Red Hat Ceph Storage 8

구성 가이드

Red Hat Ceph Storage 구성 설정

Last Updated: 2025-12-03

Red Hat Ceph Storage 8 구성 가이드

Red Hat Ceph Storage 구성 설정

Legal Notice

Abstract

이 문서에서는 부팅 시 및 런타임 시 Red Hat Ceph Storage를 구성하는 방법을 설명합니다. 구성 참조 정보도 제공합니다. 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 향후 여러 릴리스에 대해 단계적으로 구현될 예정입니다. 자세한 내용은 CTO Chris Wright의 메시지에서 참조하십시오.

Table of Contents

1장. CEPH 구성의 기본 사항

스토리지 관리자는 Ceph 구성을 보는 방법과 Red Hat Ceph Storage 클러스터에 대한 Ceph 구성 옵션을 설정하는 방법을 기본적으로 이해해야 합니다. 런타임 시 Ceph 구성 옵션을 보고 설정할 수 있습니다.

사전 요구 사항

- Red Hat Ceph Storage 소프트웨어 설치.

1.1. CEPH 구성

모든 Red Hat Ceph Storage 클러스터에는 다음과 같은 구성이 있습니다.

- 클러스터 ID
- 인증 설정
- Ceph 데몬
- 네트워크 구성
- 노드 이름 및 주소
- 인증 키 경로
- OSD 로그 파일의 경로
- 기타 런타임 옵션

cephadm 과 같은 배포 툴은 일반적으로 초기 Ceph 구성 파일을 생성합니다. 그러나 배포 도구를 사용하지 않고 Red Hat Ceph Storage 클러스터를 부트스트랩하려는 경우 직접 생성할 수 있습니다.

추가 리소스

- **cephadm** 및 Ceph 오케스트레이터에 대한 자세한 내용은 *Red Hat Ceph Storage Operations Guide* 를 참조하십시오.

1.2. CEPH 구성 데이터베이스

Ceph 모니터는 전체 스토리지 클러스터의 구성 옵션을 저장하여 구성 관리를 중앙 집중화하는 Ceph 옵션의 구성 데이터베이스를 관리합니다. 따라서 데이터베이스의 Ceph 구성을 중앙 집중화하면 스토리지 클러스터 관리가 간소화됩니다.

Ceph에서 옵션을 설정하는 데 사용하는 우선순위 순서는 다음과 같습니다.

- 컴파일된 기본값
- Ceph 클러스터 구성 데이터베이스
- 로컬 **ceph.conf** 파일
- **ceph 데몬** ***DAEMON-NAME*** **구성 세트 또는 ceph tell** ***DAEMON-NAME*** **injectargs** 명령을 사용하여 런타임 덮어쓰기

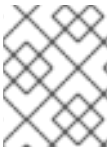
여전히 로컬 Ceph 구성 파일에 정의할 수 있는 몇 가지 Ceph 옵션이 있습니다. 기본적으로 **/etc/ceph/ceph.conf** 입니다. 그러나 **ceph.conf** 는 Red Hat Ceph Storage 8에서 더 이상 사용되지 않습니다.

cephadm 은 Ceph 모니터에 연결, 인증 및 구성 정보를 가져오기 위한 최소한의 옵션 세트만 포함된 기본 **ceph.conf** 파일을 사용합니다. 대부분의 경우 **cephadm** 은 **mon_host** 옵션만 사용합니다. **mon_host** 옵션에만 **ceph.conf** 를 사용하지 않으려면 DNS SRV 레코드를 사용하여 모니터로 작업을 수행합니다.

중요

assimilate-conf 관리 명령을 사용하여 유효한 옵션을 **ceph.conf** 파일에서 구성 데이터베이스로 이동하는 것이 좋습니다. assimilate **-conf에** 대한 자세한 내용은 Commands를 참조하십시오.

Ceph를 사용하면 런타임 시 데몬 구성을 변경할 수 있습니다. 이 기능은 디버그 설정을 활성화하거나 비활성화하여 로깅 출력을 늘리거나 줄이는 데 유용할 수 있으며 런타임 최적화에도 사용할 수 있습니다.



