Red Hat Ceph Storage 5

관리 가이드

Red Hat Ceph Storage 관리

Last Updated: 2022-08-14

Red Hat Ceph Storage 5 관리 가이드

Red Hat Ceph Storage 관리

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

법적 공지

초록

이 문서에서는 프로세스를 관리하고, 클러스터 상태를 모니터링하며, 사용자를 관리하며, Red Hat Ceph Storage 데몬을 추가 및 제거하는 방법을 설명합니다. Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 CTO Chris Wright의 메시지를 참조하십시오.

차례

1장. CEPH 관리

Red Hat Ceph Storage 클러스터는 모든 Ceph 배포를 위한 기반입니다. Red Hat Ceph Storage 클러스터를 배포한 후 Red Hat Ceph Storage 클러스터를 정상 상태로 유지하고 최적으로 실행하기 위한 관리 작업이 있습니다.

Red Hat Ceph Storage Administration Guide는 스토리지 관리자가 다음과 같은 작업을 수행할 수 있도록 지원합니다.

- Red Hat Ceph Storage 클러스터의 상태를 어떻게 확인합니까?
- Red Hat Ceph Storage 클러스터 서비스를 시작하고 중지하려면 어떻게 해야 합니까?
- 실행 중인 Red Hat Ceph Storage 클러스터에서 OSD를 추가하거나 제거하려면 어떻게 해야 합니까?
- Red Hat Ceph Storage 클러스터에 저장된 개체에 대한 사용자 인증 및 액세스 제어를 관리하려면 어떻게 해야 합니까?
- Red Hat Ceph Storage 클러스터에서 재정의를 사용하는 방법을 이해하고 싶습니다.
- Red Hat Ceph Storage 클러스터의 성능을 모니터링하려고 합니다.

기본 Ceph 스토리지 클러스터는 다음 두 가지 유형의 데몬으로 구성됩니다.

- Ceph OSD(오브젝트 스토리지 장치)는 데이터를 OSD에 할당된 배치 그룹 내의 오브젝트로 저장합니다.
- Ceph Monitor는 클러스터 맵의 마스터 사본을 유지 관리합니다.

프로덕션 시스템에는 고가용성을 위해 3개 이상의 Ceph 모니터가 있으며 일반적으로 허용 가능한 로드 밸런싱, 데이터 재조정 및 데이터 복구를 위해 최소 50개의 OSD가 있습니다.

추가 리소스

- Red Hat Ceph Storage 설치 가이드

2장. CEPH를 위한 프로세스 관리 이해

스토리지 관리자는 Red Hat Ceph Storage 클러스터에서 유형 또는 인스턴스를 통해 다양한 Ceph 데몬을 조작할 수 있습니다. 이러한 데몬을 조작하면 필요에 따라 모든 Ceph 서비스를 시작, 중지 및 다시 시작할 수 있습니다.

2.1. 사전 요구 사항

- Red Hat Ceph Storage 소프트웨어 설치.

2.2. CEPH 프로세스 관리

Red Hat Ceph Storage에서는 Systemd 서비스를 통해 모든 프로세스 관리를 수행합니다. Ceph 데몬을 **시작**,**재시작** 및 **중지** 하려는 때마다 데몬 유형 또는 데몬 인스턴스를 지정해야 합니다.

추가 리소스

- systemd 사용에 대한 자세한 내용은 Red Hat Enterprise Linux 8용 *기본 시스템 설정 구성 가이드의 systemd 장과 systemctl을 사용하여 시스템서비스 관리* 장을 참조하십시오.

2.3. 모든 CEPH 데몬 시작, 중지 및 재시작

Ceph 데몬을 중지하려는 호스트에서 모든 Ceph 데몬을 root 사용자로 시작, 중지 및 다시 시작할 수 있습니다.

사전 요구 사항

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

절차

1. 데몬을 시작, 중지 및 다시 시작하려는 호스트에서 systemctl 서비스를 실행하여 서비스의 *SERVICE_ID* 를 가져옵니다.

   예제

   ```
   [root@host01 ~]# systemctl --type=service
   ceph-499829b4-832f-11eb-8d6d-001a4a000635@mon.host01.service
   ```

2. 모든 Ceph 데몬을 시작합니다.

   구문

   ```
   systemctl start SERVICE_ID
   ```

   예제

   ```
   [root@host01 ~]# systemctl start ceph-499829b4-832f-11eb-8d6d-
   001a4a000635@mon.host01.service
   ```

3. 모든 Ceph 데몬 중지:

구문

```
systemctl stop SERVICE_ID
```

예제

```
[root@host01 ~]# systemctl stop ceph-499829b4-832f-11eb-8d6d-
001a4a000635@mon.host01.service
```

4. 모든 Ceph 데몬을 다시 시작하십시오.

구문

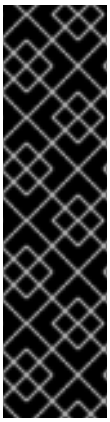
```
systemctl restart SERVICE_ID
```

예제

```
[root@host01 ~]# systemctl restart ceph-499829b4-832f-11eb-8d6d-
001a4a000635@mon.host01.service
```

2.4. 모든 CEPH 서비스 시작, 중지 및 다시 시작

Ceph 서비스는 동일한 Red Hat Ceph Storage 클러스터에서 실행되도록 구성된 동일한 유형의 Ceph 데몬의 논리 그룹입니다. Ceph의 오케스트레이션 계층을 사용하면 이러한 서비스를 중앙 집중식으로 관리할 수 있으므로 동일한 논리 서비스에 속하는 모든 Ceph 데몬에 영향을 주는 작업을 쉽게 실행할 수 있습니다. 각 호스트에서 실행 중인 Ceph 데몬은 Systemd 서비스를 통해 관리됩니다. Ceph 서비스를 관리하려는 호스트에서 모든 Ceph 서비스를 시작, 중지 및 다시 시작할 수 있습니다.



중요

특정 호스트에서 특정 Ceph 데몬을 시작, 중지 또는 다시 시작하려면 SystemD 서비스를 사용해야 합니다. 특정 호스트에서 실행 중인 SystemD 서비스 목록을 가져오려면 호스트에 연결하고 다음 명령을 실행합니다.

예제

```
[root@host01 ~]# systemctl list-units “ceph*”
```

출력은 각 Ceph 데몬을 관리하는 데 사용할 수 있는 서비스 이름 목록을 제공합니다.

사전 요구 사항

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

절차

1. Cephadm 쉘에 로그인합니다.

예제

```
[root@host01 ~]# cephadm shell
```

2. **ceph orch ls** 명령을 실행하여 Red Hat Ceph Storage 클러스터에 구성된 Ceph 서비스 목록을 가져오고 특정 서비스 ID를 가져옵니다.

예제

```
[ceph: root@host01 /]# ceph orch ls
NAME                       RUNNING  REFRESHED  AGE  PLACEMENT  IMAGE NAME
IMAGE ID
alertmanager               1/1  4m ago     4M   count:1    registry.redhat.io/openshift4/ose-
prometheus-alertmanager:v4.5   b7bae610cd46
crash                      3/3  4m ago     4M   *          registry.redhat.io/rhceph-alpha/rhceph-5-
rhel8:latest           c88a5d60f510
grafana                    1/1  4m ago     4M   count:1    registry.redhat.io/rhceph-alpha/rhceph-5-
dashboard-rhel8:latest  bd3d7748747b
mgr                        2/2  4m ago     4M   count:2    registry.redhat.io/rhceph-alpha/rhceph-5-
rhel8:latest           c88a5d60f510
mon                        2/2  4m ago     10w  count:2    registry.redhat.io/rhceph-alpha/rhceph-5-
rhel8:latest           c88a5d60f510
nfs.foo                    0/1  -          -    count:1    <unknown>
<unknown>
node-exporter              1/3  4m ago     4M   *          registry.redhat.io/openshift4/ose-
prometheus-node-exporter:v4.5  mix
osd.all-available-devices  5/5  4m ago     3M   *          registry.redhat.io/rhceph-
alpha/rhceph-5-rhel8:latest            c88a5d60f510
prometheus                 1/1  4m ago     4M   count:1    registry.redhat.io/openshift4/ose-
prometheus:v4.6            bebb0ddef7f0
rgw.test_realm.test_zone   2/2  4m ago     3M   count:2    registry.redhat.io/rhceph-
alpha/rhceph-5-rhel8:latest            c88a5d60f510
```

3. 특정 서비스를 시작하려면 다음 명령을 실행합니다.

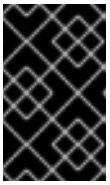
구문

```
ceph orch start SERVICE_ID
```

예제

```
[ceph: root@host01 /]# ceph orch start node-exporter
```

4. 특정 서비스를 중지하려면 다음 명령을 실행합니다.



중요

ceph orch stop SERVICE_ID 명령을 실행하면 MON 및 MGR 서비스에 대해서만 Red Hat Ceph Storage 클러스터에 액세스할 수 없습니다. **systemctl stop SERVICE_ID** 명령을 사용하여 호스트의 특정 데몬을 중지하는 것이 좋습니다.

구문

```
ceph orch stop SERVICE_ID
```

예제

```
[ceph: root@host01 /]# ceph orch stop node-exporter
```

예제에서 **ceph orch stop node-exporter** 명령은 **노드 내보내기** 서비스의 모든 데몬을 제거합니다.

5. 특정 서비스를 다시 시작하려면 다음 명령을 실행합니다.

구문

```
ceph orch restart SERVICE_ID
```

예제

```
[ceph: root@host01 /]# ceph orch restart node-exporter
```

2.5. 컨테이너에서 실행되는 CEPH 데몬의 로그 파일 보기

컨테이너 호스트의 **journald** 데몬을 사용하여 컨테이너에서 Ceph 데몬의 로그 파일을 확인합니다.

사전 요구 사항

- Red Hat Ceph Storage 소프트웨어 설치.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. 전체 Ceph 로그 파일을 보려면 다음 형식으로 구성된 **root** 로 **journalctl** 명령을 실행합니다.

구문

```
journalctl -u ceph SERVICE_ID
```

```
[root@host01 ~]# journalctl -u ceph-499829b4-832f-11eb-8d6d-
001a4a000635@osd.8.service
```

위의 예제에서는 ID가 **osd.8** 인 OSD의 전체 로그를 볼 수 있습니다.

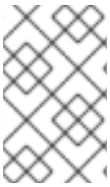
2. 최근 저널 항목만 표시하려면 **-f** 옵션을 사용합니다.

구문

```
journalctl -fu SERVICE_ID
```

예제

```
[root@host01 ~]# journalctl -fu ceph-499829b4-832f-11eb-8d6d-
001a4a000635@osd.8.service
```



참고

sosreport 유틸리티를 사용하여 **journald** 로그를 볼 수도 있습니다. SOS 보고서에 대한 자세한 내용은 sosreport 및 Red Hat Enterprise Linux에서 새로 만드는 방법을 참조하십시오. Red Hat 고객 포털 기반의 솔루션.

추가 리소스

- **journalctl** 도움말 페이지.

2.6. RED HAT CEPH STORAGE 클러스터 전원 끄기 및 재부팅

Ceph 클러스터의 전원을 끄고 재부팅하려면 다음 절차를 따르십시오.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- **루트** 액세스 권한이 있어야 합니다.

절차

Red Hat Ceph Storage 클러스터 전원 끄기

1. 클라이언트가 이 클러스터 및 기타 클라이언트에서 RBD 이미지 및 RADOS 게이트웨이를 사용하지 않도록 중지합니다.
2. 계속하기 전에 클러스터가 정상 상태(**Health_OK** 및 모든 PGs **active+clean**)여야 합니다. 클라이언트 키(예: Ceph Monitor 또는 OpenStack 컨트롤러 노드)가 있는 호스트에서 **ceph 상태를** 실행하여 클러스터가 정상인지 확인합니다.
3. Ceph 파일 시스템(**CephFS**)을 사용하는 경우 **CephFS** 클러스터를 종료해야 합니다.

   구문

   ```
   ceph fs set FS_NAME max_mds 1
   ceph fs fail FS_NAME
   ceph status
   ceph fs set FS_NAME joinable false
   ```

   예제

   ```
   [ceph: root@host01 /]# ceph fs set cephfs max_mds 1
   [ceph: root@host01 /]# ceph fs fail cephfs
   [ceph: root@host01 /]# ceph status
   [ceph: root@host01 /]# ceph fs set cephfs joinable false
   ```

4. **noout**,**norecover**,**norebalance**,**nobackfill**,**nodown** 및 **pause** 플래그를 설정합니다. 클라이언트 인증 키를 사용하여 노드에서 다음을 실행합니다. 예를 들어 Ceph Monitor 또는 OpenStack 컨트롤러 노드는 다음과 같습니다.

예제

```
[ceph: root@host01 /]# ceph osd set noout
[ceph: root@host01 /]# ceph osd set norecover
[ceph: root@host01 /]# ceph osd set norebalance
[ceph: root@host01 /]# ceph osd set nobackfill
[ceph: root@host01 /]# ceph osd set nodown
[ceph: root@host01 /]# ceph osd set pause
```

5. MDS 및 RGW Ceph Object Gateway 노드가 자체 전용 노드에 있는 경우 전원을 끕니다.

6. OSD 노드를 하나씩 종료합니다.

예제

```
[root@host01 ~]# systemctl stop ceph-499829b4-832f-11eb-8d6d-
001a4a000635@osd.2.service
```

7. 모니터 노드를 하나씩 종료합니다.

예제

```
[root@host01 ~]#  systemctl stop ceph-499829b4-832f-11eb-8d6d-
001a4a000635@mon.host01.service
```

8. admin 노드를 종료합니다.

Red Hat Ceph Storage 클러스터 재부팅

1. 네트워크 장비가 관련된 경우 Ceph 호스트 또는 노드의 전원을 켜기 전에 전원이 켜져 있고 안정적인지 확인하십시오.

2. 관리 노드의 전원을 켭니다.

3. 모니터 노드의 전원을 켭니다.

예제

```
[root@host01 ~]# systemctl start ceph-499829b4-832f-11eb-8d6d-
001a4a000635@mon.host01.service
```

4. OSD 노드의 전원을 켭니다.

예제

```
[root@host01 ~]# systemctl start ceph-499829b4-832f-11eb-8d6d-
001a4a000635@osd.2.service
```

5. 모든 노드가 나타날 때까지 기다립니다. 모든 서비스가 작동 중이며 노드 간에 연결이 제대로 작동하는지 확인합니다.

6. **noout**,**norecover**,**norebalance**,**nobackfill**,**nodown** 및 **pause** 플래그를 설정 해제합니다. 클라이언트 인증 키를 사용하여 노드에서 다음을 실행합니다. 예를 들어 Ceph Monitor 또는 OpenStack 컨트롤러 노드는 다음과 같습니다.

예제

```
[ceph: root@host01 /]# ceph osd unset noout
[ceph: root@host01 /]# ceph osd unset norecover
[ceph: root@host01 /]# ceph osd unset norebalance
[ceph: root@host01 /]# ceph osd unset nobackfill
[ceph: root@host01 /]# ceph osd unset nodown
[ceph: root@host01 /]# ceph osd unset pause
```

7. Ceph 파일 시스템(**CephFS**)을 사용하는 경우 **조인 가능한** 플래그를 **true** 로 설정하여 **CephFS** 클러스터를 다시 가져와야 합니다.

   구문

   ```
   ceph fs set FS_NAME joinable true
   ```

   예제

   ```
   [ceph: root@host01 /]# ceph fs set cephfs joinable true
   ```

8. 클러스터가 정상 상태인지 확인합니다(**Health_OK** 및 모든 PGs **active+clean**). 클라이언트 인증 키가 있는 노드에서 **ceph 상태를** 실행합니다. 예를 들어 클러스터가 정상인지 확인하기 위해 Ceph Monitor 또는 OpenStack 컨트롤러 노드

추가 리소스

- Ceph 설치에 대한 자세한 내용은 *Red Hat Ceph Storage 설치 가이드* 를 참조하십시오.

3장. CEPH 스토리지 클러스터 모니터링

스토리지 관리자는 Ceph의 개별 구성 요소의 상태를 모니터링하고 Red Hat Ceph Storage 클러스터의 전반적인 상태를 모니터링할 수 있습니다.

Red Hat Ceph Storage 클러스터를 실행 중이면 스토리지 클러스터 모니터링을 시작하여 Ceph Monitor 및 Ceph OSD 데몬이 실행 중인지 확인할 수 있습니다. Ceph 스토리지 클러스터 클라이언트는 Ceph 모니터에 연결하고 최신 버전의 스토리지 클러스터 맵을 수신하므로 스토리지 클러스터 내의 Ceph 풀에 데이터를 읽고 쓸 수 있습니다. 따라서 Ceph 클라이언트가 데이터를 읽고 쓸 수 있으려면 모니터 클러스터가 클러스터 상태에 대해 합의해야 합니다.

Ceph OSD는 보조 OSD에서 배치 그룹 복사본과 기본 OSD의 배치 그룹을 피어링해야 합니다. 오류가 발생하면 피어링은 **활성 + 클린** 상태 이외의 다른 상태를 반영합니다.

3.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.

3.2. CEPH 스토리지 클러스터의 고급 모니터링

스토리지 관리자는 Ceph 데몬의 상태를 모니터링하여 실행 중인지 확인할 수 있습니다. 또한 높은 수준의 모니터링에는 스토리지 클러스터가 **전체 비율** 을 초과하지 않도록 스토리지 클러스터 용량을 확인하는 작업이 포함됩니다. Red Hat Ceph Storage 대시보드 는 고급 모니터링을 수행하는 가장 일반적인 방법입니다. 그러나 명령줄 인터페이스, Ceph 관리자 소켓 또는 Ceph API를 사용하여 스토리지 클러스터를 모니터링할 수도 있습니다.

3.2.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.

3.2.2. Ceph 명령 인터페이스 사용

ceph 명령줄 유틸리티를 사용하여 Ceph 스토리지 클러스터와 대화형으로 상호 작용할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. 를 사용하여 **ceph** 유틸리티를 대화형 모드로 실행합니다.

   구문

   ```
   podman exec -it ceph-mon-MONITOR_NAME /bin/bash
   ```

   replace

   - **podman ps** 명령을 실행하여 찾은 Ceph Monitor 컨테이너의 이름이 있는 *MONITOR_NAME* 입니다.

예제

```
[root@host01 ~]# podman exec -it ceph-499829b4-832f-11eb-8d6d-001a4a000635-mon.host01 /bin/bash
```

이 예제에서는 **mon.host01** 에서 대화형 터미널 세션을 엽니다. 여기에서 Ceph 대화형 쉘을 시작할 수 있습니다.

3.2.3. 스토리지 클러스터 상태 확인

Ceph 스토리지 클러스터를 시작하고 데이터를 읽거나 쓰기 전에 스토리지 클러스터의 상태를 먼저 확인합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. Cephadm 쉘에 로그인합니다.

   예제

   ```
   root@host01 ~]# cephadm shell
   ```

2. 다음 명령을 사용하여 Ceph 스토리지 클러스터의 상태를 확인할 수 있습니다.

   예제

   ```
   [ceph: root@host01 /]# ceph health
   HEALTH_OK
   ```

3. **ceph status** 명령을 실행하여 Ceph 스토리지 클러스터의 상태를 확인할 수 있습니다.

   예제

   ```
   [ceph: root@host01 /]# ceph status
   ```

   출력은 다음과 같은 정보를 제공합니다.

   - 클러스터 ID
   - 클러스터 상태
   - 모니터 맵 epoch 및 모니터 쿼럼의 상태.
   - OSD는 epoch 및 OSD의 상태를 매핑합니다.
   - Ceph Manager의 상태입니다.
   - 오브젝트 게이트웨이의 상태입니다.

- 배치 그룹 맵 버전.
- 배치 그룹 및 풀 수입니다.
- 저장된 데이터 양과 저장된 오브젝트 수입니다.
- 저장된 총 데이터 양입니다.
  Ceph 클러스터를 시작하면 **HEALTH_WARN XXX 배치 그룹이 오래된** 상태 경고가 발생할 수 있습니다. 잠시 기다렸다가 다시 확인하십시오. 스토리지 클러스터가 준비되면 **ceph 상태가 HEALTH _OK** 와 같은 메시지를 반환해야 합니다. 이 시점에서 클러스터 사용을 시작하는 것이 좋습니다.

3.2.4. 스토리지 클러스터 이벤트 감시

명령줄 인터페이스를 사용하여 Ceph 스토리지 클러스터에서 발생하는 이벤트를 확인할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. Cephadm 쉘에 로그인합니다.

   예제

   ```
   root@host01 ~]# cephadm shell
   ```

2. 클러스터의 진행 중인 이벤트를 조사하려면 다음 명령을 실행합니다.

   예제

   ```
   [ceph: root@host01 /]# ceph -w
    cluster:
      id:     8c9b0072-67ca-11eb-af06-001a4a0002a0
      health: HEALTH_OK

    services:
      mon: 2 daemons, quorum Ceph5-2,Ceph5-adm (age 3d)
      mgr: Ceph5-1.nqikfh(active, since 3w), standbys: Ceph5-adm.meckej
      osd: 5 osds: 5 up (since 2d), 5 in (since 8w)
      rgw: 2 daemons active (test_realm.test_zone.Ceph5-2.bfdwcn,
   test_realm.test_zone.Ceph5-adm.acndrh)

    data:
      pools:   11 pools, 273 pgs
      objects: 459 objects, 32 KiB
      usage:   2.6 GiB used, 72 GiB / 75 GiB avail
      pgs:     273 active+clean

    io:
      client:   170 B/s rd, 730 KiB/s wr, 0 op/s rd, 729 op/s wr
   ```

```
2021-06-02 15:45:21.655871 osd.0 [INF] 17.71 deep-scrub ok
2021-06-02 15:45:47.880608 osd.1 [INF] 1.0 scrub ok
2021-06-02 15:45:48.865375 osd.1 [INF] 1.3 scrub ok
2021-06-02 15:45:50.866479 osd.1 [INF] 1.4 scrub ok
2021-06-02 15:45:01.345821 mon.0 [INF] pgmap v41339: 952 pgs: 952 active+clean; 17130
MB data, 115 GB used, 167 GB / 297 GB avail
2021-06-02 15:45:05.718640 mon.0 [INF] pgmap v41340: 952 pgs: 1
active+clean+scrubbing+deep, 951 active+clean; 17130 MB data, 115 GB used, 167 GB /
297 GB avail
2021-06-02 15:45:53.997726 osd.1 [INF] 1.5 scrub ok
2021-06-02 15:45:06.734270 mon.0 [INF] pgmap v41341: 952 pgs: 1
active+clean+scrubbing+deep, 951 active+clean; 17130 MB data, 115 GB used, 167 GB /
297 GB avail
2021-06-02 15:45:15.722456 mon.0 [INF] pgmap v41342: 952 pgs: 952 active+clean; 17130
MB data, 115 GB used, 167 GB / 297 GB avail
2021-06-02 15:46:06.836430 osd.0 [INF] 17.75 deep-scrub ok
2021-06-02 15:45:55.720929 mon.0 [INF] pgmap v41343: 952 pgs: 1
active+clean+scrubbing+deep, 951 active+clean; 17130 MB data, 115 GB used, 167 GB /
297 GB avail
```

3.2.5. Ceph에서 데이터 사용량을 계산하는 방법

사용된 값은 사용된 *실제* 원시 스토리지의 양을 반영합니다. **xxx GB / xxx GB** 값은 클러스터의 전체 스토리지 용량 중 두 숫자의 사용 가능한 양을 의미합니다. 알림 번호는 저장된 데이터의 크기를 복제, 복제 또는 스냅샷하기 전에 반영합니다. 따라서 Ceph에서 데이터의 복제본을 만들고 복제 및 스냅샷을 위해 스토리지 용량을 사용할 수 있기 때문에 실제로 저장된 데이터 양이 실제로 저장된 데이터 양을 초과합니다.

3.2.6. 스토리지 클러스터 사용량 통계 이해

풀 간 클러스터의 데이터 사용량 및 데이터 배포를 확인하려면 **df** 옵션을 사용합니다. 이는 Linux **df** 명령과 유사합니다. **ceph df** 명령 또는 **ceph df 세부 정보** 명령을 실행할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph df
RAW STORAGE:
    CLASS     SIZE       AVAIL      USED       RAW USED     %RAW USED
    hdd       90 GiB     84 GiB     100 MiB      6.1 GiB          6.78
    TOTAL     90 GiB     84 GiB     100 MiB      6.1 GiB          6.78

POOLS:
    POOL                    ID     STORED      OBJECTS     USED       %USED     MAX AVAIL
    .rgw.root               1     1.3 KiB           4     768 KiB         0        26 GiB
    default.rgw.control     2         0 B           8         0 B         0        26 GiB
    default.rgw.meta        3     2.5 KiB          12     2.1 MiB         0        26 GiB
    default.rgw.log         4     3.5 KiB         208     6.2 MiB         0        26 GiB
    default.rgw.buckets.index     5     2.4 KiB          33     2.4 KiB         0        26 GiB
    default.rgw.buckets.data      6     9.6 KiB          15     1.7 MiB         0        26 GiB
    testpool                10       231 B           5     384 KiB         0        40 GiB
```

ceph df detail 명령은 할당량 오브젝트, 할당량 바이트, 사용된 압축 및 압축과 같은 기타 풀 통계에 대한 세부 정보를 제공합니다.

예제

-

```
[ceph: root@host01 /]# ceph df detail
RAW STORAGE:
    CLASS     SIZE       AVAIL      USED       RAW USED     %RAW USED
    hdd       90 GiB     84 GiB     100 MiB      6.1 GiB          6.78
    TOTAL     90 GiB     84 GiB     100 MiB      6.1 GiB          6.78

POOLS:
    POOL                    ID     STORED      OBJECTS     USED       %USED     MAX AVAIL
QUOTA OBJECTS     QUOTA BYTES     DIRTY     USED COMPR     UNDER COMPR
    .rgw.root                1     1.3 KiB          4     768 KiB        0       26 GiB     N/A            N/A
4           0 B           0 B
    default.rgw.control      2         0 B          8         0 B        0       26 GiB     N/A            N/A
8           0 B           0 B
    default.rgw.meta         3     2.5 KiB         12     2.1 MiB        0       26 GiB     N/A            N/A
12          0 B           0 B
    default.rgw.log          4     3.5 KiB        208     6.2 MiB        0       26 GiB     N/A            N/A
208         0 B           0 B
    default.rgw.buckets.index      5     2.4 KiB         33     2.4 KiB        0       26 GiB     N/A            N/A
33          0 B           0 B
    default.rgw.buckets.data       6     9.6 KiB         15     1.7 MiB        0       26 GiB     N/A            N/A
15          0 B           0 B
    testpool                10       231 B          5     384 KiB        0       40 GiB     N/A            N/A
5           0 B           0 B
```

출력의 **RAW STORAGE** 섹션에서는 스토리지 클러스터가 데이터에 사용하는 스토리지 양에 대한 개요를 제공합니다.

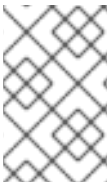
- **CLASS:** 사용되는 장치의 유형입니다.
- **SIZE:** 스토리지 클러스터에서 관리하는 전체 스토리지 용량입니다.
  위의 예에서 **SIZE** 가 90GiB이면 복제 요소가 없는 총 크기이며 이는 기본적으로 3입니다. 복제 인수가 있는 사용 가능한 총 용량은 90GiB/3 = 30GiB입니다. 기본적으로 0.85%인 전체 비율에 따라 사용 가능한 최대 공간은 30GiB * 0.85 = 25.5GiB입니다.
- **AVAIL:** 스토리지 클러스터에서 사용 가능한 여유 공간의 크기입니다.
  위의 예에서 **SIZE** 가 90GiB이고 **USED** 공간이 6GiB이면 **AVAIL** 공간은 84GiB입니다. 복제 인수가 있는 사용 가능한 총 공간(기본적으로 3개)은 84GiB/3 = 28GiB입니다.
- **USED:** 사용자 데이터, 내부 오버헤드 또는 예약된 용량에서 사용하는 스토리지 클러스터에서 사용된 공간의 양입니다.
  위의 예에서 100MiB는 복제 요소를 고려한 후 사용 가능한 총 공간입니다. 사용 가능한 실제 크기는 33MiB입니다.
- **RAW USED: USED** 공간 합계와 **db** 및 **wal** BlueStore 파티션이 할당된 공간.
- **% RAW USED: RAW USED** 의 백분율입니다. 스토리지 클러스터의 용량에 도달하지 않도록 이 수를 **전체 비율 과 거의 전체 비율** 과 함께 사용하십시오.

출력의 **POOLS** 섹션에서는 풀 목록과 각 풀의 주요 사용량을 제공합니다. 이 섹션의 출력 **은** 복제본, 복제 또는 스냅샷을 반영하지 않습니다. 예를 들어 1MB의 데이터가 있는 오브젝트를 저장하는 경우 알림 사용은 1MB이지만 실제 사용은 복제본 수(예: **size = 3**, clone 및 snapshots)에 따라 3MB 이상일 수 있습니다.

- **POOL:** 풀의 이름입니다.
- **id:** 풀 ID입니다.

- **STORED:** 풀에 사용자가 저장한 데이터의 실제 양입니다.
- **OBJECTS:** 풀당 저장된 오브젝트의 알림 수입니다. 이는 **STORED** 크기 * 복제 요소입니다.
- **USED:** 메가바이트 또는 **G** 에 대해 **M** 을 추가하지 않는 한 킬로바이트에 저장된 데이터의 기본 용량입니다.
- **%USED:** 풀당 사용되는 스토리지의 개념 백분율입니다.
- **MAX AVAIL:** 이 풀에 기록될 수 있는 개념적 양의 데이터를 추정합니다. 첫 번째 OSD가 가득 차기 전에 사용할 수 있는 데이터의 양입니다. CRUSH 맵의 디스크 전체에 데이터가 예상되는 것으로 간주되며 첫 번째 OSD를 사용하여 대상으로 채워집니다.
  위의 예에서 **MAX AVAIL** 은 기본적으로 세 가지 복제 요소를 고려하지 않고 153.85MB입니다.

  MAX AVAIL의 값을 계산하기 *위해 단순 복제 풀의 경우 ceph df* ***MAX AVAIL*** *이라는* Red Hat 지식베이스 문서를 참조하십시오.
- **QUOTA OBJECTS:** 할당량 오브젝트 수입니다.
- **QUOTA BYTES:** 할당량 오브젝트의 바이트 수입니다.
- **USED COMPR:** 압축된 데이터, 할당, 복제 및 삭제 코딩 오버헤드를 포함하여 압축 데이터를 위해 할당된 공간의 양입니다.
- **UNDER COMPR:** 압축을 통해 전달되는 데이터의 양과 압축된 형식으로 저장할 수 있을 만큼 유용합니다.



참고

POOLS 섹션의 숫자는 중요하지 않습니다. 복제본, 스냅샷 또는 복제 수가 포함되지 않습니다. 결과적으로 **USED** 및 **%USED** 수량의 합계는 출력의 **GLOBAL** 섹션에 **RAW USED** 및 **%RAW USED** 용량에 추가되지 않습니다.

참고

MAX AVAIL 값은 사용된 복제 또는 삭제 코드의 복잡한 기능이며, 스토리지를 장치에 매핑하는 CRUSH 규칙, 구성된 **mon_osd_full_ratio**.

추가 리소스

- 자세한 내용은 Ceph에서 데이터 사용량을 계산하는 방법을 참조하십시오.
- 자세한 내용은 OSD 사용량 통계 이해를 참조하십시오.

3.2.7. OSD 사용량 통계 이해

ceph osd df 명령을 사용하여 OSD 사용률 통계를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd df
ID CLASS WEIGHT  REWEIGHT SIZE    USE     DATA    OMAP    META    AVAIL   %USE VAR
PGS
 3   hdd 0.90959  1.00000  931GiB 70.1GiB 69.1GiB      0B    1GiB  861GiB 7.53 2.93  66
```

```
 4   hdd 0.90959  1.00000  931GiB 1.30GiB  308MiB     0B    1GiB  930GiB 0.14 0.05  59
 0   hdd 0.90959  1.00000  931GiB 18.1GiB 17.1GiB     0B    1GiB  913GiB 1.94 0.76  57
MIN/MAX VAR: 0.02/2.98  STDDEV: 2.91
```

- **ID:** OSD의 이름입니다.
- **CLASS:** OSD에서 사용하는 장치의 유형입니다.
- **WEoctets:** CRUSH 맵에 있는 OSD의 가중치입니다.
- 기본 다시 가중치 값.
- **SIZE: OSD의 전체 스토리지 용량입니다.**
- **USE: OSD 용량.**
- **DATA: 사용자 데이터에 사용되는 OSD 용량의 크기입니다.**
- **OMAP: 개체 맵(omap) 데이터를 저장하는 데 사용되는bluefs 스토리지의 추정치 값(파이퍼스 db에 저장된 키 쌍)입니다.**
- **META: bluefs 공간 또는 bluestore_bluefs_min 매개변수에 설정된 값(bluestore_bluefs_min 매개변수)은 bluefs 에서 할당된 총 공간을 기준으로 계산된 내부 메타데이터의 경우 예상omap 데이터 크기를 뺀다.**
- **AVAIL: OSD에서 사용 가능한 여유 공간의 크기입니다.**
- **%USE: OSD에서 사용하는 스토리지의 알림 백분율**
- **VAR: 평균 사용률보다 높거나 낮은 변동입니다.**
- **PGS: OSD의 배치 그룹 수입니다.**
- **MIN/MAX VAR: 모든 OSD의 최소 및 최대 변형입니다.**

추가 리소스

- 자세한 내용은 *Ceph에서 데이터 사용량을 계산하는*방법을 참조하십시오.
- 자세한 *내용은 OSD 사용량 통계*이해를 참조하십시오.
- 자세한 내용은 *CRUSH Weights* in *Red Hat Ceph Storage Strategies Guide*를 참조하십시오.

