシステムを再インストールし、障害が発生したノードから Ceph OSD ディスクを使用します。
  1. バックフィルを無効にします。
  2. Ceph 設定のバックアップを作成します。
  3. ノードを置き換え、障害が発生したノードから Ceph OSD ディスクを追加します。
     a. ディスクを JBOD として設定
  4. オペレーティングシステムをインストールします。
  5. Ceph の設定を復元します。
  6. **ceph-ansible** を実行します。
  7. バックフィルを有効にします。
- ノードを置き換え、オペレーティングシステムを再インストールし、すべての新規 Ceph OSD ディスクを使用します。
  1. バックフィルを無効にします。
  2. 障害のあるノードのすべての OSD をストレージクラスターから削除します。
  3. Ceph 設定のバックアップを作成します。
  4. ノードを置き換え、障害が発生したノードから Ceph OSD ディスクを追加します。
     a. ディスクを JBOD として設定
  5. オペレーティングシステムをインストールします。
  6. **ceph-ansible** を実行します。
  7. バックフィルを有効にします。

3.1. 前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- 障害のあるノード。

3.2. ノードの追加または削除前の考慮事項

Ceph の未処理の機能の 1 つは、ランタイム時に Ceph OSD ノードを追加または削除できる機能です。つまり、ストレージクラスターの容量のサイズを変更したり、ストレージクラスターを縮小せずにハードウェアを置き換えることができることを意味します。クラスターの状態が劣化 (**degraded**) している間に Ceph クライアントを提供する機能にも運用上の利点があります。たとえば、残業したり週末に作業したりするのではなく、通常の営業時間内にハードウェアを追加、削除、または交換できます。ただし、Ceph OSD ノードの追加および削除により、パフォーマンスに大きな影響を与える可能性があり、実行する前にストレージクラスターのハードウェアを追加、削除、または交換することによるパフォーマンスへの影響を考慮する必要があります。

容量の観点からは、ノードを削除するとノードに含まれる OSD が削除され、実質的にストレージクラスターの容量が低下します。ノードを追加するとノードに含まれる OSD が追加され、実質的にストレージクラスターの容量が拡張されます。ストレージクラスターの容量を拡張または縮小しても、Ceph OSD ノードを追加または削除すると、クラスターがリバランスする際にバックフィルが実行されます。このリバランス期間中に、Ceph は追加のリソースを使用します。これにより、ストレージクラスターのパフォーマンスに影響が出る可能性があります。

各ノードに OSD が 4 つある Ceph ノードが含まれるストレージクラスターを想像してみてください。16 の OSD を持つ、4 つのノードのストレージクラスターでは、ノードを削除すると 4 つの OSD が削除され、容量が 25% 削減されます。12 の OSD を持つ、3 つのノードのストレージクラスターでは、ノードを追加すると 4 つの OSD が追加され、容量が 33% 増加します。

実稼働用 Ceph Storage クラスターでは、Ceph OSD ノードに特定のタイプのストレージストラテジーを容易にする特定のハードウェア設定があります。詳細は、Red Hat Ceph Storage 3 のStorage Strategies Guide を参照してください。

Ceph OSD ノードは CRUSH 階層の一部であるため、ノードの追加や削除によるパフォーマンスへの影響は、通常その CRUSH 階層を使用するプール (つまり CRUSH ルールセット) のパフォーマンスに影響します。

3.3. パフォーマンスに関する考慮事項

以下の要素は通常、Ceph OSD ノードの追加時または削除時に、ストレージクラスターのパフォーマンスに影響します。

影響を受けるプールの現在のクライアント負荷:

Ceph クライアントは、Ceph への I/O インターフェイスに負荷を配置します。つまり、プールへの負荷。プールは CRUSH ルールセットにマップします。基礎となる CRUSH 階層により、Ceph は障害ドメインにデータを配置できます。基礎となる Ceph OSD ノードにクライアント負荷が高いプールが含まれる場合は、クライアントの負荷が復元時間やパフォーマンスに大きな影響を与える可能性があります。具体的には、書き込み操作には持続性のためにデータのレプリケーションが必要になるため、書き込み集約型クライアント負荷により、ストレージクラスターの復元に要する時間が長くなる可能性があります。

追加または削除される容量:

一般的に、追加または削除する容量のクラスター全体のパーセンテージが、ストレージクラスターの復元時間に影響を及ぼします。さらに、追加または削除するノードのストレージ密度は、復元する時間に