참고

구성 데이터베이스와 Ceph 구성 파일에 동일한 옵션이 있는 경우 구성 데이터베이스 옵션은 Ceph 구성 파일에 설정된 것보다 우선 순위가 낮습니다.

섹션 및 Cryostat

Ceph 옵션을 전역적으로, 데몬 유형별로 또는 Ceph 구성 파일의 특정 데몬에 의해 구성할 수 있는 것처럼 다음 섹션에 따라 구성 데이터베이스에서 Ceph 옵션을 구성할 수도 있습니다.

섹션	설명
global	모든 데몬 및 클라이언트에 영향을 미칩니다.
월요일	모든 Ceph 모니터에 영향을 미칩니다.
mgr	모든 Ceph Manager에 영향을 미칩니다.
OSD	모든 Ceph OSD에 영향을 미칩니다.
mds	모든 Ceph 메타데이터 서버에 영향을 미칩니다.
클라이언트	마운트된 파일 시스템, 블록 장치, RADOS 게이트웨이를 포함한 모든 Ceph 클라이언트에 영향을 미칩니다.

Ceph 구성 옵션에는 마스크가 연결되어 있을 수 있습니다. 이러한 마스크는 옵션이 적용되는 데몬 또는 클라이언트를 추가로 제한할 수 있습니다.

마스크에는 두 가지 유형이 있습니다.

type:location

유형은 CRUSH 속성입니다(예: **rack** 또는 **host**). **위치는** 속성 유형의 값입니다. 예를 들어 **host:foo** 는 옵션을 **foo** 호스트에서 실행되는 데몬 또는 클라이언트로만 제한합니다.

예

```
ceph config set osd/host:magna045 debug_osd 20
```

class:device-class

device-class 는 CRUSH 장치 클래스의 이름입니다(예: **hdd** 또는 **ssd**). 예를 들어 **class:ssd** 는 SSD(Solid State Drive)에서 지원하는 Ceph OSD로만 옵션을 제한합니다. 이 마스크는 클라이언트의 OSD 데몬에는 영향을 미치지 않습니다.

예

```
ceph config set osd/class:hdd osd_max_backfills 8
```

관리 명령

Ceph 구성 데이터베이스는 하위 명령 **ceph config *ACTION*** 을 사용하여 수행할 수 있습니다. 다음은 수행할 수 있는 작업입니다.

ls

사용 가능한 구성 옵션을 나열합니다.

dump

스토리지 클러스터에 대한 옵션의 전체 구성 데이터베이스를 덤프합니다.

가져 오기

특정 데몬 또는 클라이언트의 구성을 덤프합니다. 예를 들어 kafka 는 **mds.a** 와 같은 데몬일 수 있습니다.

설정 옵션 VALUE

Ceph 구성 데이터베이스에서 구성 옵션을 설정합니다. 여기서 Cryostat는 대상 데몬이고 *OPTION* 은 설정할 옵션입니다. *VALUE* 는 원하는 값입니다.

보기 kafka

실행 중인 데몬에 대해 보고된 실행 중인 구성이 표시됩니다. 이러한 옵션은 사용 중인 로컬 구성 파일 또는 옵션이 명령줄 또는 런타임에 재정의된 경우 Ceph Monitor에서 저장한 옵션과 다를 수 있습니다. 또한 옵션 값의 소스는 출력의 일부로 보고됩니다.

assimilate-conf -i INPUT_FILE -o OUTPUT_FILE

INPUT_FILE 에서 구성 파일을 결합하고 유효한 옵션을 Ceph 모니터의 구성 데이터베이스로 이동합니다. 인식되지 않거나 유효하지 않거나 Ceph Monitor에서 제어할 수 없는 옵션은 OUTPUT_FILE 에 저장된 축약된 구성 파일로 반환됩니다. 이 명령은 기존 구성 파일에서 중앙 집중식 구성 데이터베이스로 전환하는 데 유용할 수 있습니다. 구성과 모니터 또는 기타 데몬의 경우 동일한 옵션 집합에 대해 서로 다른 구성 값을 설정한 경우 최종 결과는 파일이 동화되는 순서에 따라 달라집니다.