3.2.8. 스토리지 클러스터 상태 확인

명령줄 인터페이스에서 Red Hat Ceph Storage 클러스터의 상태를 확인할 수 있습니다. **status** 하위 명령 또는 **-s** 인수는 스토리지 클러스터의 현재 상태를 표시합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. Cephadm 쉘에 로그인합니다.

   예제

   ```
   [root@host01 ~]# cephadm shell
   ```

2. 스토리지 클러스터 상태를 확인하려면 다음을 실행합니다.

   예제

   ```
   [ceph: root@host01 /]# ceph status
   ```

   또는

   예제

   ```
   [ceph: root@host01 /]# ceph -s
   ```

3. 대화형 모드에서 **ceph** 를 입력하고 Enter 를 누릅니다.

   예제

   ```
   [ceph: root@host01 /]# ceph
   ceph> status
    cluster:
      id:     499829b4-832f-11eb-8d6d-001a4a000635
      health: HEALTH_WARN
              1 stray daemon(s) not managed by cephadm
              1/3 mons down, quorum host03,host02
              too many PGs per OSD (261 > max 250)

    services:
      mon:     3 daemons, quorum host03,host02 (age 3d), out of quorum: host01
      mgr:     host01.hdhzwn(active, since 9d), standbys: host05.eobuuv, host06.wquwpj
      osd:     12 osds: 11 up (since 2w), 11 in (since 5w)
      rgw:     2 daemons active (test_realm.test_zone.host04.hgbvnq,
   test_realm.test_zone.host05.yqqilm)
      rgw-nfs: 1 daemon active (nfs.foo.host06-rgw)

    data:
      pools:   8 pools, 960 pgs
      objects: 414 objects, 1.0 MiB
      usage:   5.7 GiB used, 214 GiB / 220 GiB avail
      pgs:     960 active+clean

    io:
      client:   41 KiB/s rd, 0 B/s wr, 41 op/s rd, 27 op/s wr

   ceph> health
   HEALTH_WARN 1 stray daemon(s) not managed by cephadm; 1/3 mons down, quorum
   host03,host02; too many PGs per OSD (261 > max 250)

   ceph> mon stat
   e3: 3 mons at {host01=[v2:10.74.255.0:3300/0,v1:10.74.255.0:6789/0],host02=
   ```

```
[v2:10.74.249.253:3300/0,v1:10.74.249.253:6789/0],host03=
[v2:10.74.251.164:3300/0,v1:10.74.251.164:6789/0]}, election epoch 6688, leader 1 host03,
quorum 1,2 host03,host02
```

3.2.9. Ceph Monitor 상태 확인

스토리지 클러스터에 프로덕션 Red Hat Ceph Storage 클러스터에 대한 요구 사항인 여러 Ceph Monitor가 있는 경우 스토리지 클러스터를 시작한 후 Ceph Monitor 쿼럼 상태를 확인하고 데이터를 읽거나 쓰기 전에 Ceph Monitor 쿼럼 상태를 확인할 수 있습니다.

여러 Ceph 모니터가 실행 중인 경우 쿼럼이 있어야 합니다.

Ceph Monitor 상태를 주기적으로 확인하여 실행 중인지 확인합니다. Ceph Monitor에 문제가 있는 경우 이는 스토리지 클러스터 상태에 대한 계약을 방지하기 때문에 Ceph 클라이언트가 데이터를 읽고 쓰는 것을 방지할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. Cephadm 쉘에 로그인합니다.

   예제

   ```
   [root@host01 ~]# cephadm shell
   ```

2. Ceph 모니터 맵을 표시하려면 다음을 실행합니다.

   예제

   ```
   [ceph: root@host01 /]# ceph mon stat
   ```

   또는

   예제

   ```
   [ceph: root@host01 /]# ceph mon dump
   ```

3. 스토리지 클러스터의 쿼럼 상태를 확인하려면 다음을 실행합니다.

   ```
   [ceph: root@host01 /]# ceph quorum_status -f json-pretty
   ```

   Ceph에서 쿼럼 상태를 반환합니다.

   예제

   ```
   {
       "election_epoch": 6686,
       "quorum": [
   ```

```
        0,
        1,
        2
    ],
    "quorum_names": [
        "host01",
        "host03",
        "host02"
    ],
    "quorum_leader_name": "host01",
    "quorum_age": 424884,
    "features": {
        "quorum_con": "4540138297136906239",
        "quorum_mon": [
            "kraken",
            "luminous",
            "mimic",
            "osdmap-prune",
            "nautilus",
            "octopus",
            "pacific",
            "elector-pinging"
        ]
    },
    "monmap": {
        "epoch": 3,
        "fsid": "499829b4-832f-11eb-8d6d-001a4a000635",
        "modified": "2021-03-15T04:51:38.621737Z",
        "created": "2021-03-12T12:35:16.911339Z",
        "min_mon_release": 16,
        "min_mon_release_name": "pacific",
        "election_strategy": 1,
        "disallowed_leaders: ": "",
        "stretch_mode": false,
        "features": {
            "persistent": [
                "kraken",
                "luminous",
                "mimic",
                "osdmap-prune",
                "nautilus",
                "octopus",
                "pacific",
                "elector-pinging"
            ],
            "optional": []
        },
        "mons": [
            {
                "rank": 0,
                "name": "host01",
                "public_addrs": {
                    "addrvec": [
                        {
                            "type": "v2",
                            "addr": "10.74.255.0:3300",
```

```
                "nonce": 0
            },
            {
                "type": "v1",
                "addr": "10.74.255.0:6789",
                "nonce": 0
            }
        ]
    },
    "addr": "10.74.255.0:6789/0",
    "public_addr": "10.74.255.0:6789/0",
    "priority": 0,
    "weight": 0,
    "crush_location": "{}"
},
{
    "rank": 1,
    "name": "host03",
    "public_addrs": {
        "addrvec": [
            {
                "type": "v2",
                "addr": "10.74.251.164:3300",
                "nonce": 0
            },
            {
                "type": "v1",
                "addr": "10.74.251.164:6789",
                "nonce": 0
            }
        ]
    },
    "addr": "10.74.251.164:6789/0",
    "public_addr": "10.74.251.164:6789/0",
    "priority": 0,
    "weight": 0,
    "crush_location": "{}"
},
{
    "rank": 2,
    "name": "host02",
    "public_addrs": {
        "addrvec": [
            {
                "type": "v2",
                "addr": "10.74.249.253:3300",
                "nonce": 0
            },
            {
                "type": "v1",
                "addr": "10.74.249.253:6789",
                "nonce": 0
            }
        ]
    },
    "addr": "10.74.249.253:6789/0",
```

```
            "public_addr": "10.74.249.253:6789/0",
            "priority": 0,
            "weight": 0,
            "crush_location": "{}"
        }
    ]
  }
}
```

3.2.10. Ceph 관리 소켓 사용

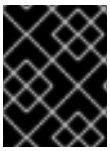
UNIX 소켓 파일을 사용하여 제공된 데몬과 직접 상호 작용하려면 관리 소켓을 사용합니다. 예를 들어 소켓은 다음을 수행할 수 있습니다.

- 런타임 시 Ceph 구성 나열
- 모니터에 의존하지 않고 런타임 시 구성 값을 직접 설정합니다. 이는 모니터가 다운 될 때 유용합니다.
- 기록 작업 덤프
- 작업 우선 순위 큐 상태 덤프
- 재부팅하지 않고 작업 덤프
- 성능 카운터 덤프

또한 소켓을 사용하면 Ceph 모니터 또는 OSD와 관련된 문제를 해결할 때 유용합니다.

데몬이 실행 중이 아닌 경우 관리 소켓을 사용하려고 할 때 다음 오류가 반환됩니다.

```
Error 111: Connection Refused
```



중요

관리 소켓은 데몬이 실행되는 동안에만 사용할 수 있습니다. 데몬을 올바르게 종료하면 관리 소켓이 제거됩니다. 그러나 데몬이 예기치 않게 종료되면 관리 소켓이 유지될 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. Cephadm 쉘에 로그인합니다.

   예제

   ```
   [root@host01 ~]# cephadm shell
   ```

2. 소켓을 사용하려면 다음을 수행합니다.

구문

```
ceph daemon MONITOR_ID COMMAND
```

replace:

- 데몬의 ***MONITOR_ID***
- 실행할 명령과 함께 ***COMMAND*** 입니다. 도움말 을 사용하여 지정된 데몬에 사용 가능한 명령을 나열합니다.
  Ceph 모니터의 상태를 보려면 다음을 수행합니다.

  예제

```
[ceph: root@host01 /]# ceph daemon mon.host01 help
{
    "add_bootstrap_peer_hint": "add peer address as potential bootstrap peer for cluster bringup",
    "add_bootstrap_peer_hintv": "add peer address vector as potential bootstrap peer for cluster bringup",
    "compact": "cause compaction of monitor's leveldb/rocksdb storage",
    "config diff": "dump diff of current config and default config",
    "config diff get": "dump diff get <field>: dump diff of current and default config setting <field>",
    "config get": "config get <field>: get the config value",
    "config help": "get config setting schema and descriptions",
    "config set": "config set <field> <val> [<val> ...]: set a config variable",
    "config show": "dump current config settings",
    "config unset": "config unset <field>: unset a config variable",
    "connection scores dump": "show the scores used in connectivity-based elections",
    "connection scores reset": "reset the scores used in connectivity-based elections",
    "dump_historic_ops": "dump_historic_ops",
    "dump_mempools": "get mempool stats",
    "get_command_descriptions": "list available commands",
    "git_version": "get git sha1",
    "heap": "show heap usage info (available only if compiled with tcmalloc)",
    "help": "list available commands",
    "injectargs": "inject configuration arguments into running daemon",
    "log dump": "dump recent log entries to log file",
    "log flush": "flush log entries to log file",
    "log reopen": "reopen log file",
    "mon_status": "report status of monitors",
    "ops": "show the ops currently in flight",
    "perf dump": "dump perfcounters value",
    "perf histogram dump": "dump perf histogram values",
    "perf histogram schema": "dump perf histogram schema",
    "perf reset": "perf reset <name>: perf reset all or one perfcounter name",
    "perf schema": "dump perfcounters schema",
    "quorum enter": "force monitor back into quorum",
    "quorum exit": "force monitor out of the quorum",
    "sessions": "list existing sessions",
    "smart": "Query health metrics for underlying device",
    "sync_force": "force sync of and clear monitor store",
    "version": "get ceph version"
}
```

예제

```
[ceph: root@host01 /]# ceph daemon mon.host01 mon_status

{
    "name": "host01",
    "rank": 0,
    "state": "leader",
    "election_epoch": 120,
    "quorum": [
        0,
        1,
        2
    ],
    "quorum_age": 206358,
    "features": {
        "required_con": "2449958747317026820",
        "required_mon": [
            "kraken",
            "luminous",
            "mimic",
            "osdmap-prune",
            "nautilus",
            "octopus",
            "pacific",
            "elector-pinging"
        ],
        "quorum_con": "4540138297136906239",
        "quorum_mon": [
            "kraken",
            "luminous",
            "mimic",
            "osdmap-prune",
            "nautilus",
            "octopus",
            "pacific",
            "elector-pinging"
        ]
    },
    "outside_quorum": [],
    "extra_probe_peers": [],
    "sync_provider": [],
    "monmap": {
        "epoch": 3,
        "fsid": "81a4597a-b711-11eb-8cb8-001a4a000740",
        "modified": "2021-05-18T05:50:17.782128Z",
        "created": "2021-05-17T13:13:13.383313Z",
        "min_mon_release": 16,
        "min_mon_release_name": "pacific",
        "election_strategy": 1,
        "disallowed_leaders: ": "",
        "stretch_mode": false,
        "features": {
            "persistent": [
                "kraken",
                "luminous",
```

```
            "mimic",
            "osdmap-prune",
            "nautilus",
            "octopus",
            "pacific",
            "elector-pinging"
        ],
        "optional": []
    },
    "mons": [
        {
            "rank": 0,
            "name": "host01",
            "public_addrs": {
                "addrvec": [
                    {
                        "type": "v2",
                        "addr": "10.74.249.41:3300",
                        "nonce": 0
                    },
                    {
                        "type": "v1",
                        "addr": "10.74.249.41:6789",
                        "nonce": 0
                    }
                ]
            },
            "addr": "10.74.249.41:6789/0",
            "public_addr": "10.74.249.41:6789/0",
            "priority": 0,
            "weight": 0,
            "crush_location": "{}"
        },
        {
            "rank": 1,
            "name": "host02",
            "public_addrs": {
                "addrvec": [
                    {
                        "type": "v2",
                        "addr": "10.74.249.55:3300",
                        "nonce": 0
                    },
                    {
                        "type": "v1",
                        "addr": "10.74.249.55:6789",
                        "nonce": 0
                    }
                ]
            },
            "addr": "10.74.249.55:6789/0",
            "public_addr": "10.74.249.55:6789/0",
            "priority": 0,
            "weight": 0,
            "crush_location": "{}"
        },
```

```
            {
                "rank": 2,
                "name": "host03",
                "public_addrs": {
                    "addrvec": [
                        {
                            "type": "v2",
                            "addr": "10.74.249.49:3300",
                            "nonce": 0
                        },
                        {
                            "type": "v1",
                            "addr": "10.74.249.49:6789",
                            "nonce": 0
                        }
                    ]
                },
                "addr": "10.74.249.49:6789/0",
                "public_addr": "10.74.249.49:6789/0",
                "priority": 0,
                "weight": 0,
                "crush_location": "{}"
            }
        ]
    },
    "feature_map": {
        "mon": [
            {
                "features": "0x3f01cfb9fffdffff",
                "release": "luminous",
                "num": 1
            }
        ],
        "osd": [
            {
                "features": "0x3f01cfb9fffdffff",
                "release": "luminous",
                "num": 3
            }
        ]
    },
    "stretch_mode": false
}
```


3. **또는 소켓 파일을 사용하여 Ceph 데몬을 지정합니다.**

   구문

   ```
   ceph daemon /var/run/ceph/SOCKET_FILE COMMAND
   ```

4. **osd.2 라는 Ceph OSD의 상태를 보려면 다음을 수행합니다.**

   예제

   ```
   [ceph: root@host01 /]# ceph daemon /var/run/ceph/ceph-osd.2.asok status
   ```

5. Ceph 프로세스의 모든 소켓 파일을 나열하려면 다음을 수행합니다.

   예제

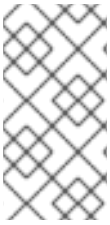
```
[ceph: root@host01 /]# ls /var/run/ceph
```

추가 리소스

- 자세한 내용은 *Red Hat Ceph Storage 문제 해결 가이드*를 참조하십시오.

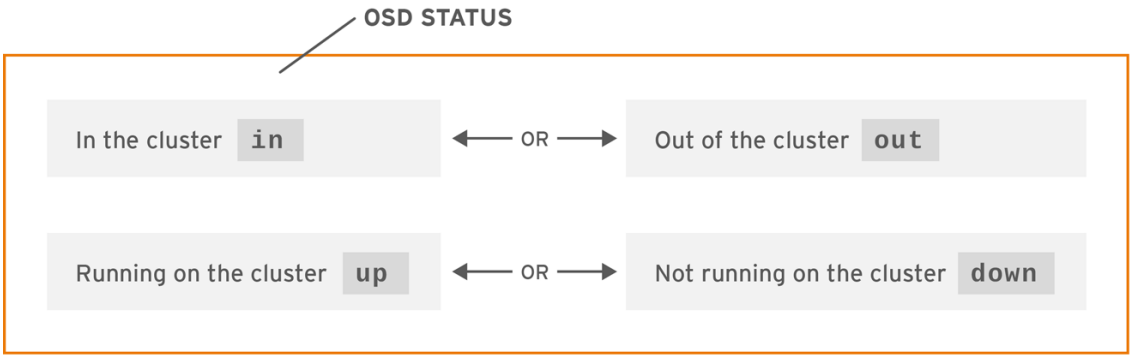
3.2.11. Ceph OSD 상태 이해

Ceph OSD의 상태는 스토리지 클러스터 또는 스토리지 클러스터 외부에 있습니다. 실행 중이거나 실행 중이거나 실행 중이지 않습니다. **Ceph OSD**가 가동 중인 경우 스토리지 클러스터에서 데이터를 읽고 쓸 수 있거나 스토리지 클러스터가 부족 해질 수 있습니다. 스토리지 클러스터에 있고 최근에 스토리지 클러스터에서 나가면 **Ceph** 에서 다른 **Ceph OSD**로 배치 그룹을 마이그레이션하기 시작합니다. **Ceph OSD**가 스토리지 클러스터 가 아닌 경우 **CRUSH**는 **Ceph OSD**에 배치 그룹을 할당하지 않습니다. **Ceph OSD**가 다운 된 경우 이기도 합니다.



참고

Ceph OSD가 다운 되어 있는 경우 문제가 있으며 스토리지 클러스터는 정상 상태가 아닙니다.



ceph 상태,**ceph -s** 또는 **ceph -w** 와 같은 명령을 실행하는 경우 스토리지 클러스터가 항상 **HEALTH OK** 를 에코하지 않을 수 있습니다. 패닉을 일으키지 마십시오. **Ceph OSD**와 관련하여 몇 가지 예상되는 상황에서 스토리지 클러스터가 **HEALTH OK** 를 에코 하지 않을 것으로 예상할 수 있습니다.

- 스토리지 클러스터를 아직 시작하지 않았으며 응답하지 않습니다.
- 스토리지 클러스터를 시작하거나 다시 시작했으며 배치 그룹이 생성되고 **Ceph OSD**가 피어링 프로세스에 있기 때문에 아직 준비되지 않았습니다.

- **Ceph OSD**를 방금 추가하거나 제거했습니다.
- 스토리지 클러스터 맵을 방금 수정했습니다.

Ceph OSD를 모니터링하는 중요한 측면은 스토리지 클러스터가 스토리지 클러스터에 있는 모든 **Ceph OSD**를 가동 및 실행하는 경우에도 실행 중인지 확인하는 것입니다.

모든 **OSD**가 실행 중인지 확인하려면 다음을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph osd stat
```

또는

예제

```
[ceph: root@host01 /]# ceph osd dump
```

그 결과 맵 **epoch, eNNNN,** 총 **OSD** 수**, x,** 몇 개**, y, up,** 개수**, z** 는 다음과 같습니다.

```
eNNNN: x osds: y up, z in
```

스토리지 클러스터에 있는 **Ceph OSD**의 수가 **up** 인 **Ceph OSD** 수보다 많은 경우**.** 다음 명령을 실행하여 실행 중이 아닌 **ceph-osd** 데몬을 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree

# id    weight  type name   up/down reweight
-1  3   pool default
-3  3       rack mainrack
-2  3           host osd-host
0   1               osd.0   up  1
1   1               osd.1   up  1
2   1               osd.2   up  1
```

작은 정보

잘 설계된 **CRUSH** 계층 구조를 통해 검색하는 기능은 물리적 위치를 더 빠르게 식별하여 스토리지 클러스터 문제를 해결하는 데 도움이 될 수 있습니다.

Ceph OSD가 다운되면 노드에 연결하여 노드를 시작합니다. **Red Hat Storage Console**을 사용하여 **Ceph OSD** 데몬을 다시 시작하거나 명령줄을 사용할 수 있습니다.

구문

```
systemctl start CEPH_OSD_SERVICE_ID
```

예제

```
[root@host01 ~]# systemctl start ceph-499829b4-832f-11eb-8d6d-001a4a000635@osd.6.service
```

추가 리소스

- 자세한 내용은 *Red Hat Ceph Storage 대시보드 가이드* 를 참조하십시오.

3.3. CEPH 스토리지 클러스터의 낮은 수준의 모니터링

스토리지 관리자는 낮은 수준의 관점에서 **Red Hat Ceph Storage** 클러스터의 상태를 모니터링할 수 있습니다. 일반적으로 낮은 수준의 모니터링에는 **Ceph OSD**가 올바르게 피어링되었는지 확인하는 작업이 포함됩니다. 피어링 오류가 발생하면 배치 그룹이 성능이 저하된 상태에서 작동합니다. 이 성능 저하 상태는 하드웨어 오류, 중단된 **Ceph** 데몬, 네트워크 대기 시간 또는 완전한 사이트 중단과 같은 다양한 문제로 인해 발생할 수 있습니다.

3.3.1. 사전 요구 사항

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

3.3.2. 배치 그룹 세트 모니터링

CRUSH가 배치 그룹을 **Ceph OSD**에 할당할 때 풀의 복제본 수를 확인하고 각 배치 그룹을 **Ceph OSD**에 할당하여 배치 그룹을 다른 **Ceph OSD**에 할당합니다. 예를 들어 풀에 배치 그룹의 복제본이 세 개 필요한 경우 **CRUSH**가 **osd.1**,**osd.2** 및 **osd.3** 에 각각 할당할 수 있습니다. **CRUSH**는 실제로 **CRUSH** 맵에 설정한 실패 도메인을 고려하기 위한 의사랜덤 배치를 검색하므로 대규모 클러스터에서 가장 가까운 **Ceph OSD**에 할당된 배치 그룹은 거의 볼 수 없습니다. 특정 배치 그룹의 복제본을 동작 설정으로 포함해야 하는 **Ceph OSD** 세트를 나타냅니다. 경우에 따라 활성 집합의 **OSD**가 다운되었거나 그렇지 않으면 배치 그룹의 개체에 대한 요청을 서비스할 수 없습니다. 이러한 상황이 발생하면 패닉을 일으키지 마십시오. 일반적인 예는 다음과 같습니다.

- **OSD**를 추가 또는 제거했습니다. 그런 다음 **CRUSH**가 배치 그룹을 다른 **Ceph OSD**에 다시 할당하여 작동 중인 세트의 구성을 변경하고 "백필" 프로세스를 사용하여 데이터 마이그레이션을 생성합니다.

- **Ceph OSD**가 다운 되어 이제 복구 중입니다.

- 작업 세트의 **Ceph OSD**가 다운 되거나 서비스 요청을 할 수 없으며 다른 **Ceph OSD**에서 일시적으로 업무를 수행한다고 가정합니다.

Ceph에서는 요청을 실제로 처리하는 **Ceph OSD** 세트인 **Up Set** 을 사용하여 클라이언트 요청을 처리합니다. 대부분의 경우 **up set** 및 **Acting Set**은 사실상 동일합니다. **Ceph**가 데이터를 마이그레이션하는 경우 **Ceph OSD**가 복구 중이거나 문제가 있음을 나타낼 수 있습니다. 즉, **Ceph**는 일반적으로 이러한 시나리오에서 "더하기 오래된" 메시지와 함께 **HEALTH WARN** 상태를 에코합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. **Cephadm** 쉘에 로그인합니다.

   예제

   ```
   [root@host01 ~]# cephadm shell
   ```

2. 배치 그룹 목록을 검색하려면 다음을 수행합니다.

   예제

   ```
   [ceph: root@host01 /]# ceph pg dump
   ```

3. 지정된 배치 그룹에 대해 동작 집합 또는 **Up Set** 에 있는 **Ceph OSD**를 확인합니다.

   구문

   ```
   ceph pg map PG_NUM
   ```

예제

```
[ceph: root@host01 /]# ceph pg map 128
```

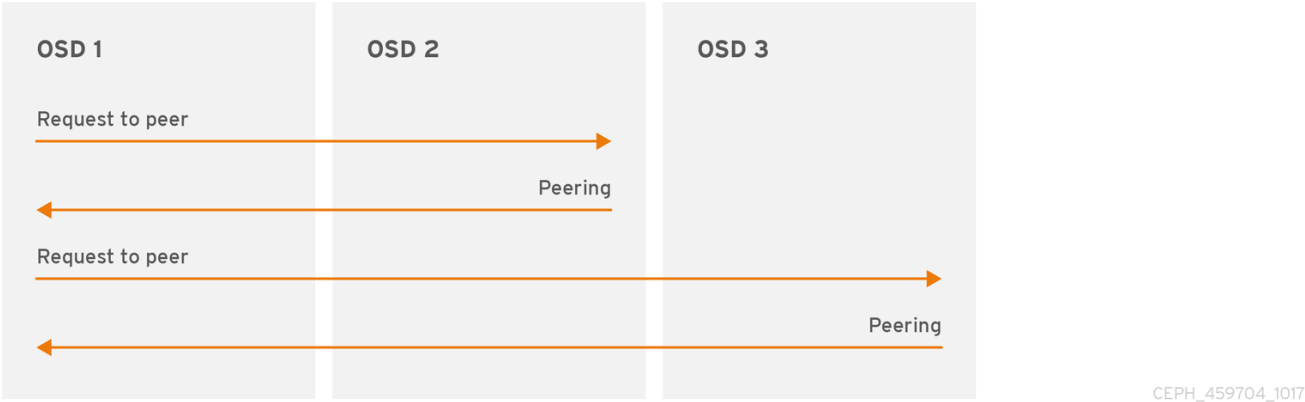
참고

Up Set 및 **Acting Set**이 일치하지 않는 경우, 스토리지 클러스터의 자체 또는 스토리지 클러스터에 잠재적인 문제가 있음을 나타내는 표시일 수 있습니다.

3.3.3. Ceph OSD 피어링

배치 그룹에 데이터를 작성하려면 먼저 활성 상태여야 하며 클린 상태에 있어야 합니다. **Ceph**가 배치 그룹의 현재 상태, 즉 작동 세트의 첫 번째 **OSD**인 배치 그룹의 기본 **OSD**, 보조 **OSD** 및 **tertiary OSD**와의 피어를 확인하여 배치 그룹의 현재 상태에 대한 계약을 설정합니다. **PG**의 복제본이 세 개 있는 풀을 가정합니다.

그림 3.1. 피어링



3.3.4. 배치 그룹 상태

ceph 상태,**ceph -s** 또는 **ceph -w** 와 같은 명령을 실행하는 경우 클러스터가 항상 **HEALTH OK** 를 에코하는 것은 아닙니다. **OSD**가 실행 중인지 확인한 후 배치 그룹 상태도 확인해야 합니다. 클러스터가 여러 배치 그룹 피어링 관련 상황에서 **HEALTH OK** 를 **echo** 하지 않을 것으로 예상해야 합니다.

- 아직 풀과 배치 그룹을 만들었지만 아직 피어링되지 않았습니다.

- 배치 그룹은 복구됩니다.
- 클러스터에서 **OSD**를 추가하거나 제거했습니다.
- **CRUSH** 맵을 방금 수정했으며 배치 그룹이 마이그레이션되고 있습니다.
- 배치 그룹의 다른 복제본에는 일치하지 않는 데이터가 있습니다.
- **Ceph**에서 배치 그룹의 복제본을 스크럽합니다.
- **Ceph**에는 백필 작업을 완료하기에 충분한 스토리지 용량이 없습니다.

계속되는 상황 중 하나가 **Ceph**에서 **WARN**을 에코 하게 되는 경우 패닉 상태가 발생하지 않습니다. 대부분의 경우 클러스터가 자체적으로 복구됩니다. 경우에 따라 조치를 취해야 할 수도 있습니다. 모니터링 배치 그룹의 중요한 측면은 클러스터가 가동 및 실행되고 모든 배치 그룹이 활성 상태이고 바람직하게는 정리 상태에 있는지 확인하는 것입니다.

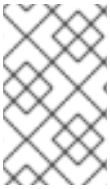
모든 배치 그룹의 상태를 보려면 다음을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph pg stat
```

그 결과 배치 그룹 맵 버전, **vNNNNNN**, 총 배치 그룹 수, **x**, **y** 의 수(예: **active+clean**)와 같은 특정 상태에 있는지 확인해야 합니다.