影響を与える可能性があります。たとえば、36 OSD のノードは、12 OSD のノードと比較して、通常復元により長い時間がかかります。ノードを削除する際には、十分な容量を確保して、**フル比率** または **ほぼフル比率** に到達しないようにします。ストレージクラスターが **フル比率** になると、Ceph は書き込み動作を一時停止してデータの喪失を防ぎます。

プールと CRUSH ルールセット:

Ceph OSD ノードは、少なくとも 1 つの Ceph CRUSH 階層にマッピングし、階層は少なくとも 1 つのプールにマップされます。Ceph OSD ノードの追加または削除を行う CRUSH 階層 (ルールセット) を使用する各プールは、パフォーマンスに影響します。

プールタイプおよび持続性:

レプリケーションプールは、データのディープコピーを複製するネットワーク帯域幅を使用する傾向がありますが、イレイジャーコーディングプールはより多くの CPU を使用して **k+m** コーディングのチャンクを計算する傾向があります。データのコピー (サイズやより多くの **k+m** チャンクなど) が多いほど、ストレージクラスターが復元するまでにかかる時間が長くなります。

合計スループット特性:

ドライブ、コントローラー、およびネットワークインターフェイスカードはすべて、復元時間に影響を与える可能性があるスループット特性を持ちます。一般に、10 Gbps や SSD などのスループット特性が高いノードは、1 Gbps や SATA ドライブなどのスループット特性が低いノードよりも迅速に復元します。

3.4. ノードの追加または削除に関する推奨事項

ノードの障害により、ノードを変更する前に一度に 1 つの OSD を削除するのが妨げられる場合があります。状況によっては、Ceph OSD ノードを追加または削除する際のパフォーマンスへの悪影響を軽減することができます。Red Hat は、ノード内で一度に 1 つの OSD を追加または削除し、次の OSD に進む前にクラスターが復元できるようにすることを推奨します。OSD の削除に関する詳細は、以下を参照してください。

- Ansible の使用
- コマンドラインインターフェイスの使用

Ceph ノードを追加する場合も、Red Hat では OSD を一度に 1 つずつ追加することを推奨します。OSD の追加に関する詳細は、以下を参照してください。

- Ansible の使用
- コマンドラインインターフェイスの使用

Ceph OSD ノードを追加または削除する場合は、実行中のその他のプロセスがパフォーマンスに影響することを考慮してください。クライアント I/O への影響を減らすために、Red Hat では以下を推奨します。

容量を計算する:

Ceph OSD ノードを削除する前に、ストレージクラスターが**フル比率**に達する**ことなく**すべての OSD のコンテンツをバックフィルできることを確認してください。**フル比率** に達すると、クラスターは書き込み操作を拒否するようになります。

一時的にスクラビングを無効にする:

スクラビングはストレージクラスターのデータの持続性を確保するために不可欠ですが、リソース集約型です。Ceph OSD ノードを追加または削除する前に、スクラビングおよびディープスクラビングを無効にし、先に進む前に現在のスクラビング操作を完了させます。以下に例を示します。

```
ceph osd set noscrub
ceph osd set nodeep-scrub
```

Ceph OSD ノードを追加または削除すると、ストレージクラスターが **active+clean** 状態に戻り、**noscrub** および **nodeep-scrub** の設定を解除します。

バックフィルおよび復元を制限する:

osd pool default size = 3 や **osd pool default min size = 2** などの妥当なデータの持続性がある場合は、パフォーマンスが劣化 (**degraded**) した状態での動作に問題はありません。可能な限り早い復元時間用にストレージクラスターを調整することができますが、これにより Ceph クライアントの I/O パフォーマンスが著しく影響を受ける可能性があります。最大の Ceph クライアント I/O パフォーマンスを維持するには、バックフィルと復元の操作を制限し、その操作に長い時間をかけられるようにします。以下に例を示します。

```
osd_max_backfills = 1
osd_recovery_max_active = 1
osd_recovery_op_priority = 1
```

osd_recovery_sleep などの sleep パラメーターおよび delay パラメーターを設定することもできます。

最後に、ストレージクラスターのサイズを拡張する場合は、配置グループの数を増やすことが必要となる場合があります。配置グループの数を拡張する必要がある場合、Red Hat はプレースメントグループの数を段階的に増やすことをお勧めします。配置グループの数を大幅に増やすと、パフォーマンスが大幅に低下します。

3.5. CEPH OSD ノードの追加

Red Hat Ceph Storage クラスターの容量を拡張するには、OSD ノードを追加します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- ネットワーク接続が割り当てられたプロビジョニングされたノード
- Red Hat Enterprise Linux 7 または Ubuntu 16.04 のインストール
- Installation Guide for Red Hat Enterprise Linux または Ubuntu の **Requirements for Installing Red Hat Ceph Storage**の章を参照してください。