도움말 옵션 -f json-pretty

JSON 형식의 출력을 사용하여 특정 OPTION 에 대한 도움말을 표시합니다.

추가 리소스

- *명령에 대한 자세한 내용은 런타임 시 특정 구성 설정을 참조하십시오.*

1.3. CEPH 메타 변수 사용

메타 변수는 Ceph 스토리지 클러스터 구성을 크게 단순화합니다. 구성 값에 메타 변수가 설정된 경우 Ceph는 메타 변수를 구체적인 값으로 확장합니다.

메타 변수는 Ceph 구성 파일의 **[global]**, **[osd]**, **[mon]** 또는 **[client]** 섹션에서 사용할 때 매우 강력합니다. 그러나 관리 소켓과 함께 사용할 수도 있습니다. Ceph 메타 변수는 Bash 쉘 확장과 유사합니다.

Ceph는 다음 메타 변수를 지원합니다.

$cluster

설명

Ceph 스토리지 클러스터 이름으로 확장됩니다. 동일한 하드웨어에서 여러 Ceph 스토리지 클러스터를 실행할 때 유용합니다.

예

/etc/ceph/$cluster.keyring

기본

Ceph

$type

설명

인스턴트 데몬 유형에 따라 **osd** 또는 **mon** 중 하나로 확장됩니다.

예

/var/lib/ceph/$type

$id

설명

데몬 식별자로 확장합니다. **osd.0** 의 경우 이 값은 **0** 입니다.

예

/var/lib/ceph/$type/$cluster-$id

$host

설명

인스턴트 데몬의 호스트 이름으로 확장합니다.

$name

설명

를 **$type.$id** 로 확장합니다.

예

/var/run/ceph/$cluster-$name.asok

1.4. 런타임 시 CEPH 구성 보기

Ceph 구성 파일은 부팅 시 및 런타임 시 볼 수 있습니다.

사전 요구 사항

- Ceph 노드에 대한 루트 수준 액세스.
- 관리자 인증 키에 액세스합니다.

프로세스

1. 런타임 구성을 보려면 데몬을 실행하는 Ceph 노드에 로그인하고 다음을 실행합니다.

   구문

   ```
   ceph daemon DAEMON_TYPE.ID config show
   ```

   osd.0 에 대한 구성을 보려면 **osd.0** 이 포함된 노드에 로그인하고 다음 명령을 실행합니다.

   예

   ```
   [root@osd ~]# ceph daemon osd.0 config show
   ```

2. 추가 옵션의 경우 데몬과 **도움말** 을 지정합니다.

   예

   ```
   [root@osd ~]# ceph daemon osd.0 help
   ```

1.5. 런타임 시 특정 구성 보기

Red Hat Ceph Storage의 구성 설정은 런타임 시 Ceph Monitor 노드에서 볼 수 있습니다.

사전 요구 사항

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

프로세스

1. Ceph 노드에 로그인하고 다음을 실행합니다.

   구문

   ```
   ceph daemon DAEMON_TYPE.ID config get PARAMETER
   ```

   예

   ```
   [root@mon ~]# ceph daemon osd.0 config get public_addr
   ```

1.6. 런타임 시 특정 구성 설정

런타임 시 특정 Ceph 구성을 설정하려면 **ceph config set** 명령을 사용합니다.

사전 요구 사항

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

프로세스

1. *모든 Monitor 또는 OSD 데몬에서 구성을 설정합니다.*

   구문

   ```
   ceph config set DAEMON CONFIG-OPTION VALUE
   ```

   예

   ```
   [root@mon ~]# ceph config set osd debug_osd 10
   ```

2. *옵션과 값이 설정되어 있는지 확인합니다.*

   예

   ```
   [root@mon ~]# ceph config dump
   osd      advanced debug_osd  10/10
   ```