```
vNNNNNN: x pgs: y active+clean; z bytes data, aa MB used, bb GB / cc GB avail
```



참고

Ceph에서 배치 그룹의 여러 상태를 보고하는 것이 일반적입니다.

스냅 샷 trimming PG States

스냅샷이 존재하는 경우 두 개의 추가 **PG** 상태가 보고됩니다.

- **snaptrim :** 현재 **PG**가 트리밍 중입니다.
- **snaptrim_wait : PG**는 트리밍 대기 중입니다.

출력 예**:**

```
244 active+clean+snaptrim_wait
 32 active+clean+snaptrim
```

Ceph는 배치 그룹 상태 외에도 사용되는 데이터 양**, aa,** 남아 있는 스토리지 용량**, bb,** 배치 그룹의 총 스토리지 용량도 에코합니다**.** 이 숫자는 몇 가지 경우에 중요할 수 있습니다**.**

- 가까운 전체 비율 또는 전체 비율에 도달하고 있습니다**.**
- **CRUSH** 구성의 오류로 인해 데이터가 클러스터에 배포되지 않습니다**.**

배치 그룹 ID

배치 그룹 **ID**는 풀 이름이 아닌 풀 번호**,** 그 뒤에 마침표**(.)** 및 배치 그룹 **ID-a 16**진수로 구성됩니다**. ceph osd lspools** 출력에서 풀 번호와 해당 이름을 볼 수 있습니다**.** 기본 풀 이름은 데이터 이름**,**메타데이터 및 **rbd** 는 각각 **0,1** 및 **2** 풀에 해당합니다**.** 정규화된 배치 그룹 **ID**의 형식은 다음과 같습니다**.**

구문

```
POOL_NUM.PG_ID
```

출력 예:

```
0.1f
```

- **배치 그룹 목록을 검색하려면 다음을 수행합니다.**

  예제

  ```
  [ceph: root@host01 /]# ceph pg dump
  ```

- **출력을 JSON 형식으로 포맷하고 파일에 저장하려면 다음을 수행합니다.**

  구문

  ```
  ceph pg dump -o FILE_NAME --format=json
  ```

  예제

  ```
  [ceph: root@host01 /]# ceph pg dump -o test --format=json
  ```

- **특정 배치 그룹을 쿼리합니다.**

  구문

```
ceph pg POOL_NUM.PG_ID query
```

예제

```
[ceph: root@host01 /]#  ceph pg 5.fe query
{
    "snap_trimq": "[]",
    "snap_trimq_len": 0,
    "state": "active+clean",
    "epoch": 2449,
    "up": [
        3,
        8,
        10
    ],
    "acting": [
        3,
        8,
        10
    ],
    "acting_recovery_backfill": [
        "3",
        "8",
        "10"
    ],
    "info": {
        "pgid": "5.ff",
        "last_update": "0'0",
        "last_complete": "0'0",
        "log_tail": "0'0",
        "last_user_version": 0,
        "last_backfill": "MAX",
        "purged_snaps": [],
        "history": {
            "epoch_created": 114,
            "epoch_pool_created": 82,
            "last_epoch_started": 2402,
            "last_interval_started": 2401,
            "last_epoch_clean": 2402,
            "last_interval_clean": 2401,
            "last_epoch_split": 114,
            "last_epoch_marked_full": 0,
            "same_up_since": 2401,
            "same_interval_since": 2401,
            "same_primary_since": 2086,
            "last_scrub": "0'0",
```

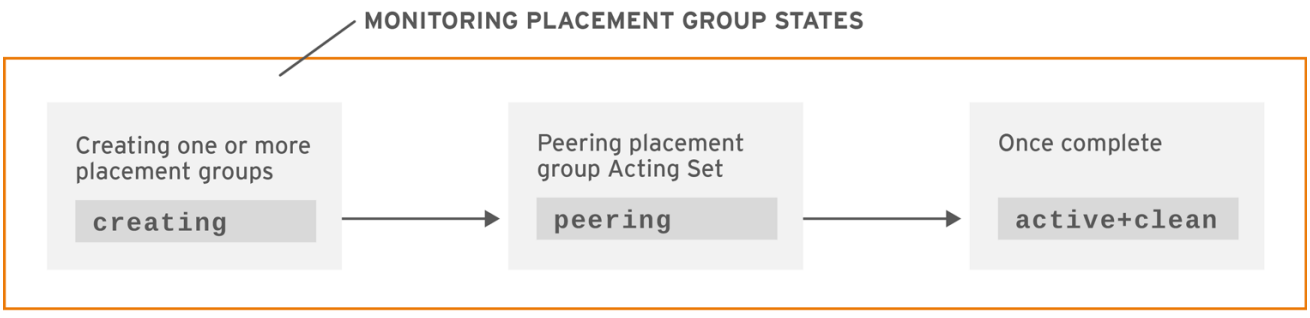
```
            "last_scrub_stamp": "2021-06-17T01:32:03.763988+0000",
            "last_deep_scrub": "0'0",
            "last_deep_scrub_stamp": "2021-06-17T01:32:03.763988+0000",
            "last_clean_scrub_stamp": "2021-06-17T01:32:03.763988+0000",
            "prior_readable_until_ub": 0
        },
        "stats": {
            "version": "0'0",
            "reported_seq": "2989",
            "reported_epoch": "2449",
            "state": "active+clean",
            "last_fresh": "2021-06-18T05:16:59.401080+0000",
            "last_change": "2021-06-17T01:32:03.764162+0000",
            "last_active": "2021-06-18T05:16:59.401080+0000",
....
```

추가 리소스

- 스냅샷 트리밍 설정에 대한 자세한 내용은 ***Red Hat Ceph Storage*** *구성 가이드의* ***OSD*** *오브젝트 스토리지 데몬 구성 가이드의* ***OSD*** *오브젝트 스토리지 데몬 구성 옵션 섹션에서* ***OSD OSD(Object Storage Daemon)*** 구성 옵션을 참조하십시오.

3.3.5. 배치 그룹 생성 상태

풀을 생성하면 지정한 배치 그룹 수가 생성됩니다. Ceph는 하나 이상의 배치 그룹을 생성할 때 생성을 에코합니다. 생성된 OSD는 배치 그룹의 acting Set에 속하는 OSD를 피어링합니다. 피어링이 완료되면 배치 그룹 상태가 active+clean 이어야 합니다. 즉, Ceph 클라이언트가 배치 그룹에 쓰기를 시작할 수 있습니다.



CEPH_459704_1017

3.3.6. 배치 그룹 피어링 상태

Ceph가 배치 그룹을 피어링하는 경우 Ceph는 배치 그룹의 복제본을 배치 그룹의 오브젝트 및 메타데이터 의 상태에 대한 동의 로 저장하는 OSD를 가져옵니다. Ceph가 피어링을 완료하면 배치 그룹을 저장

하는 OSD가 배치 그룹의 현재 상태에 대해 동의합니다. 그러나 피어링 프로세스 완료는 각 복제본에 최신 콘텐츠가 있음을 의미하는 것은 아닙니다.

신뢰할 수 있는 기록

Ceph는 대기 세트의 모든 OSD가 쓰기 작업이 될 때까지 클라이언트에 쓰기 작업을 인식하지 않습니다. 이 관행은 마지막으로 성공한 피어링 작업 이후 최소 하나의 동작 세트의 멤버가 모든 인식된 쓰기 작업의 레코드를 갖도록 보장합니다.

Ceph는 확인된 각 쓰기 작업의 정확한 레코드를 사용하여 배치 그룹의 새로운 권한 있는 기록을 구성하고 배포할 수 있습니다. 완료되면 OSD의 배치 그룹의 사본을 최신으로 가져오는 완료 및 완전히 정렬된 작업 집합입니다.

3.3.7. 배치 그룹 활성 상태

Ceph가 피어링 프로세스를 완료하면 배치 그룹이 활성화 될 수 있습니다. 활성 상태는 배치 그룹의 데이터를 일반적으로 기본 배치 그룹 및 읽기 및 쓰기 작업에 대한 복제본에서 사용할 수 있음을 의미합니다.

3.3.8. 배치 그룹 정리 상태

배치 그룹이 정리 상태에 있으면 기본 OSD와 복제본 OSD가 성공적으로 피어링되고 배치 그룹의 스프레이 복제본이 없습니다. Ceph는 배치의 모든 오브젝트를 올바른 횟수만큼 복제했습니다.

3.3.9. 배치 그룹 성능 저하 상태

클라이언트가 기본 OSD에 오브젝트를 쓸 때 기본 OSD는 복제본 OSD에 복제본을 작성합니다. 기본 OSD에서 오브젝트를 스토리지에 기록한 후 기본 OSD가 복제본 OSD에서 복제본 오브젝트를 성공적으로 생성할 때까지 배치 그룹은 성능이 저하된 상태로 유지됩니다.

배치 그룹이 active+degraded 될 수 있는 이유는 아직 모든 객체를 보유하지 않더라도 OSD를 활성화할 수 있기 때문입니다. OSD가 다운 되면 Ceph는 OSD에 할당된 각 배치 그룹을 성능 저하 로 표시합니다. Ceph OSD가 다시 온라인 상태가 되면 Ceph OSD를 다시 피어링해야 합니다. 그러나 클라이언트는 활성 상태인 경우에도 성능이 저하된 배치 그룹에 새 개체를 쓸 수 있습니다.

OSD가 다운 되고 성능이 저하 된 상태가 지속되면 Ceph에서 다운 OSD를 클러스터 외부에서 표시하고 다운 OSD에서 데이터를 다른 OSD로 다시 매핑할 수 있습니다. 표시된 후 표시된 시간은 mon_osd_down _ out _interval 에 의해 제어되며 기본적으로 600 초로 설정됩니다.

Ceph에서 배치 그룹에 있어야 한다고 생각하는 하나 이상의 오브젝트를 찾을 수 없기 때문에 배치 그룹도 성능이 저하 될 수 있습니다. unfound 개체를 읽거나 쓸 수는 없지만 성능이 저하된 배치 그룹의 다른 모든 오브젝트에 계속 액세스할 수 있습니다.

예를 들어 세 개의 복제본 풀에 9개의 OSD가 있는 경우 다음과 같습니다. OSD 번호 9가 다운되면 OSD 9에 할당된 PG가 성능 저하 상태가 됩니다. OSD 9가 복구되지 않으면 스토리지 클러스터와 스토리지 클러스터의 균형을 유지합니다. 이 시나리오에서 PG는 성능이 저하된 다음 활성 상태로 복구됩니다.

3.3.10. 상태 복구 배치 그룹

Ceph는 하드웨어 및 소프트웨어 문제가 지속적으로 발생하는 규모에 따라 내결함성을 위해 설계되었습니다. OSD가 다운 되면 해당 콘텐츠는 배치 그룹의 다른 복제본의 현재 상태 뒤에 있을 수 있습니다. OSD가 백업 되면 현재 상태를 반영하도록 배치 그룹의 콘텐츠를 업데이트해야 합니다. 이 기간 동안 OSD는 복구 상태를 반영할 수 있습니다.

하드웨어 장애로 인해 여러 Ceph OSD가 잘못될 수 있기 때문에 복구가 쉽지 않은 경우가 있습니다. 예를 들어 랙 또는 cache의 네트워크 스위치가 실패할 수 있으며 이로 인해 여러 호스트 시스템의 OSD가 스토리지 클러스터의 현재 상태 뒤에 떨어질 수 있습니다. 오류가 해결되면 OSD의 각 OSD를 복구해야 합니다.

Ceph는 새 서비스 요청 간의 리소스 경합과 배치 그룹을 복구하고 배치 그룹을 현재 상태로 복원하는 필요성 간의 리소스 경합을 균형을 유지하는 다양한 설정을 제공합니다. osd 복구 지연 시작 설정을 사용하면 OSD에서 복구 프로세스를 시작하기 전에 일부 재생 요청을 다시 시작, 재 피어링 및 처리할 수 있습니다. osd 복구 스레드 설정은 기본적으로 하나의 스레드로 복구 프로세스의 스레드 수를 제한합니다. osd 복구 스레드 시간 초과 는 여러 Ceph OSD가 실패할 수 있으므로 정지된 속도로 다시 시작 및 재피어할 수 있기 때문에 스레드 시간 초과를 설정합니다. osd 복구 max active 설정은 Ceph OSD가 동시에 작동하는 복구 요청 수를 제한하여 Ceph OSD가 서비스를 제공하지 못하도록 합니다. osd recovery max chunk 설정은 네트워크 혼잡을 방지하기 위해 복구된 데이터 청크의 크기를 제한합니다.

3.3.11. 백 채우기 상태

새 Ceph OSD가 스토리지 클러스터에 참여하면 CRUSH는 클러스터의 OSD에서 새로 추가된 Ceph OSD로 배치 그룹을 다시 할당합니다. 새 OSD가 다시 할당된 배치 그룹을 즉시 수락하도록 강제하면 새 Ceph OSD에 과도한 부하를 줄 수 있습니다. OSD를 배치 그룹으로 백필하면 이 프로세스가 백그라운드에서 시작될 수 있습니다. 백필링이 완료되면 새 OSD가 준비되면 새 OSD가 요청을 제공하기 시작합니다.

백필 작업 중에 여러 상태 중 하나가 표시될 수 있습니다.

- backfill_wait 는 백필 작업이 보류 중이지만 아직 진행 중이 아님을 나타냅니다.

- 백필(backfill)은 백필 작업이 진행 중임을 나타냅니다.

- backfill_too_full 은 백필 작업이 요청되었지만 스토리지 용량이 부족하여 완료할 수 없음을 나타냅니다.

배치 그룹을 다시 채울 수 없는 경우 불완전한 것으로 간주될 수 있습니다.

Ceph는 Ceph OSD, 특히 새 Ceph OSD에 배치 그룹을 다시 할당하는 것과 관련된 부하 급증을 관리하는 다양한 설정을 제공합니다. 기본적으로 osd_max_backfills 는 최대 동시 백필 수를 Ceph OSD에서 10으로 설정합니다. OSD가 전체 비율에 도달하면 osd 백필 전체 비율을 사용하면 Ceph OSD에서 전체 비율에 도달하면 백필 요청을 거부할 수 있습니다. OSD에서 백필 요청을 거부하는 경우 osd backfill 재시도 간격을 사용하면 OSD에서 기본적으로 요청을 10초 후에 재시도할 수 있습니다. OSD는 osd backfill 검사 min 및 osd backfill 검사 max 를 설정하여 기본적으로 64 및 512로 검사 간격을 관리할 수도 있습니다.

일부 워크로드의 경우 일반 복구를 완전히 방지하고 대신 백필을 사용하는 것이 좋습니다. 백그라운드에서 백필링되므로 I/O가 OSD의 개체를 계속 진행할 수 있습니다. osd_min_pg_log_entries 옵션을 1 로 설정하고 osd_max_pg_log_entries 옵션을 2 로 설정하여 복구 대신 백필을 강제 적용할 수 있습니다. 이러한 상황이 워크로드에 적합한 경우 Red Hat 지원 계정 팀에 문의하십시오.

3.3.12. 배치 그룹 다시 매핑된 상태

배치 그룹이 변경되는 서비스를 설정하면 데이터가 새로운 동작 세트로 설정된 이전 동작에서 마이그레이션됩니다. 새로운 기본 OSD가 서비스 요청에 시간이 걸릴 수 있습니다. 따라서 배치 그룹 마이그레이션이 완료될 때까지 이전 주에서 계속 요청을 서비스하도록 요청할 수 있습니다. 데이터 마이그레이션이 완료되면 매핑은 새 작동 세트의 기본 OSD를 사용합니다.

3.3.13. 배치 그룹 오래된 상태

Ceph는 하트비트를 사용하여 호스트 및 데몬이 실행 중인지 확인하는 동안 ceph-osd 데몬도 시기 적절하게 통계를 보고하지 않는 고정 상태가 될 수 있습니다. 예를 들어, 일시적인 네트워크 오류입니다. 기본적으로 OSD 데몬은 배치 그룹, up thru, boot 및 failure 통계를 반 초 단위로 보고합니다. 즉, 0.5 이며 이는 하트비트 임계값보다 더 자주 발생합니다. 배치 그룹의 기본 OSD 가 모니터에 보고되지 않거나 다른 OSD에서 기본 OSD 다운 을 보고한 경우 모니터에서 배치 그룹 stale 를 표시합니다.

스토리지 클러스터를 시작하면 피어 프로세스가 완료될 때까지 오래된 상태를 확인하는 것이 일반적입니다. 오래된 상태의 배치 그룹을 확인하여 스토리지 클러스터가 실행된 후 해당 배치 그룹의 기본 OSD가 다운되었거나 배치 그룹 통계를 모니터에 보고하지 않음을 나타냅니다.

3.3.14. 배치 그룹 잘못된 위치 상태

PG가 **OSD**에 일시적으로 매핑되는 몇 가지 임시 백필링 시나리오가 있습니다. 임시 상황이 더 이상 발생하지 않아야 하는 경우 **PG**는 여전히 임시 위치에 있고 적절한 위치에 있지 않을 수 있습니다. 이 경우, 그들은 잘못된 것으로 간주됩니다. 올바른 수의 추가 사본이 실제로 존재하지만 하나 이상의 복사본이 잘못된 위치에 있기 때문입니다.

예를 들어 **OSD 0,1,2**가 **3**개 있으며 모든 **PG**는 이러한 세 가지 변경 사항에 매핑됩니다. 다른 **OSD(OSD 3)**를 추가하면 일부 **PG**는 이제 다른 하나의 **OSD** 대신 **OSD 3**에 매핑됩니다. 그러나 **OSD 3**가 다시 입력될 때까지 **PG**는 이전 매핑에서 **I/O**를 계속 제공할 수 있는 임시 매핑을 제공합니다. 이 기간 동안 **PG**는 **3**개의 사본이 있기 때문에 임시 매핑이 있지만 성능 저하 가 없기 때문에 **PG**가 잘못 배치되어 있습니다.

예제

```
pg 1.5: up=acting: [0,1,2]
ADD_OSD_3
pg 1.5: up: [0,3,1] acting: [0,1,2]
```

[0,1,2]는 임시 매핑이므로 **up** 세트는 작동 세트와 같지 않으며 **PG**는 **[0,1,2]**이 여전히 세 복사본이므로 성능이 저하되지는 않습니다.

예제

```
pg 1.5: up=acting: [0,3,1]
```

이제 **OSD 3**이 백필되어 있으며 임시 매핑이 제거되고 성능이 저하되지 않고 잘못 배치되지 않습니다.

3.3.15. 배치 그룹 불완전한 상태

PG는 불완전 한 콘텐츠가 있고 피어링에 실패할 때 불완전한 상태가 됩니다. 즉, 현재 복구 수행에 충분한 완전한 **OSD**가 없는 경우입니다.

OSD 1, 2, 3은 OSD 1, 4, 3으로 전환하고 OSD 1, 4, 3으로 전환한 다음 osd.1 은 OSD 1, 2, 3의 임시 동작 세트를 요청한다는 것을 의미합니다. 이 기간 동안 OSD 1, 2, 3이 모두 아래로 내려지면 osd.4 가 모든 데이터를 완전히 다시 채우지 않았을 수 있는 유일한 왼쪽입니다. 현재 PG는 복구 수행에 충분한 완전한 OSD가 없음을 나타냅니다.

또는 osd.4 가 관련이 없고 작동 중인 세트가 OSD 1, 2 및 3인 경우, PG는 작동 세트가 변경된 이후 해당 PG에서 해당 PG에서 아무것도 듣지 않았음을 나타내는 오래된 것으로 나타났습니다. OSD가 새 OSD에 알 수 없는 이유는 다음과 같습니다.

3.3.16. 중단된 배치 그룹 확인

배치 그룹은 active+clean 상태가 아니기 때문에 반드시 문제가 되는 것은 아닙니다. 일반적으로 배치 그룹이 중단될 때 Ceph를 자체 복구할 수 있는 기능이 작동하지 않을 수 있습니다. 정지 상태에는 다음이 포함됩니다.

- unclean: 배치 그룹에는 원하는 횟수를 복제하지 않은 오브젝트가 포함되어 있습니다. 그들은 회복되어야 합니다.
- inactive: 배치 그룹은 최신 데이터가 있는 OSD가 백업 될 때까지 대기하고 있기 때문에 읽기 또는 쓰기를 처리할 수 없습니다.
- stale : 배치 그룹은 호스트한 OSD가 잠시 동안 모니터 클러스터에 보고되지 않았으며 mon osd 보고서 시간 제한 설정으로 구성할 수 없기 때문에 알 수 없는 상태입니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. 정지된 배치 그룹을 식별하려면 다음을 실행합니다.

구문

```
ceph pg dump_stuck {inactive|unclean|stale|undersized|degraded
[inactive|unclean|stale|undersized|degraded...]} {<int>}
```

예제

```
[ceph: root@host01 /]# ceph pg dump_stuck stale
OK
```

3.3.17. 개체의 위치 찾기

Ceph 클라이언트는 최신 클러스터 맵을 검색하고 CRUSH 알고리즘은 오브젝트를 배치 그룹에 매핑하는 방법을 계산한 다음 배치 그룹을 동적으로 할당하는 방법을 계산합니다.

사전 요구 사항

- **실행 중인 Red Hat Ceph Storage 클러스터.**
- **노드에 대한 루트 수준 액세스입니다.**

절차

1. **오브젝트 위치를 찾으려면 오브젝트 이름과 풀 이름이 필요합니다.**

구문

```
ceph osd map POOL_NAME OBJECT_NAME
```

예제

```
[ceph: root@host01 /]# ceph osd map mypool myobject
```

4장. CEPH 동작 덮어쓰기

스토리지 관리자는 **Red Hat Ceph Storage** 클러스터에 대한 재정의를 사용하여 런타임 중에 **Ceph** 옵션을 변경하는 방법을 이해해야 합니다.

4.1. 사전 요구 사항

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

4.2. CEPH 덮어쓰기 설정 및 설정 해제 옵션

Ceph의 기본 동작을 재정의하기 위해 **Ceph** 옵션을 설정하고 설정 해제할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. **Ceph**의 기본 동작을 재정의하려면 **ceph osd set** 명령과 재정의하려는 동작을 사용합니다.

   구문

   ```
   ceph osd set FLAG
   ```

   동작을 설정하면 **ceph** 상태가 클러스터에 설정된 덮어쓰기를 반영합니다.

   예제

```
[ceph: root@host01 /]# ceph osd set noout
```

2. **Ceph의 기본 동작 재정의를 중단하려면 ceph osd unset 명령과 중단하려는 덮어쓰기를 사용합니다.**

구문

```
ceph osd unset FLAG
```

예제

```
[ceph: root@host01 /]# ceph osd unset noout
```

플래그	설명
noin	OSD가 클러스터에서 로 취급되지 않도록 합니다.
noout	OSD가 클러스터에서 처리되지 않도록 합니다.
noup	OSD가 실행 중 **으로** 처리되지 않도록 합니다.
nodown	OSD가 **다운된** 것으로 취급되지 않도록 합니다.
full	클러스터가 **full_ratio** 에 도달한 것으로 표시되고 이로 인해 쓰기 작업이 방지됩니다.

플래그	설명
pause	Ceph는 읽기 및 쓰기 작업을 중지하지만 ,**out** 또는 **down 상태의 OSD에는** 영향을 미치지 않습니다.
nobackfill	Ceph는 새 백필(backfill) 작업을 방지합니다.
norebalance	Ceph는 새로운 재조정 작업을 방지합니다.
norecover	Ceph는 새로운 복구 작업을 방지합니다.
noscrub	Ceph는 새로운 스크럽 작업을 방지합니다.
nodeep-scrub	Ceph는 새로운 딥 스크럽 작업을 방지합니다.
notieragent	Ceph는 cold/dirty 오브젝트를 플러시하고 제거할 프로세스를 비활성화합니다.

4.3. CEPH 덮어쓰기 사용 사례

- **noin: 새로 고침 OSD를 해결하기 위해 noout 과 함께 일반적으로 사용됩니다.**
- **no out : mon osd 보고서 시간 초과가 초과되고 OSD가 모니터에 보고되지 않은 경우 OSD 가 표시됩니다. 오류가 발생하면 문제를 해결하는 동안 OSD가 표시되지 않도록 noout 를 설정할 수 있습니다.**
- **noup: 일반적으로 nodown 과 함께 사용하여 플로팅 OSD를 해결합니다.**
- **nodown: 네트워킹 문제는 Ceph 'heartbeat' 프로세스를 중단할 수 있으며 OSD는 시작되었지만 계속 표시될 수 있습니다. 문제를 해결하는 동안 OSD가 표시되지 않도록 nodown 을 설정할 수 있습니다.**
- **full : 클러스터가 full_ratio 에 도달하면 클러스터를 선점하여 용량을 넓히도록 설정할 수 있습니다.**

참고

클러스터를 **full** 로 설정하면 쓰기 작업이 금지됩니다.

- **일시 중지:** 클라이언트가 데이터를 읽고 작성하지 않고 실행 중인 **Ceph** 클러스터의 문제를 해결해야 하는 경우 클라이언트 작업을 방지하기 위해 일시 중지 하도록 클러스터를 설정할 수 있습니다.

- **nobackfill:** 예를 들어 데몬을 업그레이드하고 데몬을 업그레이드해야하는 경우 **OSD**가 다운되는 동안 **Ceph**가 다시 입력되지 않도록 **nobackfill** 을 설정할 수 있습니다.

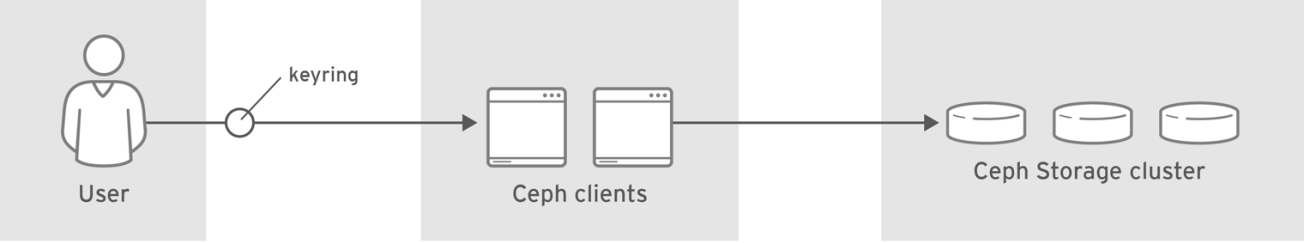
- **noecover: OSD** 디스크를 교체해야 하며 **PG**가 디스크를 핫스팅하는 동안 다른 **OSD**로 복구할 필요가 없는 경우 다른 **OSD**가 새 **PG** 세트를 다른 **OSD**에 복사하지 못하도록 하거나 무효화할 수 있습니다.

- **noscrub** 및 **nodeep-scrubb:** 예를 들어 높은 부하, 복구, 백필링 및 재조정 중에 오버헤드를 줄이기 위해 **noscrub** 및/또는 **nodeep-scrub** 를 설정하여 클러스터가 정지되지 않도록 할 수 있습니다.

- **notieragent:** 계층 에이전트 프로세스가 백업 스토리지 계층으로 플러시하도록 콜드 오브젝트를 찾는 것을 중지하려면 **notieragent** 를 설정할 수 있습니다.

5장. CEPH 사용자 관리

스토리지 관리자는 인증을 제공하고 **Red Hat Ceph Storage** 클러스터의 오브젝트에 대한 액세스 제어를 통해 **Ceph** 사용자 기반을 관리할 수 있습니다.



5.1. 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ceph Monitor** 또는 **Ceph** 클라이언트 노드에 액세스할 수 있습니다.

중요

Cephadm은 클라이언트가 **Cephadm** 범위 내에 있는 경우 **Red Hat Ceph Storage** 클러스터의 클라이언트 인증 키를 관리합니다. 문제를 해결하지 않는 한 사용자가 **Cephadm**에서 관리하는 인증 키를 수정하지 않아야 합니다.

5.2. CEPH 사용자 관리 배경

Ceph가 인증 및 권한 부여를 활성화하면 사용자 이름을 지정해야 합니다. 사용자 이름을 지정하지 않으면 **Ceph**에서 **client.admin** 관리 사용자를 기본 사용자 이름으로 사용합니다.

또는**CEPH _ARGS** 환경 변수를 사용하여 사용자 이름 및 시크릿을 다시 입력하지 않도록 할 수 있습니다.

Ceph 클라이언트 유형(예: 블록 장치, 오브젝트 저장소, 파일 시스템, 네이티브 **API** 또는 **Ceph** 명령줄)에 관계없이 **Ceph**는 모든 데이터를 풀 내의 오브젝트로 저장합니다. **Ceph** 사용자는 데이터를 읽고 쓰려면 풀에 액세스할 수 있어야 합니다. 또한 관리 **Ceph** 사용자는 **Ceph**의 관리 명령을 실행할 수 있는 권한이 있어야 합니다.

다음 개념은 **Ceph** 사용자 관리를 이해하는 데 도움이 될 수 있습니다.

스토리지 클러스터 사용자

Red Hat Ceph Storage 클러스터의 사용자는 개별 또는 애플리케이션으로 사용됩니다. 사용자를 생성하면 스토리지 클러스터, 해당 풀 및 해당 풀 내의 데이터에 액세스할 수 있는 사용자를 제어할 수 있습니다.

Ceph에는 사용자 유형의 개념이 있습니다. 사용자 관리의 목적으로 유형은 항상 클라이언트 가 됩니다. **Ceph**는 사용자 유형과 사용자 **ID**로 구성된 마침표(.)로 구분된 형식의 사용자를 식별합니다. 예를 들면 **TYPE.ID,client.admin** 또는 **client.user1** 입니다. 사용자 입력의 이유는 **Ceph Monitors**가 **Cephx** 프로토콜도 사용하지만 클라이언트가 아니기 때문입니다. 사용자 유형을 구분하면 클라이언트 사용자와 기타 사용자를 구별하는 데 도움이 되며 액세스 제어, 사용자 모니터링 및 추적 기능이 간소화됩니다.

Ceph 명령줄을 사용하면 명령줄 사용량에 따라 또는 유형 없이 사용자를 지정할 수 있으므로 **Ceph**의 사용자 유형이 혼동될 수 있습니다. **--user** 또는 **--id** 를 지정하는 경우 유형을 생략할 수 있습니다. 따라서 **client.user1** 은 **user1** 로서 간단히 입력할 수 있습니다. **--name** 또는 **-n** 을 지정하는 경우 **type** 및 **name(예: client.user1)**을 지정해야 합니다. 유형 및 이름을 가능한 한 모범 사례로 사용하는 것이 좋습니다.

참고

Red Hat Ceph Storage 클러스터 사용자는 **Ceph Object Gateway** 사용자와 동일하지 않습니다. 오브젝트 게이트웨이는 **Red Hat Ceph Storage** 클러스터 사용자를 사용하여 게이트웨이 데몬과 스토리지 클러스터 간 통신하지만 게이트웨이에는 최종 사용자를 위한 자체 사용자 관리 기능이 있습니다.

권한 부여 기능

Ceph는 **"caps"**라는 용어를 사용하여 인증된 사용자 인증을 통해 **Ceph** 모니터 및 **OSD**의 기능을 수행합니다. 또한 기능은 풀 내의 데이터 또는 풀 내의 네임스페이스에 대한 액세스를 제한할 수 있습니다. **Ceph** 관리 사용자는 사용자를 생성하거나 업데이트할 때 사용자의 기능을 설정합니다. 기능 구문은 형식을 따릅니다.

구문

```
DAEMON_TYPE 'allow CAPABILITY' [DAEMON_TYPE 'allow CAPABILITY']
```

- **모니터 기능: 모니터 기능에는 r,w,x,allow 프로필 *CAP*, rbd 등이 있습니다.**

예제

```
mon 'allow rwx`
mon 'allow profile osd'
```

- **OSD 캡처: OSD 기능에는 r,w,x,class-read,class-write,프로필 osd,profile rbd, profile rbd-only 가 포함됩니다. 또한 OSD 기능은 풀 및 네임스페이스 설정을 허용합니다. :**

구문

