Red Hat Ceph Storage 3

운영 가이드

Red Hat Ceph Storage 운영 작업

Last Updated: 2023-08-18

Red Hat Ceph Storage 3 운영 가이드

Red Hat Ceph Storage 운영 작업

법적 공지

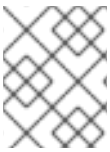
초록

이 문서에서는 Red Hat Ceph Storage 운영 작업을 수행하는 방법을 설명합니다.

차례

1장. 스토리지 클러스터 크기 관리

스토리지 관리자는 스토리지 용량이 확장되거나 축소될 때 Ceph 모니터 또는 OSD를 추가하거나 제거하여 스토리지 클러스터 크기를 관리할 수 있습니다.



참고

스토리지 클러스터를 처음 부트 스트랩하는 경우 Red Hat Enterprise Linux 또는 Ubuntu 용 Red Hat Ceph Storage 3 설치 가이드를 참조하십시오.

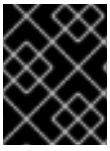
1.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.

1.2. CEPH 모니터

Ceph 모니터는 클러스터 맵의 마스터 사본을 유지 관리하는 경량의 프로세스입니다. 모든 Ceph 클라이언트는 Ceph 모니터에 연락하여 클러스터 맵의 현재 사본을 검색하여 클라이언트가 풀에 바인딩하고 데이터를 읽고 쓸 수 있습니다.

Ceph 모니터는 Paxos 프로토콜의 변형을 사용하여 클러스터 전체에서 맵 및 기타 중요한 정보에 대한 합의를 설정합니다. Paxos의 특성으로 인해 Ceph는 쿼럼을 구축하기 위해 실행 중인 대부분의 모니터가 필요하므로 합의를 설정해야합니다.



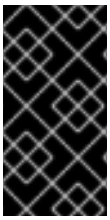
중요

프로덕션 클러스터에 대한 지원을 받으려면 별도의 호스트에 3개 이상의 모니터가 필요합니다.

홀수의 모니터 배포를 권장합니다. 홀수의 모니터는 짝수의 모니터보다 실패에 더 높은 복원력을 갖습니다. 예를 들어, 2개의 모니터 배포에서 쿼럼을 유지하기 위해 Ceph는 어떠한 실패도 허용할 수 없습니다. 모니터 3개, 모니터 4개, 1개의 실패가 있는 경우, 모니터 5개와 함께 오류가 발생합니다. 따라서 홀수의 숫자를 사용하는 것이 좋습니다. Ceph에서는 대부분의 모니터가 실행되고 있고 서로 통신할 수 있어야 하며, 3개 중 2개, 4개 중 3개 등에서 통신할 수 있어야 합니다.

멀티 노드 Ceph 스토리지 클러스터의 초기 배포를 위해 Red Hat은 세 개 이상의 모니터가 있는 경우 한 번에 2개 이상의 모니터를 늘려야 합니다.

모니터는 경량화되어 있으므로 OpenStack 노드와 동일한 호스트에서 실행할 수 있습니다. 그러나 Red Hat은 별도의 호스트에서 모니터를 실행하는 것이 좋습니다.



중요

Red Hat은 동일한 노드에서 Ceph 모니터 및 OSD의 배치를 지원하지 않습니다. 이렇게 하면 스토리지 클러스터 성능에 부정적인 영향을 미칠 수 있습니다.

Red Hat은 컨테이너화된 환경에서 Ceph 서비스를 조합하여만 지원합니다.

스토리지 클러스터에서 모니터를 제거하는 경우 Ceph 모니터에서 Paxos 프로토콜을 사용하여 마스터 스토리지 클러스터 맵에 대한 합의를 설정하는 것이 좋습니다. 쿼럼을 설정하려면 충분한 모니터가 있어야 합니다.

추가 리소스

- 지원되는 모든 Ceph *구성은 Red Hat Ceph Storage 지원 구성 지식 베이스 문서를* 참조하십시오.

1.2.1. 새 Ceph Monitor 노드 준비

스토리지 클러스터에 새 Ceph Monitor를 추가할 때 별도의 노드에 배포합니다. 스토리지 클러스터의 모든 모니터 노드에 대해 노드 하드웨어가 균일해야 합니다.

사전 요구 사항

- 네트워크 연결.
- 새 노드에 **root** 액세스 권한이 있어야 합니다.
- *Red Hat Enterprise Linux 또는 Ubuntu 설치 가이드의 Red Hat Ceph Storage 설치 요구 사항을* 검토하십시오.

절차

1. 새 노드를 서버 랙에 추가합니다.
2. 새 노드를 네트워크에 연결합니다.
3. 새 노드에 Red Hat Enterprise Linux 7 또는 Ubuntu 16.04를 설치합니다.
4. NTP를 설치하고 안정적인 시간 소스를 구성합니다.

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

5. 방화벽을 사용하는 경우 TCP 포트 6789를 엽니다.

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
```

우분투

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

우분투 예

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

1.2.2. Ansible을 사용하여 Ceph 모니터 추가

홀수의 모니터를 유지하기 위해 한 번에 두 개의 모니터를 추가하는 것이 좋습니다. 예를 들어 스토리지 클러스터에 모니터 3개가 있는 경우 Red Hat은 5개의 모니터로 확장할 것을 권장합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.

- 새 노드에 대한 **루트** 액세스 권한 보유.

절차

1. **[mons]** 섹션의 **/etc/ansible/hosts** Ansible 인벤토리 파일에 새 Ceph Monitor 노드를 추가합니다.

   예제

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

2. Ansible이 Ceph 노드에 연결할 수 있는지 확인합니다.

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

3. 디렉터리를 Ansible 구성 디렉터리로 변경합니다.

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

4. Ansible Playbook을 실행합니다.

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

   Ceph의 컨테이너화된 배포에 새 모니터를 추가하는 경우 **site-docker.yml** 플레이북을 실행합니다.

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

5. Ansible 플레이북이 완료되면 새 모니터 노드가 스토리지 클러스터에 있습니다.

1.2.3. 명령줄 인터페이스를 사용하여 Ceph Monitor 추가

홀수의 모니터를 유지하기 위해 한 번에 두 개의 모니터를 추가하는 것이 좋습니다. 예를 들어 스토리지 클러스터에 모니터 3개가 있는 경우 Red Hat은 너무 5개의 모니터를 확장할 것을 권장합니다.

중요

Red Hat은 노드당 하나의 Ceph 모니터 데몬만 실행하는 것이 좋습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 실행 중인 Ceph Monitor 노드와 새 모니터 노드에 대한 **root** 액세스 권한이 있어야 합니다.

절차

1. Red Hat Ceph Storage 3 모니터 리포지토리를 추가합니다.

Red Hat Enterprise Linux

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

우분투

```
[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. 새 Ceph Monitor 노드에 **ceph-mon** 패키지를 설치합니다.

Red Hat Enterprise Linux

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

우분투

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

3. 스토리지 클러스터가 시작 또는 재시작 시 모니터를 식별할 수 있도록 Ceph 구성 파일에 모니터의 IP 주소를 추가합니다.
스토리지 클러스터의 기존 모니터 노드에 있는 Ceph 구성 파일의 **[mon]** 또는 **[global]** 섹션에 새 모니터를 추가하려면 다음을 수행합니다. DNS 확인 가능한 호스트 이름 또는 IP 주소 목록인 **mon_host** 설정: "," 또는 ";" 또는 ". 선택적으로 새 모니터 노드에 대해 Ceph 구성 파일에 특정 섹션을 생성할 수도 있습니다.

구문

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

또는

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

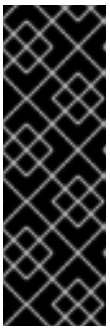
초기 쿼럼 그룹의 모니터를 수행하려면 Ceph 구성 파일의 **[global]** 섹션에 있는 **mon_initial_members** 매개변수에 호스트 이름도 추가해야 합니다.

예제

```
[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
```



중요

프로덕션 스토리지 클러스터 REQUIRE는 고가용성을 보장하기 위해 **mon_initial_members** 및 **mon_host** 에 설정된 세 개 이상의 모니터입니다. 초기 모니터가 한 개뿐인 스토리지 클러스터에서 두 개의 모니터를 더 추가하지만 **mon_initial_members** 및 **mon_host** 에 추가하지 않는 경우 초기 모니터의 실패로 인해 스토리지 클러스터가 잠길 수 있습니다. 추가하는 모니터가 **mon_initial_members** 및 **mon_host** 의 일부인 모니터를 사용하는 경우 새 모니터를 **mon_initial_members** 및 **mon_host** 에 추가해야 합니다.

4. 업데이트된 Ceph 구성 파일을 모든 Ceph 노드 및 Ceph 클라이언트에 복사합니다.

   구문

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

   예제

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

5. 새 모니터 노드에 모니터의 데이터 디렉터리를 생성합니다.

   구문

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

   예제

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

6. 실행 중인 모니터 노드와 새 모니터 노드에서 임시 디렉터리를 생성하여 이 프로세스에 필요한 파일을 보관합니다. 이 디렉터리는 이전 단계에서 만든 모니터의 기본 디렉터리와 달라야 하며 모든 단계가 완료된 후 제거할 수 있습니다.

   구문

   ```
   mkdir $TEMP_DIRECTORY
   ```

   예제

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

7. **ceph** 명령을 실행할 수 있도록 실행 중인 모니터 노드의 admin 키를 새 모니터 노드로 복사합니다.

   구문

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

   예제

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

8. 실행 중인 모니터 노드에서 모니터 인증 키를 검색합니다.

   구문

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

   예제

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

9. 실행 중인 모니터 노드에서 모니터 맵을 검색합니다.

   구문

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

   예제

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

10. 수집한 모니터 데이터를 새 모니터 노드에 복사합니다.