   - *모든 데몬에서 구성 옵션을 제거하려면 다음을 수행합니다.*

     구문

     ```
     ceph config rm DAEMON CONFIG-OPTION VALUE
     ```

     예

     ```
     [root@mon ~]# ceph config rm osd debug_osd
     ```

   - *특정 데몬의 구성을 설정하려면 다음을 수행합니다.*

     구문

     ```
     ceph config set DAEMON.DAEMON-NUMBER CONFIG-OPTION VALUE
     ```

     예

     ```
     [root@mon ~]# ceph config set osd.0 debug_osd 10
     ```

   - *구성이 지정된 데몬에 대해 설정되어 있는지 확인하려면 다음을 수행합니다.*

     예

     ```
     [root@mon ~]# ceph config dump
     osd.0      advanced debug_osd     10/10
     ```

   - *특정 데몬의 구성을 제거하려면 다음을 수행합니다.*

구문

```
ceph config rm DAEMON.DAEMON-NUMBER CONFIG-OPTION
```

예

```
[root@mon ~]# ceph config rm osd.0 debug_osd
```

참고

*구성 데이터베이스에서 옵션 읽기를 지원하지 않는 클라이언트를 사용하거나 **ceph.conf** 를 사용하여 다른 이유로 클러스터 구성을 변경해야 하는 경우 다음 명령을 실행합니다.*

```
ceph config set mgr mgr/cephadm/manage_etc_ceph_ceph_conf false
```

*스토리지 클러스터에 **ceph.conf** 파일을 유지 관리하고 배포해야 합니다.*

1.7. OSD 메모리 대상

*BlueStore는 **osd_memory_target** 구성 옵션을 사용하여 OSD 힙 메모리 사용량을 지정된 대상 크기 아래에 유지합니다.*

***osd_memory_target** 옵션은 시스템에서 사용 가능한 RAM에 따라 OSD 메모리를 설정합니다. TCMalloc이 메모리al Cryostat로 구성되어 있고 BlueStore의 **bluestore_cache_autotune** 옵션이 **true** 로 설정된 경우 이 옵션을 사용합니다.*

블록 장치가 느릴 때 Ceph OSD 메모리 캐싱이 더 중요합니다. 예를 들어 캐시 적중의 이점은 솔리드 스테이트 드라이브를 사용하는 것보다 훨씬 길기 때문입니다. 그러나 OSD를 하이퍼 컨버지드 인프라(하이퍼 컨버지드 인프라) 또는 기타 애플리케이션과 같이 다른 서비스와 배치하기 위한 결정에 직면해야 합니다.

1.7.1. OSD 메모리 대상 설정

***osd_memory_target** 옵션을 사용하여 스토리지 클러스터의 모든 OSD 또는 특정 OSD의 최대 메모리 임계값을 설정합니다. **osd_memory_target** 옵션이 16GB로 설정된 OSD는 최대 16GB의 메모리를 사용할 수 있습니다.*

참고

개별 OSD의 구성 옵션은 모든 OSD의 설정보다 우선합니다.

사전 요구 사항

- *실행 중인 Red Hat Ceph Storage 클러스터.*
- *스토리지 클러스터의 모든 호스트에 대한 루트 수준 액세스.*

프로세스

- *스토리지 클러스터의 모든 OSD에 대해 **osd_memory_target** 을 설정하려면 다음을 수행합니다.*

  구문

```
ceph config set osd osd_memory_target VALUE
```

VALUE 는 스토리지 클러스터의 각 OSD에 할당할 GBytes의 메모리 수입니다.

- 스토리지 클러스터의 특정 OSD에 대해 **osd_memory_target** 을 설정하려면 다음을 수행합니다.

  구문

  ```
  ceph config set osd.id osd_memory_target VALUE
  ```

  .ID는 OSD의 ID이며 VALUE 는 지정된 OSD에 할당할 메모리 GB 수입니다. 예를 들어 최대 16GB의 메모리를 사용하도록 OSD를 ID 8로 구성하려면 다음을 수행합니다.

  예

  ```
  [ceph: root@host01 /]# ceph config set osd.8 osd_memory_target 16G
  ```

- 하나의 최대 메모리를 사용하고 나머지 OSD를 다른 양을 사용하도록 개별 OSD를 설정하려면 먼저 개별 OSD를 지정합니다.