```
osd 'allow CAPABILITY' [pool=POOL_NAME] [namespace=NAMESPACE_NAME]
```

참고

Ceph Object Gateway 데몬(radosgw)은 Ceph 스토리지 클러스터 클러스터 클러스터의 클라이언트이므로 Ceph 스토리지 클러스터 데몬 유형으로 표시되지 않습니다.

다음 항목에서는 각 기능을 설명합니다.

allow	데몬의 액세스 설정 우선 순위입니다.
r	사용자에게 읽기 액세스 권한을 부여합니다. CRUSH 맵을 검색하려면 모니터에 필요합니다.
w	사용자에게 오브젝트에 대한 쓰기 액세스 권한을 제공합니다.
x	사용자에게 클래스 메서드(즉, 읽기 및 쓰기)를 호출하고 모니터에서 **인증** 작업을 수행할 수 있는 기능을 제공합니다.
class-read	사용자에게 클래스 읽기 메서드를 호출할 수 있는 기능을 제공합니다. **x** 의 서브 세트입니다.

class-write	사용자에게 클래스 쓰기 메서드를 호출할 수 있는 기능을 제공합니다. **x** 의 서브 세트입니다.
*	사용자에게 특정 데몬 또는 풀에 대한 읽기, 쓰기, 실행 권한과 admin 명령을 실행할 수 있는 기능을 제공합니다.
프로필 osd	사용자에게 OSD로 다른 OSD 또는 모니터에 연결할 수 있는 권한을 제공합니다. OSD에서 복제 하트비트 트래픽 및 상태 보고를 처리할 수 있도록 OSD에 대한 승인.
프로필 bootstrap-osd	OSD를 부트스트랩할 때 키를 추가할 수 있는 권한이 있도록 OSD를 부트스트랩할 수 있는 권한을 사용자에게 제공합니다.
rbd 프로필	사용자에게 Ceph 블록 장치에 대한 읽기-쓰기 액세스 권한을 부여합니다.
rbd-read-only 프로필	사용자에게 Ceph 블록 장치에 대한 읽기 전용 액세스 권한을 부여합니다.

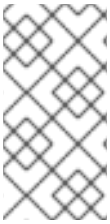
pool

풀은 Ceph 클라이언트에 대한 스토리지 전략을 정의하고 해당 전략의 논리 파티션 역할을 합니다.

Ceph 배포에서는 다양한 유형의 사용 사례를 지원하는 풀을 생성하는 것이 일반적입니다. 예를 들어 클라우드 볼륨 또는 이미지, 오브젝트 스토리지, 핫 스토리지, 콜드 스토리지 등이 있습니다. Ceph를 OpenStack의 백엔드로 배포하는 경우 일반적인 배포에는 볼륨, 이미지, 백업 및 가상 머신의 풀과 client.glance,client.cinder 등과 같은 사용자가 있습니다.

네임스페이스

풀 내의 오브젝트는 풀 내의 개체의 네임스페이스-일a 논리 그룹에 연결할 수 있습니다. 사용자가 풀 내에서 읽고 쓸 수 있도록 풀에 대한 사용자의 액세스 권한을 네임스페이스와 연결할 수 있습니다. 풀 내의 네임스페이스에 작성된 오브젝트는 네임스페이스에 액세스할 수 있는 사용자만 액세스할 수 있습니다.

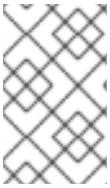


참고

현재 네임스페이스는 librados 상단에 작성된 애플리케이션에만 유용합니다. 블록 장치 및 오브젝트 스토리지와 같은 Ceph 클라이언트는 현재 이 기능을 지원하지 않습니다.

네임스페이스의 논리는 풀이 OSD에 매핑되는 배치 그룹 집합을 생성하므로 사용 사례별로 데이터를 분리하는 컴퓨팅 비용이 많이 드는 방법일 수 있다는 것입니다. 여러 풀이 동일한 CRUSH 계층 구조와 규칙 세트를 사용하는 경우 로드가 증가함에 따라 OSD 성능이 저하될 수 있습니다.

예를 들어, 풀에는 OSD당 약 100개의 배치 그룹이 있어야 합니다. 따라서 1000개의 OSD가 있는 예시적 클러스터에는 하나의 풀에 대해 10,000개의 배치 그룹이 있습니다. 동일한 CRUSH 계층 구조에 매핑된 각 풀과 ruleset은 예시 클러스터에 또 다른 10,000개의 배치 그룹을 생성합니다. 반대로 네임스페이스에 오브젝트를 작성하면 네임스페이스를 별도의 풀의 계산 오버헤드와 개체 이름에 연결하면 됩니다. 사용자 또는 사용자 집합에 대해 별도의 풀을 생성하는 대신 네임스페이스를 사용할 수 있습니다.



참고

현재 librados 를 통해서만 사용 가능합니다.

추가 리소스

- 인증 사용 구성에 대한 자세한 내용은 *Red Hat Ceph Storage 구성 가이드* 를 참조하십시오.

5.3. CEPH 사용자 관리

스토리지 관리자는 사용자를 생성, 수정, 삭제 및 가져와 Ceph 사용자를 관리할 수 있습니다. Ceph 클라이언트 사용자는 Ceph 클라이언트를 사용하여 Red Hat Ceph Storage 클러스터 데몬과 상호 작용하는 개인 또는 애플리케이션일 수 있습니다.

5.3.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- Ceph Monitor 또는 Ceph 클라이언트 노드에 액세스할 수 있습니다.

5.3.2. Ceph 사용자 나열

명령줄 인터페이스를 사용하여 스토리지 클러스터의 사용자를 나열할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. **스토리지 클러스터의 사용자를 나열하려면 다음을 실행합니다.**

예제

```
[ceph: root@host01 /]# ceph auth list
installed auth entries:

osd.10
 key: AQBW7U5gqOsEExAAg/CxSwZ/gSh8iOsDV3iQOA==
 caps: [mgr] allow profile osd
 caps: [mon] allow profile osd
 caps: [osd] allow *
osd.11
 key: AQBX7U5gtj/JIhAAPsLBNG+SfC2eMVEFkl3vfA==
 caps: [mgr] allow profile osd
 caps: [mon] allow profile osd
 caps: [osd] allow *
osd.9
 key: AQBV7U5g1XDULhAAKo2tw6ZhH1jki5aVui2v7g==
 caps: [mgr] allow profile osd
 caps: [mon] allow profile osd
 caps: [osd] allow *
client.admin
 key: AQADYEtgFfD3ExAAwH+C1qO7MSLE4TWRfD2g6g==
 caps: [mds] allow *
 caps: [mgr] allow *
 caps: [mon] allow *
 caps: [osd] allow *
client.bootstrap-mds
 key: AQAHYEtgpbkANBAANqoFlvzEXFwD8oB0w3TF4Q==
 caps: [mon] allow profile bootstrap-mds
client.bootstrap-mgr
 key: AQAHYEtg3dcANBAAVQf6brq3sxTSrCrPe0pKVQ==
 caps: [mon] allow profile bootstrap-mgr
client.bootstrap-osd
 key: AQAHYEtgD/QANBAATS9DuP3DbxEl86MTyKEmdw==
 caps: [mon] allow profile bootstrap-osd
client.bootstrap-rbd
 key: AQAHYEtgjxEBNBAANho25V9tWNNvIKnHknW59A==
 caps: [mon] allow profile bootstrap-rbd
client.bootstrap-rbd-mirror
 key: AQAHYEtgdE8BNBAAr6rLYxZci0b2hoIgH9GXYw==
 caps: [mon] allow profile bootstrap-rbd-mirror
client.bootstrap-rgw
 key: AQAHYEtgwGkBNBAAuRzI4WSrnowBhZxr2XtTFg==
 caps: [mon] allow profile bootstrap-rgw
client.crash.host04
 key: AQCQYEtgz8lGGhAAy5bJS8VH9fMdxuAZ3CqX5Q==
 caps: [mgr] profile crash
```

```
 caps: [mon] profile crash
client.crash.host02
 key: AQDuYUtgqgfdOhAAsyX+Mo35M+HFpURGad7nJA==
 caps: [mgr] profile crash
 caps: [mon] profile crash
client.crash.host03
 key: AQB98E5g5jHZAxAAklWSvmDsh2JaL5G7FvMrrA==
 caps: [mgr] profile crash
 caps: [mon] profile crash
client.nfs.foo.host03
 key: AQCgTk9gm+HvMxAAHbjG+XpdwL6prM/uMcdPdQ==
 caps: [mon] allow r
 caps: [osd] allow rw pool=nfs-ganesha namespace=foo
client.nfs.foo.host03-rgw
 key: AQCgTk9g8sJQNhAAPykcoYUuPc7ljubaFx09HQ==
 caps: [mon] allow r
 caps: [osd] allow rwx tag rgw *=*
client.rgw.test_realm.test_zone.host01.hgbvnq
 key: AQD5RE9gAQKdCRAAJzxDwD/dJObbInp9J95sXw==
 caps: [mgr] allow rw
 caps: [mon] allow *
 caps: [osd] allow rwx tag rgw *=*
client.rgw.test_realm.test_zone.host02.yqqilm
 key: AQD0RE9gkxA4ExAAFXp3pLJWdIhsyTe2ZR6Ilw==
 caps: [mgr] allow rw
 caps: [mon] allow *
 caps: [osd] allow rwx tag rgw *=*
mgr.host01.hdhzwn
 key: AQAEYEtg3lhIBxAAmHodoIpdvnxK0llWF80ltQ==
 caps: [mds] allow *
 caps: [mon] profile mgr
 caps: [osd] allow *
mgr.host02.eobuuv
 key: AQAn6U5gzUuiABAA2Fed+jPM1xwb4XDYtrQxaQ==
 caps: [mds] allow *
 caps: [mon] profile mgr
 caps: [osd] allow *
mgr.host03.wquwpj
 key: AQAd6U5gIzWsLBAAbOKUKZlUcAVe9kBLfajMKw==
 caps: [mds] allow *
 caps: [mon] profile mgr
 caps: [osd] allow *
```

참고

사용자에 대한 **TYPE.ID** 표기법은 **osd.0** 유형의 사용자이고 해당 **ID**는 **0** 이며,**client.admin** 은 **client** 유형의 사용자이며, 즉 기본 **client.admin** 사용자는 **admin** 입니다. 또한 각 항목에 ***VALUE*** 항목이 있고 하나 이상의 **caps:** 항목이 있습니다.

ceph auth list 와 함께 **-o *FILE_NAME*** 옵션을 사용하여 출력을 파일에 저장할 수 있습니다.

5.3.3. Ceph 사용자 정보 표시

명령줄 인터페이스를 사용하여 **Ceph**의 사용자 정보를 표시할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. 특정 사용자, 키 및 기능을 검색하려면 다음을 실행합니다.

   구문

   ```
   ceph auth export TYPE.ID
   ```

   예제

   ```
   [ceph: root@host01 /]# ceph auth export mgr.host02.eobuuv
   ```

2. **-o *FILE_NAME*** 옵션을 사용할 수도 있습니다.

   구문

```
ceph auth export TYPE.ID -o FILE_NAME
```

예제

```
[ceph: root@host01 /]# ceph auth export osd.9 -o filename
export auth(key=AQBV7U5g1XDULhAAKo2tw6ZhH1jki5aVui2v7g==)
```

auth export 명령은 **auth get** 와 동일하지만 최종 사용자와 관련이 없는 내부 **auid** 도 출력합니다.

5.3.4. 새 Ceph 사용자 추가

사용자를 추가하면 사용자 이름(**type *.ID***), 시크릿 키, 사용자를 생성하는 데 사용하는 명령에 포함된 모든 기능이 생성됩니다.

사용자의 키를 사용하면 **Ceph** 스토리지 클러스터로 인증할 수 있습니다. 사용자의 기능은 사용자가 **Ceph** 모니터(**mon**), **Ceph OSD**(**osd**) 또는 **Ceph** 메타데이터 서버(**mds**)에서 읽기, 쓰기 또는 실행할 수 있는 권한을 부여합니다.

사용자를 추가하는 몇 가지 방법이 있습니다.

- **Ceph auth add:** 이 명령은 사용자를 추가하는 표준 방법입니다. 사용자를 생성하고 키를 생성하고 지정된 기능을 추가합니다.
- **Ceph auth get-or-create:** 이 명령은 사용자 이름(대괄호) 및 키를 사용하여 키 파일 형식을 반환하기 때문에 사용자를 생성하는 가장 편리한 방법입니다. 사용자가 이미 존재하는 경우 이 명령은 키 파일 형식으로 사용자 이름과 키를 반환하기만 하면 됩니다. **-o *FILE_NAME*** 옵션을 사용하여 출력을 파일에 저장할 수 있습니다.
- **Ceph auth get-or-create-key:** 이 명령은 사용자를 생성하고 사용자의 키만 반환하는 편리한 방법입니다. 이 명령은 키만 필요한 클라이언트(예: **libvirt**)에 유용합니다. 사용자가 이미 있는 경

우 이 명령은 키를 반환하기만 하면 됩니다. **-o *FILE_NAME*** 옵션을 사용하여 출력을 파일에 저장할 수 있습니다.

클라이언트 사용자를 생성할 때 기능 없이 사용자를 생성할 수 있습니다. 기능이 없는 사용자는 단순히 인증보다 쓸모가 없습니다. 클라이언트는 모니터에서 클러스터 맵을 검색할 수 없기 때문입니다. 그러나 **ceph auth caps** 명령을 사용하여 나중에 기능 추가를 미끄러면 기능이 없는 사용자를 생성할 수 있습니다.

일반적인 사용자는 **Ceph** 모니터에서 최소한 읽기 및 쓰기 기능을 **Ceph OSD**에서 읽고 쓸 수 있습니다. 또한 사용자의 **OSD** 권한은 특정 풀에 액세스하는 경우가 많습니다. :

```
[ceph: root@host01 /]# ceph auth add client.john mon 'allow r' osd 'allow rw pool=mypool'
[ceph: root@host01 /]# ceph auth get-or-create client.paul mon 'allow r' osd 'allow rw pool=mypool'
[ceph: root@host01 /]# ceph auth get-or-create client.george mon 'allow r' osd 'allow rw pool=mypool'
-o george.keyring
[ceph: root@host01 /]# ceph auth get-or-create-key client.ringo mon 'allow r' osd 'allow rw
pool=mypool' -o ringo.key
```

중요

OSD에 기능이 있지만 특정 풀에 대한 액세스를 제한하지 않으면 사용자는 클러스터의 모든 풀에 액세스할 수 있습니다.

5.3.5. Ceph 사용자 수정

ceph auth caps 명령을 사용하면 사용자를 지정하고 사용자의 기능을 변경할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. 기능을 추가하려면 다음 양식을 사용하십시오.

구문

```
ceph auth caps USERTYPE.USERID DAEMON 'allow [r|w|x|*|...] [pool=POOL_NAME]
[namespace=NAMESPACE_NAME]'
```

예제

```
[ceph: root@host01 /]# ceph auth caps client.john mon 'allow r' osd 'allow rw pool=mypool'
[ceph: root@host01 /]# ceph auth caps client.paul mon 'allow rw' osd 'allow rwx pool=mypool'
[ceph: root@host01 /]# ceph auth caps client.brian-manager mon 'allow *' osd 'allow *'
```

2. 기능을 제거하려면 기능을 재설정할 수 있습니다. 사용자가 이전에 설정된 특정 데몬에 대한 액세스 권한이 없도록 하려면 빈 문자열을 지정합니다.

예제

```
[ceph: root@host01 /]# ceph auth caps client.ringo mon ' ' osd ' '
```

추가 리소스

- *기능에 대한 자세한 내용은 권한 부여* 기능을 참조하십시오.

5.3.6. Ceph 사용자 삭제

명령줄 인터페이스를 사용하여 **Ceph** 스토리지 클러스터에서 사용자를 삭제할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. 사용자를 삭제하려면 **ceph auth del** 을 사용합니다.

구문

```
ceph auth del TYPE.ID
```

예제

```
[ceph: root@host01 /]# ceph auth del osd.6
```

5.3.7. Ceph 사용자 키 출력

명령줄 인터페이스를 사용하여 **Ceph** 사용자의 주요 정보를 표시할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1.

1. 사용자의 인증 키를 표준 출력에 출력하려면 다음을 실행합니다.

구문

```
ceph auth print-key TYPE.ID
```

예제

```
[ceph: root@host01 /]# ceph auth print-key osd.6

AQBQ7U5gAry3JRAA3NoPrqBBThpFMcRL6Sr+5w==[ceph: root@host01 /]#
```

2. 사용자 키를 인쇄하는 것은 사용자 키를 사용하여 클라이언트 소프트웨어를 채워야 할 때 유용합니다(예: libvirt).

구문

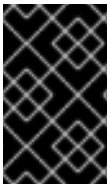
```
mount -t ceph HOSTNAME:/MOUNT_POINT -o name=client.user,secret=ceph auth print-key
client.user
```

예제

```
[ceph: root@host01 /]#  mount -t ceph host02:/ceph -o name=client.user,secret=`ceph auth
print-key client.user`
```

6장. CEPH-VOLUME 유틸리티

스토리지 관리자는 **ceph-volume** 유틸리티를 사용하여 **Ceph OSD**를 준비, 나열, 생성, 활성화, 활성화, 비활성화, 배치, 트리거, **zap**, 마이그레이션할 수 있습니다. **ceph-volume** 유틸리티는 논리 볼륨을 **OSD**로 배포하는 단일 용도의 명령줄 툴입니다. 플러그인 유형 프레임워크를 사용하여 다양한 장치 기술로 **OSD**를 배포합니다. **ceph-volume** 유틸리티는 **OSD**를 배포할 수 있는 **ceph-disk** 유틸리티의 유사한 워크플로를 따르며, **OSD**를 준비, 활성화 및 시작하는 데 필요한 강력한 방법으로 **OSD**를 배포합니다. 현재 **ceph-volume** 유틸리티는 향후 다른 기술을 지원할 계획과 함께 **lvm** 플러그인만 지원합니다.



중요

ceph-disk 명령은 더 이상 사용되지 않습니다.

6.1. 사전 요구 사항

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

6.2. CEPH 볼륨 LVM 플러그인

LVM 태그를 사용하면 **lvm** 하위 명령을 사용하여 **OSD**와 연결된 장치를 쿼리하여 해당 태그를 활성화하고 다시 검색할 수 있습니다. 여기에는 **dm-cache** 와 같은 **lvm** 기반 기술도 지원됩니다.

ceph-volume 를 사용하는 경우 **dm-cache** 를 사용할 수 없으며 논리 볼륨과 같이 **dm-cache** 를 처리합니다. **dm-cache** 를 사용할 때 성능 향상 및 손실은 특정 워크로드에 따라 달라집니다. 일반적으로 임의의 읽기 및 순차적 읽기는 작은 블록 크기에서 성능이 향상되는 것을 볼 수 있습니다. 무작위 및 순차적 쓰기는 더 큰 블록 크기에서 성능이 저하되는 것을 볼 수 있습니다.

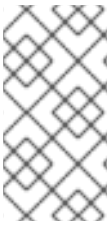
LVM 플러그인을 사용하려면 **cephadm** 쉘 내의 **ceph-volume** 명령에 **lvm** 을 하위 명령으로 추가합니다.

```
[ceph: root@host01 /]# ceph-volume lvm
```

lvm 하위 명령은 다음과 같습니다.

- 준비 **- LVM** 장치를 포맷하여 **OSD**와 연결합니다.

- 활성화 - OSD ID와 연결된 LVM 장치를 검색하고 마운트하고 Ceph OSD를 시작합니다.
- list - Ceph와 연결된 논리 볼륨 및 장치를 나열합니다.
- 일괄 처리 - 최소한의 상호 작용을 통해 다중 OSD 프로비저닝을 위한 자동 장치 크기.
- 비활성화 - OSD 비활성화.
- Create - LVM 장치에서 새 OSD를 만듭니다.
- Trigger - OSD를 활성화하는 systemd 도우미입니다.
- zap - 논리 볼륨 또는 파티션에서 모든 데이터와 파일 시스템을 제거합니다.
- 마이그레이션 - BlueFS 데이터를 다른 LVM 장치로 마이그레이션합니다.
- new-wal - 지정된 논리 볼륨에서 OSD에 대한 새 WAL 볼륨을 할당합니다.
- new-db - 지정된 논리 볼륨에서 OSD의 새 DB 볼륨을 할당합니다.



참고

create 하위 명령을 사용하면 prepare 및 activate 하위 명령이 하나의 하위 명령으로 결합됩니다.

추가 리소스

- 자세한 내용은 create 하위 명령 섹션 을 참조하십시오.

6.3. CEPH-VOLUME 이 CEPH-DISK 를 대체하는 이유는 무엇입니까?

이전 버전의 **Red Hat Ceph Storage**에서는 **ceph-disk** 유틸리티를 사용하여 **OSD**를 준비, 활성화 및 생성했습니다. **Red Hat Ceph Storage 5**부터 **ceph-disk** 는 **OSD**로 논리 볼륨을 배포하는 단일 용도의 명령줄 툴을 목표로 하고 **OSD**를 준비, 활성화 및 생성할 때 **ceph-disk**와 유사한 **API**를 유지하는 것을 목표로 하는 **ceph-volume** 유틸리티로 교체됩니다.

ceph-volume 은 어떻게 작동합니까?

ceph-volume 은 현재 하드웨어 장치, 레거시 **ceph-disk** 장치 및 **LVM(Logical Volume Manager)** 장치를 프로비저닝하는 두 가지 방법을 지원하는 모듈식 툴입니다. **ceph-volume lvm** 명령은 **LVM** 태그를 사용하여 **Ceph** 관련 장치와 **OSD**와의 관계에 대한 정보를 저장합니다. 나중에 이러한 태그를 사용하여 **OSDS**와 연결된 장치를 다시 검색하고 쿼리하여 활성화할 수 있습니다. **LVM** 및 **dm-cache** 를 기반으로 하는 기술도 지원합니다.

ceph-volume 유틸리티는 **dm-cache** 를 투명하게 사용하고 이를 논리 볼륨으로 처리합니다. 처리 중인 특정 워크로드에 따라 **dm-cache** 를 사용할 때 성능 향상 및 손실을 고려할 수 있습니다. 일반적으로 무작위 및 순차적 읽기 작업의 성능은 더 작은 블록 크기로 증가하며, 무작위 및 순차적 쓰기 작업의 성능은 더 큰 블록 크기로 감소합니다. **ceph-volume** 을 사용하면 상당한 성능 저하가 발생하지 않습니다.

중요

ceph-disk 유틸리티는 더 이상 사용되지 않습니다.

참고

ceph-volume simple 명령은 이러한 장치를 아직 사용 중인 경우 기존 **ceph-disk** 장치를 처리할 수 있습니다.

ceph-disk 는 어떻게 작동합니까?

ceph-disk 유틸리티는 **upstart** 또는 **sysvinit** 와 같은 다양한 **init** 시스템을 지원하는 동시에 장치를 검색할 수 있어야 합니다. 이러한 이유로 **ceph-disk** 는 **GUID** 파티션 테이블**(GPT)** 파티션에만 집중합니다. 특히 다음과 같은 질문에 답변할 수 있는 고유한 방법으로 장치에 레이블을 지정하는 **GPT GUID**입니다.

- 이 장치는 저널 입니까?
- 이 장치는 암호화된 데이터 파티션입니까?
- 장치가 부분적으로 준비됩니까?

이러한 문제를 해결하기 위해 **ceph-disk** 는 **UDEV** 규칙을 사용하여 **GUID**와 일치시킵니다.

ceph-disk 사용의 단점은 무엇입니까?

UDEV 규칙을 사용하여 **ceph-disk** 를 호출하면 **ceph-disk systemd** 장치와 **ceph-disk** 실행 파일 사이에 **back-and-forth**가 발생할 수 있습니다. 이 프로세스는 매우 신뢰할 수 없으며 시간이 오래 걸리며 노드의 부팅 프로세스 중에 **OSD**가 전혀 발생하지 않을 수 있습니다. 또한 **UDEV**의 비동기 동작을 고려하여 디버깅하거나 이러한 문제를 복제하기가 어렵습니다.

ceph-disk 는 **GPT** 파티션에서만 작동하므로 **LVM(Logical Volume Manager)** 볼륨 또는 유사한 장치 매퍼 장치와 같은 다른 기술을 지원할 수 없습니다.

GPT 파티션이 장치 검색 워크플로우에서 올바르게 작동하도록 하려면 **ceph-disk** 를 사용하려면 많은 수의 특수 플래그를 사용해야 합니다. 또한 이러한 파티션은 **Ceph**에서 장치를 독점적으로 소유해야 합니다.

6.4. CEPH-VOLUME을 사용하여 CEPH OSD 준비

prepare 하위 명령은 **OSD** 백엔드 오브젝트 저장소를 준비하고 **OSD** 데이터와 저널 모두에 논리 볼륨(**LV**)을 사용합니다. **LVM**을 사용하여 일부 메타데이터 태그를 추가하는 경우를 제외하고 논리 볼륨은 수정하지 않습니다. 이러한 태그를 사용하면 볼륨을 더 쉽게 검색할 수 있으며, 볼륨을 **Ceph Storage** 클러스터의 일부로 식별하고 스토리지 클러스터에서 해당 볼륨의 역할도 식별합니다.

BlueStore OSD 백엔드는 다음 구성을 지원합니다.

- 블록 장치, **block.wal** 장치, **block.db** 장치
- 블록 장치 및 **block.wal** 장치
- 블록 장치 및 **block.db** 장치
- 단일 블록 장치

prepare 하위 명령은 전체 장치 또는 파티션 또는 블록에 논리 볼륨을 허용합니다.

사전 요구 사항

- **OSD** 노드에 대한 루트 수준 액세스.
- 선택적으로 논리 볼륨을 생성합니다. 물리 장치에 대한 경로를 제공하면 하위 명령이 장치를 논리 볼륨으로 전환합니다. 이 방법은 더 간단하지만 논리 볼륨이 생성되는 방식을 구성하거나 변경할 수 없습니다.

절차

1. **LVM** 볼륨을 준비합니다.

   구문

   ```
   ceph-volume lvm prepare --bluestore --data VOLUME_GROUP/LOGICAL_VOLUME
   ```

   예제

   ```
   [ceph: root@host01 /]# ceph-volume lvm prepare --bluestore --data example_vg/data_lv
   ```

   a. 필요한 경우 **MigsDB**에 대해 별도의 장치를 사용하려면 **--block.db** 및 **--block.wal** 옵션을 지정합니다.

      구문

      ```
      ceph-volume lvm prepare --bluestore --block.db --block.wal --data
      VOLUME_GROUP/LOGICAL_VOLUME
      ```

예제

```
[ceph: root@host01 /]# ceph-volume lvm prepare --bluestore --block.db --block.wal --
data example_vg/data_lv
```

b. **선택적으로 데이터를 암호화하려면 --dmcrypt 플래그를 사용합니다.**

구문

```
ceph-volume lvm prepare --bluestore --dmcrypt --data
VOLUME_GROUP/LOGICAL_VOLUME
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm prepare --bluestore --dmcrypt --data
example_vg/data_lv
```

추가 리소스

- 자세한 내용은 ***Red Hat Ceph Storage 관리 가이드에서 'ceph-volume'을 사용하여 CephOSD 활성화*** 섹션을 참조하십시오.
- 자세한 내용은 ***Red Hat Ceph Storage 관리 가이드의 'ceph-volume'을 사용하여 CephOSD 생성*** 섹션을 참조하십시오.

6.5. CEPH-VOLUME을 사용하여 장치 나열

ceph-volume lvm list 하위 명령을 사용하여 **Ceph** 클러스터와 연결된 논리 볼륨 및 장치를 나열할 수 있습니다. 이에는 해당 검색을 허용하기에 충분한 메타데이터가 포함되어 있습니다. 출력은 장치와 연결된 **OSD ID**로 그룹화됩니다. 논리 볼륨의 경우 **devices** 키는 논리 볼륨과 연결된 물리 장치로 채워집니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ceph OSD** 노드에 대한 루트 수준 액세스.

절차

- **Ceph** 클러스터의 장치를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph-volume lvm list


====== osd.6 =======

  [block]       /dev/ceph-83909f70-95e9-4273-880e-5851612cbe53/osd-block-7ce687d9-07e7-4f8f-a34e-d1b0efb89920

      block device              /dev/ceph-83909f70-95e9-4273-880e-5851612cbe53/osd-block-7ce687d9-07e7-4f8f-a34e-d1b0efb89920
      block uuid                4d7gzX-Nzxp-UUG0-bNxQ-Jacr-l0mP-IPD8cX
      cephx lockbox secret
      cluster fsid              1ca9f6a8-d036-11ec-8263-fa163ee967ad
      cluster name              ceph
      crush device class        None
      encrypted                 0
      osd fsid                  7ce687d9-07e7-4f8f-a34e-d1b0efb89920
      osd id                    6
      osdspec affinity          all-available-devices
      type                      block
      vdo                       0
      devices                   /dev/vdc
```

6.6. CEPH-VOLUME을 사용하여 CEPH OSD 활성화

활성화 프로세스를 사용하면 부팅 시 **systemd** 장치를 활성화하므로 올바른 **OSD** 식별자와 해당 **UUID**를 활성화 및 마운트할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ceph OSD** 노드에 대한 루트 수준 액세스.
- **ceph-volume** 유틸리티로 준비된 **Ceph OSD.**

절차

1. **OSD** 노드에서 **OSD ID** 및 **OSD FSID**를 가져옵니다.

```
[ceph: root@host01 /]# ceph-volume lvm list
```

2. **OSD**를 활성화합니다.

구문

```
ceph-volume lvm activate --bluestore OSD_ID OSD_FSID
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm activate --bluestore 10 7ce687d9-07e7-4f8f-a34e-d1b0efb89920
```

활성화용으로 준비된 모든 **OSD**를 활성화하려면 **--all** 옵션을 사용합니다.

예제

```
[ceph: root@host01 /]# ceph-volume lvm activate --all
```

3. 선택적으로 **trigger** 하위 명령을 사용할 수 있습니다. 이 명령은 직접 사용할 수 없으며, **ceph-volume lvm**에 대한 입력을 프록시하도록 **systemd** 에서 사용합니다. 그러면 **systemd** 및 **startup**에서 들어오는 메타데이터를 구문 분석하고 **OSD**와 연결된 **UUID** 및 **ID**를 탐지합니다.

구문

```
ceph-volume lvm trigger SYSTEMD_DATA
```

여기서 ***SYSTEMD_DATA*** 는 ***OSD_ID-OSD_FSID*** 형식입니다.

예제

```
[ceph: root@host01 /]# ceph-volume lvm trigger 10 7ce687d9-07e7-4f8f-a34e-d1b0efb89920
```

추가 리소스

- 자세한 내용은 ***Red Hat Ceph Storage 관리 가이드의 'ceph-volume'을 사용하여 CephOSD 준비*** 섹션을 참조하십시오.
- 자세한 내용은 ***Red Hat Ceph Storage 관리 가이드의 'ceph-volume'을 사용하여 CephOSD***

생성 섹션을 참조하십시오.

6.7. CEPH-VOLUME을 사용하여 CEPH OSD 비활성화

ceph-volume lvm 하위 명령을 사용하여 **Ceph OSD**를 비활성화할 수 있습니다. 이 하위 명령은 볼륨 그룹과 논리 볼륨을 제거합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ceph OSD** 노드에 대한 루트 수준 액세스.
- **Ceph OSD**는 **ceph-volume** 유틸리티를 사용하여 활성화됩니다.

절차

1. **OSD** 노드에서 **OSD ID**를 가져옵니다.

```
[ceph: root@host01 /]# ceph-volume lvm list
```

2. **OSD**를 비활성화합니다.

구문

```
ceph-volume lvm deactivate OSD_ID
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm deactivate 16
```

추가 리소스

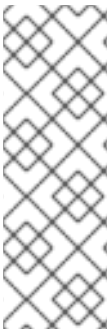
- 자세한 내용은 *Red Hat Ceph Storage 관리 가이드에서 'ceph-volume'을 사용하여 CephOSD 활성화* 섹션을 참조하십시오.
- 자세한 내용은 *Red Hat Ceph Storage 관리 가이드의 'ceph-volume'을 사용하여 CephOSD 준비* 섹션을 참조하십시오.
- 자세한 내용은 *Red Hat Ceph Storage 관리 가이드의 'ceph-volume'을 사용하여 CephOSD 생성* 섹션을 참조하십시오.

6.8. CEPH-VOLUME을 사용하여 CEPH OSD 생성

create 하위 명령은 **prepare** 하위 명령을 호출한 다음 **activate** 하위 명령을 호출합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ceph OSD** 노드에 대한 루트 수준 액세스.



참고

생성 프로세스를 더 많이 제어하려는 경우, **create**를 사용하는 대신 **prepare** 하위 명령을 별도로 사용하여 **OSD**를 만들 수 있습니다. 두 개의 하위 명령을 사용하여 대량의 데이터를 재조정하는 것을 피하면서 점진적으로 새 **OSD**를 스토리지 클러스터에 도입할 수 있습니다. 두 접근 방식은 **create** 하위 명령을 사용하면 **OSD**가 완료 후 즉시 가동되는 것을 제외하고는 동일한 방식으로 작동합니다.