    구문

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

    예제

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

11. 이전에 수집한 데이터에서 새 모니터의 데이터 디렉토리를 준비합니다. 모니터 맵의 경로를 지정하여 모니터 맵에서 쿼럼 정보를 검색해야 합니다. **모니터 인증 키의 경로도 지정해야 합니다.**

    구문

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

예제

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

12. **사용자 지정 이름이 있는 스토리지 클러스터의 경우 /etc/sysconfig/ceph 파일에 다음 행을 추가합니다.**

Red Hat Enterprise Linux

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

우분투

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

13. **새 모니터 노드에서 소유자 및 그룹 권한을 업데이트합니다.**

구문

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

예제

```
[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. **새 모니터 노드에서 ceph-mon 프로세스를 활성화하고 시작합니다.**

구문

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

예제

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

추가 리소스

- ***Red Hat Enterprise Linux 또는 Ubuntu 용 설치 가이드의 Red Hat Ceph Storage 리포지토리 활성화 섹션을* 참조하십시오.**

1.2.4. Ansible을 사용하여 Ceph 모니터 제거

Ansible을 사용하여 **Ceph** 모니터를 제거하려면 **shrink-mon.yml** 플레이북을 사용합니다.

사전 요구 사항

- **Ansible** 관리 노드.
- **Ansible**에서 배포한 실행 중인 **Red Hat Ceph Storage** 클러스터.

절차

1. **/usr/share/ceph-ansible/** 디렉토리로 변경합니다.

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

2. **shrink-mon.yml** 플레이북을 **infrastructure-playbooks** 디렉터리에서 현재 디렉터리로 복사합니다.

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

3. **Red Hat Ceph Storage**의 일반 또는 컨테이너화된 배포에 대해 **shrink-mon.yml** 플레이북을 실행합니다.

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

교체:

- 모니터 노드의 짧은 호스트 이름이 있는 **<hostname>** 입니다. 더 많은 모니터를 제거하려면 호스트 이름을 쉼표로 구분합니다.
- **Ansible** 사용자 이름이 있는 **<ansible-user>**

예를 들어 **monitor1** 호스트 이름이 있는 노드에 있는 모니터를 제거하려면 다음을 수행합니다.

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

4. 클러스터의 모든 **Ceph** 구성 파일에서 **Monitor** 항목을 제거합니다.

5. 모니터가 성공적으로 제거되었는지 확인합니다.

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

추가 리소스

- **Red Hat Ceph Storage** 설치에 대한 자세한 내용은 **Red Hat Enterprise Linux** 또는 **Ubuntu** 설치 가이드 를 참조하십시오.

1.2.5. 명령줄 인터페이스를 사용하여 Ceph Monitor 제거

Ceph 모니터를 제거하려면 스토리지 클러스터에서 **ceph-mon** 데몬을 제거하고 스토리지 클러스터 맵을 업데이트해야 합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 모니터 노드에 **root** 액세스 권한이 있어야 합니다.

절차

1. **monitor** 서비스를 중지합니다.

구문

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

예제

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

2. **스토리지 클러스터에서 모니터를 삭제합니다.**

구문

```
ceph mon remove $MONITOR_ID
```

예제

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

3. **기본적으로 Ceph 구성 파일인 /etc/ceph/ceph.conf 에서 monitor 항목을 제거합니다.**

4. **스토리지 클러스터의 나머지 모든 Ceph 노드에 Ceph 구성 파일을 재배포합니다.**

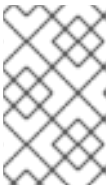
구문

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

예제

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

5. **컨테이너만 해당. monitor 서비스를 비활성화합니다.**



참고

컨테이너를 사용하는 경우에만 5-9단계를 수행합니다.

구문

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

예제

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

6. **컨테이너만 해당. systemd에서 서비스를 제거합니다.**

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

–

7. **컨테이너만 해당. systemd 관리자 설정을 다시로드합니다.**

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

8. **컨테이너만 해당. 실패한 모니터 장치의 상태를 재설정합니다.**

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

9. **컨테이너만 해당. ceph-mon RPM을 제거합니다.**

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

10. **모니터 데이터를 보관합니다.**

구문

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

예제

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

11. **모니터 데이터를 삭제합니다.**

구문

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

예제

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

추가 리소스

- 자세한 내용은 **Knowledgebase** 솔루션 ***How to re-deploy Ceph Monitor in a director deployed Ceph cluster*** 에서 참조하십시오.

1.2.6. 비정상적인 스토리지 클러스터에서 Ceph 모니터 제거

이 절차에서는 비정상 스토리지 클러스터에서 **ceph-mon** 데몬을 제거합니다. 배치 그룹이 영구적으로 활성 상태가 아닌 비정상적인 스토리지 클러스터 + 정리.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 모니터 노드에 **root** 액세스 권한이 있어야 합니다.
- 실행 중인 하나 이상의 **Ceph Monitor** 노드.

절차

1. 남아 있는 모니터를 식별하고 해당 노드에 로그인합니다.

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

2. **ceph-mon** 데몬을 중지하고 **monmap** 파일의 사본을 추출합니다. :

구문

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

예제

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

3. 지원되지 않는 모니터(s)를 제거합니다.

구문

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

예제

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

4. 제거된 모니터(s)와 함께 Surviving 모니터 맵을urviving 모니터에 삽입합니다.

구문

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

예제

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

1.3. CEPH OSD

Red Hat Ceph Storage 클러스터가 가동되어 실행되면 런타임 시 스토리지 클러스터에 **OSD**를 추가할 수 있습니다.

Ceph OSD는 일반적으로 스토리지 드라이브 **1**대와 노드 내에서 연결된 저널을 위한 하나의 **ceph-osd** 데몬으로 구성됩니다. 노드에 여러 스토리지 드라이브가 있는 경우 각 드라이브에 대해 하나의 **ceph-osd** 데몬을 매핑합니다.

스토리지 용량의 상위에 도달하고 있는지 확인하기 위해 클러스터의 용량을 정기적으로 확인하는 것이 좋습니다. 스토리지 클러스터가 전체 비율에 도달하므로 하나 이상의 **OSD**를 추가하여 스토리지 클러스터의 용량을 확장합니다.

Red Hat Ceph Storage 클러스터의 크기를 줄이거나 하드웨어를 교체하려는 경우 런타임 시 **OSD**를 제거할 수도 있습니다. 노드에 스토리지 드라이브가 여러 개 있는 경우 해당 드라이브의 **ceph-osd** 데몬 중 하나를 제거해야 할 수도 있습니다. 일반적으로 스토리지 클러스터의 용량을 확인하여 용량의 상단에 도달했는지 확인하는 것이 좋습니다. 스토리지 클러스터가 거의 전체 비율에 있지 않은 **OSD**를 제거해야 합니다.

중요

OSD를 추가하기 전에 스토리지 클러스터가 전체 비율에 도달할 수 없도록 하십시오. 스토리지 클러스터가 거의 전체 비율에 도달한 후 발생하는 **OSD** 오류는 스토리지 클러스터가 전체 비율을 초과할 수 있습니다 . **Ceph**는 스토리지 용량 문제를 해결할 때까지 데이터를 보호하기 위한 쓰기 액세스를 차단합니다. 먼저 전체 비율에 미치는 영향을 고려하지 않고 **OSD**를 제거하지 마십시오.

1.3.1. Ceph OSD 노드 구성

OSD를 사용할 풀의 스토리지 전략으로 **Ceph OSD** 및 지원 하드웨어를 유사하게 구성해야 합니다. **Ceph**에서는 일관된 성능 프로필을 위해 풀 간에 균일한 하드웨어를 선호합니다. 최상의 성능을 위해 동일한 유형 또는 크기의 드라이브가 있는 **CRUSH** 계층 구조를 고려하십시오. 자세한 내용은 스토리지 전략 가이드를 참조하십시오.

dissimilar 크기의 드라이브를 추가하는 경우 그에 따라 가중치를 조정해야 합니다. **OSD**를 **CRUSH** 맵에 추가할 때 새 **OSD**의 가중치를 고려합니다. 하드 드라이브 용량은 연간 약 **40%** 증가하므로 최신 **OSD** 노드는 스토리지 클러스터의 이전 노드보다 더 큰 하드 드라이브를 사용할 수 있습니다. 즉, 가중치가 더 클 수 있습니다.

새 설치를 수행하기 전에 **Red Hat Enterprise Linux** 또는 **Ubuntu** *용 설치 가이드의* ***Red Hat Ceph Storage*** *설치 요구 사항* 장을 검토하십시오.

1.3.2. 컨테이너 OSD ID를 드라이브에 매핑

컨테이너화된 **OSD**가 사용 중인 드라이브를 식별해야 하는 경우가 있습니다. 예를 들어 **OSD**에 문제가 있는 경우 드라이브 상태를 확인하는 데 사용하는 드라이브를 알아야 할 수 있습니다. 또한 컨테이너화되지 않은 **OSD**의 경우 **OSD ID**를 참조하여 시작하고 중지하지만 컨테이너화된 **OSD**를 시작하고 중지하려면 사용하는 드라이브를 참조해야 합니다.

사전 요구 사항

- 컨테이너화된 환경에서 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 컨테이너 호스트에 대한 루트 액세스 권한이 있어야 합니다.

절차

1. 컨테이너 이름을 찾습니다. 예를 들어 **osd.5**와 연결된 드라이브를 식별하려면 **osd.5** 가 실행 중인 컨테이너 노드에서 터미널을 연 다음 **docker ps** 를 실행하여 모든 컨테이너를 나열합니

다.

예제

```
[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** 를 사용하여 이전 출력의 **OSD** 컨테이너 이름에 **ceph-volume lvm** 목록을 실행합니다.

예제

```
[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                 SIEgHe-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
```

이 출력에서 **osd.5** 가 **/dev/sdb** 와 연결되어 있음을 확인할 수 있습니다.