  예

  ```
  [ceph: root@host01 /]# ceph config set osd osd_memory_target 16G
  [ceph: root@host01 /]# ceph config set osd.8 osd_memory_target 8G
  ```

추가 리소스

- OSD 메모리 사용량을 자동으로 조정하도록 Red Hat Ceph Storage를 구성하려면 운영 가이드의 OSD 메모리 자동 튜닝 을 참조하십시오.

1.8. OSD 메모리 자동 튜닝

OSD 데몬은 **osd_memory_target** 구성 옵션을 기반으로 메모리 사용을 조정합니다. **osd_memory_target** 옵션은 시스템에서 사용 가능한 RAM에 따라 OSD 메모리를 설정합니다.

Red Hat Ceph Storage가 다른 서비스와 메모리를 공유하지 않는 전용 노드에 배포된 경우 **cephadm** 은 총 RAM 양 및 배포된 OSD 수에 따라 자동으로 OSD 소비를 조정합니다.

중요

기본적으로 **osd_memory_target_autotune** 매개변수는 Red Hat Ceph Storage 클러스터에서 **true** 로 설정됩니다.

구문

```
ceph config set osd osd_memory_target_autotune true
```

cephadm 은 **mgr/cephadm/autotune_memory_target_ratio** 로 시작합니다. 기본값은 시스템의 총 RAM으로, 비 자동 조정되지 않은 데몬에서 사용하는 메모리를 제거하고 **osd_memory_target_autotune** 이 false인 OSD로 나눕니다.

osd_memory_target 매개변수는 다음과 같이 계산됩니다.

구문

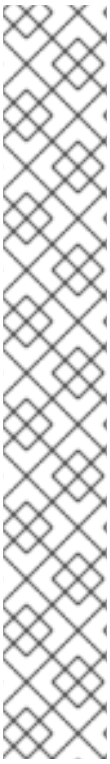
```
osd_memory_target = TOTAL_RAM_OF_THE_OSD * (1048576) * (autotune_memory_target_ratio) / NUMBER_OF_OSDS_IN_THE_OSD_NODE - (SPACE_ALLOCATED_FOR_OTHER_DAEMONS)
```

SPACE_ALLOCATED_FOR_OTHER_DAEMONS 는 선택적으로 다음 데몬 공간 할당을 포함할 수 있습니다.

- *Alertmanager: 1GB*
- *Grafana: 1GB*
- *Ceph Manager: 4GB*
- *Ceph 모니터: 2GB*
- *node-exporter: 1GB*
- *Prometheus: 1GB*

예를 들어 노드에 24개의 OSD가 있고 251GB RAM 공간이 있는 경우 osd_memory_target 은 7860684936 입니다.

최종 대상은 옵션을 사용하여 구성 데이터베이스에 반영됩니다. MEM LIMIT 열의 ceph orch ps 출력에서 각 데몬에서 사용하는 제한 및 현재 메모리를 볼 수 있습니다.



참고

***osd_memory_target_autotune true** 의 기본 설정은 컴퓨팅 및 **Ceph** 스토리지 서비스가 배치되는 하이퍼 컨버지드 인프라에 적합하지 않습니다. 하이퍼컨버지드 인프라에서 **autotune_memory_target_ratio** 를 **0.2** 로 설정하여 **Ceph**의 메모리 사용량을 줄일 수 있습니다.*

예

```
[ceph: root@host01 /]# ceph config set mgr
mgr/cephadm/autotune_memory_target_ratio 0.2
```

*스토리지 클러스터에서 **OSD**의 특정 메모리 대상을 수동으로 설정할 수 있습니다.*

예

```
[ceph: root@host01 /]# ceph config set osd.123 osd_memory_target 7860684936
```

*스토리지 클러스터에서 **OSD** 호스트의 특정 메모리 대상을 수동으로 설정할 수 있습니다.*

구문

```
ceph config set osd/host:HOSTNAME osd_memory_target TARGET_BYTES
```

예

```
[ceph: root@host01 /]# ceph config set osd/host:host01 osd_memory_target 1000000000
```

참고

osd_memory_target_autotune 을 활성화하면 기존 수동 **OSD** 메모리 대상 설정을 덮어씁니다. **osd_memory_target_autotune** 옵션 또는 기타 유사한 옵션이 활성화된 경우에도 데몬 메모리가 튜닝되지 않도록 하려면 호스트에서 **_no_autotune_memory** 레이블을 설정합니다.