手順

1. ストレージクラスターの他のノードが、短縮ホスト名で新規ノードに到達できることを確認します。
2. スクラビングを一時的に無効にします。

```
[root@monitor ~]# ceph osd set noscrub
[root@monitor ~]# ceph osd set nodeep-scrub
```

3. バックフィルおよび復元機能を制限します。

構文

```
ceph tell $DAEMON_TYPE.* injectargs --$OPTION_NAME $VALUE [--$OPTION_NAME $VALUE]
```

例

```
[root@monitor ~]# ceph tell osd.* injectargs --osd-max-backfills 1 --osd-recovery-max-active 1 --osd-recovery-op-priority 1
```

4. 新規ノードを CRUSH マップに追加します。

構文

```
ceph osd crush add-bucket $BUCKET_NAME $BUCKET_TYPE
```

例

```
[root@monitor ~]# ceph osd crush add-bucket node2 host
```

5. ノードの各ディスクの OSD をストレージクラスターに追加します。

- Ansible の使用
- コマンドラインインターフェイスの使用

重要

OSD ノードを Red Hat Ceph Storage クラスターに追加する場合、Red Hat では、ノード内の一度に 1 つの OSD を追加して、次の OSD に進む前に、クラスターが **active+clean** 状態に回復できるようにすることを推奨します。

関連情報

- 詳細は、Red Hat Ceph Storage Configuration Guide のSetting a Specific Configuration Setting at Runtimeセクションを参照してください。
- CRUSH 階層の適切な場所にノードを配置するための詳細は、Red Hat Ceph Storage Storage Strategies Guide のAdd a BucketおよびMove a Bucketセクションを参照してください。

3.6. CEPH OSD ノードの削除

ストレージクラスターの容量を減らすには、OSD ノードを削除します。

警告

Ceph OSD ノードを削除する前に、ストレージクラスターがフル比率に達することなくすべての OSD のコンテンツをバックフィルできることを確認してください。フル比率に達すると、クラスターは書き込み操作を拒否するようになります。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター。

手順

1. ストレージクラスターの容量を確認します。

```
[root@monitor ~]# ceph df
[root@monitor ~]# rados df
[root@monitor ~]# ceph osd df
```

2. スクラビングを一時的に無効にします。

```
[root@monitor ~]# ceph osd set noscrub
[root@monitor ~]# ceph osd set nodeep-scrub
```

3. バックフィルおよび復元機能を制限します。

構文

```
ceph tell $DAEMON_TYPE.* injectargs --$OPTION_NAME $VALUE [--$OPTION_NAME $VALUE]
```

例

```
[root@monitor ~]# ceph tell osd.* injectargs --osd-max-backfills 1 --osd-recovery-max-active 1 --osd-recovery-op-priority 1
```

4. ノード上の各 OSD をストレージクラスターから削除します。

- Ansible の使用
- コマンドラインインターフェイスの使用

重要

ストレージクラスターから OSD ノードを削除する場合、Red Hat は、ノード内の一度に 1 つの OSD を削除してから、次の OSD に進む前にクラスターが **active+clean** 状態に回復できるようにすることを推奨します。

a. OSD を削除したら、ストレージクラスターがほぼフル比率に達していないことを確認します。

```
[root@monitor ~]# ceph -s
[root@monitor ~]# ceph df
```

b. ノードのすべての OSD がストレージクラスターから削除されるまでこの手順を繰り返します。

5. すべての OSD が削除されると、CRUSH マップからホストバケットを削除します。

構文

```
ceph osd crush rm $BUCKET_NAME
```

例

```
[root@monitor ~]# ceph osd crush rm node2
```

関連情報

- * 詳細は、Red Hat Ceph Storage Configuration Guide のSetting a Specific Configuration Setting at Runtimeセクションを参照してください。

3.7. ノードの障害のシミュレーション

ハードノードの障害をシミュレーションするには、ノードの電源をオフにし、オペレーティングシステムを再インストールします。

前提条件

- 正常かつ実行中の Red Hat Ceph Storage クラスター

手順

1. ストレージ容量をチェックして、ストレージクラスターに対するノード削除の影響について確認します。

```
# ceph df
# rados df
# ceph osd df
```

2. 必要に応じて、復元およびバックフィルを無効にします。

```
# ceph osd set noout
# ceph osd set noscrub
# ceph osd set nodeep-scrub
```