추가 리소스

- 자세한 내용은 실패한 **OSD** 디스크 배치를 참조하십시오.

1.3.3. 동일한 디스크 토폴로지가 있는 Ansible을 사용하여 Ceph OSD 추가

디스크 토폴로지가 동일한 **Ceph OSD**의 경우 **Ansible**은 **/usr/share/ceph-ansible/group_vars/osds** 파일의 **devices:** 섹션에 지정된 동일한 장치 경로를 사용하여 다른 **OSD** 노드와 동일한 개수의 **OSD**를 추가합니다.

참고

새 **Ceph OSD** 노드는 나머지 **OSD**와 동일한 구성을 갖습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- ***Red Hat Enterprise Linux* 또는 *Ubuntu* 설치 가이드의 *Red Hat Ceph Storage* 설치 *요구 사항*을** 검토하십시오.
- 새 노드에 대한 루트 액세스 권한 보유.
- 스토리지 클러스터의 다른 **OSD** 노드와 동일한 **OSD** 데이터 드라이브 수입니다.

절차

1. **Ceph OSD 노드를 [osds] 섹션 아래의 /etc/ansible/hosts 파일에 추가합니다.**

   예제

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

2. **Ansible이 Ceph 노드에 연결할 수 있는지 확인합니다.**

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

3. **Ansible 구성 디렉터리로 이동합니다.**

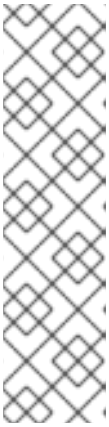
```
[user@admin ~]$ cd /usr/share/ceph-ansible
```

4. **add-osd.yml 파일을 /usr/share/ceph-ansible/ 디렉터리에 복사합니다.**

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

5. **Ceph의 일반 또는 컨테이너화된 배포에 대해 Ansible 플레이북을 실행합니다.**

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



참고

OSD를 추가할 때 **Placement s**와 함께 플레이북이 **active+clean**으로 보고되지 않은 경우 **all.yml** 파일에서 다음 변수를 구성하여 재시도 및 지연을 조정합니다.

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

1.3.4. 다양한 디스크 토폴로지가 있는 Ansible을 사용하여 Ceph OSD 추가

디스크 토폴로지가 다른 **Ceph OSD**의 경우 새 **OSD** 노드를 기존 스토리지 클러스터에 추가하는 두 가지 방법이 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- ***Red Hat Enterprise Linux* 또는 *Ubuntu* 설치 가이드의 *Red Hat Ceph Storage* 설치 *요구 사항*을** 검토하십시오.
- 새 노드에 대한 루트 액세스 권한 보유.

절차

1. 첫 번째 접근 방식

   a. **[osds]** 섹션의 **/etc/ansible/hosts** 파일에 새 **Ceph OSD** 노드를 추가합니다.

   예제

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

b.

/etc/ansible/host_vars/ 디렉터리에 스토리지 클러스터에 추가된 각 새 Ceph OSD 노드의 새 파일을 만듭니다.

구문

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

예제

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

c.

새 파일을 편집하고 devices: 및 dedicated_devices: 섹션을 파일에 추가합니다. 이러한 각 섹션 아래에 -, 공백을 추가한 다음 이 OSD 노드의 블록 장치 이름에 대한 전체 경로를 추가합니다.

예제

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

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

d. **Ansible이 모든 Ceph 노드에 연결할 수 있는지 확인합니다.**

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

e. **디렉터리를 Ansible 구성 디렉터리로 변경합니다.**

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

f. **add-osd.yml 파일을 /usr/share/ceph-ansible/ 디렉터리에 복사합니다.**

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

g. **Ansible Playbook을 실행합니다.**

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

2. **두 번째 접근 방식**

a. **새 OSD 노드 이름을 /etc/ansible/hosts 파일에 추가하고, 다른 디스크 토폴로지를 지정하여 devices 및 dedicated_devices 옵션을 사용합니다.**

예제

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

b. **Ansible이 모든 Ceph 노드에 연결할 수 있는지 확인합니다.**

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

c. 디렉터리를 **Ansible** 구성 디렉터리로 변경합니다.

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

d. **add-osd.yml** 파일을 **/usr/share/ceph-ansible/** 디렉터리에 복사합니다.

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

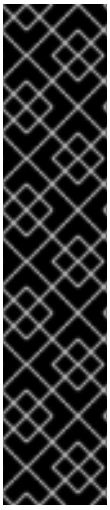
e. **Ansible Playbook**을 실행합니다.

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

1.3.5. 명령줄 인터페이스를 사용하여 Ceph OSD 추가

다음은 **Red Hat Ceph Storage**에 **OSD**를 수동으로 추가하기 위한 상위 수준 워크플로입니다.

1. **ceph-osd** 패키지를 설치하고 새 **OSD** 인스턴스 생성
2. **OSD** 데이터 및 저널 드라이브 준비 및 마운트
3. **CRUSH** 맵에 새 **OSD** 노드 추가
4. 소유자 및 그룹 권한을 업데이트
5. **ceph-osd** 데몬 활성화 및 시작



중요

ceph-disk 명령은 더 이상 사용되지 않습니다. 이제 명령줄 인터페이스에서 **OSD**를 배포하는 데 **ceph-volume** 명령이 선호됩니다. 현재 **ceph-volume** 명령은 **lvm** 플러그인만 지원합니다. **Red Hat**은 두 명령을 참조로 사용하여 이 가이드의 예제를 제공하므로 스토리지 관리자가 **ceph-disk**를 사용하는 모든 사용자 지정 스크립트를 **ceph- volume** 으로 변환할 수 있습니다.

ceph-volume 명령 사용에 대한 자세한 내용은 **Red Hat Ceph Storage** 관리 가이드를 참조하십시오.

참고

사용자 지정 스토리지 클러스터 이름의 경우 **ceph** 및 **ceph-osd** 명령과 함께 **--cluster $CLUSTER_NAME** 옵션을 사용합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- ***Red Hat Enterprise Linux 또는 Ubuntu 설치 가이드의 Red Hat Ceph Storage 설치 요구 사항을*** 검토하십시오.
- 새 노드에 대한 루트 액세스 권한 보유.

절차

1. **Red Hat Ceph Storage 3 OSD** 소프트웨어 리포지토리를 활성화합니다.

Red Hat Enterprise Linux

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

우분투

```
[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/ 디렉토리를 만듭니다.**

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

3. **새 OSD 노드에서 Ceph Monitor 노드 중 하나에서 Ceph 관리 인증 키링 및 구성 파일을 복사합니다.**

구문

```
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
```

예제

```
[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. **새 Ceph OSD 노드에 ceph-osd 패키지를 설치합니다.**

Red Hat Enterprise Linux

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

우분투

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

5. 저널을 배치하거나 새 **OSD**에 전용 저널을 사용할지 여부를 결정합니다.

참고

--filestore 옵션이 필요합니다.

a. 배치된 저널이 있는 **OSD**의 경우:

구문

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

예

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

b.

전용 저널이 있는 OSD의 경우:

구문

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

또는

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

예

```
[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 옵션을 설정합니다.

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

7.

새 OSD를 활성화합니다.

구문

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

또는

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

예제

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

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

8. **CRUSH 맵에 OSD를 추가합니다.**

구문

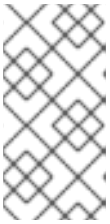
```
ceph osd crush add $OSD_ID $WEIGHT [$BUCKET_TYPE=$BUCKET_NAME ...]
```

예제

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

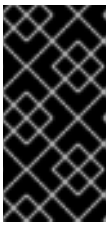
참고

버킷을 두 개 이상 지정하는 경우 명령은 지정된 버킷에 **OSD**를 배치하고 지정한 다른 버킷 아래의 버킷을 *이동합니다*.



참고

CRUSH 맵을 수동으로 편집할 수도 있습니다. **Red Hat Ceph Storage 3**용 스토리지 전략 가이드의 **CRUSH** 맵 편집 섹션을 참조하십시오.



중요

루트 버킷만 지정하면 **OSD**가 **root**에 직접 연결되지만 **CRUSH** 규칙에는 호스트 버킷 내부에 **OSD**가 있어야 합니다.

9. **noup** 옵션을 설정 해제합니다.

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

10. 새로 생성된 디렉터리에 대한 소유자 및 그룹 권한을 업데이트합니다.

구문

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

예제

```
[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. 사용자 지정 이름으로 클러스터를 사용하는 경우 적절한 파일에 다음 행을 추가합니다.

Red Hat Enterprise Linux

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

우분투

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

$CLUSTER_NAME 을 사용자 지정 클러스터 이름으로 교체합니다.

12. 새 **OSD**가 가동 되어 데이터를 수신할 준비가 되었는지 확인하려면 **OSD** 서비스를 활성화하고 시작합니다.

구문

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

예제

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

1.3.6. Ansible을 사용하여 Ceph OSD 제거

경우에 따라 **Red Hat Ceph Storage** 클러스터의 용량을 축소해야 할 수도 있습니다. **Ansible**을 사용하여 **Red Hat Ceph Storage** 클러스터에서 **OSD**를 제거하고 사용되는 **OSD** 시나리오에 따라 **shrink-osd.yml** 또는 **shrink-osd-ceph-disk.yml** 플레이북을 실행합니다. **osd_scenario** 가 **collocated** 또는 **non-collocated** 로 설정된 경우 **shrink-osd-ceph-disk.yml** 플레이북을 사용합니다. **osd_scenario** 가 **lvm** 으로 설정된 경우 **shrink-osd.yml** 플레이북을 사용합니다.

중요

스토리지 클러스터에서 **OSD**를 제거하면 해당 **OSD**에 포함된 모든 데이터가 삭제됩니다.

사전 요구 사항

- **Ansible**에서 배포한 실행 중인 **Red Hat Ceph Storage.**
- 실행 중인 **Ansible** 관리 노드.
- **Ansible** 관리 노드에 대한 루트 수준 액세스입니다.