구문

```
ceph orch host label add HOSTNAME _no_autotune_memory
```

autotune 옵션을 비활성화하고 특정 메모리 대상을 설정하여 **OSD**를 메모리 자동 튜닝에서 제외할 수 있습니다.

예

```
[ceph: root@host01 /]# ceph config set osd.123 osd_memory_target_autotune false
[ceph: root@host01 /]# ceph config set osd.123 osd_memory_target 16G
```

1.9. MDS 메모리 캐시 제한

MDS 서버는 **Ceph OSD**의 사용자인 **cephfs_metadata** 라는 별도의 스토리지 풀에 메타데이터를 유지합니다. **Ceph File Systems**의 경우 **MDS** 서버는 스토리지 클러스터 내의 단일 스토리지 장치뿐만 아니라 전체 **Red Hat Ceph Storage** 클러스터를 지원해야 하므로 특히 워크로드가 소규모**-medium-size** 파

일로 구성된 경우 특히 메타데이터와 데이터 비율이 훨씬 높은 경우 메모리 요구 사항이 중요할 수 있습니다.

예: **mds_cache_memory_limit** 을 **2000000000** 바이트로 설정합니다.

```
ceph_conf_overrides:
  mds:
    mds_cache_memory_limit=2000000000
```

참고

메타데이터 집약적인 워크로드가 있는 대규모 **Red Hat Ceph Storage** 클러스터의 경우 메모리 집약적인 서비스와 동일한 노드에 **MDS** 서버를 배치하지 마십시오. 이렇게 하면 **MDS**에 더 많은 메모리를 할당할 수 있습니다(예: **100GB**보다 큰 크기).

추가 리소스

- **Red Hat Ceph Storage** 파일 시스템 가이드의 메타데이터 서버 캐시 크기 제한을 참조하십시오.
- 특정 옵션 설명 및 사용은 구성 옵션 의 일반 **Ceph** 구성 옵션을 참조하십시오.

2장. CEPH 네트워크 구성

스토리지 관리자는 **Red Hat Ceph Storage** 클러스터가 작동할 네트워크 환경을 이해하고 그에 따라 **Red Hat Ceph Storage**를 구성해야 합니다. **Ceph** 네트워크 옵션을 이해하고 구성하면 전체 스토리지 클러스터의 최적 성능과 안정성이 보장됩니다.

사전 요구 사항

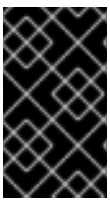
- 네트워크 연결.
- **Red Hat Ceph Storage** 소프트웨어 설치.

2.1. CEPH의 네트워크 구성

네트워크 구성은 고성능 **Red Hat Ceph Storage** 클러스터를 구축하는 데 중요합니다. **Ceph** 스토리지 클러스터는 **Ceph** 클라이언트를 대신하여 요청 라우팅 또는 디스패치를 수행하지 않습니다. 대신 **Ceph** 클라이언트는 **Ceph OSD** 데몬에 직접 요청합니다. **Ceph OSD**는 **Ceph** 클라이언트를 대신하여 데이터 복제를 수행합니다. 즉, 복제 및 기타 요인은 **Ceph** 스토리지 클러스터 네트워크에 추가 로드를 부과합니다.

Ceph에는 모든 데몬에 적용되는 하나의 네트워크 구성 요구 사항이 있습니다. **Ceph** 구성 파일은 각 데몬의 호스트 를 지정해야 합니다.

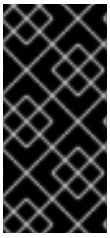
cephadm 과 같은 일부 배포 유틸리티는 구성 파일을 생성합니다. 배포 유틸리티에서 해당 값을 수행하는 경우 이 값을 설정하지 마십시오.



중요

host 옵션은 **FQDN**이 아닌 노드의 짧은 이름입니다. **IP** 주소가 아닙니다.