절차

1. *새 OSD를 생성하려면 다음을 수행합니다.*

구문

```
ceph-volume lvm create --bluestore --data VOLUME_GROUP/LOGICAL_VOLUME
```

예제

```
[root@osd ~]# ceph-volume lvm create --bluestore --data example_vg/data_lv
```

추가 리소스

- *자세한 내용은 **Red Hat Ceph Storage** 관리 가이드의 **'ceph-volume'**을 사용하여 **CephOSD** 준비 섹션을 참조하십시오.*
- *자세한 내용은 **Red Hat Ceph Storage** 관리 가이드에서 **'ceph-volume'**을 사용하여 **CephOSD** 활성화 섹션을 참조하십시오.*

6.9. BLUEFS 데이터 마이그레이션

***migrate LVM** 하위 명령을 사용하여 소스 볼륨에서 대상 볼륨으로**tonDB** 데이터인 **BlueStore** 파일 시스템(**BlueFS**) 데이터를 마이그레이션할 수 있습니다. 기본 볼륨을 제외한 소스 볼륨은 성공 시 제거됩니다.*

***LVM** 볼륨은 주로 대상 전용입니다.*

*새 볼륨이 **OSD**에 연결되어 소스 드라이브 중 하나를 대체합니다.*

*다음은 **LVM** 볼륨에 대한 배치 규칙입니다.*

- *소스 목록에 **DB** 또는 **WAL** 볼륨이 있는 경우 대상 장치에서 대체됩니다.*

- *소스 목록에 볼륨 속도가 느린 경우 **new-db** 또는 **new-wal** 명령을 사용하여 명시적 할당이 필요합니다.*

***new-db** 및 **new-wal** 명령은 지정된 논리 볼륨을 각각 **DB** 또는 **WAL** 볼륨으로 지정된 **OSD**에 연결합니다.*

사전 요구 사항

- *실행 중인 **Red Hat Ceph Storage** 클러스터.*
- ***Ceph OSD** 노드에 대한 루트 수준 액세스.*
- ***ceph-volume** 유틸리티로 준비된 **Ceph OSD.***
- *볼륨 그룹 및 논리 볼륨이 생성됩니다.*

절차

1. ***cephadm** 쉘에 로그인합니다.*

   예제

   ```
   [root@host01 ~]# cephadm shell
   ```

2. ***DB** 또는 **WAL** 장치를 추가해야 하는 **OSD**를 중지합니다.*

   예제

   ```
   [ceph: root@host01 /]# ceph orch stop osd.1
   ```

3. *컨테이너에 새 장치를 마운트합니다.*

예제

```
[root@host01 ~]# cephadm shell --mount /var/lib/ceph/72436d46-ca06-11ec-9809-ac1f6b5635ee/osd.1:/var/lib/ceph/osd/ceph-1
```

4. *지정된 논리 볼륨을 **DB/WAL** 장치로 **OSD**에 연결합니다.*

참고

***OSD**에 연결된 **DB**가 있는 경우 이 명령이 실패합니다.*

구문

```
ceph-volume lvm new-db --osd-id OSD_ID --osd-fsid OSD_FSID --target VOLUME_GROUP_NAME/LOGICAL_VOLUME_NAME
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm new-db --osd-id 1 --osd-fsid 7ce687d9-07e7-4f8f-a34e-d1b0efb89921 --target vgname/new_db
[ceph: root@host01 /]# ceph-volume lvm new-wal --osd-id 1 --osd-fsid 7ce687d9-07e7-4f8f-a34e-d1b0efb89921 --target vgname/new_wal
```

5. ***BlueFS 데이터를 다음과 같은 방법으로 마이그레이션할 수 있습니다.***

- ***BlueFS 데이터를 DB로 이미 연결된 LV로 이동합니다.***

구문

```
ceph-volume lvm migrate --osd-id OSD_ID --osd-fsid OSD_UUID --from data --target VOLUME_GROUP_NAME/LOGICAL_VOLUME_NAME
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm migrate --osd-id 1 --osd-fsid 0263644D-0BF1-4D6D-BC34-28BD98AE3BC8 --from data --target vgname/db
```

- ***BlueFS 공유 장치에서 새로운 DB로 연결된 LV로 데이터를 이동합니다.***

구문

```
ceph-volume lvm migrate --osd-id OSD_ID --osd-fsid OSD_UUID --from data --target VOLUME_GROUP_NAME/LOGICAL_VOLUME_NAME
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm migrate --osd-id 1 --osd-fsid 0263644D-0BF1-4D6D-BC34-28BD98AE3BC8 --from data --target vgname/new_db
```

- ***BlueFS 데이터를 DB 장치에서 새 LV로 이동하고 DB 장치를 교체합니다.***

구문

```
ceph-volume lvm migrate --osd-id OSD_ID --osd-fsid OSD_UUID --from db --target VOLUME_GROUP_NAME/LOGICAL_VOLUME_NAME
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm migrate --osd-id 1 --osd-fsid 0263644D-0BF1-4D6D-BC34-28BD98AE3BC8 --from db --target vgname/new_db
```

- ***BlueFS 데이터를 기본 및 DB 장치에서 새 LV로 이동하고 DB 장치를 교체하십시오.***

구문

```
ceph-volume lvm migrate --osd-id OSD_ID --osd-fsid OSD_UUID --from data db --target VOLUME_GROUP_NAME/LOGICAL_VOLUME_NAME
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm migrate --osd-id 1 --osd-fsid 0263644D-0BF1-4D6D-BC34-28BD98AE3BC8 --from data db --target vgname/new_db
```

- ***BlueFS 데이터를 기본, DB 및 WAL 장치에서 새 LV로 이동하고, WAL 장치를 제거한 후 DB 장치를 교체합니다.***

구문

```
ceph-volume lvm migrate --osd-id OSD_ID --osd-fsid OSD_UUID --from data db wal --target VOLUME_GROUP_NAME/LOGICAL_VOLUME_NAME
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm migrate --osd-id 1 --osd-fsid 0263644D-0BF1-4D6D-BC34-28BD98AE3BC8 --from data db --target vgname/new_db
```

- ***BlueFS 데이터를 기본, DB 및 WAL 장치에서 기본 장치로 이동하여 WAL 및 DB 장치를 제거하십시오.***

구문

```
ceph-volume lvm migrate --osd-id OSD_ID --osd-fsid OSD_UUID --from db wal --target VOLUME_GROUP_NAME/LOGICAL_VOLUME_NAME
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm migrate --osd-id 1 --osd-fsid 0263644D-0BF1-4D6D-BC34-28BD98AE3BC8 --from db wal --target vgname/data
```

6.10. CEPH-VOLUME과 함께 배치 모드 사용

batch 하위 명령은 단일 장치를 제공하면 여러 **OSD** 생성을 자동화합니다.

ceph-volume 명령은 드라이브 유형에 따라 **OSD**를 생성하는 데 사용할 최적의 방법을 결정합니다. **Ceph OSD** 최적화는 사용 가능한 장치에 따라 다릅니다.

- 모든 장치가 기존 하드 드라이브인 경우 배치 는 장치당 하나의 **OSD**를 생성합니다.
- 모든 장치가 솔리드 상태 드라이브인 경우 배치 는 장치당 두 개의 **OSD**를 생성합니다.
- 기존 하드 드라이브와 솔리드 스테이트 드라이브가 혼합된 경우 일괄 처리는 데이터를 위해 기존 하드 드라이브를 사용하고 솔리드 스테이트 드라이브에서 가능한 가장 큰 저널(**block.db**)을 생성합니다.

참고

batch 하위 명령은 **write-ahead-log(block.wal)** 장치에 대해 별도의 논리 볼륨 생성을 지원하지 않습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ceph OSD** 노드에 대한 루트 수준 액세스.

절차

1. 여러 드라이브에서 **OSD**를 생성하려면 다음을 수행합니다.

구문

```
ceph-volume lvm batch --bluestore PATH_TO_DEVICE [PATH_TO_DEVICE]
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm batch --bluestore /dev/sda /dev/sdb /dev/nvme0n1
```

추가 리소스

- 자세한 내용은 **Red Hat Ceph Storage** 관리 가이드의 **'ceph-volume'**을 사용하여 **CephOSD** 생성 섹션을 참조하십시오.

6.11. CEPH-VOLUME을 사용하여 데이터 ZAPPING

zap 하위 명령은 논리 볼륨 또는 파티션에서 모든 데이터와 파일 시스템을 제거합니다.

zap 하위 명령을 사용하여 **Ceph OSD**에서 재사용하기 위해 사용하는 논리 볼륨, 파티션 또는 원시 장치를 **zap** 하위 명령을 사용할 수 있습니다. 지정된 논리 볼륨 또는 파티션에 있는 파일 시스템이 제거되고 모든 데이터가 제거됩니다.

필요한 경우 **--destroy** 플래그를 사용하여 논리 볼륨, 파티션 또는 물리 장치를 완전히 제거할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ceph OSD** 노드에 대한 루트 수준 액세스.

절차

- *논리 볼륨을 **zap**합니다.*

  구문

  ```
  ceph-volume lvm zap VOLUME_GROUP_NAME/LOGICAL_VOLUME_NAME [--destroy]
  ```

  예제

  ```
  [ceph: root@host01 /]# ceph-volume lvm zap osd-vg/data-lv
  ```

- *파티션을 **zap**합니다.*

  구문

  ```
  ceph-volume lvm zap DEVICE_PATH_PARTITION [--destroy]
  ```

  예제

  ```
  [ceph: root@host01 /]# ceph-volume lvm zap /dev/sdc1
  ```

- ***raw device*를 *zap*합니다.**

구문

```
ceph-volume lvm zap DEVICE_PATH --destroy
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm zap /dev/sdc --destroy
```

- ***OSD ID*를 사용하여 여러 장치를 삭제합니다.**

구문

```
ceph-volume lvm zap --destroy --osd-id OSD_ID
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm zap --destroy --osd-id 16
```

참고

*모든 상대 장치는 **zapped**입니다.*

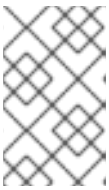
- ***FSID**로 **OSD**를 삭제합니다.*

구문

```
ceph-volume lvm zap --destroy --osd-fsid OSD_FSID
```

예제

```
[ceph: root@host01 /]# ceph-volume lvm zap --destroy --osd-fsid 65d7b6b1-e41a-4a3c-b363-83ade63cb32b
```



참고

*모든 상대 장치는 **zapped**입니다.*

7장. CEPH 성능 벤치마크

스토리지 관리자는 **Red Hat Ceph Storage** 클러스터의 성능을 벤치마크할 수 있습니다. 이 섹션의 목적은 **Ceph** 관리자에게 **Ceph** 기본 벤치마킹 툴에 대한 기본적인 이해를 제공하는 것입니다. 이러한 툴은 **Ceph** 스토리지 클러스터 수행 방법에 대한 몇 가지 통찰력을 제공합니다. 이는 **Ceph** 성능 벤치마킹에 대한 확실한 안내서도 아니며 **Ceph**를 적절하게 조정하는 방법에 대한 가이드도 아닙니다.

7.1. 사전 요구 사항

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

7.2. 성능 기준

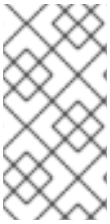
저널, 디스크, 네트워크 처리량을 포함한 **OSD**에는 각각 비교할 성능 기준선이 있어야 합니다. 기본 성능 데이터를 **Ceph** 기본 툴의 데이터와 비교하여 잠재적인 튜닝 기회를 파악할 수 있습니다. **Red Hat Enterprise Linux**에는 다양한 기본 제공 도구가 있으며 이러한 작업을 수행할 수 있도록 다양한 오픈 소스 커뮤니티 툴이 제공됩니다.

추가 리소스

- 사용 가능한 도구 중 일부에 대한 자세한 내용은 이 지식베이스 문서를 참조하십시오.

7.3. CEPH 성능 벤치마킹

Ceph에는 **RADOS** 스토리지 클러스터에서 성능 벤치마킹을 수행하는 **rados bench** 명령이 포함되어 있습니다. 명령은 쓰기 테스트와 두 가지 유형의 읽기 테스트를 실행합니다. **--no-cleanup** 옵션은 읽기 및 쓰기 성능을 모두 테스트할 때 사용하는 것이 중요합니다. 기본적으로 **rados bench** 명령은 스토리지 풀에 작성된 오브젝트를 삭제합니다. 이 오브젝트 뒤에 남겨 두는 두 개의 읽기 테스트를 통해 순차적 및 임의의 읽기 성능을 측정할 수 있습니다.



참고

이러한 성능 테스트를 실행하기 전에 다음을 실행하여 모든 파일 시스템 캐시를 삭제합니다.

예제

```
[ceph: root@host01 /]# echo 3 | sudo tee /proc/sys/vm/drop_caches && sudo sync
```

사전 요구 사항

- *실행 중인 **Red Hat Ceph Storage** 클러스터.*
- *노드에 대한 루트 수준 액세스입니다.*

절차

1. *새 스토리지 풀을 생성합니다.*

   예제

   ```
   [ceph: root@host01 /]# ceph osd pool create testbench 100 100
   ```

2. *새로 생성된 스토리지 풀에 대해 **10**초 동안 쓰기 테스트를 실행합니다.*

   예제

   ```
   [ceph: root@host01 /]# rados bench -p testbench 10 write --no-cleanup

   Maintaining 16 concurrent writes of 4194304 bytes for up to 10 seconds or 0 objects
    Object prefix: benchmark_data_cephn1.home.network_10510
      sec Cur ops   started  finished  avg MB/s  cur MB/s  last lat   avg lat
        0       0         0         0         0         0         -         0
        1      16        16         0         0         0         -         0
        2      16        16         0         0         0         -         0
        3      16        16         0         0         0         -         0
        4      16        17         1  0.998879         1   3.19824   3.19824
        5      16        18         2   1.59849         4   4.56163   3.87993
        6      16        18         2   1.33222         0         -   3.87993
        7      16        19         3   1.71239         2   6.90712     4.889
   ```

```
    8     16       25        9   4.49551       24   7.75362   6.71216
    9     16       25        9   3.99636        0         -   6.71216
   10     16       27       11   4.39632        4   9.65085   7.18999
   11     16       27       11   3.99685        0         -   7.18999
   12     16       27       11   3.66397        0         -   7.18999
   13     16       28       12   3.68975  1.33333   12.8124   7.65853
   14     16       28       12   3.42617        0         -   7.65853
   15     16       28       12   3.19785        0         -   7.65853
   16     11       28       17   4.24726  6.66667   12.5302   9.27548
   17     11       28       17   3.99751        0         -   9.27548
   18     11       28       17   3.77546        0         -   9.27548
   19     11       28       17   3.57683        0         -   9.27548
 Total time run:         19.505620
Total writes made:      28
Write size:             4194304
Bandwidth (MB/sec):     5.742

Stddev Bandwidth:       5.4617
Max bandwidth (MB/sec): 24
Min bandwidth (MB/sec): 0
Average Latency:        10.4064
Stddev Latency:         3.80038
Max latency:            19.503
Min latency:            3.19824
```

3. ***10*초 동안 스토리지 풀에 대해 순차적 읽기 테스트를 실행합니다.**

예제

```
[ceph: root@host01 /]# rados bench -p testbench 10 seq

sec Cur ops   started  finished  avg MB/s  cur MB/s  last lat   avg lat
  0       0         0         0         0         0         -         0
Total time run:        0.804869
Total reads made:      28
Read size:             4194304
Bandwidth (MB/sec):    139.153

Average Latency:       0.420841
Max latency:           0.706133
Min latency:           0.0816332
```

4.

*스토리지 풀에 **10**초 동안 임의의 읽기 테스트를 실행합니다.*

예제

```
[ceph: root@host01 /]# rados bench -p testbench 10 rand

sec Cur ops   started  finished  avg MB/s  cur MB/s  last lat   avg lat
  0       0         0         0         0         0         -         0
  1      16        46        30   119.801       120  0.440184  0.388125
  2      16        81        65   129.408       140  0.577359  0.417461
  3      16       120       104   138.175       156  0.597435  0.409318
  4      15       157       142   141.485       152  0.683111  0.419964
  5      16       206       190   151.553       192  0.310578  0.408343
  6      16       253       237   157.608       188 0.0745175  0.387207
  7      16       287       271   154.412       136  0.792774   0.39043
  8      16       325       309   154.044       152  0.314254   0.39876
  9      16       362       346   153.245       148  0.355576  0.406032
 10      16       405       389   155.092       172   0.64734  0.398372
Total time run:        10.302229
Total reads made:      405
Read size:             4194304
Bandwidth (MB/sec):    157.248

Average Latency:       0.405976
Max latency:           1.00869
Min latency:           0.0378431
```

5.

*동시 읽기 및 쓰기 수를 늘리려면 기본값이 **16**개 스레드인 **-t** 옵션을 사용합니다. 또한 **-b** 매개 변수는 작성 중인 오브젝트의 크기를 조정할 수 있습니다. 기본 오브젝트 크기는 **4MB**입니다. 안전한 최대 오브젝트 크기는 **16MB**입니다. **Red Hat**은 서로 다른 풀에 이러한 벤치마크 테스트의 여러 사본을 실행하는 것이 좋습니다. 이 작업을 수행하면 여러 클라이언트의 성능 변경 사항이 표시됩니다.*

__--run-name LABEL__ 옵션을 추가하여 벤치마크 테스트 중에 기록되는 오브젝트의 이름을 제어합니다. 실행 중인 각 명령 인스턴스에 대해 __--run-name__ 레이블을 변경하여 여러 __rados bench__ 명령을 동시에 실행할 수 있습니다. 이렇게 하면 여러 클라이언트가 동일한 오브젝트에 액세스하고 다른 클라이언트가 다른 오브젝트에 액세스하려고 할 때 발생할 수 있는 잠재적인 __I/O__ 오류가 발생하지 않습니다. __--run-name__ 옵션은 실제 워크로드를 시뮬레이션하려고 할 때 유용합니다.

예제

```
[ceph: root@host01 /]# rados bench -p testbench 10 write -t 4 --run-name client1

Maintaining 4 concurrent writes of 4194304 bytes for up to 10 seconds or 0 objects
 Object prefix: benchmark_data_node1_12631
   sec Cur ops   started  finished  avg MB/s  cur MB/s  last lat   avg lat
     0       0         0         0         0         0         -         0
     1       4         4         0         0         0         -         0
     2       4         6         2   3.99099         4   1.94755   1.93361
     3       4         8         4   5.32498         8     2.978   2.44034
     4       4         8         4   3.99504         0         -   2.44034
     5       4        10         6   4.79504         4   2.92419    2.4629
     6       3        10         7   4.64471         4   3.02498    2.5432
     7       4        12         8   4.55287         4   3.12204   2.61555
     8       4        14        10    4.9821         8   2.55901   2.68396
     9       4        16        12   5.31621         8   2.68769   2.68081
    10       4        17        13   5.18488         4   2.11937   2.63763
    11       4        17        13   4.71431         0         -   2.63763
    12       4        18        14   4.65486         2    2.4836   2.62662
    13       4        18        14   4.29757         0         -   2.62662
Total time run:         13.123548
Total writes made:      18
Write size:             4194304
Bandwidth (MB/sec):     5.486

Stddev Bandwidth:       3.0991
Max bandwidth (MB/sec): 8
Min bandwidth (MB/sec): 0
Average Latency:        2.91578
Stddev Latency:         0.956993
Max latency:            5.72685
Min latency:            1.91967
```

6. ***rados bench*** *명령으로 생성된 데이터를 제거합니다.*

예제

```
[ceph: root@host01 /]# rados -p testbench cleanup
```

7.4. CEPH 블록 성능 벤치마킹

Ceph*에는 블록 장치 측정 처리량 및 대기 시간에 대한 순차적 쓰기를 테스트하는* ***rbd bench-write*** *명*

*령이 포함되어 있습니다. 기본 바이트 크기는 **4096**이며, **I/O** 스레드의 기본 수는 **16**이며, 쓸 기본 총 바이트 수는 **1GB**입니다. 이러한 기본값은 각각 **--io-size,--io-threads** 및 **--io-total** 옵션을 통해 수정할 수 있습니다.*

사전 요구 사항

- *실행 중인 **Red Hat Ceph Storage** 클러스터.*
- *노드에 대한 루트 수준 액세스입니다.*

절차

1. *아직 로드되지 않은 경우 **rbd** 커널 모듈을 로드합니다.*

   예제

   ```
   [root@host01 ~]# modprobe rbd
   ```

2. ***testbench** 풀에 **1GB rbd** 이미지 파일을 만듭니다.*

   예제

   ```
   [root@host01 ~]# rbd create image01 --size 1024 --pool testbench
   ```

3. *이미지 파일을 장치 파일에 매핑합니다.*

   예제

```
[root@host01 ~]# rbd map image01 --pool testbench --name client.admin
```

4. *블록 장치에 **ext4** 파일 시스템을 생성합니다.*

예제

```
[root@host01 ~]# mkfs.ext4 /dev/rbd/testbench/image01
```

5. *새 디렉터리를 생성합니다.*

예제

```
[root@host01 ~]# mkdir /mnt/ceph-block-device
```

6. */**mnt/ceph-block-device**/ 에 블록 장치를 마운트합니다.*

예제

```
[root@host01 ~]# mount /dev/rbd/testbench/image01 /mnt/ceph-block-device
```

7. *블록 장치에 대해 쓰기 성능 테스트 실행*

예제

```
[root@host01 ~]# rbd bench --io-type write image01 --pool=testbench

bench-write  io_size 4096 io_threads 16 bytes 1073741824 pattern seq
  SEC       OPS   OPS/SEC   BYTES/SEC
    2     11127   5479.59  22444382.79
    3     11692   3901.91  15982220.33
    4     12372   2953.34  12096895.42
    5     12580   2300.05  9421008.60
    6     13141   2101.80  8608975.15
    7     13195    356.07  1458459.94
    8     13820    390.35  1598876.60
    9     14124    325.46  1333066.62
    ..
```

추가 리소스

- ***rbd** 명령에 대한 자세한 내용은 **Red Hat Ceph Storage Block Device Guide** 의 **Ceph** 블록 장치 장을 참조하십시오.*

8장. CEPH 성능 카운터

스토리지 관리자는 **Red Hat Ceph Storage** 클러스터의 성능 지표를 수집할 수 있습니다. **Ceph** 성능 카운터는 내부 인프라 지표의 컬렉션입니다. 이 지표 데이터의 수집, 집계 및 그래프는 도구의 정렬을 통해 수행할 수 있으며 성능 분석에 유용할 수 있습니다.

8.1. 사전 요구 사항

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

8.2. CEPH 성능 카운터에 액세스

성능 카운터는 **Ceph** 모니터 및 **OSD**의 소켓 인터페이스를 통해 사용할 수 있습니다. 각 데몬의 소켓 파일은 기본적으로 **/var/run/ceph** 에 있습니다. 성능 카운터는 컬렉션 이름으로 그룹화됩니다. 이러한 컬렉션 이름은 하위 시스템 또는 하위 시스템의 인스턴스를 나타냅니다.

다음은 각 모니터에 대한 간단한 설명과 **OSD** 컬렉션 이름 범주의 전체 목록입니다.

모니터링 수집 이름 범주

- 클러스터 메트릭 **-** 스토리지 클러스터에 대한 정보를 표시합니다**: Monitors, OSD, Pools, PGs**
- 수준 데이터베이스 지표 **-** 백엔드 **KeyValueStore** 데이터베이스에 대한 정보를 표시
- 지표 모니터링 **-** 일반 모니터 정보 표시
- **Paxos Metrics -** 클러스터 쿼럼 관리에 대한 정보 표시
- **throttle Metrics -** 모니터가 제한되는 방법에 대한 통계 표시

OSD 컬렉션 이름 범주

- **Throttle Metrics** 작성 **-** 쓰기 **back throttle**이 손상되지 않은 **IO**를 추적하는 방법에 대한 통계 표시

- 수준 데이터베이스 지표 - 백엔드 KeyValueStore 데이터베이스에 대한 정보를 표시
- Objecter 지표 - 다양한 오브젝트 기반 작업에 대한 정보를 표시
- 읽기 및 쓰기 작업 지표 - 다양한 읽기 및 쓰기 작업에 대한 정보 표시
- 복구 상태 지표 - 다양한 복구 상태에 대한 대기 시간 표시
- OSD Throttle 지표 - OSD가 제한되는 방법에 대한 통계 표시

RADOS 게이트웨이 컬렉션 이름 범주

- Object Gateway Client Metrics - GET 및 PUT 요청에 대한 통계를 표시
- Objecter 지표 - 다양한 오브젝트 기반 작업에 대한 정보를 표시
- Object Gateway Throttle Metrics - OSD가 제한되는 방법에 대한 통계 표시

8.3. CEPH 성능 카운터 표시

ceph 데몬 DAE. 16.0_NAME perf schema 명령은 사용 가능한 지표를 출력합니다. 각 메트릭에는 관련 비트 필드 값 유형이 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. *메트릭의 스키마를 보려면 다음을 수행합니다.*

   Synatx

   ```
   ceph daemon DAEMON_NAME perf schema
   ```

   참고

   *데몬을 실행하는 노드에서 **ceph daemon** 명령을 실행해야 합니다.*

2. *모니터 노드에서 **ceph** 데몬 **DAE.16.0_NAME perf schema** 명령을 실행합니다.*

   예제

   ```
   [ceph: root@host01 /]# ceph daemon mon.host01 perf schema
   ```

3. ***OSD** 노드에서 **ceph** 데몬 **DAEtekton_NAME perf** 스키마 명령을 실행합니다.*

   예제

   ```
   [ceph: root@host01 /]# ceph daemon osd.11 perf schema
   ```

*표 **8.1.** 비트 필드 값 정의*

Bit	meaning
1	부동 소수점 값
2	서명되지 않은 64비트 정수 값
4	평균 (Sum + Count)
8	카운터

각 값에는 부동 소수점 또는 정수 값 중 하나를 나타내는 비트 1 또는 2가 설정됩니다. 비트 4가 설정되면 두 개의 값, 즉 합계와 개수가 있습니다. 비트 8이 설정되면 이전 간격의 평균이 합계 델타가 됩니다. 또는 값을 올바르게 나누면 수명 평균 값을 제공할 수 있습니다. 일반적으로 이들은 대기 시간, 요청 수 및 요청 대기 시간 합계를 측정하는 데 사용됩니다. 일부 비트 값은 조합됩니다(예: 5, 6 및 10). 5의 비트 값은 비트 1과 비트 4의 조합입니다. 즉, 평균은 부동 소수점 값이 됩니다. 6 비트 값은 비트 2와 비트 4의 조합입니다. 즉, 평균 값은 정수입니다. 비트 값 10은 비트 2와 비트 8의 조합입니다. 즉, 카운터 값이 정수 값이 됩니다.

추가 리소스

- *자세한 내용은 **Red Hat Ceph Storage** 관리 가이드 의 평균 수 및 합계 섹션을 참조하십시오.*

8.4. CEPH 성능 카운터 덤프

ceph 데몬 ... perf dump 명령은 현재 값을 출력하고 각 하위 시스템의 컬렉션 이름 아래에 지표를 그룹화합니다.

사전 요구 사항

- *실행 중인 **Red Hat Ceph Storage** 클러스터.*
- *노드에 대한 루트 수준 액세스입니다.*

절차

1. *현재 메트릭 데이터를 보려면 다음을 수행합니다.*

구문

```
ceph daemon DAEMON_NAME perf dump
```

참고

데몬을 실행하는 노드에서 **ceph daemon** 명령을 실행해야 합니다.

2. 모니터 노드에서 **ceph** 데몬 **... perf dump** 명령을 실행합니다.

```
[ceph: root@host01 /]# ceph daemon mon.host01 perf dump
```

3. **OSD** 노드에서 **ceph** 데몬 **... perf dump** 명령을 실행합니다.

```
[ceph: root@host01 /]# ceph daemon osd.11 perf dump
```

추가 리소스

- 사용 가능한 각 모니터 메트릭에 대한 간단한 설명을 보려면 **Ceph 모니터 메트릭 테이블** 을 참조하십시오.

8.5. 평균 수 및 합계

모든 대기 시간 번호에는 비트 필드 값이 **5**입니다. 이 필드에는 평균 수 및 합계에 대한 부동 소수점 값이 포함되어 있습니다. **avgcount** 는 이 범위 내의 작업 수이며 합계 는 총 대기 시간(초)입니다. 합계 를 **avgcount** 로 나눌 때 작업당 대기 시간에 대한 아이디어를 제공합니다.

추가 리소스

- 사용 가능한 각 **OSD** 지표에 대한 간략한 설명을 보려면 **Ceph OSD 테이블** 을 참조하십시오.