절차

1. **/usr/share/ceph-ansible/** 디렉토리로 변경합니다.

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

2. **Ceph Monitor** 노드의 **/etc/ceph/** 에서 관리자 인증 키를 삭제하려는 **OSD**가 포함된 노드로 복사합니다.

3. **infrastructure-playbooks** 디렉터리에서 현재 디렉터리에 적절한 플레이북을 복사합니다.

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

또는

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

4. 베어 메탈 또는 컨테이너 배포의 경우 적절한 **Ansible** 플레이북을 실행합니다.

구문

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

또는

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

교체:

- **OSD**의 **ID**가 있는 **$ID** 입니다. 더 많은 **OSD**를 제거하려면 **OSD ID**를 쉼표로 구분합니다.
- **$ANSIBLE_USER** 및 **Ansible** 사용자 이름

예제

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

또는

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

5. **OSD**가 성공적으로 제거되었는지 확인합니다.

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

추가 리소스

- 자세한 내용은 ***Red Hat Enterprise Linux*** 또는 ***Ubuntu*** 용 **Red Hat Ceph Storage** 설치 가이드 를 참조하십시오.

1.3.7. 명령줄 인터페이스를 사용하여 Ceph OSD 제거

스토리지 클러스터에서 **OSD**를 제거하려면 클러스터 맵을 업데이트하고, 인증 키를 제거하고, **OSD** 맵에서 **OSD**를 제거하고, **ceph.conf** 파일에서 **OSD**를 제거해야 합니다. 노드에 여러 개의 드라이브가 있는 경우 이 절차를 반복하여 각 드라이브에 대해 **OSD**를 제거해야 할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 스토리지 클러스터가 거의 가득 차지 않도록 사용 가능한 **OSD**가 충분합니다.
- **OSD** 노드에 **root** 액세스 권한이 있어야 합니다.

절차

1. **OSD** 서비스를 비활성화하고 중지합니다.

구문

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

예제

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

OSD가 중지되면 종료됩니다.

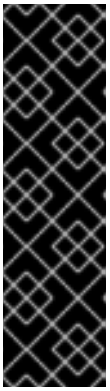
2. **스토리지 클러스터에서 OSD를 제거합니다.**

구문

```
ceph osd out $OSD_ID
```

예제

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



중요

OSD가 부족하면 **Ceph**가 스토리지 클러스터의 다른 **OSD**에 데이터 재조정 및 복사를 시작합니다. 다음 단계를 진행하기 전에 스토리지 클러스터가 **active+clean** 이 될 때까지 기다리는 것이 좋습니다. 데이터 마이그레이션을 관찰하려면 다음 명령을 실행합니다.

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

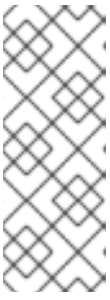
3. **CRUSH** 맵에서 **OSD**를 제거하여 더 이상 데이터를 받지 않도록 합니다.

구문

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

예제

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



참고

CRUSH 맵을 컴파일하고, 장치 목록에서 **OSD**를 제거하고, 호스트 버킷의 항목으로 장치를 제거하거나 호스트 버킷을 제거할 수도 있습니다. **CRUSH** 맵에 있고 호스트를 제거하려는 경우 맵을 다시 컴파일하여 설정합니다. 자세한 내용은 스토리지 전략 가이드 를 참조하십시오.

4. **OSD** 인증 키를 제거합니다.

구문

-

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

예제

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

5. **OSD를 제거합니다.**

구문

```
ceph osd rm $OSD_ID
```

예제

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

6. **스토리지 클러스터의 구성 파일을 기본적으로 /etc/ceph/ceph.conf 로 편집하고 OSD 항목이 있는 경우 삭제합니다.**

예제

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

7. **OSD**를 수동으로 추가한 경우 **/etc/fstab** 파일에서 **OSD**에 대한 참조를 제거합니다.

8. 업데이트된 구성 파일을 스토리지 클러스터에 있는 다른 모든 노드의 **/etc/ceph/** 디렉터리에 복사합니다.

구문

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

예제

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

1.3.8. 명령줄 인터페이스를 사용하여 저널 교체

저널 및 데이터 장치가 동일한 물리적 장치에 있을 때 저널을 교체하는 절차는 (예: **osd_scenario: collocated**) 전체 **OSD**를 교체해야 합니다. 그러나 저널이 데이터 장치와 별도의 물리적 장치에 있는 **OSD**에서 (예: **osd_scenario: non-collocated**)를 사용하면 저널 장치만 교체할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 새 파티션 또는 스토리지 장치.

절차

1. 백필을 방지하려면 클러스터를 **noout** 로 설정합니다.

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

2. 저널이 변경되는 **OSD**를 중지합니다.

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

3. **OSD**에서 저널을 플러시합니다.

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

4. 기존 저널 파티션을 제거하여 파티션 **UUID**가 새 파티션과 충돌하지 않도록 합니다.

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

replace

- **$OLD_PART_NUM -** 기존 저널 장치의 파티션 번호입니다.
- **$OLD_DEV_PATH** 및 이전 저널 장치 경로가 사용됩니다.

예제

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

5. 새 장치에 새 저널 파티션을 생성합니다. 이 **sgdisk** 명령은 사용 가능한 다음 파티션 번호를 자동으로 사용합니다.

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

replace

- 환경에 적합한 저널 크기(예: **10240M**)가 있는 **$JOUECDHEL_SIZE.**
- 새 저널에 사용할 장치 경로가 포함된 **NEW_DEV_PATH.**

참고

저널의 최소 기본 크기는 **5GB**입니다. 일반적으로 **10GB** 이상의 값은 필요하지 않습니다. 자세한 내용은 **Red Hat** 지원에 문의하십시오.

예제

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

6. 새 파티션에 적절한 매개변수를 설정합니다.

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

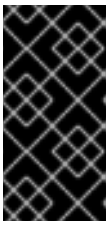
replace

- **$OLD_PART_UUID** 및 관련 **OSD**의 **journal_uuid** 파일에 **UUID**를 사용합니다. 예를 들어 **OSD 0** 의 경우 **/var/lib/ceph/osd/ceph-0/journal_uuid** 의 **UUID**를 사용합니다.
- 새 저널에 사용할 장치 경로가 포함된 **NEW_DEV_PATH.**

예제

```
[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
```

위의 **sgdisk** 명령을 실행하면 새 저널 파티션이 **Ceph**에 대해 준비되어 있고 저널을 생성할 수 있습니다.



중요

sgdisk 의 제한으로 인해 파티션을 올바르게 생성되지 않기 때문에 이 명령을 파티션 생성 명령과 결합할 수 없습니다.

7. 새 저널을 생성합니다.

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

8. **OSD**를 시작합니다.

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

1. **OSD**에서 **noout** 플래그를 제거합니다.

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

2. 저널이 올바른 장치와 연결되어 있는지 확인합니다.

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

1.3.9. 데이터 마이그레이션 관찰

OSD를 **CRUSH** 맵에 추가하거나 제거하면 **Ceph**에서 배치 그룹을 새 **OSD** 또는 기존 **OSD**로 마이그레이션하여 데이터 리밸런싱을 시작합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.

- 최근에 **OSD**를 추가하거나 제거했습니다.

절차

1. 데이터 마이그레이션을 관찰하려면 다음을 수행합니다.

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

2. 마이그레이션이 완료되면 배치 그룹 상태가 **active+clean** 에서 활성, 일부 저하된 개체 및 마지막으로 활성**+clean** 으로 변경되는 것을 봅니다.

3. 유틸리티를 종료하려면 **Ctrl + C** 를 누릅니다.

1.4. 배치 그룹 재계산

배치 그룹**(PG)**은 사용 가능한 **OSD**에서 풀 데이터의 분배를 정의합니다. 배치 그룹은 사용할 지정된 중복 알고리즘에 빌드됩니다. **3**방향 복제의 경우 중복은 세 개의 다른 **OSD**를 사용하도록 정의됩니다. **Earsure-coded** 풀의 경우 사용할 **OSD** 수는 청크 수로 정의됩니다.

풀을 정의할 때 배치 그룹의 수는 사용 가능한 모든 **OSD**에 걸쳐 데이터가 분산되는 세분화 등급을 정의합니다. 크기가 클수록 용량 부하의 동등화가 더 클 수 있습니다. 그러나 데이터를 재구성하는 경우 배치 그룹을 처리하는 것도 중요하므로 신중하게 선택하는 것이 중요합니다. 계산 작업을 지원하기 위해 툴을 사용하여 민첩한 환경을 생성할 수 있습니다.

스토리지 클러스터의 라이프사이클 동안 풀은 초기에 예상되는 한계보다 증가할 수 있습니다. 드라이브 수가 늘어나면 재계산이 권장됩니다. **OSD**당 배치 그룹 수는 약 **100**개여야 합니다. 스토리지 클러스터에 **OSD**를 더 추가하면 **OSD**당 **DC** 수가 시간이 지남에 따라 줄어듭니다. 스토리지 클러스터에서 처음 **120**개의 드라이브로 시작하고 풀의 **pg_num** 을 **4000**개로 설정하면 **OSD**당 **100**개의 **Placements**가 되며, **3**개의 복제 요소가 적용됩니다. 시간이 지남에 따라 **OSD** 수를 **10**배로 늘릴 경우 **OSD**당 **DC** 수는 **10**개로 줄어듭니다. **OSD**당 배치 수가 적지 않게 분산된 용량이기 때문에 풀당 **PPP**를 조정하는 것이 좋습니다.

배치 그룹의 수를 조정하는 작업은 온라인으로 수행할 수 있습니다. 재계산은 **PPP** 숫자를 재계산하는 것뿐만 아니라 데이터 재배치가 포함될 것이며, 이는 긴 프로세스가 될 것입니다. 그러나 데이터 가용성은 언제든지 유지됩니다.