3. ノードをシャットダウンします。

4. ホスト名が変更された場合は、CRUSH マップからノードを削除します。

```
[root@ceph1 ~]# ceph osd crush rm ceph3
```

5. クラスターのステータスを確認します。

```
[root@ceph1 ~]# ceph -s
```

6. ノードにオペレーティングシステムを再インストールします。

7. Ansible ユーザーおよび SSH キーを追加します。

```
[root@ceph3 ~]# useradd ansible
[root@ceph3 ~]# passwd ansible
[root@ceph3 ~]# cat << EOF > /etc/sudoers.d/ansible
ansible ALL = (root) NOPASSWD:ALL
Defaults:ansible !requiretty
EOF
[root@ceph3 ~]# su - ansible
[ansible@ceph3 ~]# ssh-keygen
```

8. 管理ノードから、**ansible** ユーザーの SSH キーをコピーします。

```
[ansible@admin ~]$ ssh-copy-id ceph3
```

9. 管理ノードから、Ansible Playbook を再実行します。

```
[ansible@admin ~]$ cd /usr/share/ceph-ansible
[ansible@admin ~]$ ansible-playbook site.yml
```

出力例

```
PLAY RECAP ********************************************************************
ceph1                      : ok=368  changed=2    unreachable=0    failed=0
ceph2                      : ok=284  changed=0    unreachable=0    failed=0
ceph3                      : ok=284  changed=15   unreachable=0    failed=0
```

10. 必要に応じて、復元およびバックフィルを有効にします。

```
[root@ceph3 ~]# ceph osd unset noout
[root@ceph3 ~]# ceph osd unset noscrub
[root@ceph3 ~]# ceph osd unset nodeep-scrub
```

11. Ceph のヘルスを確認します。

```
[root@ceph3 ~]# ceph -s
    cluster 1e0c9c34-901d-4b46-8001-0d1f93ca5f4d
     health HEALTH_OK
     monmap e1: 3 mons at
{ceph1=192.168.122.81:6789/0,ceph2=192.168.122.82:6789/0,ceph3=192.168.122.83:6789/0}

            election epoch 36, quorum 0,1,2 ceph1,ceph2,ceph3
     osdmap e95: 3 osds: 3 up, 3 in
            flags sortbitwise
      pgmap v1190: 152 pgs, 12 pools, 1024 MB data, 441 objects
            3197 MB used, 293 GB / 296 GB avail
                 152 active+clean
```

関連情報

- Red Hat Ceph Storage のインストールについての詳しい情報は、以下のドキュメントを参照してください。
  - Red Hat Enterprise Linux
  - Ubuntu

第4章 データセンター障害の処理

Red Hat Ceph Storage は、ストレッチクラスターで 3 つのデータセンターのいずれかを失うなど、インフラストラクチャーに非常に致命的な障害がある場合があります。標準のオブジェクトストアのユースケースでは、3 つのデータセンターすべての設定は、それらの間にレプリケーションを設定して個別に実行できます。このシナリオでは、各データセンターのクラスター設定は異なり、ローカルの機能と依存関係を反映する可能性があります。

配置階層の論理構造を考慮する必要があります。適切な CRUSH マップは使用でき、インフラストラクチャー内の障害ドメインの階層構造が反映されます。論理階層定義を使用すると、標準の階層定義を使用することではなく、ストレージクラスターの信頼性が向上します。障害ドメインは CRUSH マップで定義されます。デフォルトの CRUSH マップには、フラットな階層のすべてのノードが含まれます。

ストレッチクラスターの 3 つのデータセンター環境の例では、ノードの配置は、1 つのデータセンターが停止できるように管理する必要がありますが、ストレージクラスターは稼働したままです。データに対して 3 方向レプリケーションを使用する場合に、ノードのある障害ドメインを考慮してください。1 つのデータセンターの障害の場合、データの一部を 1 つのコピーで残すことができる場合があります。このシナリオでは、2 つのオプションがあります。

- 標準設定で、データは読み取り専用ステータスのままにします。
- ライブは、停止期間に 1 つのコピーのみを行います。

標準設定では、ノード間でのデータ配置のランダム性のため、すべてのデータが影響を受けるわけではありませんが、一部のデータは 1 つのコピーしか持つことができず、ストレージクラスターは読み取り専用モードに戻ります。

以下の例では、作成されるマップは 6 つの OSD ノードを持つクラスターの初期設定から派生しています。この例では、すべてのノードが 1 つのディスクを持つため、1 つの OSD しかありません。すべてのノードは、階層ツリーの標準 **root** であるデフォルトの **root** 下に分類されます。2 つの OSD に重みが割り当てられているため、これらの OSD は他の OSD よりも少ないデータチャンクを受け取ります。これらのノードは、最初の OSD ディスクよりも大きなディスクを持つ後から導入されました。これは、ノードのグループが失敗しているデータ配置には影響しません。