8.6. CEPH MONITOR 메트릭

- *클러스터 지표 테이블*
- *수준 데이터베이스 지표 테이블*
- *일반 모니터 지표 테이블*
- ***Paxos** 지표 테이블*
- *대규모 지표 테이블*

표 8.2. 클러스터 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
cluster	**num_mon**	2	모니터 수
	num_mon_quorum	2	쿼럼의 모니터 수
	num_osd	2	총 OSD 수
	num_osd_up	2	가동 중인 OSD 수
	num_osd_in	2	클러스터에 있는 OSD 수
	osd_epoch	2	OSD 맵의 현재 epoch
	osd_bytes	2	총 클러스터 용량(바이트)
	osd_bytes_used	2	클러스터에서 사용되는 바이트 수
	osd_bytes_avail	2	클러스터에서 사용 가능한 바이트 수
	num_pool	2	풀 수
	num_pg	2	총 배치 그룹 수
	num_pg_active_clean	2	active+clean 상태의 배치 그룹 수

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	num_pg_active	2	활성 상태의 배치 그룹 수
	num_pg_peering	2	피어링 상태의 배치 그룹 수
	num_object	2	클러스터의 총 오브젝트 수
	num_object_degraded	2	성능이 저하된 (복제 복제) 오브젝트 수
	num_object_misplaced	2	잘못 배치된(클러스터의 위치) 오브젝트 수
	num_object_unfound	2	unfound 오브젝트 수
	num_bytes	2	모든 오브젝트의 총 바이트 수
	num_mds_up	2	작동 중인 MDS 수
	num_mds_in	2	클러스터에 있는 MDS 수
	num_mds_failed	2	실패한 MDS 수
	mds_epoch	2	MDS 맵의 현재 epoch

표 8.3. 수준 데이터베이스 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
leveldb	**leveldb_get**	10	gets
	leveldb_transaction	10	트랜잭션
	leveldb_compact	10	압축
	leveldb_compact_range	10	범위별 컴팩트
	leveldb_compact_queue_merge	10	콤팩트 큐에서 범위 제거
	leveldb_compact_queue_len	2	압축 큐의 길이

표 8.4. 일반 모니터 메트릭 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
Mon	**num_sessions**	2	현재 열린 모니터 세션의 수
	session_add	10	생성된 모니터 세션 수
	session_rm	10	monitor의 remove_session 호출 수
	session_trim	10	트리밍 모니터 세션 수
	num_elections	10	여러 선거 모니터가 참여했습니다.
	election_call	10	모니터에서 시작한 선택 수
	election_win	10	모니터에서 얻은 선택 수입니다.
	election_lose	10	모니터에 의해 손실된 선택 수

표 8.5. Paxos 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
Paxos	**start_leader**	10	리더 역할에서 시작
	start_peon	10	peon 역할에서 시작
	재시작	10	재시작
	새로 고침	10	새로 고침
	refresh_latency	5	대기 시간 새로 고침
	begin	10	시작 및 처리
	begin_keys	6	시작 시 트랜잭션의 키
	begin_bytes	6	트랜잭션 시작 시 데이터
	begin_latency	5	시작 작업의 대기 시간
	commit	10	commit
	commit_keys	6	커밋 시 트랜잭션의 키
	commit_bytes	6	커밋 시 트랜잭션의 데이터

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	commit_latency	5	커밋 대기 시간
	collect	10	peon 수집
	collect_keys	6	peon 수집 트랜잭션의 키
	collect_bytes	6	peon 수집 트랜잭션의 데이터
	collect_latency	5	peon 수집 대기 시간 수집
	collect_uncommitted	10	시작 및 처리에서 커밋되지 않은 값 수집
	collect_timeout	10	시간 제한 수집
	accept_timeout	10	시간 제한 수락
	lease_ack_timeout	10	임대 승인 시간 초과
	lease_timeout	10	리스 제한 시간
	store_state	10	디스크에 공유 상태 저장
	store_state_keys	6	저장된 상태의 트랜잭션의 키
	store_state_bytes	6	저장된 상태의 트랜잭션의 데이터
	store_state_latency	5	상태 대기 시간 저장
	share_state	10	자주하는 질문
	share_state_keys	6	공유 상태의 키
	share_state_bytes	6	공유 상태의 데이터
	new_pn	10	새로운 제안 번호 쿼리
	new_pn_latency	5	대기 시간을 가져오는 새로운 제안 번호

표 8.6. 제한 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
throttle-*	**val**	10	현재 사용 가능한 제한
	max	10	throttle의 최대 값
	get	10	gets
	get_sum	10	데이터 가져오기
	get_or_fail_fail	10	get_or_fail 중에 차단됨
	get_or_fail_success	10	get_or_fail 중에 성공
	take	10	takes
	take_sum	10	가져온 데이터
	put	10	put
	put_sum	10	데이터 배치
	wait	5	대기 시간

8.7. CEPH OSD 지표

- *Back Throttle 지표 테이블 작성*
- *수준 데이터베이스 지표 테이블*
- *오브젝트 지표 테이블*
- *읽기 및 작업 지표 테이블 읽기 및 쓰기*
- *복구 상태 지표 테이블*
- *OSD Throttle 지표 테이블*

표 8.7. Throttle 지표 테이블 작성

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
WBThrottle	**bytes_dirtied**	2	더티 데이터
	bytes_wb	2	기록된 데이터
	ios_dirtied	2	더티 작업
	ios_wb	2	작성된 작업
	inodes_dirtied	2	쓰기 대기 중인 항목
	inodes_wb	2	기록된 항목

표 8.8. 수준 데이터베이스 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
leveldb	**leveldb_get**	10	gets
	leveldb_transaction	10	트랜잭션
	leveldb_compact	10	압축
	leveldb_compact_range	10	범위별 컴팩트
	leveldb_compact_queue_merge	10	콤팩트 큐에서 범위 제거
	leveldb_compact_queue_len	2	압축 큐의 길이

표 8.9. 오브젝트 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
objecter	**op_active**	2	활성 작업
	op_laggy	2	laggy 작업
	op_send	10	전송된 작업
	op_send_bytes	10	전송된 데이터

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	op_resend	10	Resent operations
	op_ack	10	콜백 커밋
	op_commit	10	작업 커밋
	op	10	작업
	op_r	10	읽기 작업
	op_w	10	쓰기 작업
	op_rmw	10	읽기-modify-write 작업
	op_pg	10	PG 작업
	osdop_stat	10	stat 작업
	osdop_create	10	오브젝트 작업 생성
	osdop_read	10	읽기 작업
	osdop_write	10	쓰기 작업
	osdop_writefull	10	전체 오브젝트 작업 작성
	osdop_append	10	추가 작업
	osdop_zero	10	오브젝트를 0개의 작업으로 설정
	osdop_truncate	10	개체 작업 정리
	osdop_delete	10	오브젝트 작업 삭제
	osdop_mapext	10	맵 범위 작업
	osdop_sparse_read	10	스파스 읽기 작업
	osdop_clonerange	10	복제 범위 작업
	osdop_getxattr	10	xattr 작업 가져오기
	osdop_setxattr	10	xattr 작업 설정

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	osdop_cmpxattr	10	xattr 비교 작업
	osdop_rmxattr	10	xattr 작업 제거
	osdop_resetxattrs	10	xattr 작업 재설정
	osdop_tmap_up	10	TMAP 업데이트 작업
	osdop_tmap_put	10	TMAP 작업
	osdop_tmap_get	10	TMAP 실행
	osdop_call	10	호출(실행) 작업
	osdop_watch	10	오브젝트 작업 조사
	osdop_notify	10	오브젝트 작업에 대한 알림
	osdop_src_cmpxattr	10	다중 작업의 확장 속성 비교
	osdop_other	10	기타 작업
	linger_active	2	활성 언어 작업
	linger_send	10	전송된 언어 작업
	linger_resend	10	관련 언어 작업 반복
	linger_ping	10	작업을 제어하기 위해 ping 전송
	poolop_active	2	활성 풀 작업
	poolop_send	10	전송된 풀 작업
	poolop_resend	10	Resent pool 작업
	poolstat_active	2	활성 get pool stat 작업
	poolstat_send	10	전송된 풀 stat 작업
	poolstat_resend	10	Resent pool 통계
	statfs_active	2	statfs 작업

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	statfs_send	10	전송 FS 통계
	statfs_resend	10	Resent FS 통계
	command_active	2	활성 명령
	command_send	10	전송된 명령
	command_resend	10	Resent 명령
	map_epoch	2	OSD 맵 epoch
	map_full	10	수신된 전체 OSD 맵
	map_inc	10	수신된 증분 OSD 맵
	osd_sessions	2	공개 세션
	osd_session_open	10	세션이 열려 있습니다.
	osd_session_close	10	세션 종료
	osd_laggy	2	지연 OSD 세션

*표 **8.10.** 운영 지표 테이블 읽기 및 쓰기*

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
osd	**op_wip**	2	현재 처리 중인 복제 작업(primary)
	op_in_bytes	10	클라이언트 작업 전체 쓰기 크기
	op_out_bytes	10	클라이언트 작업 전체 읽기 크기
	op_latency	5	클라이언트 작업 대기 시간(초기 시간 포함)
	op_process_latency	5	클라이언트 작업 대기 시간 (예약 대기열 시간 제외)
	op_r	10	클라이언트 읽기 작업
	op_r_out_bytes	10	클라이언트 데이터 읽기

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	op_r_latency	5	읽기 작업의 대기 시간(예: 대기열 시간 포함)
	op_r_process_latency	5	읽기 작업의 대기 시간(예: 대기열 시간 제외)
	op_w	10	클라이언트 쓰기 작업
	op_w_in_bytes	10	작성된 클라이언트 데이터
	op_w_rlat	5	클라이언트 쓰기 작업 읽기/ 적용 대기 시간
	op_w_latency	5	쓰기 작업의 대기 시간(초기 시간 포함)
	op_w_process_latency	5	쓰기 작업의 대기 시간 (프레드 시간 제외)
	op_rw	10	클라이언트 읽기-수정-쓰기 작업
	op_rw_in_bytes	10	클라이언트 읽기-modify-write 작업 쓰기
	op_rw_out_bytes	10	클라이언트 읽기-modify-write 작업 읽기
	op_rw_rlat	5	클라이언트 읽기-modify-write 작업 읽기/ 적용 대기 시간
	op_rw_latency	5	읽기-modify-write 작업의 대기 시간 (큐 시간 포함)
	op_rw_process_latency	5	읽기-modify-write 작업의 대기 시간 (예: 대기열 시간 제외)
	subop	10	suboperations
	subop_in_bytes	10	하위 작업 총 크기
	subop_latency	5	하위 작업 대기 시간
	subop_w	10	복제된 쓰기
	subop_w_in_bytes	10	복제된 쓰기 데이터 크기

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	subop_w_latency	5	복제된 쓰기 대기 시간
	subop_pull	10	하위 작업 요청 가져오기
	subop_pull_latency	5	하위 작업 대기 시간
	subop_push	10	하위 작업 푸시 메시지
	subop_push_in_bytes	10	하위 작업 크기 푸시됨
	subop_push_latency	5	하위 작업 푸시 대기 시간
	pull	10	전송된 가져오기 요청
	push	10	푸시 메시지 전송
	push_out_bytes	10	푸시된 크기
	push_in	10	인바운드 푸시 메시지
	push_in_bytes	10	인바운드 푸시 크기
	recovery_ops	10	복구 작업 시작
	loadavg	2	CPU 로드
	buffer_bytes	2	할당된 총 버퍼 크기
	numpg	2	배치 그룹
	numpg_primary	2	이 osd가 기본인 배치 그룹
	numpg_replica	2	이 osd가 복제본인 배치 그룹
	numpg_stray	2	이 osd에서 삭제할 준비가 된 배치 그룹
	heartbeat_to_peers	2	하트비트(ping) 피어를 보냅니다.
	heartbeat_from_peers	2	하트비트(ping) 피어에서 복구
	map_messages	10	OSD 맵 메시지

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	map_message_epochs	10	OSD 맵 epochs
	map_message_epoch_dups	10	OSD 맵 중복
	stat_bytes	2	OSD 크기
	stat_bytes_used	2	사용된 공간
	stat_bytes_avail	2	사용 가능한 공간
	copyFrom	10	RADOS의 'copy-from' 작업
	tier_promote	10	계층 프로모션
	tier_flush	10	계층 플러시
	tier_flush_fail	10	실패한 계층 플러시
	tier_try_flush	10	계층 플러시 시도
	tier_try_flush_fail	10	실패한 계층 플러시 시도
	tier_evict	10	계층 제거
	tier_whiteout	10	계층 백아웃
	tier_dirty	10	더티 계층 플래그 세트
	tier_clean	10	더티 계층 플래그 정리
	tier_delay	10	계층 지연(주요 대기)
	tier_proxy_read	10	계층 프록시 읽기
	agent_wake	10	계층화 에이전트 발생
	agent_skip	10	에이전트에서 건너뛰는 오브젝트
	agent_flush	10	계층화 에이전트 플러시
	agent_evict	10	계층화 에이전트 제거

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	object_ctx_cache_hit	10	오브젝트 컨텍스트 캐시 적중
	object_ctx_cache_total	10	오브젝트 컨텍스트 캐시 조회

표 8.11. 복구 상태 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
recoverystate_perf	**initial_latency**	5	초기 복구 상태 대기 시간
	started_latency	5	시작 복구 상태 대기 시간
	reset_latency	5	복구 상태 대기 시간 재설정
	start_latency	5	복구 상태 대기 시간 시작
	primary_latency	5	기본 복구 상태 대기 시간
	peering_latency	5	복구 상태 대기 시간 피어링
	backfilling_latency	5	복구 상태 대기 시간 백 채우기
	waitremotebackfillreserved_latency	5	원격 backfill 예약된 복구 상태 대기 시간
	waitlocalbackfillreserved_latency	5	로컬 backfill 예약된 복구 상태 대기 시간
	notbackfilling_latency	5	Notbackfilling 복구 상태 대기 시간
	repnotrecovering_latency	5	복구 상태 대기 시간 복원
	repwaitrecoveryreserved_latency	5	복구 대기 상태 대기 시간
	repwaitbackfillreserved_latency	5	백필 대기 예약된 복구 상태 대기 대기
	RepRecovering_latency	5	복구 상태 대기 시간 복원

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	activating_latency	5	복구 상태 대기 시간 활성화
	waitlocalrecoveryreserved_latency	5	로컬 복구 예약된 복구 상태 대기 시간
	waitremoterecoveryreserved_latency	5	원격 복구 예약된 복구 상태 대기 시간 대기
	recovering_latency	5	복구 상태 대기 시간 복구
	recovered_latency	5	복구 상태 복구 대기 시간
	clean_latency	5	정리 복구 상태 대기 시간
	active_latency	5	활성 복구 상태 대기 시간
	replicaactive_latency	5	Replicaactive 복구 상태 대기 시간
	stray_latency	5	대기 시간 복구 상태 대기 시간
	getinfo_latency	5	Getinfo 복구 상태 대기 시간
	getlog_latency	5	Getlog 복구 상태 대기 시간
	waitactingchange_latency	5	변경 내용의 복구 상태 대기 시간
	incomplete_latency	5	불완전한 복구 상태 대기 시간
	getmissing_latency	5	Getmissing 복구 상태 대기 시간
	waitupthru_latency	5	Waitupthru 복구 상태 대기 시간

표 8.12. OSD Throttle 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
throttle-*	**val**	10	현재 사용 가능한 제한
	max	10	throttle의 최대 값
	get	10	gets

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	get_sum	10	데이터 가져오기
	get_or_fail_fail	10	get_or_fail 중에 차단됨
	get_or_fail_success	10	get_or_fail 중에 성공
	take	10	takes
	take_sum	10	가져온 데이터
	put	10	put
	put_sum	10	데이터 배치
	wait	5	대기 시간

8.8. CEPH OBJECT GATEWAY 지표

- *Ceph Object Gateway 클라이언트 테이블*
- *오브젝트 지표 테이블*
- *Ceph Object Gateway Throttle Metrics Table*

표 8.13. Ceph Object Gateway 클라이언트 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
client.rgw.<rgw_node_name>	**req**	10	요구 사항
	failed_req	10	중단된 요청
	get	10	gets
	get_b	10	Get의 크기
	get_initial_lat	5	대기 시간 가져오기

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	put	10	put
	put_b	10	줄 크기
	put_initial_lat	5	대기 시간 배치
	qlen	2	대기열 길이
	qactive	2	활성 요청 대기열
	cache_hit	10	캐시 적중
	cache_miss	10	캐시 누락
	keystone_token_cache_hit	10	Keystone 토큰 캐시 적중
	keystone_token_cache_miss	10	Keystone 토큰 캐시 누락

표 8.14. 오브젝트 지표 테이블

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
objecter	**op_active**	2	활성 작업
	op_laggy	2	laggy 작업
	op_send	10	전송된 작업
	op_send_bytes	10	전송된 데이터
	op_resend	10	Resent operations
	op_ack	10	콜백 커밋
	op_commit	10	작업 커밋
	op	10	작업
	op_r	10	읽기 작업
	op_w	10	쓰기 작업
	op_rmw	10	읽기-modify-write 작업

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	op_pg	10	PG 작업
	osdop_stat	10	stat 작업
	osdop_create	10	오브젝트 작업 생성
	osdop_read	10	읽기 작업
	osdop_write	10	쓰기 작업
	osdop_writefull	10	전체 오브젝트 작업 작성
	osdop_append	10	추가 작업
	osdop_zero	10	오브젝트를 0개의 작업으로 설정
	osdop_truncate	10	개체 작업 정리
	osdop_delete	10	오브젝트 작업 삭제
	osdop_mapext	10	맵 범위 작업
	osdop_sparse_read	10	스파스 읽기 작업
	osdop_clonerange	10	복제 범위 작업
	osdop_getxattr	10	xattr 작업 가져오기
	osdop_setxattr	10	xattr 작업 설정
	osdop_cmpxattr	10	xattr 비교 작업
	osdop_rmxattr	10	xattr 작업 제거
	osdop_resetxattrs	10	xattr 작업 재설정
	osdop_tmap_up	10	TMAP 업데이트 작업
	osdop_tmap_put	10	TMAP 작업
	osdop_tmap_get	10	TMAP 실행
	osdop_call	10	호출(실행) 작업

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	osdop_watch	10	오브젝트 작업 조사
	osdop_notify	10	오브젝트 작업에 대한 알림
	osdop_src_cmpxattr	10	다중 작업의 확장 속성 비교
	osdop_other	10	기타 작업
	linger_active	2	활성 언어 작업
	linger_send	10	전송된 언어 작업
	linger_resend	10	관련 언어 작업 반복
	linger_ping	10	작업을 제어하기 위해 ping 전송
	poolop_active	2	활성 풀 작업
	poolop_send	10	전송된 풀 작업
	poolop_resend	10	Resent pool 작업
	poolstat_active	2	활성 get pool stat 작업
	poolstat_send	10	전송된 풀 stat 작업
	poolstat_resend	10	Resent pool 통계
	statfs_active	2	statfs 작업
	statfs_send	10	전송 FS 통계
	statfs_resend	10	Resent FS 통계
	command_active	2	활성 명령
	command_send	10	전송된 명령
	command_resend	10	Resent 명령
	map_epoch	2	OSD 맵 epoch
	map_full	10	수신된 전체 OSD 맵

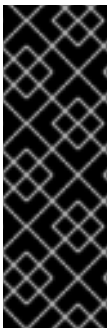
컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
	map_inc	10	수신된 증분 OSD 맵
	osd_sessions	2	공개 세션
	osd_session_open	10	세션이 열려 있습니다.
	osd_session_close	10	세션 종료
	osd_laggy	2	지연 OSD 세션

표 8.15. Ceph Object Gateway Throttle Metrics Table

컬렉션 이름	메트릭 이름	비트 필드 값	짧은 설명
throttle-*	**val**	10	현재 사용 가능한 제한
	max	10	throttle의 최대 값
	get	10	gets
	get_sum	10	데이터 가져오기
	get_or_fail_fail	10	get_or_fail 중에 차단됨
	get_or_fail_success	10	get_or_fail 중에 성공
	take	10	takes
	take_sum	10	가져온 데이터
	put	10	put
	put_sum	10	데이터 배치
	wait	5	대기 시간

9장. BLUESTORE

bluestore는 **OSD** 데몬의 백엔드 오브젝트 저장소이며 블록 장치에 개체를 직접 배치합니다.



중요

bluestore는 프로덕션 환경의 **OSD** 데몬에 대한 고성능 백엔드를 제공합니다. 기본적으로 **BlueStore**는 자체 조정하도록 구성되어 있습니다. **BlueStore**가 수동으로 튜닝된 환경에서 더 잘 작동하는 것으로 판단되는 경우 **Red Hat** 지원팀에 연락하여 자동 튜닝 기능을 개선하는 데 도움이 되도록 설정 세부 정보를 공유하십시오. **Red Hat**은 고객의 의견을 기다리고 귀하의 권장 사항을 알려 드립니다.

9.1. CEPH BLUESTORE

다음은 **BlueStore** 사용의 주요 기능 중 일부입니다.

스토리지 장치 직접 관리

bluestore는 원시 블록 장치 또는 파티션을 사용합니다. 이렇게 하면 성능을 제한하거나 복잡성을 추가할 수 있는 **XFS**와 같은 로컬 파일 시스템과 같은 추상화 계층을 방해할 수 있습니다.

RobsDB의 메타데이터 관리

bluestore는 **RocksDB** 키-값 데이터베이스를 사용하여 개체 이름에서 디스크의 위치 블록 위치와 같은 내부 메타데이터를 관리합니다.

전체 데이터 및 메타데이터 체크섬

기본적으로 **BlueStore**에 기록된 모든 데이터 및 메타데이터는 하나 이상의 체크섬으로 보호됩니다. 데이터 또는 메타데이터는 확인 없이 디스크에서 읽거나 사용자에게 반환되지 않습니다.

효율적인 **COW(Copy-On-Write)**

Ceph 블록 장치 및 **Ceph** 파일 시스템 스냅샷은 **BlueStore**에서 효율적으로 구현된 **COW(Copy-On-Write)** 복제 메커니즘을 사용합니다. 그러면 효율적인 **2**단계 커밋을 구현하기 위해 복제를 사용하는 일반 스냅샷과 삭제 코드 풀의 경우 **I/O**를 효율적으로 수행할 수 있습니다.

큰 이중 쓰기 없음

bluestore는 먼저 블록 장치의 할당되지 않은 공간에 새 데이터를 작성한 다음, 개체 메타데이터를 업데이트하여 디스크의 새 영역을 참조하도록 **viewingsDB** 트랜잭션을 커밋합니다. 쓰기 작업이 구성 가능한 크기 임계값 미만인 경우에만 쓰기 작업이 쓰기 계획으로 대체됩니다.

다중 장치 지원

bluestore는 다른 데이터를 저장하는 데 여러 블록 장치를 사용할 수 있습니다. 예: 데이터의 하드 디스크 드라이브(HDD) 메타데이터, NVMe(Non-volatile Memory) 또는 NVRAM(Non-volatile random-access memory)에 대한 SSD(하드 디스크 드라이브) 또는 pxesDB 쓰기-ahead 로그(WAL) 용 영구 메모리입니다. 자세한 내용은 Ceph BlueStore 장치를 참조하십시오.

효율적인 블록 장치 사용

BlueStore는 파일 시스템을 사용하지 않으므로 스토리지 장치 캐시를 지우는 필요성을 최소화합니다.

9.2. CEPH BLUESTORE 장치

이 섹션에서는 BlueStore 백엔드에서 사용하는 블록 장치에 대해 설명합니다.

bluestore는 하나, 두 개 또는 세 개의 저장 장치를 관리합니다.

- primary
- WAL
- DB

가장 간단한 경우 BlueStore는 단일 (primary) 스토리지 장치를 사용합니다. 스토리지 장치는 다음을 포함하는 두 부분으로 분할됩니다.

- OSD 메타데이터: OSD의 기본 메타데이터가 포함된 XFS로 포맷된 작은 파티션입니다. 이 데이터 디렉터리에는 식별자, 클러스터가 속한 클러스터 및 개인 인증 키와 같은 OSD에 대한 정보가 포함되어 있습니다.
- Data: BlueStore에서 직접 관리하고 모든 OSD 데이터가 포함된 나머지 장치를 처리하는 큰 파티션입니다. 이 기본 장치는 데이터 디렉터리의 블록 심볼릭 링크로 식별됩니다.

두 개의 추가 장치를 사용할 수도 있습니다.

-

WAL (Write-ahead-log) 장치: BlueStore 내부 저널 또는 쓰기 로그를 저장하는 장치입니다. 이는 데이터 디렉토리에서 **block.wal** 심볼릭 링크로 식별됩니다. 장치가 기본 장치보다 빠른 경우에만 **WAL** 장치를 사용하는 것이 좋습니다. 예를 들어 **WAL** 장치는 **SSD** 디스크를 사용하고 기본 장치는 **HDD** 디스크를 사용합니다.

- **DB 장치: BlueStore** 내부 메타데이터를 저장하는 장치입니다. 임베디드 **backsDB** 데이터베이스는 성능을 개선하기 위해 기본 장치가 아닌 **DB** 장치에 가능한 한 많은 메타데이터를 배치합니다. **DB** 장치가 가득 차 있으면 기본 장치에 메타데이터 추가를 시작합니다. 장치가 기본 장치보다 빠른 경우에만 **DB** 장치를 사용하는 것이 좋습니다.

주의

빠른 장치에서 사용할 수 있는 기가바이트 미만의 스토리지가 있는 경우. **Red Hat**은 이를 **WAL** 장치로 사용하는 것이 좋습니다. 더 빠른 장치를 사용할 수 있는 경우 **DB** 장치로 사용하십시오. **BlueStore** 저널은 항상 가장 빠른 장치에 배치되므로 **DB** 장치를 사용하면 추가 메타데이터를 저장할 수 있는 것과 동일한 이점이 있습니다.

9.3. CEPH BLUESTORE 캐싱

BlueStore 캐시는 구성에 따라 **OSD** 데몬에서 읽거나 디스크에 쓰기 때문에 데이터로 채울 수 있는 버퍼 컬렉션입니다. 기본적으로 **Red Hat Ceph Storage**에서 **BlueStore**는 읽기 시 캐시되지만 쓰기는 하지 않습니다. 이는 **cache** 제거와 관련된 잠재적인 오버헤드를 방지하기 위해 **bluestore_default_buffered_write** 옵션이 **false** 로 설정되어 있기 때문입니다.

bluestore_default_buffered_write 옵션이 **true** 로 설정된 경우 데이터를 먼저 버퍼에 쓴 다음 디스크에 커밋됩니다. 그 후 해당 데이터가 제거될 때까지 캐시에 이미 있는 데이터에 대한 액세스를 더 빠르게 읽을 수 있도록 클라이언트에게 쓰기 승인이 전송됩니다.

읽기-복구 워크로드에서는 **BlueStore** 캐싱의 즉각적인 이점이 표시되지 않습니다. 더 많은 읽기가 완료되면 시간이 지남에 따라 캐시가 증가하고 후속 읽기는 성능이 향상됩니다. 캐시가 빠르게 채워지는 방법은 **BlueStore** 블록 및 데이터베이스 디스크 유형 및 클라이언트의 워크로드 요구 사항에 따라 달라집니다.

중요

bluestore_default_buffered_write 옵션을 활성화하기 전에 **Red Hat** 지원에 문의하십시오.

9.4. CEPH BLUESTORE에 대한 크기 조정 고려 사항

BlueStore OSD를 사용하여 기존 및 솔리드 스테이트 드라이브를 혼합할 때 RobsDB 논리 볼륨 (block.db)의 크기를 적절히 조정해야 합니다. populars DB 논리 볼륨은 오브젝트, 파일, 혼합 워크로드의 블록 크기 중 4% 미만이 될 것을 권장합니다. Red Hat은 RocksDB 및 OpenStack 블록 워크로드를 사용하여 BlueStore 블록 크기의 1%를 지원합니다. 예를 들어, 블록 크기가 오브젝트 워크로드의 경우 1TB인 경우 최소 40GBsDB 논리 볼륨을 생성합니다.

드라이브 유형을 혼합하지 않는 경우 별도의 telnetsDB 논리 볼륨을 보유할 필요가 없습니다. bluestore는 로크DB의 크기를 자동으로 관리합니다.

bluestore의 캐시 메모리는centossDB, BlueStore 메타데이터 및 오브젝트 데이터의 키-값 쌍 메타데이터에 사용됩니다.

참고

BlueStore 캐시 메모리 값은 OSD에서 이미 소비 중인 메모리 풋프린트 외에 있습니다.

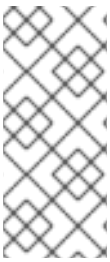
9.5. BLUESTORE_MIN_ALLOC_SIZE 매개변수를 사용하여 CEPH BLUESTORE 튜닝

BlueStore에서 원시 파티션은 bluestore_min_alloc_size 의 청크로 할당 및 관리됩니다. 기본적으로 bluestore_min_alloc_size 는 4096 이며 HDD 및 SSD의 경우 4KiB와 동일합니다. 각 청크의 작성되지 않은 영역은 원시 파티션에 작성될 때 0으로 채워집니다. 예를 들어 작은 오브젝트를 작성할 때와 같이 워크로드에 대해 올바르게 크기가 조정되지 않은 경우 사용되지 않은 공간을 낭비할 수 있습니다.

가장 작은 쓰기와 일치하도록 bluestore_min_alloc_size 를 설정하는 것이 가장 좋은 방법입니다.

중요

bluestore_min_alloc_size 값을 변경하는 것은 권장되지 않습니다. 도움이 필요한 경우 Red Hat 지원에 문의하십시오.



참고

bluestore_min_alloc_size_sd 및 bluestore_min_alloc_size_hdd 설정은 SSD 및 HDD에 고유하지만 bluestore_min_alloc_size 를 설정하면 이를 재정의할 필요가 없습니다.

사전 요구 사항

- *실행 중인 **Red Hat Ceph Storage** 클러스터.*
- ***Ceph** 모니터 및 관리자가 클러스터에 배포됩니다.*
- ***OSD** 노드로 새로 프로비저닝할 수 있는 서버 또는 노드*
- *기존 **Ceph OSD** 노드를 재배포하는 경우 **Ceph Monitor** 노드의 관리자 인증 키입니다.*

절차

1. *부트스트랩 노드에서 **bluestore_min_alloc_size** 매개변수 값을 변경합니다.*

   구문

   ```
   ceph config set osd.OSD_ID bluestore_min_alloc_size_DEVICE_NAME_ VALUE
   ```

   예제

   ```
   [ceph: root@host01 /]# ceph config set osd.4 bluestore_min_alloc_size_hdd 6144
   ```

   ***bluestore_min_alloc_size** 가 **6KiB**에 해당하는 **6144** 바이트로 설정되어 있음을 확인할 수 있습니다.*

2. ***OSD** 서비스를 다시 시작합니다.*

   구문

```
systemctl restart SERVICE_ID
```

예제

```
[ceph: root@host01 /]# systemctl restart ceph-499829b4-832f-11eb-8d6d-
001a4a000635@osd.4.service
```

검증

- ***ceph daemon** 명령을 사용하여 설정을 확인합니다.*

구문

```
ceph daemon osd.OSD_ID config get bluestore_min_alloc_size__DEVICE_
```

예제

```
[ceph: root@host01 /]# ceph daemon osd.4 config get bluestore_min_alloc_size_hdd

ceph daemon osd.4 config get bluestore_min_alloc_size
{
    "bluestore_min_alloc_size": "6144"
}
```

9.6. BLUESTORE 관리자 툴을 사용하여 ROBSDB 데이터베이스 재하드

BlueStore 관리자 도구를 사용하여 데이터베이스를 다시 지정할 수 있습니다. BlueStore의 RobsDB 데이터베이스를 OSD를 재배포하지 않고 한 가지 모양에서 다른 열 제품군으로 변환합니다. 열 제품군은 전체 데이터베이스와 동일한 기능을 가지고 있지만 사용자가 작은 데이터 세트에서 작동하고 다른 옵션을 적용할 수 있습니다. 저장된 다양한 키 수명을 활용합니다. 새 키를 생성하거나 기존 키를 삭제하지 않고 변환 중에 키가 이동합니다.

사전 요구 사항

- *실행 중인 Red Hat Ceph Storage 클러스터.*
- *BlueStore로 구성된 오브젝트 저장소입니다.*
- *호스트에 배포된 OSD 노드입니다.*
- *모든 호스트에 대한 루트 수준 액세스.*
- *모든 호스트에서 ceph-common 및 cephadm 패키지가 제거되었습니다.*

절차

1. *cephadm 쉘에 로그인합니다.*

   예제

   ```
   [root@host01 ~]# cephadm shell
   ```

2. *관리 노드에서 OSD_ID 및 호스트 세부 정보를 가져옵니다.*

   예제

-

```
[ceph: root@host01 /]# ceph orch ps
```

3. *각 호스트에 **root** 사용자로 로그인하고 **OSD**를 중지합니다.*

구문

```
cephadm unit --name OSD_ID stop
```

예제

```
[root@host02 ~]# cephadm unit --name osd.0 stop
```

4. *중지된 **OSD** 데몬 컨테이너에 입력합니다.*

구문

```
cephadm shell --name OSD_ID
```

예제

```
[root@host02 ~]# cephadm shell --name osd.0
```

5. ***cephadm*** *셸에 로그인하고 파일 시스템의 일관성을 확인합니다.*

구문

```
ceph-bluestore-tool --path/var/lib/ceph/osd/ceph-OSD_ID/ fsck
```

예제

```
[ceph: root@host02 /]# ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-0/ fsck

fsck success
```

6. ***OSD*** *노드의 분할 상태를 확인합니다.*

구문

```
ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-OSD_ID/ show-sharding
```

예제

```
[ceph: root@host02 /]# ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-6/ show-sharding

m(3) p(3,0-12) O(3,0-13) L P
```

7. ***ceph-bluestore-tool** 명령을 실행하여 재정의할 수 있습니다. 명령에 지정된 매개 변수를 사용하는 것이 좋습니다.*

구문

```
ceph-bluestore-tool --log-level 10 -l log.txt --path /var/lib/ceph/osd/ceph-OSD_ID/ --sharding="m(3) p(3,0-12) O(3,0-13)=block_cache={type=binned_lru} L P" reshard
```

예제

```
[ceph: root@host02 /]# ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-6/ --sharding="m(3) p(3,0-12) O(3,0-13)=block_cache={type=binned_lru} L P" reshard

reshard success
```

8. ***OSD** 노드의 분할 상태를 확인하려면 **show-sharding** 명령을 실행합니다.*

구문

```
ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-OSD_ID/ show-sharding
```

예제

```
[ceph: root@host02 /]# ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-6/ show-sharding
m(3) p(3,0-12) O(3,0-13)=block_cache={type=binned_lru} L P
```

9. ***cephadm** 쉘을 종료합니다.*

```
[ceph: root@host02 /]# exit
```

10. *각 호스트에 **root** 사용자로 로그인하고 **OSD**를 시작합니다.*

구문

```
cephadm unit --name OSD_ID start
```

예제

```
[root@host02 ~]# cephadm unit --name osd.0 start
```

추가 리소스

- *자세한 내용은 **Red Hat Ceph Storage** 설치 가이드 를 참조하십시오.*

9.7. BLUESTORE 조각화 툴

*스토리지 관리자는 **BlueStore OSD**의 조각화 수준을 주기적으로 확인하려고 합니다. 오프라인 또는 온라인 **OSD**에 대해 하나의 간단한 명령으로 조각화 수준을 확인할 수 있습니다.*

9.7.1. 사전 요구 사항

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

9.7.2. BlueStore 조각화 도구는 무엇입니까?

BlueStore OSD의 경우 사용 가능한 공간은 기본 스토리지 장치의 시간이 지남에 따라 조각화됩니다. 일부 조각화는 정상이지만 과도한 조각화가 발생하면 성능이 저하됩니다.

BlueStore 조각화 툴은 **BlueStore OSD**의 조각화 수준에서 점수를 생성합니다. 이 조각화 점수는 **0**에서 **1**까지의 범위로 지정됩니다. 점수 **0**은 조각화를 의미하고 **1**의 점수는 심각한 조각화를 의미합니다.