실패한 **OSD**에 있는 모든 **DC**의 재구성이 즉시 시작되므로 **OSD**당 **VDO** 수가 매우 많으면 안 됩니다. 많은 **IOPS**가 적시에 재구성을 수행해야 하므로 사용할 수 없을 수 있습니다. 이로 인해 깊은 **I/O** 대기열과

스토리지 클러스터를 사용할 수 없게 되거나 복구 시간이 길어집니다.

추가 리소스

- 지정된 사용 사례로 값을 계산하기 위한 PG 계산기 를 참조하십시오.
- 자세한 내용은 **Red Hat Ceph Storage Strategies Guide**의 Erasure Code Pools 장을 참조하십시오.

1.5. CEPH MANAGER 밸런서 모듈 사용

밸런서는 **OSD**에서 배치 그룹 배치 또는 **PG**를 자동 또는 감독 방식으로 수행하기 위해 **Ceph Manager** 용 모듈입니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터

밸런서를 시작합니다.

1. **balancer** 모듈이 활성화되었는지 확인합니다.

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

2. **balancer** 모듈을 켭니다.

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

모드

현재 지원되는 두 가지 밸런서 모드가 있습니다.

- **crush-compat: CRUSH compat** 모드는 **Ceph Luminous**에 도입된 **compat weight-set** 기능을 사용하여 **CRUSH** 계층 구조의 장치에 대한 다른 가중치 집합을 관리합니다. 일반 가중치는 장치에 저장하려는 데이터의 대상 양을 반영하기 위해 장치의 크기로 설정되어야 합니다. 그런 다음 밸런서는 가중치 설정 값을 최적화하여 대상 배포와 일치하는 배포를 가능한 한 가깝게 달성하

기 위해 작은 증분으로 조정 또는 축소합니다. **PG** 배치는 **pseudorandom** 프로세스이기 때문에 배치에는 자연 양의 변동이 있습니다. 가중치를 최적화하면 이러한 자연 변동이 발생합니다.

이 모드는 이전 클라이언트와 완전히 호환됩니다. **OSDMap** 및 **CRUSH** 맵이 이전 클라이언트와 공유되면 밸런서에서 최적화된 가중치를 실제 가중치로 제공합니다.

이 모드의 주요 제한 사항은 계층 구조의 하위 트리가 **OSD**를 공유하는 경우 다른 배치 규칙으로 밸런서가 여러 개의 **CRUSH** 계층을 처리할 수 없다는 것입니다. 이 구성에서는 공유 **OSD**에서 공간 사용률을 관리하기가 어렵기 때문에 일반적으로 권장되지 않습니다. 따라서 이 제한 사항은 일반적으로 문제가 아닙니다.

- **upmap: Luminous**부터 **OSDMap**은 개별 **OSD**의 명시적 매핑을 일반 **CRUSH** 배치 계산에 예외적으로 저장할 수 있습니다. 이러한 **upmap** 항목은 **PG** 매핑을 세밀하게 제어할 수 있습니다. 이 **CRUSH** 모드는 분산 배포를 수행하기 위해 개별 **PG**의 배치를 최적화합니다. 대부분의 경우 이 배포는 각 **OSD +/-1 PG**에서 동일한 개수의 **PG**가 있는 "완성"입니다.

중요

upmap을 사용하려면 모든 클라이언트가 **Red Hat Ceph Storage 3.x** 이상 및 **Red Hat Enterprise Linux 7.5** 이상을 실행해야 합니다.

이 기능을 사용하려면 다음을 사용하여 동종 또는 이후 클라이언트만 지원하는 데 필요하다는 사실을 클러스터에 알려주어야 합니다.

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

사전 요구 클라이언트 또는 데몬이 모니터에 연결된 경우 이 명령은 실패합니다.

알려진 문제로 인해 **kernel CephFS** 클라이언트는 **jewel** 클라이언트를 보고합니다. 이 문제를 해결하려면 **--yes-i-really-mean-it** 플래그를 사용하십시오.

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

에서 사용 중인 클라이언트 버전을 확인할 수 있습니다.

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

주의

Red Hat Ceph Storage 3.x에서 **upmap** 기능은 클러스터가 사용되는 **PG**의 밸런싱을 위해 **Ceph Manager** 밸런서 모듈에서만 사용할 수 있습니다. **upmap** 기능을 사용하여 **PG** 수동 재조정은 **Red Hat Ceph Storage 3.x**에서 지원되지 않습니다.

기본 모드는 **crush-compat** 입니다. 모드는 다음을 사용하여 변경할 수 있습니다.

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

또는 다음을 수행합니다.

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

상태

pod의 현재 상태는 다음을 사용하여 언제든지 확인할 수 있습니다.

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

자동 분산

기본적으로 **balancer** 모듈을 활성화하면 자동 분산이 사용됩니다.

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

다음을 사용하여 밸런서를 다시 끌 수 있습니다.

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

이 명령은 이전 클라이언트와 역호환되는 **crush-compat** 모드를 사용하며 시간이 지남에 따라 데이터 배포를 약간 변경하여 **OSD**를 동일하게 활용합니다.

제한

클러스터가 성능이 저하되고 **OSD**가 실패하여 시스템이 아직 복구되지 않은 경우와 같이 **cluster**가 저하된 경우는 **Placement** 배포에서 조정되지 않습니다.

클러스터가 정상이면 밸런서에서 잘못된 배치 또는 이동해야 하는 **PC**의 백분율이 기본적으로 **5%** 미만인 만큼 해당 변경 사항을 제한합니다. 이 백분율은 **max_misplaced** 설정을 사용하여 조정할 수 있습니다. 예를 들어 임계값을 **7%**로 늘리려면 다음을 수행합니다.

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

감독된 최적화

밸런서 작업은 다음과 같은 몇 가지 단계로 나뉩니다.

1. **계획수립**
2. 현재 **PG** 배포에 대한 데이터 배포 품질 평가 또는 계획을실행한 후 발생하는 **PG** 배포를 평가합니다.
3. **계획실행**
   - 현재 배포를 평가하고 점수를 매기려면 다음을 수행합니다.

     ```
     [root@mon ~]# ceph balancer eval
     ```
   - 단일 풀의 배포를 평가하려면 다음을 수행합니다.

     ```
     [root@mon ~]# ceph balancer eval <pool-name>
     ```
   - 평가에 대한 세부 정보를 보려면 다음을 수행합니다.

     ```
     [root@mon ~]# ceph balancer eval-verbose ...
     ```
   - 현재 구성된 모드를 사용하여 계획을 생성하려면 다음을 수행합니다.

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

<plan-name> 을 사용자 정의 계획 이름으로 바꿉니다.

- 계획 내용을 보려면 다음을 수행합니다.

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

- 이전 계획을 삭제하려면 다음을 수행합니다.

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

- 현재 기록된 계획을 보려면 상태 명령을 사용합니다.

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

- 계획을 실행한 후 발생하는 배포 품질을 계산하려면 다음을 수행합니다.

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

- 계획을 실행하려면 다음을 수행합니다.

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

[NOTE]: 배포를 개선할 것으로 예상되는 경우에만 계획을 실행합니다. 실행 후 계획이 취소됩니다.

1.6. 추가 리소스

- 자세한 내용은 **Red Hat Ceph Storage Strategies Guide** 의 **PG(배치 그룹)** 장을 참조하십시오.

2장. 디스크 오류 처리

스토리지 관리자는 스토리지 클러스터 수명 동안 일정 시점에서 디스크 오류를 처리해야 합니다. 실제 오류가 발생하기 전에 디스크 오류를 테스트하고 시뮬레이션하면 실제 일이 발생할 때 대비할 수 있습니다.

다음은 실패한 디스크를 교체하는 상위 수준 워크플로입니다.

1. 실패한 OSD를 찾습니다.
2. OSD 제거.
3. 노드에서 OSD 데몬을 중지합니다.
4. Ceph 상태를 확인합니다.
5. CRUSH 맵에서 OSD를 제거합니다.
6. OSD 권한 부여를 삭제합니다.
7. 스토리지 클러스터에서 OSD를 제거합니다.
8. 노드에서 파일 시스템을 마운트 해제합니다.
9. 실패한 드라이브를 바꿉니다.
10. OSD를 스토리지 클러스터에 다시 추가합니다.
11. Ceph 상태를 확인합니다.

2.1. 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 오류가 발생한 디스크.

2.2. 디스크 오류

Ceph는 내결함성을 위해 설계되었습니다. 즉, 데이터를 손실하지 않고 **Ceph**가 성능이 저하된 상태에서 작동할 수 있습니다. 데이터 스토리지 드라이브가 실패해도 **Ceph**가 계속 작동할 수 있습니다. 성능이 저하된 상태는 다른 **OSD**에 저장된 추가 데이터 사본이 스토리지 클러스터의 다른 **OSD**로 자동으로 다시 입력됨을 의미합니다. **OSD**가 아래로 표시되면 드라이브가 실패할 수 있습니다.

드라이브에 오류가 발생하면 초기에 **OSD** 상태가 다운 되지만 스토리지 클러스터에서는 여전히 **OSD** 상태가 중지 됩니다. 네트워킹 문제가 실제로 작동 중이더라도 **OSD**를 다운 상태로 표시할 수도 있습니다. 먼저 환경의 네트워크 문제를 확인합니다. 네트워킹이 확인되면 **OSD** 드라이브가 실패할 가능성이 큽니다.

최신 서버는 일반적으로 핫 스왑 가능 드라이브로 배포하므로 오류가 발생한 드라이브를 가져와 노드를 중단하지 않고 새 드라이브로 교체할 수 있습니다. 그러나 **Ceph**를 사용하면 **OSD**의 소프트웨어 정의 부분도 제거해야 합니다.