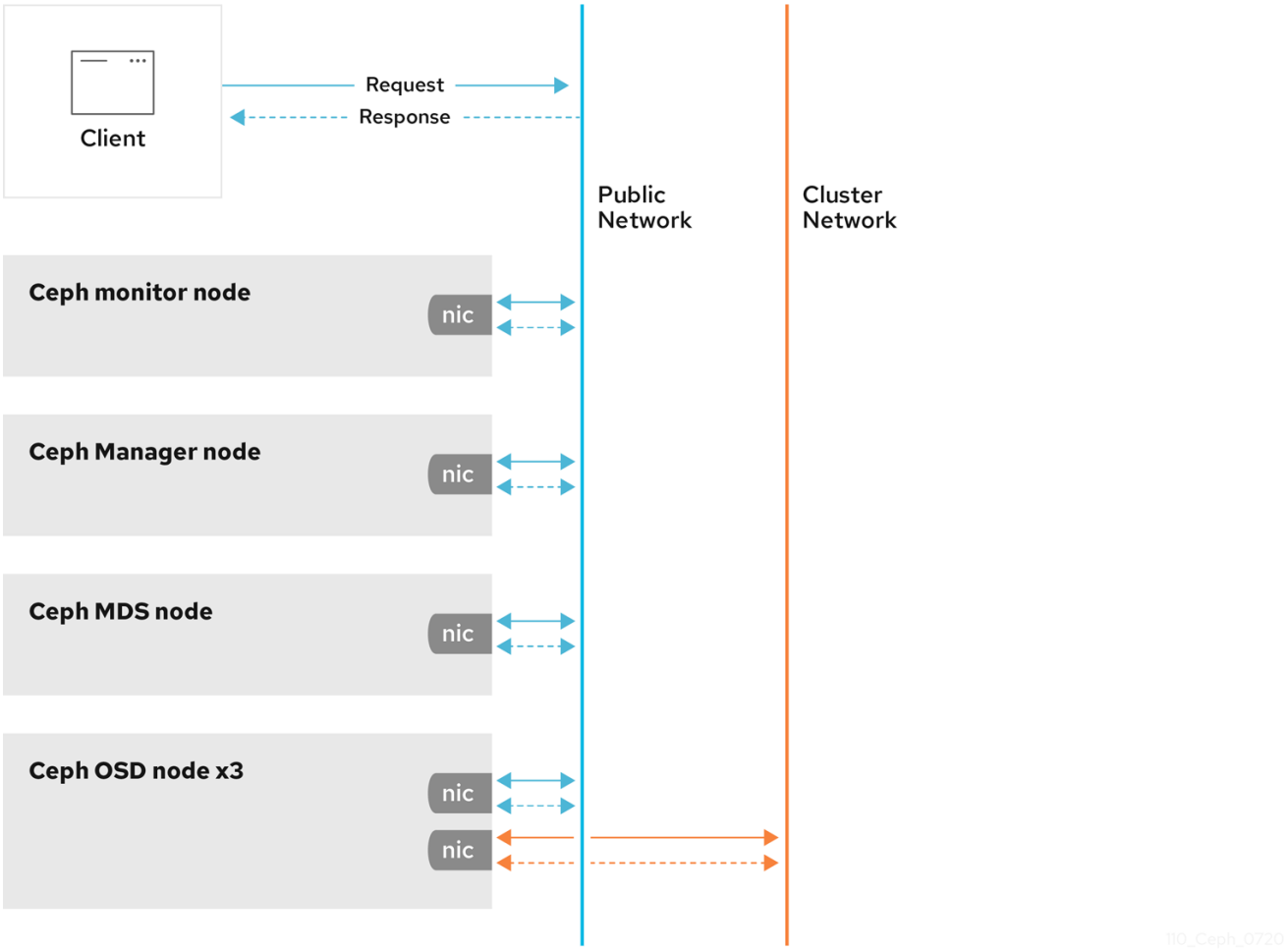
모든 **Ceph** 클러스터는 공용 네트워크를 사용해야 합니다. 그러나 내부 클러스터 네트워크를 지정하지 않으면 **Ceph**는 단일 공용 네트워크를 가정합니다. **Ceph**는 공용 네트워크에서만 작동할 수 있지만 대규모 스토리지 클러스터의 경우 클러스터 관련 트래픽만 전송하기 위해 두 번째 프라이빗 네트워크에서 성능이 크게 향상되었습니다.