표 **9.1.** 조각 모음 점수의 의미

점수	조각화 Amount
0.0 - 0.4	작은 조각화는 하지 않습니다.
0.4 - 0.7	작고 허용 가능한 조각화입니다.
0.7 - 0.9	상당한 경우도 있지만 안전한 조각화는 가능합니다.
0.9 - 1.0	심각한 조각화를 수행할 수 있으며 이로 인해 성능 문제가 발생합니다.

중요

심각한 조각화가 있고 문제 해결에 도움이 필요한 경우 **Red Hat** 지원에 문의하십시오.

9.7.3. 조각화 확인

BlueStore OSD의 조각화 수준을 온라인으로 또는 오프라인으로 확인할 수 있습니다.

사전 요구 사항

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

- ***bluestore OSD.***

*온라인 **BlueStore** 조각화 점수*

1. *실행 중인 **BlueStore OSD** 프로세스를 검사합니다.*

   a. *간단한 보고서:*

   구문

   ```
   ceph daemon OSD_ID bluestore allocator score block
   ```

   예제

   ```
   [ceph: root@host01 /]# ceph daemon osd.123 bluestore allocator score block
   ```

   b. *더 자세한 보고서:*

   구문

   ```
   ceph daemon OSD_ID bluestore allocator dump block
   ```

   예제

```
[ceph: root@host01 /]# ceph daemon osd.123 bluestore allocator dump block
```

*오프라인 **BlueStore** 조각화 점수*

1. *오프라인 조각화 점수를 확인하기 위해 재정의하는 단계를 따릅니다.*

   예제

```
[root@host01 ~]# podman exec -it 7fbd6c6293c0 /bin/bash
```

1. *실행 중이 아닌 **BlueStore OSD** 프로세스를 검사합니다.*

   a. *간단한 보고서:*

   구문

```
ceph-bluestore-tool --path PATH_TO_OSD_DATA_DIRECTORY --allocator block free-score
```

   예제

```
[root@7fbd6c6293c0 /]# ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-123 --allocator block free-score
```

   b. *더 자세한 보고서:*

   구문

```
ceph-bluestore-tool --path PATH_TO_OSD_DATA_DIRECTORY --allocator block free-dump
block:
{
    "fragmentation_rating": 0.018290238194701977
}
```

예제

```
[root@7fbd6c6293c0 /]# ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-123 --allocator block free-dump
block:
{
    "capacity": 21470642176,
    "alloc_unit": 4096,
    "alloc_type": "hybrid",
    "alloc_name": "block",
    "extents": [
        {
            "offset": "0x370000",
            "length": "0x20000"
        },
        {
            "offset": "0x3a0000",
            "length": "0x10000"
        },
        {
            "offset": "0x3f0000",
            "length": "0x20000"
        },
        {
            "offset": "0x460000",
            "length": "0x10000"
        },
```

추가 리소스

- *조각화 점수에 대한 자세한 내용은 **BlueStore Fragmentation Tool** 을 참조하십시오.*

- *Resharding에 대한 자세한 내용은 BlueStore 관리 도구를 사용하여 RocskDB 데이터베이스 Resharding을 참조하십시오.*

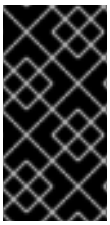
9.8. CEPH BLUESTORE BLUEFS

bluestore 블록 데이터베이스는 메타데이터를 RocksDB 데이터베이스에 키-값 쌍으로 저장합니다. 블록 데이터베이스는 스토리지 장치의 작은 BlueFS 파티션에 있습니다. BlueFS는 RocksDB 파일을 보관하도록 설계된 최소 파일 시스템입니다.

9.8.1. bluefs_buffered_io 설정 보기

스토리지 관리자는 bluefs_buffered_io 매개 변수의 현재 설정을 볼 수 있습니다.

Red Hat Ceph Storage에 대해 기본적으로 bluefs_buffered_io 옵션이 True 로 설정됩니다. 이 옵션을 사용하면 BlueFS가 일부 경우에 버퍼링된 읽기를 수행하고 커널 페이지 캐시가 RocksDB 블록 읽기와 같은 읽기의 보조 캐시 역할을 할 수 있습니다.



중요

bluefs_buffered_io 의 값을 변경하는 것은 권장되지 않습니다. bluefs_buffered_io 매개변수를 변경하기 전에 Red Hat 지원 계정 팀에 문의하십시오.

사전 요구 사항

- *실행 중인 Red Hat Ceph Storage 클러스터.*
- *Ceph Monitor 노드에 대한 루트 수준 액세스입니다.*

절차

1. *Cephadm 쉘에 로그인합니다.*

예제

```
[root@host01 ~]# cephadm shell
```

2. *다음 세 가지 방법으로 **bluefs_buffered_io** 매개변수의 현재 값을 볼 수 있습니다.*

방법 1

- *구성 데이터베이스에 저장된 값을 확인합니다.*

  예제

  ```
  [ceph: root@host01 /]# ceph config get osd bluefs_buffered_io
  ```

방법 2

- *특정 **OSD**의 구성 데이터베이스에 저장된 값을 확인합니다.*

  구문

  ```
  ceph config get OSD_ID bluefs_buffered_io
  ```

  예제

  ```
  [ceph: root@host01 /]# ceph config get osd.2 bluefs_buffered_io
  ```

방법 3

- *실행 중인 값이 구성 데이터베이스에 저장된 값과 다른 **OSD**의 실행 값을 확인합니다.*

구문

```
ceph config show OSD_ID bluefs_buffered_io
```

예제

```
[ceph: root@host01 /]# ceph config show osd.3 bluefs_buffered_io
```

10장. CEPHADM 문제 해결

스토리지 관리자는 **Red Hat Ceph Storage** 클러스터의 문제를 해결할 수 있습니다. **Cephadm** 명령이 실패하거나 특정 서비스가 제대로 실행되지 않는 이유를 조사해야 하는 경우가 있습니다.

10.1. 사전 요구 사항

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

10.2. CEPHADM 일시 중지 또는 비활성화

Cephadm이 예상대로 작동하지 않는 경우 다음 명령을 사용하여 대부분의 백그라운드 활동을 일시 중지할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph orch pause
```

이렇게 하면 변경 사항이 중지되지만 **Cephadm**은 주기적으로 호스트를 확인하여 데몬 및 장치의 인벤토리를 새로 고칩니다.

Cephadm을 완전히 비활성화하려면 다음 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph orch backend
ceph mgr module disable cephadm
```

이전에 배포한 데몬 컨테이너는 이전과 마찬가지로 계속 존재하며 시작합니다.

10.3. 서비스당 및 데몬 이벤트당

Cephadm은 실패한 데몬 배포를 디버깅하기 위해 서비스당 이벤트를 저장하고 데몬별로 저장합니다. 이러한 이벤트에는 종종 관련 정보가 포함됩니다.

서비스당

구문

```
ceph orch ls --service_name SERVICE_NAME --format yaml
```

예제

```
[ceph: root@host01 /]# ceph orch ls --service_name alertmanager --format yaml
service_type: alertmanager
service_name: alertmanager
placement:
  hosts:
  - unknown_host
status:
  ...
  running: 1
  size: 1
events:
- 2021-02-01T08:58:02.741162 service:alertmanager [INFO] "service was created"
- '2021-02-01T12:09:25.264584 service:alertmanager [ERROR] "Failed to apply: Cannot
  place <AlertManagerSpec for service_name=alertmanager> on unknown_host: Unknown hosts"'
```

데몬당

구문

```
ceph orch ps --service-name SERVICE_NAME --daemon-id DAEMON_ID --format yaml
```

예제

```
[ceph: root@host01 /]# ceph orch ps --service-name mds --daemon-id cephfs.hostname.ppdhsz --
format yaml
daemon_type: mds
daemon_id: cephfs.hostname.ppdhsz
hostname: hostname
status_desc: running
...
events:
- 2021-02-01T08:59:43.845866 daemon:mds.cephfs.hostname.ppdhsz [INFO] "Reconfigured
  mds.cephfs.hostname.ppdhsz on host 'hostname'"
```

10.4. CEPHADM 로그 확인

*다음 명령을 사용하여 **Cephadm** 로그를 실시간으로 모니터링할 수 있습니다.*

예제

```
[ceph: root@host01 /]# ceph -W cephadm
```

다음 명령을 사용하여 마지막 몇 개의 메시지를 볼 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph log last cephadm
```

파일에 로깅을 활성화한 경우 모니터 호스트에서 **ceph.cephadm.log** 라는 **Cephadm** 로그 파일을 볼 수 있습니다.

10.5. 로그 파일 수집

journalctl 명령을 사용하여 모든 데몬의 로그 파일을 수집할 수 있습니다.

참고

cephadm 쉘 외부에서 이러한 명령을 모두 실행해야 합니다.

참고

기본적으로 **Cephadm**은 **journald**에 로그를 저장하므로 **/var/log/ceph** 에서 데몬 로그를 더 이상 사용할 수 없습니다.

- 특정 데몬의 로그 파일을 읽으려면 다음 명령을 실행합니다.

구문

```
cephadm logs --name DAEMON_NAME
```

예제

```
[root@host01 ~]# cephadm logs --name cephfs.hostname.ppdhsz
```

참고

이 명령은 데몬이 실행 중인 동일한 호스트에서 실행됩니다.

- *다른 호스트에서 실행 중인 특정 데몬의 로그 파일을 읽으려면 다음 명령을 실행합니다.*

구문

```
cephadm logs --fsid FSID --name DAEMON_NAME
```

예제

```
[root@host01 ~]# cephadm logs --fsid 2d2fd136-6df1-11ea-ae74-002590e526e8 --name
cephfs.hostname.ppdhsz
```

*여기서 **fsid** 는 **ceph status** 명령에서 제공하는 클러스터 **ID**입니다.*

- *지정된 호스트에서 모든 데몬의 모든 로그 파일을 가져오려면 다음 명령을 실행합니다.*

구문

```
for name in $(cephadm ls | python3 -c "import sys, json; [print(i['name']) for i in
json.load(sys.stdin)]") ; do cephadm logs --fsid FSID_OF_CLUSTER --name "$name" >
$name; done
```

예제

```
[root@host01 ~]# for name in $(cephadm ls | python3 -c "import sys, json; [print(i['name']) for i in json.load(sys.stdin)]") ; do cephadm logs --fsid 57bddb48-ee04-11eb-9962-001a4a000672 --name "$name" > $name; done
```

10.6. SYSTEMD 상태 수집

- ***systemd** 장치의 상태를 출력하려면 다음 명령을 실행합니다.*

예제

```
[root@host01 ~]$ systemctl status ceph-a538d494-fb2a-48e4-82c8-b91c37bb0684@mon.host01.service
```

10.7. 다운로드한 모든 컨테이너 이미지 나열

호스트에서 다운로드한 모든 컨테이너 이미지를 나열하려면 다음 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# podman ps -a --format json | jq '.[].Image'
"docker.io/library/rhel8"
"registry.redhat.io/rhceph-alpha/rhceph-5-rhel8@sha256:9aaea414e2c263216f3cdcb7a096f57c3adf6125ec9f4b0f5f65fa8c43987155"
```

10.8. 수동으로 컨테이너 실행

Cephadm은 컨테이너를 실행하는 작은 래퍼를 작성합니다. 컨테이너 실행 명령을 실행하려면 **/var/lib/ceph/CLUSTER_FSID/SERVICE_NAME/unit** 을 참조하십시오.

SSH 오류 분석

다음과 같은 오류가 발생하는 경우:

예제

```
execnet.gateway_bootstrap.HostNotFound: -F /tmp/cephadm-conf-73z09u6g -i /tmp/cephadm-identity-ky7ahp_5 root@10.10.1.2
...
raise OrchestratorError(msg) from e
orchestrator._interface.OrchestratorError: Failed to connect to 10.10.1.2 (10.10.1.2).
Please make sure that the host is reachable and accepts connections using the cephadm SSH key
```

다음 옵션을 사용하여 문제를 해결하십시오.

- **Cephadm**에 **SSH ID** 키가 있는지 확인하려면 다음 명령을 실행합니다.

  예제

  ```
  [ceph: root@host01 /]# ceph config-key get mgr/cephadm/ssh_identity_key > ~/cephadm_private_key
  INFO:cephadm:Inferring fsid f8edc08a-7f17-11ea-8707-000c2915dd98
  INFO:cephadm:Using recent ceph image docker.io/ceph/ceph:v15 obtained 'mgr/cephadm/ssh_identity_key'
  [root@mon1 ~] # chmod 0600 ~/cephadm_private_key
  ```

  위의 명령이 실패하면 **Cephadm**에 키가 없습니다. **SSH** 키를 생성하려면 다음 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# chmod 0600 ~/cephadm_private_key
```

또는

예제

```
[ceph: root@host01 /]# cat ~/cephadm_private_key | ceph cephadm set-ssk-key -i-
```

- ***SSH** 구성이 올바른지 확인하려면 다음 명령을 실행합니다.*

예제

```
[ceph: root@host01 /]# ceph cephadm get-ssh-config
```

- *호스트 연결을 확인하려면 다음 명령을 실행합니다.*

예제

```
[ceph: root@host01 /]# ssh -F config -i ~/cephadm_private_key root@host01
```

*공개 키가 **authorized_keys** 에 있는지 확인합니다.*

*공개 키가 **authorized_keys** 파일에 있는지 확인하려면 다음 명령을 실행합니다.*

예제

```
[ceph: root@host01 /]# ceph cephadm get-pub-key
[ceph: root@host01 /]# grep "`cat ~/ceph.pub`" /root/.ssh/authorized_keys
```

10.9. CIDR 네트워크 오류

***CIDR(Classless inter domain routing)은 IP(Internet Protocol)** 주소를 할당하는 메서드이며, **Cephadm** 로그 항목은 주소 배포의 효율성을 향상시키고 클래스 **A**, 클래스 **B** 및 클래스 **C** 네트워크를 기반으로 이전 시스템을 대체하는 현재 상태를 보여줍니다. 다음 오류 중 하나가 표시되면 다음을 수행하십시오.*

***ERROR: Failed to infer CIDR network for mon ip *; 나중에 구성할 --skip-mon-network**를 전달합니다.*

또는

***public_network config** 옵션을 설정하거나 **CIDR** 네트워크, **ceph addrvec** 또는 일반 **IP**를 지정해야 합니다.*

다음 명령을 실행해야 합니다.

예제

```
[ceph: root@host01 /]# ceph config set host public_network hostnetwork
```

10.10. ADMIN 소켓에 액세스

각 Ceph 데몬은 MON을 바이패스하는 관리 소켓을 제공합니다.

관리 소켓에 액세스하려면 호스트에 데몬 컨테이너를 입력합니다.

예제

```
[ceph: root@host01 /]# cephadm enter --name cephfs.hostname.ppdhsz
[ceph: root@mon1 /]# ceph --admin-daemon /var/run/ceph/ceph-cephfs.hostname.ppdhsz.asok
config show
```

10.11. 수동으로 MGR 데몬 배포

Cephadm은 Red Hat Ceph Storage 클러스터를 관리하기 위해 mgr 데몬이 필요합니다. Red Hat Ceph Storage 클러스터의 마지막 mgr 데몬이 제거된 경우 Red Hat Ceph Storage 클러스터의 임의의 호스트에 mgr 데몬을 수동으로 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준 액세스.
- 호스트는 클러스터에 추가됩니다.

절차

1. Cephadm 쉘에 로그인합니다.

예제

```
[root@host01 ~]# cephadm shell
```

2. *다음 명령으로 **Cephadm** 스케줄러를 비활성화하여 **Cephadm**이 새 **MGR** 데몬을 제거하지 못하도록 합니다.*

예제

```
[ceph: root@host01 /]# ceph config-key set mgr/cephadm/pause true
```

3. *새 **MGR** 데몬에 대한 **auth** 항목을 가져오거나 생성합니다.*

예제

```
[ceph: root@host01 /]# ceph auth get-or-create mgr.host01.smfvfd1 mon "profile mgr" osd
"allow *" mds "allow *"
[mgr.host01.smfvfd1]
key = AQDhcORgW8toCRAAlMzlqWXnh3cGRjqYEa9ikw==
```

4. ***ceph.conf** 파일을 엽니다.*

예제

```
[ceph: root@host01 /]# ceph config generate-minimal-conf
# minimal ceph.conf for 8c9b0072-67ca-11eb-af06-001a4a0002a0
```

```
[global]
fsid = 8c9b0072-67ca-11eb-af06-001a4a0002a0
mon_host = [v2:10.10.200.10:3300/0,v1:10.10.200.10:6789/0]
[v2:10.10.10.100:3300/0,v1:10.10.200.100:6789/0]
```

5. *컨테이너 이미지를 가져옵니다.*

예제

```
[ceph: root@host01 /]# ceph config get "mgr.host01.smfvfd1" container_image
```

6. ***config-json.json** 파일을 생성하고 다음을 추가합니다.*

참고

***ceph config generate-minimal-conf** 명령의 출력에서 값을 사용합니다.*

예제

```
{
  {
  "config": "# minimal ceph.conf for 8c9b0072-67ca-11eb-af06-001a4a0002a0\n[global]\n\tfsid
= 8c9b0072-67ca-11eb-af06-001a4a0002a0\n\tmon_host =
[v2:10.10.200.10:3300/0,v1:10.10.200.10:6789/0]
[v2:10.10.10.100:3300/0,v1:10.10.200.100:6789/0]\n",
  "keyring": "[mgr.Ceph5-2.smfvfd1]\n\tkey =
AQDhcORgW8toCRAAlMzlqWXnh3cGRjqYEa9ikw==\n"
}
}
```

7. ***Cephadm** 쉘을 종료합니다.*

예제

```
[ceph: root@host01 /]# exit
```

8. ***MGR** 데몬을 배포합니다.*

예제

```
[root@host01 ~]# cephadm --image registry.redhat.io/rhceph-alpha/rhceph-5-rhel8:latest deploy --fsid  8c9b0072-67ca-11eb-af06-001a4a0002a0 --name mgr.host01.smfvfd1 --config-json config-json.json
```

검증

***Cephadm** 쉘에서 다음 명령을 실행합니다.*

예제

```
[ceph: root@host01 /]# ceph -s
```

*새로운 **mgr** 데몬이 추가되었음을 알 수 있습니다.*

11장. CEPHADM 작업

스토리지 관리자는 **Red Hat Ceph Storage** 클러스터에서 **Cephadm** 작업을 수행할 수 있습니다.

11.1. 사전 요구 사항

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

11.2. CEPHADM 로그 메시지 모니터링

Cephadm 로그는 **cephadm** 클러스터 로그 채널에 기록되므로 실시간으로 진행 상황을 모니터링할 수 있습니다.

- 실시간 진행 상황을 모니터링하려면 다음 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph -W cephadm
```

예제

```
2022-06-10T17:51:36.335728+0000 mgr.Ceph5-1.nqikfh [INF] refreshing Ceph5-adm facts
2022-06-10T17:51:37.170982+0000 mgr.Ceph5-1.nqikfh [INF] deploying 1 monitor(s) instead
of 2 so monitors may achieve consensus
2022-06-10T17:51:37.173487+0000 mgr.Ceph5-1.nqikfh [ERR] It is NOT safe to stop
['mon.Ceph5-adm']: not enough monitors would be available (Ceph5-2) after stopping mons
[Ceph5-adm]
2022-06-10T17:51:37.174415+0000 mgr.Ceph5-1.nqikfh [INF] Checking pool "nfs-ganesha"
exists for service nfs.foo
2022-06-10T17:51:37.176389+0000 mgr.Ceph5-1.nqikfh [ERR] Failed to apply nfs.foo spec
NFSServiceSpec({'placement': PlacementSpec(count=1), 'service_type': 'nfs', 'service_id':
'foo', 'unmanaged': False, 'preview_only': False, 'pool': 'nfs-ganesha', 'namespace': 'nfs-ns'}):
Cannot find pool "nfs-ganesha" for service nfs.foo
Traceback (most recent call last):
  File "/usr/share/ceph/mgr/cephadm/serve.py", line 408, in _apply_all_services
    if self._apply_service(spec):
```

```
  File "/usr/share/ceph/mgr/cephadm/serve.py", line 509, in _apply_service
    config_func(spec)
  File "/usr/share/ceph/mgr/cephadm/services/nfs.py", line 23, in config
    self.mgr._check_pool_exists(spec.pool, spec.service_name())
  File "/usr/share/ceph/mgr/cephadm/module.py", line 1840, in _check_pool_exists
    raise OrchestratorError(f'Cannot find pool "{pool}" for '
orchestrator._interface.OrchestratorError: Cannot find pool "nfs-ganesha" for service nfs.foo
2022-06-10T17:51:37.179658+0000 mgr.Ceph5-1.nqikfh [INF] Found osd claims -> {}
2022-06-10T17:51:37.180116+0000 mgr.Ceph5-1.nqikfh [INF] Found osd claims for
drivegroup all-available-devices -> {}
2022-06-10T17:51:37.182138+0000 mgr.Ceph5-1.nqikfh [INF] Applying all-available-devices
on host Ceph5-adm...
2022-06-10T17:51:37.182987+0000 mgr.Ceph5-1.nqikfh [INF] Applying all-available-devices
on host Ceph5-1...
2022-06-10T17:51:37.183395+0000 mgr.Ceph5-1.nqikfh [INF] Applying all-available-devices
on host Ceph5-2...
2022-06-10T17:51:43.373570+0000 mgr.Ceph5-1.nqikfh [INF] Reconfiguring node-
exporter.Ceph5-1 (unknown last config time)...
2022-06-10T17:51:43.373840+0000 mgr.Ceph5-1.nqikfh [INF] Reconfiguring daemon node-
exporter.Ceph5-1 on Ceph5-1
```

- *기본적으로 로그는 정보 수준 이벤트 이상을 표시합니다. 디버그 수준 메시지를 보려면 다음 명령을 실행합니다.*

  예제

  ```
  [ceph: root@host01 /]# ceph config set mgr mgr/cephadm/log_to_cluster_level debug
  [ceph: root@host01 /]# ceph -W cephadm --watch-debug
  [ceph: root@host01 /]# ceph -W cephadm --verbose
  ```

- *디버깅 수준을 기본 정보로 반환:*

  예제

  ```
  [ceph: root@host01 /]# ceph config set mgr mgr/cephadm/log_to_cluster_level info
  ```

- 최근 이벤트를 보려면 다음 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph log last cephadm
```

이러한 이벤트는 모니터 호스트 및 모니터 데몬의 **stderr**에 있는 **ceph.cephadm.log** 파일에도 기록됩니다.

11.3. CEPH 데몬 로그

stderr 또는 파일을 통해 **Ceph** 데몬 로그를 볼 수 있습니다.

stdout에 로깅

일반적으로 **Ceph** 데몬은 **/var/log/ceph** 에 기록되었습니다. 기본적으로 **Cephadm** 데몬은 **stderr** 에 기록되고 로그는 컨테이너 런타임 환경에서 캡처됩니다. 대부분의 시스템에서 이러한 로그는 **journald** 로 전송되고 **journalctl** 명령을 통해 액세스할 수 있습니다.

- 예를 들어 **ID**가 **5c5a50ae-272a-455d-99e9-32c6a013e694** 인 스토리지 클러스터의 **host01** 에서 데몬의 로그를 보려면 다음을 수행합니다.

예제

```
[ceph: root@host01 /]# journalctl -u ceph-5c5a50ae-272a-455d-99e9-
32c6a013e694@host01
```

*이는 로깅 수준이 낮은 경우 일반 **Cephadm** 작업에 적합합니다.*

- ***stderr** 에 로깅을 비활성화하려면 다음 값을 설정합니다.*

예제

```
[ceph: root@host01 /]# ceph config set global log_to_stderr false
[ceph: root@host01 /]# ceph config set global mon_cluster_log_to_stderr false
```

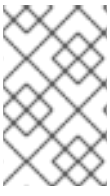
파일에 로깅

***stderr** 대신 파일에 기록하도록 **Ceph** 데몬을 구성할 수도 있습니다. 파일에 로깅할 때 **Ceph** 로그는 **/var/log/ceph/CLUSTER_FSID** 에 있습니다.*

- *파일에 로깅을 활성화하려면 다음 값을 설정합니다.*

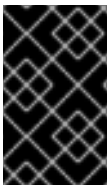
예제

```
[ceph: root@host01 /]# ceph config set global log_to_file true
[ceph: root@host01 /]# ceph config set global mon_cluster_log_to_file true
```



참고

***Red Hat**은 **stderr** 에 로깅을 비활성화하여 이중 로그를 방지하는 것이 좋습니다.*



중요

현재 기본이 아닌 경로로의 로그 회전은 지원되지 않습니다.

기본적으로 Cephadm은 이러한 파일을 순환하기 위해 각 호스트에서 로그 회전을 설정합니다. /etc/logrotate.d/ceph.CLUSTER_FSID 를 수정하여 로깅 보존 일정을 구성할 수 있습니다.

11.4. 데이터 위치

Cephadm 데몬 데이터 및 로그는 이전 버전의 Ceph와 약간 다른 위치에 있습니다.

- /var/log/ceph/CLUSTER_FSID 에는 모든 스토리지 클러스터 로그가 포함되어 있습니다. stderr 및 컨테이너 런타임을 통한 기본 Cephadm 로그로, 이러한 로그는 일반적으로 존재하지 않습니다.
- /var/lib/ceph/CLUSTER_FSID 에는 로그 외에도 모든 클러스터 데몬 데이터가 포함되어 있습니다.
- var/lib/ceph/CLUSTER_FSID/ DAE 6443_NAME 에는 특정 데몬에 대한 모든 데이터가 포함되어 있습니다.
- /var/lib/ceph/CLUSTER_FSID/crash 에는 스토리지 클러스터에 대한 충돌 보고서가 포함되어 있습니다.
- /var/lib/ceph/CLUSTER_FSID/removed 에는 Cephadm에서 제거된 상태 저장 데몬(예: monitor 또는 Prometheus)의 이전 데몬 데이터 디렉터리가 포함되어 있습니다.

디스크 사용량

일부 Ceph 데몬은 /var/lib/ceph, 특히 모니터 및 Prometheus 데몬에 상당한 양의 데이터를 저장할 수 있으므로 Red Hat은 이 디렉터리를 자체 디스크, 파티션 또는 논리 볼륨으로 이동하여 루트 파일 시스템이 가득 차지 않도록 하는 것이 좋습니다.

11.5. CEPHADM 상태 점검

스토리지 관리자는 Cephadm 모듈에서 제공하는 추가 상태 확인을 사용하여 Red Hat Ceph Storage 클러스터를 모니터링할 수 있습니다. 이는 스토리지 클러스터에서 제공하는 기본 상태 점검의 보조입니다.

11.5.1. 사전 요구 사항

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

11.5.2. Cephadm 작업 상태 점검

Cephadm 모듈이 활성화되면 상태 점검이 실행됩니다. 다음과 같은 상태 경고를 받을 수 있습니다.

CEPHADM_PAUSED

ceph orch pause 명령을 사용하여 **Cephadm** 백그라운드 작업이 일시 중지되었습니다. **Cephadm**은 호스트 및 데몬 상태 확인과 같은 수동 모니터링 활동을 계속 수행하지만 데몬 배포 또는 제거와 같은 변경은 하지 않습니다. **ceph orch resume** 명령을 사용하여 **Cephadm** 작업을 다시 시작할 수 있습니다.

CEPHADM_STRAY_HOST

하나 이상의 호스트가 **Ceph** 데몬을 실행하고 있지만 **Cephadm** 모듈에서 관리하는 호스트로 등록되지 않습니다. 즉, 해당 서비스는 현재 **Cephadm**에서 관리하지 않습니다(예: **ceph orch ps** 명령에 포함된 재시작 및 업그레이드). **ceph orch host add HOST_NAME** 명령을 사용하여 호스트를 관리할 수 있지만 원격 호스트에 대한 **SSH** 액세스가 구성되어 있는지 확인합니다. 또는 호스트에 수동으로 연결하고 해당 호스트의 서비스가 제거되거나 **Cephadm**에서 관리하는 호스트로 마이그레이션되었는지 확인할 수 있습니다. **ceph config set mgr mgr/cephadm/warn_on_stray_hosts false**설정을 사용하여 이 경고를 비활성화할 수도 있습니다.

CEPHADM_STRAY_DAEMON

하나 이상의 **Ceph** 데몬이 실행 중이지만 **Cephadm** 모듈에서는 관리되지 않습니다. 이는 다른 도구를 사용하여 배포되었거나 수동으로 시작되었기 때문일 수 있습니다. 이러한 서비스는 현재 **Cephadm**에서 관리하지 않습니다(예: **ceph orch ps** 명령에 포함된 재시작 및 업그레이드 등).

데몬이 모니터 또는 **OSD daeon**인 상태 저장 장치인 경우 **Cephadm**에서 이러한 데몬을 채택해야 합니다. 상태 비저장 데몬의 경우 **ceph orch apply** 명령을 사용하여 새 데몬을 프로비저닝한 다음 관리되지 않는 데몬을 중지할 수 있습니다.

ceph config set mgr mgr/cephadm/warn_on_stray_daemons false 를 사용하여 이 상태 경고를 비활성화할 수 있습니다.

CEPHADM_HOST_CHECK_FAILED

기본 **Cephadm** 호스트 검사에 실패한 하나 이상의 호스트가:**name: value**를 확인합니다.

- 호스트에 연결할 수 있으며 **Cephadm**을 실행할 수 있습니다.

- 호스트는 **Podman**의 작업 컨테이너 런타임 및 작업 시간 동기화와 같은 기본 사전 요구 사항을 충족합니다. 이 테스트가 실패하면 **Cephadm**이 해당 호스트에서 서비스를 관리할 수 없습니다.

ceph cephadm check-host HOST_NAME 명령을 사용하여 이 검사를 수동으로 실행할 수 있습니다. **ceph orch** 호스트 **rm HOST_NAME**명령을 사용하여 관리에서 손상된 호스트 를 제거할 수 있습니다. **ceph config set mgr mgr/cephadm/warn_on_failed_host_check false** 를 사용하여 이 상태 경고를 비활성화할 수 있습니다.

11.5.3. Cephadm 구성 상태 점검

Cephadm은 스토리지 클러스터의 각 호스트를 주기적으로 검사하여 **OS**, 디스크 및 **NIC**의 상태를 확인합니다. 이러한 팩트는 스토리지 클러스터의 호스트 간에 일관성을 분석하여 이상한 구성을 식별합니다. 구성 검사는 선택적 기능입니다.

- 다음 명령을 사용하여 이 기능을 활성화할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph config set mgr mgr/cephadm/config_checks_enabled true
```

각 호스트 검사 후에 구성 검사가 트리거됩니다. 이는 **1**분 동안 지속됩니다.

- **ceph -W cephadm** 명령은 다음과 같이 현재 상태 및 구성 검사의 로그 항목을 표시합니다.

비활성화 상태

예제

```
ALL cephadm checks are disabled, use 'ceph config set mgr
mgr/cephadm/config_checks_enabled true' to enable
```

enabled 상태

예제

```
CEPHADM 8/8 checks enabled and executed (0 bypassed, 0 disabled). No issues detected
```

설정 검사 자체는 여러 **cephadm** 하위 명령을 통해 관리됩니다.

- 구성 검사 활성화 여부를 확인하려면 다음 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph cephadm config-check status
```

이 명령은 구성 검사기의 상태를 **Enabled** 또는 **Disabled** 로 반환합니다.

- 구성 검사 및 현재 상태를 모두 나열하려면 다음 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph cephadm config-check ls
NAME            HEALTHCHECK                      STATUS   DESCRIPTION
```

```
kernel_security  CEPHADM_CHECK_KERNEL_LSM        enabled  checks
SELINUX/Apparmor profiles are consistent across cluster hosts
os_subscription  CEPHADM_CHECK_SUBSCRIPTION      enabled  checks subscription
states are consistent for all cluster hosts
public_network   CEPHADM_CHECK_PUBLIC_MEMBERSHIP  enabled  check that all hosts
have a NIC on the Ceph public_netork
osd_mtu_size     CEPHADM_CHECK_MTU               enabled  check that OSD hosts share a
common MTU setting
osd_linkspeed    CEPHADM_CHECK_LINKSPEED         enabled  check that OSD hosts
share a common linkspeed
network_missing  CEPHADM_CHECK_NETWORK_MISSING   enabled  checks that the
cluster/public networks defined exist on the Ceph hosts
ceph_release     CEPHADM_CHECK_CEPH_RELEASE      enabled  check for Ceph version
consistency - ceph daemons should be on the same release (unless upgrade is active)
kernel_version   CEPHADM_CHECK_KERNEL_VERSION    enabled  checks that the
MAJ.MIN of the kernel on Ceph hosts is consistent
```

각 구성 검사는 다음과 같이 설명되어 있습니다.