2.2.1. 실패한 OSD 디스크 교체

OSD를 교체하는 일반적인 절차에는 스토리지 클러스터에서 **OSD**를 제거하고 드라이브를 교체한 다음 **OSD**를 다시 생성해야 합니다.

중요

BlueStore OSD의 데이터베이스 파티션이 포함된 **BlueStore block.db** 디스크를 교체할 때 **Red Hat**은 **Ansible**을 사용하여 모든 **OSD**의 재배포만 지원합니다. 손상된 **block.db** 파일은 해당 **block.db** 파일에 포함된 모든 **OSD**에 영향을 미칩니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.

- 오류가 발생한 디스크.

절차

1. 스토리지 클러스터 상태를 확인합니다.

```
# ceph health
```

2. CRUSH 계층 구조에서 OSD 위치를 식별합니다.

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

3. OSD 노드에서 OSD를 시작합니다.

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

명령에서 OSD가 이미 실행 중임을 나타내는 경우 하트비트 또는 네트워킹 문제가 있을 수 있습니다. OSD를 다시 시작할 수 없는 경우 드라이브가 실패할 수 있습니다.

참고

OSD가 다운 되면 OSD가 결국 표시됩니다. 이는 Ceph Storage의 정상적인 동작입니다. OSD가 표시되지 않으면 실패한 OSD 데이터 복사본이 있는 기타 OSD가 스토리지 클러스터 내에 필요한 복사본 수가 있는지 확인하기 위해 백필을 시작합니다. 스토리지 클러스터가 백필되는 동안 클러스터가 저하된 상태가 됩니다.

4. Ceph의 컨테이너화된 배포의 경우 OSD와 연결된 드라이브를 참조하여 OSD 컨테이너를 시작합니다.

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

명령에서 OSD가 이미 실행 중임을 나타내는 경우 하트비트 또는 네트워킹 문제가 있을 수 있습니다. OSD를 다시 시작할 수 없는 경우 드라이브가 실패할 수 있습니다.

참고

OSD와 연결된 드라이브는 컨테이너 OSD ID를 드라이브에 매핑하여 확인할 수 있습니다.

5. 실패한 OSD 마운트 지점을 확인합니다.

참고

Ceph 컨테이너 배포의 경우 OSD가 다운되고 OSD 드라이브가 마운트 해제되므로 run df 를 실행하여 마운트 지점을 확인할 수 없습니다. 다른 방법을 사용하여 OSD 드라이브가 실패했는지 확인합니다. 예를 들어 컨테이너 노드의 드라이브에서 smartctl 을 실행합니다.

```
# df -h
```

OSD를 다시 시작할 수 없는 경우 마운트 지점을 확인할 수 있습니다. 마운트 지점이 더 이상 나타나지 않으면 OSD 드라이브를 다시 마운트하고 OSD를 다시 시작할 수 있습니다. 마운트 지점을 복원할 수 없는 경우 OSD 드라이브가 실패할 수 있습니다.

smartctl 유틸리티 cab를 사용하면 드라이브가 정상인지 확인합니다. 예를 들면 다음과 같습니다.

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

드라이브가 실패하면 교체해야합니다.

6. OSD 프로세스를 중지합니다.

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

a. FileStore 를 사용하는 경우 저널을 디스크로 플러시합니다.

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

7.

Ceph의 컨테이너화된 배포의 경우 **OSD**와 연결된 드라이브를 참조하여 **OSD** 컨테이너를 중지합니다.

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

8. 스토리지 클러스터에서 **OSD**를 제거합니다.

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

9. 실패한 **OSD**가 다시 입력되었는지 확인합니다.

```
# ceph -w
```

10. **CRUSH** 맵에서 **OSD**를 제거합니다.

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

참고

이 단계는 **OSD**를 영구적으로 제거하고 재배포하지 않는 경우에만 필요합니다.

11. **OSD**의 인증 키를 제거합니다.

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

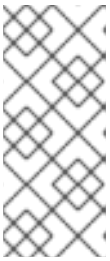
12. **OSD** 키가 나열되지 않았는지 확인합니다.

```
# ceph auth list
```

13. 스토리지 클러스터에서 **OSD**를 제거합니다.

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

14. 실패한 드라이브 경로를 마운트 해제합니다.



참고

Ceph의 컨테이너화된 배포의 경우 **OSD**가 다운되고 **OSD** 드라이브가 마운트 해제됩니다. 이 경우 마운트 해제할 항목이 없으며 이 단계를 건너뛸 수 있습니다.

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

15. 물리적 드라이브를 교체하십시오. 노드의 하드웨어 벤더 설명서를 참조하십시오. 드라이브를 핫 스왑할 수 있는 경우 오류가 발생한 드라이브를 새 드라이브로 바꾸기만 하면 됩니다. 드라이브가 핫 스왑할 수 없고 노드에 여러 **OSD**가 포함된 경우, 노드를 중단하여 실제 드라이브를 교체해야 합니다. 노드를 일시적으로 중단해야 하는 경우 백필을 방지하기 위해 클러스터를 **noout** 으로 설정할 수 있습니다.

```
# ceph osd set noout
```

드라이브를 교체하고 노드와 **OSD**를 다시 온라인 상태가 되면, **noout** 설정을 제거하십시오.

```
# ceph osd unset noout
```

계속 진행하기 전에 새 드라이브가 **/dev/** 디렉터리에 표시되도록 허용하고 드라이브 경로를 기록합니다.

16. **OSD** 드라이브를 찾아서 디스크를 포맷합니다.

17. **OSD**를 다시 생성합니다.

    a. **Ansible** 사용.

    b. **명령줄 인터페이스** 사용.

18. **CRUSH** 계층 구조를 확인하여 정확한지 확인합니다.

```
# ceph osd tree
```

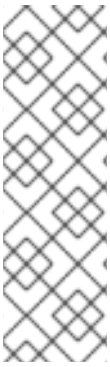
CRUSH 계층 구조의 **OSD** 위치에 만족하지 않으면 **move** 명령으로 이동할 수 있습니다.

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

19. **OSD**가 온라인 상태인지 확인합니다.

2.2.2. OSD ID를 유지하면서 OSD 드라이브 교체

실패한 **OSD** 드라이브를 교체하는 경우 원래 **OSD ID**와 **CRUSH** 맵 항목을 유지할 수 있습니다.



참고

ceph-volume lvm 명령의 기본값은 **OSD**의 **BlueStore**입니다. **FileStore OSD**를 사용하려면 **--filestore,--data** 및 **--journal** 옵션을 사용합니다.

자세한 내용은 **OSD 데이터 준비 및 저널 드라이브** 준비 섹션을 참조하십시오.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 오류가 발생한 디스크.

절차

1. **OSD**를 삭제합니다.

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

예제

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

2. 교체 디스크를 이전에 사용한 경우 선택적으로 디스크를 **zap** 해야 합니다.

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

예제

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

3. 기존 **OSD ID**를 사용하여 새 **OSD**를 생성합니다.

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

예제

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

2.3. 디스크 오류 시뮬레이션

하드 및 소프트의 두 가지 디스크 장애 시나리오가 있습니다. 하드 오류는 디스크를 교체합니다. 소프트 오류는 장치 드라이버 또는 일부 다른 소프트웨어 구성 요소에 문제가 될 수 있습니다.

소프트 오류의 경우 디스크 교체가 필요하지 않을 수 있습니다. 디스크를 교체하는 경우 오류가 발생한 디스크를 제거하고 **Ceph**에 대체 디스크를 추가하려면 단계를 따라야 합니다. 소프트 디스크 오류를 시뮬레이션하기 위해 가장 좋은 방법은 장치를 삭제하는 것입니다. 장치를 선택하고 시스템에서 장치를 삭제합니다.

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

예제

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

Ceph OSD 로그의 OSD 노드에서 Ceph는 오류를 감지하고 복구 프로세스를 자동으로 시작했습니다.

예제

```
[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 디스크 트리를 보면 디스크도 오프라인 상태임을 알 수 있습니다.