標準の CRUSH マップ

```
$ sudo ceph osd tree
ID WEIGHT  TYPE NAME          UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 0.33554 root default
-2 0.04779     host ceph-node3
 0 0.04779         osd.0           up  1.00000          1.00000
-3 0.04779     host ceph-node2
 1 0.04779         osd.1           up  1.00000          1.00000
-4 0.04779     host ceph-node1
 2 0.04779         osd.2           up  1.00000          1.00000
-5 0.04779     host ceph-node4
 3 0.04779         osd.3           up  1.00000          1.00000
-6 0.07219     host ceph-node6
 4 0.07219         osd.4           up  0.79999          1.00000
-7 0.07219     host ceph-node5
 5 0.07219         osd.5           up  0.79999          1.00000
```

論理階層定義を使用してノードを同じデータセンターにグループ化すると、データの配置の成熟度を実行できます。**root**、**datacenter**、**rack**、**row**、および **host** の定義タイプにより、3 つのデータセンターのストッククラスターで障害ドメインを反映させることができます。

- ノードの ceph-node1 および ceph-node2 がデータセンター 1 (DC1) にある。
- ノードの ceph-node3 および ceph-node5 がデータセンター 2 (DC2) にある。
- ノードの ceph-node4 および ceph-node6 はデータセンター 3 (DC3) にある。
- すべてのデータセンターが同じ構造に属する (全 DC)

ホストのすべての OSD はホスト定義に属しているため、変更は必要ありません。その他のすべての割り当ては、ストレージクラスターの実行時に以下によって調整できます。

- 以下のコマンドで **バケット** 構造を定義します。

```
ceph osd crush add-bucket allDC root
ceph osd crush add-bucket DC1 datacenter
ceph osd crush add-bucket DC2 datacenter
ceph osd crush add-bucket DC3 datacenter
```

- CRUSH マップを変更して、ノードをこの構造内の適切な場所に移動します。

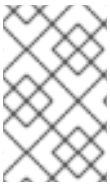
```
ceph osd crush move DC1 root=allDC
ceph osd crush move DC2 root=allDC
ceph osd crush move DC3 root=allDC
ceph osd crush move ceph-node1 datacenter=DC1
ceph osd crush move ceph-node2 datacenter=DC1
ceph osd crush move ceph-node3 datacenter=DC2
ceph osd crush move ceph-node5 datacenter=DC2
ceph osd crush move ceph-node4 datacenter=DC3
ceph osd crush move ceph-node6 datacenter=DC3
```

この構造内で、新しいホストや新しいディスクを追加することもできます。OSD を階層の右側に配置することにより、CRUSH アルゴリズムが冗長な部分を構造内の異なる障害ドメインに配置するように変更されます。

上記の例は、以下のようになります。

```
$ sudo ceph osd tree
ID  WEIGHT  TYPE NAME             UP/DOWN REWEIGHT PRIMARY-AFFINITY
 -8 6.00000 root allDC
 -9 2.00000     datacenter DC1
 -4 1.00000         host ceph-node1
  2 1.00000             osd.2          up  1.00000          1.00000
 -3 1.00000         host ceph-node2
  1 1.00000             osd.1          up  1.00000          1.00000
-10 2.00000     datacenter DC2
 -2 1.00000         host ceph-node3
  0 1.00000             osd.0          up  1.00000          1.00000
 -7 1.00000         host ceph-node5
  5 1.00000             osd.5          up  0.79999          1.00000
-11 2.00000     datacenter DC3
 -6 1.00000         host ceph-node6
  4 1.00000             osd.4          up  0.79999          1.00000
 -5 1.00000         host ceph-node4
  3 1.00000             osd.3          up  1.00000          1.00000
 -1       0 root default
```

上記の一覧には、osd ツリーを表示することで、生成される CRUSH マップが表示されます。ホストがデータセンターにどのように属し、すべてのデータセンターが同じトップレベル構造に属しているかがわかりやすくなりましたが、場所が明確に区別されています。



注記

マップに応じてデータを適切な場所に配置すると、正常なクラスター内でのみ適切に機能します。一部の OSD が利用できない状況では、置き違えが発生する可能性があります。これらの置き違えは、可能な場合は自動的に修正されます。

関連情報

- 詳細は、Red Hat Ceph Storage の Storage Strategies Guide の CRUSH administration の章を参照してください。