CEPHADM_CHECK_KERNEL_LSM

*스토리지 클러스터 내의 각 호스트는 동일한 **Linux Security Module (LSM)** 상태 내에서 작동할 것으로 예상됩니다. 예를 들어, 대다수의 호스트가 강제 모드에서 **SELINUX**를 사용하여 실행 중인 경우 이 모드에서 실행되지 않는 모든 호스트에는 예기치 않게 플래그가 지정되고 경고 상태가 있는 상태 확인이 발생합니다.*

CEPHADM_CHECK_SUBSCRIPTION

*이 점검은 공급 업체 서브스크립션의 상태와 관련이 있습니다. 이 검사는 **Red Hat Enterprise Linux**를 사용하는 호스트에 대해서만 수행되지만 패치 및 업데이트를 사용할 수 있도록 모든 호스트가 활성 서브스크립션을 통해 적용되는지 확인하는 데 도움이 됩니다.*

CEPHADM_CHECK_PUBLIC_MEMBERSHIP

*클러스터의 모든 멤버는 공용 네트워크 서브넷 중 하나 이상에 **NIC**가 구성되어 있어야 합니다. 공용 네트워크에 없는 호스트는 라우팅에 따라 성능에 영향을 미칠 수 있습니다.*

CEPHADM_CHECK_MTU

***OSD**에서 **NIC**의 최대 전송 단위(**MTU**)는 일관된 성능의 핵심 요소가 될 수 있습니다. 이 검사에서는 **OSD** 서비스를 실행하는 호스트를 검사하여 **MTU**가 클러스터 내에서 일관되게 구성되었는지 확인합니다. 이는 대부분의 호스트가 사용 중인 **MTU** 설정을 설정하여 확인되며, 이로 인해 **Ceph** 상태 확인을 초래할 수 있습니다.*

CEPHADM_CHECK_LINKSPEED

MTU 검사와 유사하게 linkspeed consistency도 클러스터 성능에 영향을 미칩니다. 이 검사는 대부분의 OSD 호스트에서 공유하는 linkspeed를 결정하여 낮은 링크 속도로 설정된 모든 호스트에 대해 상태 확인을 수행합니다.

CEPHADM_CHECK_NETWORK_MISSING

public_network 및 cluster_network 설정은 IPv4 및 IPv6에 대한 서브넷 정의를 지원합니다. 스토리지 클러스터의 호스트에서 이러한 설정을 찾을 수 없는 경우 상태 점검이 발생합니다.

CEPHADM_CHECK_CEPH_RELEASE

일반 작업에서는 Ceph 클러스터가 동일한 Ceph 릴리스에서 데몬을 실행해야 합니다(예: 모든 Red Hat Ceph Storage 클러스터 5 릴리스). 이 검사는 각 데몬의 활성 릴리스를 살펴보고 모든 이상 상태를 상태 점검으로 보고합니다. 클러스터 내에서 업그레이드 프로세스가 활성화된 경우 이 검사를 바이패스합니다.

CEPHADM_CHECK_KERNEL_VERSION

OS 커널 버전은 호스트의 일관성을 확인합니다. 다시 한번, 대부분의 호스트가 이상하게 식별되는 기준으로 사용됩니다.

12장. CEPHADM-ANSIBLE 모듈을 사용하여 RED HAT CEPH STORAGE 클러스터 관리

스토리지 관리자는 Ansible 플레이북에서 cephadm-ansible 모듈을 사용하여 Red Hat Ceph Storage 클러스터를 관리할 수 있습니다. cephadm-ansible 패키지는 cephadm 호출을 래핑하여 클러스터를 관리하기 위해 고유한 Ansible 플레이북을 작성할 수 있는 여러 모듈을 제공합니다.

참고

현재 cephadm-ansible 모듈은 가장 중요한 작업만 지원합니다. cephadm-ansible 모듈에서 다루지 않는 작업은 플레이북에서 command 또는 shell Ansible 모듈을 사용하여 완료해야 합니다.

12.1. CEPHADM-ANSIBLE 모듈

cephadm -ansible 모듈은 cephadm 및 ceph orch 명령에 래퍼를 제공하여 Ansible 플레이북 작성을 간소화하는 모듈 컬렉션입니다. 모듈을 사용하여 하나 이상의 모듈을 사용하여 클러스터를 관리하기 위해 고유한 Ansible 플레이북을 작성할 수 있습니다.

cephadm-ansible 패키지에는 다음 모듈이 포함되어 있습니다.

- cephadm_bootstrap
- ceph_orch_host
- ceph_config
- ceph_orch_apply
- ceph_orch_daemon
- cephadm_registry_login

12.2. CEPHADM-ANSIBLE 모듈 옵션

*다음 표에는 **cephadm-ansible** 모듈에 사용할 수 있는 옵션이 나열되어 있습니다. 필요에 따라 나열된 옵션을 **Ansible** 플레이북에서 모듈을 사용할 때 설정해야 합니다. 기본값인 **true** 로 나열된 옵션은 모듈을 사용할 때 옵션이 자동으로 설정되므로 플레이북에 지정할 필요가 없음을 나타냅니다. 예를 들어 **cephadm_bootstrap** 모듈의 경우 **dashboard: false** 를 설정하지 않으면 **Ceph** 대시보드가 설치됩니다.*

*표 **12.1. cephadm_bootstrap** 모듈에 사용 가능한 옵션.*

cephadm_bootstrap	설명	필수 항목	기본값
mon_ip	Ceph Monitor IP 주소.	true	
image	Ceph 컨테이너 이미지.	false	
docker	**podman** 대신 **docker** 를 사용하십시오.	false	
fsid	Ceph FSID를 정의합니다.	false	
pull	Ceph 컨테이너 이미지를 가져옵니다.	false	true
대시보드	Ceph 대시보드를 배포합니다.	false	true
dashboard_user	특정 Ceph 대시보드 사용자를 지정합니다.	false	
dashboard_password	Ceph 대시보드 암호.	false	
모니터링	모니터링 스택을 배포합니다.	false	true
firewalld	firewalld를 사용하여 방화벽 규칙을 관리합니다.	false	true
allow_overwrite	기존 --output-config, --output-keyring 또는 --output-pub-ssh-key 파일을 덮어쓸 수 있습니다.	false	false
registry_url	사용자 정의 레지스트리의 URL입니다.	false	
registry_username	사용자 정의 레지스트리의 사용자 이름.	false	

cephadm_bootstrap	설명	필수 항목	기본값
registry_password	사용자 정의 레지스트리의 암호입니다.	false	
registry_json	사용자 정의 레지스트리 로그인 정보가 있는 JSON 파일입니다.	false	
ssh_user	호스트에서 **cephadm** ssh에 사용할 SSH 사용자입니다.	false	
ssh_config	**cephadm** SSH 클라이언트의 SSH 구성 파일 경로입니다.	false	
allow_fqdn_hostname	FQDN(정규화된 도메인 이름)인 호스트 이름을 허용합니다.	false	false
cluster_network	클러스터 복제, 복구 및 하트비트에 사용할 서브넷입니다.	false	

표 12.2. ceph_orch_host 모듈에 사용 가능한 옵션입니다.

ceph_orch_host	설명	필수 항목	기본값
fsid	와 상호 작용할 Ceph 클러스터의 FSID입니다.	false	
image	사용할 Ceph 컨테이너 이미지입니다.	false	
name	추가, 제거 또는 업데이트할 호스트의 이름입니다.	true	
주소	호스트의 IP 주소입니다.	**상태가 있는** 경우 True입니다.	
set_admin_label	지정된 호스트에서 **_admin** 레이블을 설정합니다.	false	false
labels	호스트에 적용할 라벨 목록입니다.	false	[]

ceph_orch_host	설명	필수 항목	기본값
state	**present** 로 설정하면 **name** 에 지정된 이름이 있는지 확인합니다. **absent** 로 설정하면 **이름에** 지정된 호스트가 제거됩니다. **drain** 으로 설정하면 **이름에** 지정된 호스트에서 모든 데몬을 제거하도록 예약합니다.	false	present

표 12.3. ceph_config 모듈에 사용 가능한 옵션

ceph_config	설명	필수 항목	기본값
fsid	와 상호 작용할 Ceph 클러스터의 FSID입니다.	false	
image	사용할 Ceph 컨테이너 이미지입니다.	false	
작업	**옵션** 에 지정된 매개 변수를 **설정하거나** 가져올지 여부입니다.	false	set
who	구성을 로 설정하는 데몬은 무엇입니까.	true	
옵션	**설정하거나 가져올 매개변수의 이름입니다**.	true	
value	설정할 매개변수의 값입니다.	action이 **설정된**경우 true	

표 12.4. ceph_orch_apply 모듈에 사용 가능한 옵션입니다.

ceph_orch_apply	설명	필수 항목
fsid	와 상호 작용할 Ceph 클러스터의 FSID입니다.	false
image	사용할 Ceph 컨테이너 이미지입니다.	false
spec	적용할 서비스 사양입니다.	true

표 12.5. ceph_orch_daemon 모듈에 사용할 수 있는 옵션입니다.

ceph_orch_daemon	설명	필수 항목
fsid	와 상호 작용할 Ceph 클러스터의 FSID입니다.	false
image	사용할 Ceph 컨테이너 이미지입니다.	false
state	**이름에** 지정된 서비스의 원하는 상태입니다.	true 가 **시작되면** 서비스가 시작되었는지 확인합니다. **중지** 되면 서비스가 중지되었는지 확인합니다. 를 **다시** 시작하면 서비스를 다시 시작합니다.
daemon_id	서비스의 ID입니다.	true
daemon_type	서비스 유형입니다.	true

표 12.6. cephadm_registry_login 모듈에 사용 가능한 옵션

cephadm_registry_login	설명	필수 항목	기본값
state	레지스트리에서 로그인 또는 로그아웃.	false	login
docker	**podman** 대신 **docker** 를 사용하십시오.	false	
registry_url	사용자 정의 레지스트리의 URL입니다.	false	
registry_username	사용자 정의 레지스트리의 사용자 이름.	**상태가 로그인** 인 경우 **True** 입니다.	
registry_password	사용자 정의 레지스트리의 암호입니다.	**상태가 로그인** 인 경우 **True** 입니다.	
registry_json	JSON 파일의 경로입니다. 이 작업을 실행하기 전에 이 파일이 원격 호스트에 있어야 합니다. 이 옵션은 현재 지원되지 않습니다.		

12.3. CEPHADM_BOOTSTRAP 및 CEPHADM_REGISTRY_LOGIN 모듈을 사용하여 스토리지 클러스터 부트 스트랩

스토리지 관리자는 **Ansible** 플레이북에서 **cephadm_bootstrap** 및 **cephadm_registry_login** 모듈을 사용하여 **Ansible**을 사용하여 스토리지 클러스터를 부트스트랩할 수 있습니다.

사전 요구 사항

- 첫 번째 **Ceph Monitor** 컨테이너의 **IP** 주소이며, 이는 스토리지 클러스터의 첫 번째 노드의 **IP** 주소이기도 합니다.
- **registry.redhat.io** 에 대한 로그인 액세스
- **/var/lib/containers/** 용으로 최소 **10GB**의 여유 공간이 필요
- **Red Hat Enterprise Linux 8.4 EUS** 또는 **Red Hat Enterprise Linux 8.5.**
- **Ansible** 관리 노드에 **cephadm-ansible** 패키지 설치
- 암호 없는 **SSH**는 스토리지 클러스터의 모든 호스트에 설정됩니다.
- 호스트는 **CDN**에 등록됩니다.

절차

1. **Ansible** 관리 노드에 로그인합니다.
2. **Ansible** 관리 노드에서 **/usr/share/cephadm-ansible** 디렉토리로 이동합니다.

예제

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

3.

호스트 파일을 생성하고 호스트, 레이블을 추가하고, 스토리지 클러스터에서 첫 번째 호스트의 IP 주소를 모니터링합니다.

구문

```
sudo vi INVENTORY_FILE

HOST1 labels="['LABEL1', 'LABEL2']"
HOST2 labels="['LABEL1', 'LABEL2']"
HOST3 labels="['LABEL1']"

[admin]
ADMIN_HOST monitor_address=MONITOR_IP_ADDRESS labels="['ADMIN_LABEL',
'LABEL1', 'LABEL2']"
```

예제

```
[ansible@admin cephadm-ansible]$ sudo vi hosts

host02 labels="['mon', 'mgr']"
host03 labels="['mon', 'mgr']"
host04 labels="['osd']"
host05 labels="['osd']"
host06 labels="['osd']"

[admin]
host01 monitor_address=10.10.128.68 labels="['_admin', 'mon', 'mgr']"
```

4.

preflight Playbook을 실행합니다.

구문

-

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars "ceph_origin=rhcs"
```

예제

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --extra-vars "ceph_origin=rhcs"
```

5. *클러스터를 부트스트랩할 플레이북을 생성합니다.*

구문

```
sudo vi PLAYBOOK_FILENAME.yml

---
- name: NAME_OF_PLAY
  hosts: BOOTSTRAP_HOST
  become: USE_ELEVATED_PRIVILEGES
  gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
  tasks:
    -name: NAME_OF_TASK
     cephadm_registry_login:
       state: STATE
       registry_url: REGISTRY_URL
       registry_username: REGISTRY_USER_NAME
       registry_password: REGISTRY_PASSWORD

    - name: NAME_OF_TASK
      cephadm_bootstrap:
        mon_ip: "{{ monitor_address }}"
        dashboard_user: DASHBOARD_USER
        dashboard_password: DASHBOARD_PASSWORD
        allow_fqdn_hostname: ALLOW_FQDN_HOSTNAME
        cluster_network: NETWORK_CIDR
```

예제

```
[ansible@admin cephadm-ansible]$ sudo vi bootstrap.yml

---
- name: bootstrap the cluster
  hosts: host01
  become: true
  gather_facts: false
  tasks:
    - name: login to registry
      cephadm_registry_login:
        state: login
        registry_url: registry.redhat.io
        registry_username: user1
        registry_password: mypassword1

    - name: bootstrap initial cluster
      cephadm_bootstrap:
        mon_ip: "{{ monitor_address }}"
        dashboard_user: mydashboarduser
        dashboard_password: mydashboardpassword
        allow_fqdn_hostname: true
        cluster_network: 10.10.128.0/28
```

6. *플레이북을 실행합니다.*

구문

```
ansible-playbook -i INVENTORY_FILE PLAYBOOK_FILENAME.yml -vvv
```

예제

```
ansible@admin cephadm-ansible]$ ansible-playbook -i hosts bootstrap.yml -vvv
```

검증

- 플레이북을 실행한 후 **Ansible** 출력을 검토합니다.

12.4. CEPH_ORCH_HOST 모듈을 사용하여 호스트 추가 또는 제거

스토리지 관리자는 **Ansible** 플레이북에서 **ceph_orch_host** 모듈을 사용하여 스토리지 클러스터에서 호스트를 추가하고 제거할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드를 **CDN**에 등록하고 서브스크립션을 연결합니다.
- 스토리지 클러스터의 모든 노드에 대한 **sudo** 및 암호 없는 **SSH** 액세스 권한이 있는 **Ansible** 사용자.
- **Ansible** 관리 노드에 **cephadm-ansible** 패키지 설치
- 새 호스트에는 스토리지 클러스터의 공용 **SSH** 키가 있습니다. 스토리지 클러스터의 공용 **SSH** 키를 새 호스트에 복사하는 방법에 대한 자세한 내용은 **Red Hat Ceph Storage** 설치 가이드의 호스트 추가를 참조하십시오.

절차

1. 다음 절차에 따라 새 호스트를 클러스터에 추가합니다.
   a. **Ansible** 관리 노드에 로그인합니다.
   b. **Ansible** 관리 노드에서 **/usr/share/cephadm-ansible** 디렉토리로 이동합니다.

예제

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

c.

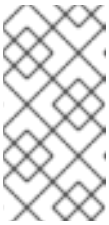
*새 호스트 및 레이블을 **Ansible** 인벤토리 파일에 추가합니다.*

구문

```
sudo vi INVENTORY_FILE

NEW_HOST1 labels="['LABEL1', 'LABEL2']"
NEW_HOST2 labels="['LABEL1', 'LABEL2']"
NEW_HOST3 labels="['LABEL1']"

[admin]
ADMIN_HOST monitor_address=MONITOR_IP_ADDRESS labels="['ADMIN_LABEL',
'LABEL1', 'LABEL2']"
```

예제

```
[ansible@admin cephadm-ansible]$ sudo vi hosts

host02 labels="['mon', 'mgr']"
host03 labels="['mon', 'mgr']"
host04 labels="['osd']"
host05 labels="['osd']"
host06 labels="['osd']"

[admin]
host01 monitor_address= 10.10.128.68 labels="['_admin', 'mon', 'mgr']"
```



참고

*이전에 새 호스트를 **Ansible** 인벤토리 파일에 추가하고 호스트에서 **preflight** 플레이북을 실행한 경우 **3** 단계로 건너뜁니다.*

d.

***--limit** 옵션을 사용하여 **preflight** 플레이북을 실행합니다.*

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars
"ceph_origin=rhcs" --limit NEWHOST
```

예제

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --
extra-vars "ceph_origin=rhcs" --limit host02
```

***preflight Playbook**은 새 호스트에 **podman,lvm2,chronyd, cephadm** 을 설치합니다. 설치가 완료되면 **cephadm**은 **/usr/sbin/** 디렉터리에 있습니다.*

e.

새 호스트를 클러스터에 추가하는 플레이북을 생성합니다.

구문

```
sudo vi PLAYBOOK_FILENAME.yml

---
- name: PLAY_NAME
  hosts: HOSTS_OR_HOST_GROUPS
  become: USE_ELEVATED_PRIVILEGES
  gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
  tasks:
    - name: NAME_OF_TASK
      ceph_orch_host:
        name: "{{ ansible_facts['hostname'] }}"
        address: "{{ ansible_facts['default_ipv4']['address'] }}"
        labels: "{{ labels }}"
      delegate_to: HOST_TO_DELEGATE_TASK_TO
```

```
- name: NAME_OF_TASK
  when: inventory_hostname in groups['admin']
  ansible.builtin.shell:
    cmd: CEPH_COMMAND_TO_RUN
  register: REGISTER_NAME

- name: NAME_OF_TASK
  when: inventory_hostname in groups['admin']
  debug:
    msg: "{{ REGISTER_NAME.stdout }}"
```

참고

*기본적으로 **Ansible**은 플레이북의 **hosts** 줄과 일치하는 호스트에서 모든 작업을 실행합니다. **ceph orch** 명령은 관리자 인증 키 및 **Ceph** 구성 파일이 포함된 호스트에서 실행해야 합니다. **delegate_to** 키워드를 사용하여 클러스터에서 관리 호스트를 지정합니다.*

예제

```
[ansible@admin cephadm-ansible]$ sudo vi add-hosts.yml

---
- name: add additional hosts to the cluster
  hosts: all
  become: true
  gather_facts: true
  tasks:
    - name: add hosts to the cluster
      ceph_orch_host:
        name: "{{ ansible_facts['hostname'] }}"
        address: "{{ ansible_facts['default_ipv4']['address'] }}"
        labels: "{{ labels }}"
      delegate_to: host01

    - name: list hosts in the cluster
      when: inventory_hostname in groups['admin']
      ansible.builtin.shell:
        cmd: ceph orch host ls
      register: host_list

    - name: print current list of hosts
      when: inventory_hostname in groups['admin']
      debug:
        msg: "{{ host_list.stdout }}"
```

이 예제에서 플레이북은 새 호스트를 클러스터에 추가하고 현재 호스트 목록을 표시합니다.

f. *플레이북을 실행하여 클러스터에 추가 호스트를 추가합니다.*

구문

```
ansible-playbook -i INVENTORY_FILE PLAYBOOK_FILENAME.yml
```

예제

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts add-hosts.yml
```

2. *다음 절차에 따라 클러스터에서 호스트를 제거합니다.*

a. ***Ansible** 관리 노드에 로그인합니다.*

b. ***Ansible** 관리 노드에서 **/usr/share/cephadm-ansible** 디렉토리로 이동합니다.*

예제

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

c.

클러스터에서 호스트 또는 호스트를 제거하는 플레이북을 생성합니다.

구문

```
sudo vi PLAYBOOK_FILENAME.yml

---
- name: NAME_OF_PLAY
  hosts: ADMIN_HOST
  become: USE_ELEVATED_PRIVILEGES
  gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
  tasks:
    - name: NAME_OF_TASK
      ceph_orch_host:
        name: HOST_TO_REMOVE
        state: STATE

    - name: NAME_OF_TASK
      ceph_orch_host:
        name: HOST_TO_REMOVE
        state: STATE
      retries: NUMBER_OF_RETRIES
      delay: DELAY
      until: CONTINUE_UNTIL
      register: REGISTER_NAME

    - name: NAME_OF_TASK
      ansible.builtin.shell:
        cmd: ceph orch host ls
      register: REGISTER_NAME

    - name: NAME_OF_TASK
        debug:
          msg: "{{ REGISTER_NAME.stdout }}"
```

예제

```
[ansible@admin cephadm-ansible]$ sudo vi remove-hosts.yml

---
```

```
- name: remove host
  hosts: host01
  become: true
  gather_facts: true
  tasks:
    - name: drain host07
      ceph_orch_host:
        name: host07
        state: drain

    - name: remove host from the cluster
      ceph_orch_host:
        name: host07
        state: absent
      retries: 20
      delay: 1
      until: result is succeeded
      register: result

     - name: list hosts in the cluster
       ansible.builtin.shell:
         cmd: ceph orch host ls
       register: host_list

     - name: print current list of hosts
       debug:
         msg: "{{ host_list.stdout }}"
```

*이 예제에서 플레이북 작업은 **host07** 의 모든 데몬을 드레이닝하고, 클러스터에서 호스트를 제거하고 현재 호스트 목록을 표시합니다.*

d.

***Playbook**을 실행하여 클러스터에서 호스트를 제거합니다.*

구문

```
ansible-playbook -i INVENTORY_FILE PLAYBOOK_FILENAME.yml
```

예제

-

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts remove-hosts.yml
```

검증

- *클러스터의 현재 호스트 목록을 표시하는 **Ansible** 작업 출력을 검토합니다.*

예제

```
TASK [print current hosts]
*****************************************************************************************************
Friday 24 June 2022  14:52:40 -0400 (0:00:03.365)       0:02:31.702 ***********
ok: [host01] =>
  msg: |-
    HOST    ADDR          LABELS         STATUS
    host01  10.10.128.68  _admin mon mgr
    host02  10.10.128.69  mon mgr
    host03  10.10.128.70  mon mgr
    host04  10.10.128.71  osd
    host05  10.10.128.72  osd
    host06  10.10.128.73  osd
```

12.5. CEPH_CONFIG 모듈을 사용하여 구성 옵션 설정

*스토리지 관리자는 **ceph_config** 모듈을 사용하여 **Red Hat Ceph Storage** 구성 옵션을 설정하거나 가져올 수 있습니다.*

사전 요구 사항

- *실행 중인 **Red Hat Ceph Storage** 클러스터.*
- *스토리지 클러스터의 모든 노드에 대한 **sudo** 및 암호 없는 **SSH** 액세스 권한이 있는 **Ansible** 사용자.*
- ***Ansible** 관리 노드에 **cephadm-ansible** 패키지 설치*

- **Ansible** 인벤토리 파일에는 클러스터와 관리 호스트가 포함되어 있습니다.

절차

1. **Ansible** 관리 노드에 로그인합니다.

2. **Ansible** 관리 노드에서 **/usr/share/cephadm-ansible** 디렉토리로 이동합니다.

   예제

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

3. 구성 변경 사항으로 플레이북을 생성합니다.

   구문

   ```
   sudo vi PLAYBOOK_FILENAME.yml

   ---
   - name: PLAY_NAME
     hosts: ADMIN_HOST
     become: USE_ELEVATED_PRIVILEGES
     gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
     tasks:
       - name: NAME_OF_TASK
         ceph_config:
           action: GET_OR_SET
           who: DAEMON_TO_SET_CONFIGURATION_TO
           option: CEPH_CONFIGURATION_OPTION
           value: VALUE_OF_PARAMETER_TO_SET

       - name: NAME_OF_TASK
         ceph_config:
           action: GET_OR_SET
           who: DAEMON_TO_SET_CONFIGURATION_TO
           option: CEPH_CONFIGURATION_OPTION
         register: REGISTER_NAME
   ```

```
- name: NAME_OF_TASK
  debug:
    msg: "MESSAGE_TO_DISPLAY {{ REGISTER_NAME.stdout }}"
```

예제

```
[ansible@admin cephadm-ansible]$ sudo vi change_configuration.yml

---
- name: set pool delete
  hosts: host01
  become: true
  gather_facts: false
  tasks:
    - name: set the allow pool delete option
      ceph_config:
        action: set
        who: mon
        option: mon_allow_pool_delete
        value: true

    - name: get the allow pool delete setting
      ceph_config:
        action: get
        who: mon
        option: mon_allow_pool_delete
      register: verify_mon_allow_pool_delete

    - name: print current mon_allow_pool_delete setting
      debug:
        msg: "the value of 'mon_allow_pool_delete' is {{ verify_mon_allow_pool_delete.stdout }}"
```

이 예제에서 플레이북은 먼저 ***mon_allow_pool_delete*** *옵션을* ***false*** *로 설정합니다. 그러면* ***Playbook****이 현재* ***mon_allow_pool_delete*** *설정을 가져와서* ***Ansible*** *출력에 값을 표시합니다.*

4. *플레이북을 실행합니다.*

구문

-

```
ansible-playbook -i INVENTORY_FILE _PLAYBOOK_FILENAME.yml
```

예제

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts change_configuration.yml
```

검증

- 플레이북 작업의 출력을 검토합니다.

예제

```
TASK [print current mon_allow_pool_delete setting]
*************************************************************
Wednesday 29 June 2022  13:51:41 -0400 (0:00:05.523)       0:00:17.953 ********
ok: [host01] =>
  msg: the value of 'mon_allow_pool_delete' is true
```

추가 리소스

- 구성 옵션에 대한 자세한 내용은 **Red Hat Ceph Storage Configuration Guide** 를 참조하십시오.

12.6. CEPH_ORCH_APPLY 모듈을 사용하여 서비스 사양 적용

스토리지 관리자는 **Ansible** 플레이북의 **ceph_orch_apply** 모듈을 사용하여 스토리지 클러스터에 서비스 사양을 적용할 수 있습니다. 서비스 사양은 **Ceph** 서비스를 배포하는 데 사용되는 서비스 속성 및 구성 설정을 지정하는 데이터 구조입니다. 서비스 사양을 사용하여 **mon,크래시,mds,mgr,osd,rdb** 또는 **rbd-mirror** 와 같은 **Ceph** 서비스 유형을 배포할 수 있습니다.

사전 요구 사항

- *실행 중인 **Red Hat Ceph Storage** 클러스터.*
- *스토리지 클러스터의 모든 노드에 대한 **sudo** 및 암호 없는 **SSH** 액세스 권한이 있는 **Ansible** 사용자.*
- ***Ansible** 관리 노드에 **cephadm-ansible** 패키지 설치*
- ***Ansible** 인벤토리 파일에는 클러스터와 관리 호스트가 포함되어 있습니다.*

절차

1. ***Ansible** 관리 노드에 로그인합니다.*
2. ***Ansible** 관리 노드에서 **/usr/share/cephadm-ansible** 디렉토리로 이동합니다.*

   예제

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

3. *서비스 사양으로 플레이북을 생성합니다.*

   구문

   ```
   sudo vi PLAYBOOK_FILENAME.yml

   ---
   - name: PLAY_NAME
     hosts: HOSTS_OR_HOST_GROUPS
     become: USE_ELEVATED_PRIVILEGES
     gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
   ```

```
tasks:
  - name: NAME_OF_TASK
    ceph_orch_apply:
      spec: |
        service_type: SERVICE_TYPE
        service_id: UNIQUE_NAME_OF_SERVICE
        placement:
          host_pattern: 'HOST_PATTERN_TO_SELECT_HOSTS'
          label: LABEL
        spec:
          SPECIFICATION_OPTIONS:
```

예제

```
[ansible@admin cephadm-ansible]$ sudo vi deploy_osd_service.yml

---
- name: deploy osd service
  hosts: host01
  become: true
  gather_facts: true
  tasks:
    - name: apply osd spec
      ceph_orch_apply:
        spec: |
          service_type: osd
          service_id: osd
          placement:
            host_pattern: '*'
            label: osd
          spec:
            data_devices:
              all: true
```

*이 예제에서 플레이북은 **osd** 레이블이 있는 모든 호스트에 **Ceph OSD** 서비스를 배포합니다.*

4. *플레이북을 실행합니다.*

구문

```
ansible-playbook -i INVENTORY_FILE _PLAYBOOK_FILENAME.yml
```

예제

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts deploy_osd_service.yml
```

검증

- *플레이북 작업의 출력을 검토합니다.*

추가 리소스

- *서비스 사양 옵션에 대한 자세한 내용은 **Red Hat Ceph Storage Operations Guide** 를 참조하십시오.*

12.7. CEPH_ORCH_DAEMON 모듈을 사용하여 CEPH 데몬 상태 관리

*스토리지 관리자는 **Ansible** 플레이북에서 **ceph_orch_daemon** 모듈을 사용하여 호스트에서 **Ceph** 데몬을 시작, 중지 및 재시작할 수 있습니다.*

사전 요구 사항

- *실행 중인 **Red Hat Ceph Storage** 클러스터.*
- *스토리지 클러스터의 모든 노드에 대한 **sudo** 및 암호 없는 **SSH** 액세스 권한이 있는 **Ansible** 사용자.*
- ***Ansible** 관리 노드에 **cephadm-ansible** 패키지 설치*

- ***Ansible** 인벤토리 파일에는 클러스터와 관리 호스트가 포함되어 있습니다.*

절차

1. ***Ansible** 관리 노드에 로그인합니다.*

2. ***Ansible** 관리 노드에서 **/usr/share/cephadm-ansible** 디렉토리로 이동합니다.*

   예제

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

3. *데몬 상태 변경 사항을 사용하여 플레이북을 생성합니다.*

   구문

   ```
   sudo vi PLAYBOOK_FILENAME.yml

   ---
   - name: PLAY_NAME
     hosts: ADMIN_HOST
     become: USE_ELEVATED_PRIVILEGES
     gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
     tasks:
       - name: NAME_OF_TASK
         ceph_orch_daemon:
           state: STATE_OF_SERVICE
           daemon_id: DAEMON_ID
           daemon_type: TYPE_OF_SERVICE
   ```

   예제

```
[ansible@admin cephadm-ansible]$ sudo vi restart_services.yml

---
- name: start and stop services
  hosts: host01
  become: true
  gather_facts: false
  tasks:
    - name: start osd.0
      ceph_orch_daemon:
        state: started
        daemon_id: 0
        daemon_type: osd

    - name: stop mon.host02
      ceph_orch_daemon:
        state: stopped
        daemon_id: host02
        daemon_type: mon
```

*이 예제에서 플레이북은 **ID**가 **0** 인 **OSD**를 시작하고, **host02 ID**가 있는 **Ceph Monitor**를 중지합니다.*

4. *플레이북을 실행합니다.*

구문

```
ansible-playbook -i INVENTORY_FILE _PLAYBOOK_FILENAME.yml
```

예제

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts restart_services.yml
```

검증

- *플레이북 작업의 출력을 검토합니다.*