```
[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
```

3장. 노드 오류 처리

스토리지 관리자는 스토리지 클러스터 내에서 전체 노드가 실패하고 노드 오류 처리와 디스크 오류 처리와 유사할 수 있습니다. 노드에 오류가 발생하면 Ceph에서 하나의 디스크에 대한 PG(위로 그룹)를 복구하는 대신 해당 노드 내의 디스크에 있는 모든 PG를 복구해야 합니다. Ceph에서 OSD가 모두 다운되었음을 감지하고 자동 복구라는 복구 프로세스를 자동으로 시작합니다.

노드 장애 시나리오는 세 가지입니다. 다음은 노드를 교체할 때 각 시나리오에 대한 상위 수준 워크플로입니다.

- 노드 교체는 하지만 오류가 발생한 노드에서 root 및 Ceph OSD 디스크를 사용합니다.

  1. 백필을 비활성화합니다.
  2. 노드를 교체하여 이전 노드에서 디스크를 가져와 새 노드에 추가합니다.
  3. 백필을 활성화합니다.

- 노드를 교체하고 운영 체제를 다시 설치하고 실패한 노드에서 Ceph OSD 디스크를 사용합니다.

  1. 백필을 비활성화합니다.
  2. Ceph 구성의 백업을 만듭니다.
  3. 노드를 교체하고 실패한 노드에서 Ceph OSD 디스크를 추가합니다.
     a. 디스크를 JBOD로 구성.
  4. 운영 체제를 설치합니다.

5. Ceph 구성을 복원합니다.

6. ceph-ansible 을 실행합니다.

7. 백필을 활성화합니다.

- 노드를 교체하고 운영 체제를 다시 설치하고 모든 새 Ceph OSD 디스크를 사용합니다.

1. 백필을 비활성화합니다.

2. 스토리지 클러스터에서 장애가 발생한 노드의 모든 OSD를 제거합니다.

3. Ceph 구성의 백업을 만듭니다.

4. 노드를 교체하고 실패한 노드에서 Ceph OSD 디스크를 추가합니다.

   a. 디스크를 JBOD로 구성.

5. 운영 체제를 설치합니다.

6. ceph-ansible 을 실행합니다.

7. 백필을 활성화합니다.

3.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.

- 실패한 노드.

3.2. 노드를 추가하거나 제거하기 전에 고려해야 할 사항

Ceph의 뛰어난 기능 중 하나는 런타임에 Ceph OSD 노드를 추가하거나 제거하는 기능입니다. 즉, 스토리지 클러스터를 중단하지 않고도 스토리지 클러스터 용량의 크기를 조정하거나 하드웨어를 교체할 수 있습니다. 클러스터가 저하된 상태에서도 Ceph 클라이언트에 서비스를 제공할 수 있는 기능으로, 예를 들어 과도한 시간이나 주말 작업 대신 정규 영업 시간 동안 하드웨어를 추가하거나 제거하거나 교체할 수 있습니다. 그러나 Ceph OSD 노드를 추가 및 제거하면 성능에 상당한 영향을 미칠 수 있으며, 작업하기 전에 스토리지 클러스터에서 하드웨어를 추가, 제거 또는 교체하는 성능에 미치는 영향을 고려해야 합니다.

용량 관점에서 노드를 제거하면 노드에 포함된 OSD가 제거되고 스토리지 클러스터의 용량을 효과적으로 줄일 수 있습니다. 노드를 추가하면 노드에 포함된 OSD가 추가되고 스토리지 클러스터의 용량을 효과적으로 확장합니다. 스토리지 클러스터 용량을 확장하거나 감소하든 Ceph OSD 노드를 추가하거나 제거하든 클러스터 재조정으로 백필링됩니다. Ceph는 이러한 재조정 기간 동안 스토리지 클러스터 성능에 영향을 미칠 수 있는 추가 리소스를 사용합니다.

각 노드에 OSD 4개가 있는 Ceph 노드가 포함된 스토리지 클러스터를 가정하십시오. 16개의 OSD가 있는 노드 4개로 구성된 스토리지 클러스터에서 노드는 4개의 OSD를 제거하고 용량을 25%로 줄입니다. OSD 12개가 있는 노드 3개로 구성된 스토리지 클러스터에서는 노드를 추가하여 OSD 4개를 추가하고 용량을 33% 늘립니다.

프로덕션 Ceph 스토리지 클러스터에서 Ceph OSD 노드에는 특정 유형의 스토리지 전략을 용이하게 하는 특정 하드웨어 구성이 있습니다. 자세한 내용은 Red Hat Ceph Storage 3에 대한 스토리지 전략 가이드를 참조하십시오.

Ceph OSD 노드는 CRUSH 계층 구조의 일부이므로 노드를 추가하거나 제거하는 경우 일반적으로 CRUSH 규칙 세트를 사용하는 풀의 성능에 영향을 미칩니다.

3.3. 성능 고려 사항

다음과 같은 요인은 일반적으로 Ceph OSD 노드를 추가하거나 제거할 때 스토리지 클러스터의 성능에 영향을 미칩니다.

현재 클라이언트가 영향을 받는 풀에서 로드됩니다.

Ceph 클라이언트는 I/O 인터페이스에서 Ceph에 로드됩니다. 즉, 풀에 로드됩니다. 풀은 CRUSH 규칙 세트에 매핑됩니다. 기본 CRUSH 계층 구조를 통해 Ceph는 장애 도메인에 데이터를 배치할 수 있습니다. 기본 Ceph OSD 노드에 클라이언트 부하가 높은 풀과 관련된 경우 클라이언트 로드는 복구 시간에 큰 영향을 미치고 성능에 영향을 미칠 수 있습니다. 특히 쓰기 작업에는 지속성을 위해 데이터 복제가 필요하므로 쓰기 집약적 클라이언트 로드는 스토리지 클러스터가 복구되는 시간이 늘어납니다.

용량 추가 또는 제거:

일반적으로 전체 클러스터의 백분율로 추가하거나 제거하는 용량은 스토리지 클러스터의 복구 시간에 영향을 미칩니다. 또한 추가 또는 제거 노드의 스토리지 밀도는 복구에 영향을 미칠 수 있습니다. 예를 들어, 36개의 OSD가 있는 노드는 일반적으로 OSD 12개가 있는 노드에 비해 복구하는 데 시간이 더 오래 걸립니다. 노드를 제거할 때 충분한 예비 용량을 확보하여 전체 비율 또는 거의 전체 비율에 도달하지 않도록 해야 합니다. 스토리지 클러스터가 전체 비율에 도달하면 Ceph에서 쓰기 작업을 일시 중지하여 데이터 손실을 방지합니다.

pool 및 CRUSH Ruleset:

Ceph OSD 노드는 하나 이상의 Ceph CRUSH 계층 구조에 매핑되며 계층 구조는 하나 이상의 풀에 매핑됩니다. Ceph OSD 노드를 추가하거나 제거하는 CRUSH 계층(ruleset)을 사용하는 각 풀은 성능에 영향을 미칩니다.

풀 유형 및 안정성:

복제 풀은 더 많은 네트워크 대역폭을 사용하여 데이터의 심층 복사본을 복제하는 경향이 있지만, 이레이저 코딩 풀은 더 많은 CPU를 사용하여 k+m 코딩 청크를 계산하는 경향이 있습니다. 데이터 복사본(예: 크기 또는 더 많은 k+m 청크)이 많을수록 스토리지 클러스터가 복구되는 데 시간이 더 오래 걸립니다.

총 처리량 특성:

드라이브, 컨트롤러 및 네트워크 인터페이스 카드에는 모두 복구 시간에 영향을 줄 수 있는 처리량 특성이 있습니다. 일반적으로 처리량이 높은 특성(예: 10Gbps 및 SSD)이 있는 노드는 처리량이 낮은 특성(예: 1Gbps 및 SATA 드라이브)을 사용하는 노드보다 더 빠르게 복구됩니다.

3.4. 노드 추가 또는 제거 권장 사항

노드 장애로 인해 노드를 변경하기 전에 한 번에 하나의 OSD를 제거할 수 있습니다. Ceph OSD 노드를 추가하거나 제거할 때 부정적인 성능 영향을 줄일 수 있습니다. Red Hat은 노드 내에서 한 번에 하나의 OSD를 추가하거나 제거하고 다음 OSD로 진행하기 전에 클러스터를 복구할 것을 권장합니다. OSD 제거에 대한 자세한 내용은 다음을 참조하십시오.

- Ansible 사용.
- 명령줄 인터페이스 사용.

Ceph 노드를 추가할 때 Red Hat은 한 번에 하나의 OSD를 추가하는 것이 좋습니다. OSD 추가에 대한 자세한 내용은 다음을 참조하십시오.

- Ansible 사용.
- 명령줄 인터페이스 사용.

Ceph OSD 노드를 추가하거나 제거할 때 다른 프로세스도 성능에 미치는 영향을 고려해야 합니다. 클라이언트 I/O에 미치는 영향을 줄이기 위해 Red Hat은 다음을 권장합니다.

용량 계산:

Ceph OSD 노드를 제거하기 전에 스토리지 클러스터가 모든 OSD WITHOUT 의 내용을 전체 비율에 다시 채울 수 있는지 확인합니다. 전체 비율에 도달하면 클러스터에서 쓰기 작업을 거부합니다.

일시적으로 Scrubbing을 비활성화:

스크럽은 스토리지 클러스터 데이터의 내구성을 보장하는 데 필수적이지만 리소스 집약적입니다. Ceph OSD 노드를 추가하거나 제거하기 전에 스크럽 및 깊은 스크럽을 비활성화하고 다음을 진행하기 전에 현재 스크럽 작업을 완료하십시오.

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

Ceph OSD 노드를 추가하거나 제거하고 스토리지 클러스터가 active+clean 상태로 반환되면 noscrub 및 nodeep-scrub 설정을 설정 해제합니다.

Backfill 및 복구 제한:

적절한 데이터 durability(예: osd pool default size = 3 및 osd pool default min size = 2)가 있는 경우 성능이 저하된 상태에서 작동하는 데 아무런 문제가 없습니다. 최대한 빠른 복구 시간을 위해 스토리지 클러스터를 조정할 수 있지만 Ceph 클라이언트 I/O 성능에 큰 영향을 미칩니다. 가장 높은 Ceph 클라이언트 I/O 성능을 유지하려면 백필(backfill) 및 복구 작업을 제한하고 예를 들어 다음과 같이 더 오래 걸릴 수 있습니다.

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

sleep 및 delay 매개변수(예: osd_recovery_sleep)를 설정할 수도 있습니다.

마지막으로 스토리지 클러스터 크기를 확장하는 경우 배치 그룹의 수를 늘려야 할 수 있습니다. 배치 그룹 수를 확장해야 한다고 판단되면 Red Hat은 배치 그룹 수를 점진적으로 늘리는 것이 좋습니다. 배치 그

룹의 수를 크게 늘리면 성능이 크게 저하됩니다.

3.5. CEPH OSD 노드 추가

Red Hat Ceph Storage 클러스터의 용량을 확장하려면 **OSD** 노드를 추가합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 네트워크 연결이 있는 프로비저닝된 노드.
- **Red Hat Enterprise Linux 7** 또는 **Ubuntu 16.04** 설치
- ***Red Hat Enterprise Linux 또는 Ubuntu 설치 가이드의 Red Hat Ceph Storage 설치 요구 사항을*** 검토하십시오.

절차

1. 스토리지 클러스터의 다른 노드가 짧은 호스트 이름으로 새 노드에 연결할 수 있는지 확인합니다.
2. 일시적으로 스크럽을 비활성화합니다.

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

3. 백필 및 복구 기능을 제한합니다.

구문

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

예제

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

4. **CRUSH 맵에 새 노드를 추가합니다.**

구문

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

예제

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

5. **노드의 각 디스크에 OSD를 스토리지 클러스터에 추가합니다.**

- **Ansible 사용.**
- **명령줄 인터페이스 사용.**

중요

Red Hat Ceph Storage 클러스터에 **OSD** 노드를 추가할 때 **Red Hat**은 노드 내에서 한 번에 하나의 **OSD**를 추가하고 다음 **OSD**를 진행하기 전에 활성**+clean** 상태로 클러스터를 복구할 것을 권장합니다.

추가 리소스

- 자세한 내용은 **Red Hat Ceph Storage** 구성 가이드의 런타임 시 특정 구성 설정 섹션을 참조하십시오.

- **CRUSH** 계층 구조의 적절한 위치에 노드를 배치하는 방법에 대한 자세한 내용은 **Red Hat Ceph Storage Storage Strategies** 가이드의 버킷 추가 및 버킷 이동 섹션을 참조하십시오.

3.6. CEPH OSD 노드 제거

스토리지 클러스터의 용량을 줄이기 위해 **OSD** 노드를 제거합니다.

주의

Ceph OSD 노드를 제거하기 전에 스토리지 클러스터가 모든 **OSD WITHOUT**의 내용을 전체 비율에 다시 채울 수 있는지 확인합니다. 전체 비율에 도달하면 클러스터에서 쓰기 작업을 거부합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.

절차

1. 스토리지 클러스터의 용량을 확인합니다.

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

2. 일시적으로 스크럽을 비활성화합니다.

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

3. 백필 및 복구 기능을 제한합니다.

구문

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

예제

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

4. 스토리지 클러스터에서 노드의 각 **OSD**를 제거합니다.

- **Ansible** 사용.
- 명령줄 인터페이스 사용.

중요

스토리지 클러스터에서 **OSD** 노드를 제거할 때 **Red Hat**은 노드 내에서 한 번에 하나의 **OSD**를 제거하고 다음 **OSD**로 진행하기 전에 활성**+clean** 상태로 클러스터를 복구할 것을 권장합니다.

a. **OSD** 검사를 제거한 후 스토리지 클러스터가 거의 전체 비율에 도달하지 않는지 확인합니다.

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

b. 노드의 모든 **OSD**가 스토리지 클러스터에서 제거될 때까지 이 단계를 반복합니다.

5. 모든 **OSD**가 제거되면 **CRUSH** 맵에서 호스트 버킷을 제거합니다.

구문

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

예제

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

추가 리소스

- * 자세한 내용은 **Red Hat Ceph Storage** 구성 가이드의 런타임 시 특정 구성 설정 섹션을 참조하십시오.

3.7. 노드 오류 시뮬레이션

하드 노드 장애를 시뮬레이션하려면 노드의 전원을 끄고 운영 체제를 다시 설치합니다.

사전 요구 사항

- 정상적인 실행 중인 **Red Hat Ceph Storage** 클러스터.

절차

1. 스토리지 용량을 확인하여 스토리지 클러스터에 노드 제거의 의미를 파악합니다.

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

2. 선택적으로 복구 및 백필을 비활성화합니다.

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

3. 노드를 종료합니다.

4. 호스트 이름이 변경되면 **CRUSH** 맵에서 노드를 제거합니다.

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

5. 클러스터 상태를 확인합니다.

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

6. 노드에 운영 체제를 다시 설치합니다.

7. **Ansible** 사용자 및 **SSH** 키를 추가합니다.

```
[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. **관리 노드에서 ansible 사용자의 SSH 키를 복사합니다.**

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

9. **관리 노드에서 Ansible 플레이북을 다시 실행합니다.**

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

출력 예

```
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. **선택적으로 복구 및 백필을 활성화합니다.**

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

11. **Ceph의 상태를 확인합니다.**

```
[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
```

추가 리소스

- **Red Hat Ceph Storage** 설치에 대한 자세한 내용은 다음을 참조하십시오.
  - Red Hat Enterprise Linux
  - 우분투

4장. 데이터 센터 오류 처리

Red Hat Ceph Storage는 확장 클러스터에서 세 개의 데이터 센터 중 하나를 손실하는 등 인프라에 치명적인 실패를 초래할 수 있습니다. 표준 오브젝트 저장소 사용 사례의 경우 세 개의 데이터 센터를 둘 간에 설정한 복제와 독립적으로 구성할 수 있습니다. 이 시나리오에서는 로컬 기능과 종속성을 반영하여 각 데이터 센터의 클러스터 구성이 다를 수 있습니다.

배치 계층 구조의 논리적 구조를 고려해야 합니다. 인프라 내에서 장애 도메인의 계층 구조를 반영하여 적절한 **CRUSH** 맵을 사용할 수 있습니다. 논리적 계층 구조 정의를 사용하면 표준 계층 구조 정의를 사용하는 대신 스토리지 클러스터의 안정성이 향상됩니다. 실패 도메인은 **CRUSH** 맵에 정의되어 있습니다. 기본 **CRUSH** 맵에는 플랫 계층 구조의 모든 노드가 포함됩니다.

3개의 데이터 센터 환경 예에서는 확장 클러스터를 사용하는 노드 배치는 하나의 데이터 센터를 중단할 수 있는 방식으로 관리되어야 하지만 스토리지 클러스터는 가동 상태를 유지해야 합니다. 데이터에 **3-**방향 복제를 사용할 때 노드가 상주하는 장애 도메인의 경우 하나의 데이터 센터에 중단되는 경우 일부 데이터를 한 복사본으로 유지할 수 있습니다. 이 시나리오가 발생하면 다음 두 가지 옵션이 있습니다.

- 표준 설정을 사용하여 데이터를 읽기 전용 상태로 둡니다.
- 정전 기간 동안 단 하나의 사본으로만 라이브.

표준 설정을 사용하고 노드에서 데이터 배치의 임의성 때문에 모든 데이터가 영향을 받는 것은 아니지만 일부 데이터는 하나의 복사본만 가질 수 있으며 스토리지 클러스터는 읽기 전용 모드로 되돌아갑니다.

아래 예제에서 결과 맵은 **6**개의 **OSD** 노드로 클러스터의 초기 설정에서 파생됩니다. 이 예에서 모든 노드에는 디스크가 하나만 있고 **OSD**가 하나만 있습니다. 모든 노드는 기본 *루트*, 즉 계층 구조 트리의 표준 *루트에* 따라 정렬됩니다. **2**개의 **OSD**에 할당된 가중치가 있으므로 이러한 **OSD**에는 다른 **OSD**보다 더 적은 데이터 청크가 부여됩니다. 이러한 노드는 초기 **OSD** 디스크보다 큰 디스크가 나중에 도입되었습니다. 이는 노드 그룹의 실패를 견디도록 데이터 배치에는 영향을 미치지 않습니다.

표준 CRUSH 맵

```
$ 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
```

논리 계층 정의를 사용하여 노드를 동일한 데이터 센터로 그룹화하면 데이터 배치 완성도를 달성할 수 있습니다. *루트*, *데이터센터*, *랙*, *행* 및 *호스트의* 가능한 정의 유형을 사용하면 세 개의 데이터 센터에 대한 장애 도메인을 반영할 수 있습니다.

- **ceph-node1** 및 **ceph-node2** 노드가 데이터 센터 **1(DC1)**에 상주합니다.
- 노드 **ceph-node3** 및 **ceph-node5**는 **DC2(**데이터 센터 **2)**에 상주합니다.
- **ceph-node4** 및 **ceph-node6** 노드가 데이터센터 **3(DC3)**에 상주합니다.
- 모든 데이터 센터는 동일한 구조(모든 **DC)**에 속합니다.

호스트의 모든 **OSD**가 호스트 정의에 속해 있으므로 변경할 필요가 없습니다. 스토리지 클러스터 런타임 중에 다음을 통해 다른 모든 할당을 조정할 수 있습니다.

- 다음 명령을 사용하여 *버킷* 구조를 정의합니다.

```
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** 맵을 수정하여 이 구조 내에서 노드를 적절한 위치로 이동합니다.

```
ceph osd crush move DC1 root=allDC
ceph osd crush move DC2 root=allDC
ceph osd crush move DC3 root=allDC
```

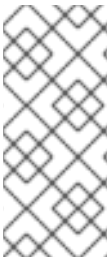
```
ceph osd crush move ceph-node1 datacenter=DC1
ceph osd crush move ceph-node2 datacenter=DC1
ceph osd crush move ceph-node3 datacenter=DC2
ceph osd crush move ceph-node5 datacenter=DC2
ceph osd crush move ceph-node4 datacenter=DC3
ceph osd crush move ceph-node6 datacenter=DC3
```

이 구조 내에서 새 호스트도 추가할 수 있으며 새 디스크도 추가할 수 있습니다. 계층 구조에서 **OSD**를 올바른 위치에 배치하면 **CRUSH** 알고리즘은 구조 내의 다른 장애 도메인에 중복 조각을 배치하도록 변경됩니다.

위 예제는 다음과 같습니다.

```
$ 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
```

위의 목록은 **osd** 트리를 표시하여 결과 **CRUSH** 맵을 보여줍니다. 이제 호스트가 데이터 센터에 속한 방법 및 모든 데이터 센터가 동일한 최상위 수준에 속하지만 위치를 명확하게 구분할 수 있습니다.



참고

맵에 따라 적절한 위치에 데이터를 배치하면 정상 클러스터 내에서만 올바르게 작동합니다. **Misplacement**는 일부 **OSD**를 사용할 수 없는 경우 경우에 따라 발생할 수 있습니다. 이러한 변경 사항은 가능한 한 자동으로 수정됩니다.

추가 리소스

- 자세한 내용은 **Red Hat Ceph Storage** 전략 가이드의 **CRUSH** 관리 장을 참조하십시오.