중요

***Red Hat**은 두 개의 네트워크가 있는 **Ceph** 스토리지 클러스터를 실행하는 것이 좋습니다. 하나의 공용 네트워크와 사설 네트워크 **1**개*

*두 개의 네트워크를 지원하려면 각 **Ceph** 노드에 **NIC**(네트워크 인터페이스 카드)가 두 개 이상 있어야 합니다.*



두 개의 별도의 네트워크 운영을 고려해야 하는 몇 가지 이유가 있습니다.

- *성능: **Ceph OSD**는 **Ceph** 클라이언트의 데이터 복제를 처리합니다. **Ceph OSD**가 데이터를 두 번 이상 복제하면 **Ceph OSD** 간에 네트워크 로드가 **Ceph** 클라이언트와 **Ceph** 스토리지 클러스터 간에 쉽게 로드됩니다. 이로 인해 대기 시간이 발생하고 성능 문제가 발생할 수 있습니다. 복구 및 재조정으로 인해 공용 네트워크에서 대기 시간이 크게 발생할 수 있습니다.*

- *보안: 대부분의 사람들은 일반적으로 민주적이지만 일부 행위자는 서비스 거부 **(DoS)** 공격으*

로 알려진 것에 참여할 것입니다. **Ceph OSD** 간 트래픽이 중단되면 피어링이 실패할 수 있으며 배치 그룹은 더 이상 활성 + 정리 상태를 반영하지 않을 수 있으므로 사용자가 데이터를 읽고 쓰는 것을 방지할 수 있습니다. 이러한 유형의 공격을 극복할 수 있는 좋은 방법은 인터넷에 직접 연결되지 않는 완전히 분리된 클러스터 네트워크를 유지하는 것입니다.

네트워크 구성 설정은 필요하지 않습니다. **Ceph** 데몬을 실행하는 모든 호스트에 공용 네트워크가 구성되어 있다고 가정하면 **Ceph**가 공용 네트워크에서만 작동할 수 있습니다. 그러나 **Ceph**를 사용하면 공용 네트워크에 대한 여러 **IP** 네트워크 및 서브넷 마스크를 포함하여 훨씬 더 구체적인 기준을 설정할 수 있습니다. **OSD** 하트비트, 오브젝트 복제 및 복구 트래픽을 처리하도록 별도의 클러스터 네트워크를 설정할 수도 있습니다.

구성에 설정한 **IP** 주소와 공용 방향 **IP** 주소 네트워크 클라이언트가 서비스에 액세스하는 데 사용할 수 있는 **IP** 주소를 혼동하지 마십시오. 일반적인 내부 **IP** 네트워크는 종종 **192.168.0.0** 또는 **10.0.0.0** 입니다.

참고

Ceph는 서브넷에 **CIDR** 표기법을 사용합니다(예: **10.0.0.0/24**).

중요

공용 또는 사설 네트워크에 대해 두 개 이상의 **IP** 주소 및 서브넷 마스크를 지정하는 경우 네트워크 내의 서브넷은 서로 라우팅할 수 있어야 합니다. 또한 **IP** 테이블에 각 **IP** 주소와 서브넷을 포함하고 필요에 따라 포트를 열어야 합니다.

네트워크를 구성하면 클러스터를 다시 시작하거나 각 데몬을 다시 시작할 수 있습니다. **Ceph** 데몬은 동적으로 바인딩되므로 네트워크 구성을 변경하는 경우 전체 클러스터를 한 번에 다시 시작할 필요가 없습니다.

추가 리소스

- 특정 옵션 설명 및 사용은 **Red Hat Ceph Storage** 구성 가이드,부록 **B** 의 일반적인 옵션을 참조하십시오.

2.2. CEPH 네트워크 메시징기

Cryostat는 **Ceph** 네트워크 계층 구현입니다. **Red Hat**은 다음 두 가지 유형의 메시징을 지원합니다.

- ***simple***

- ***async***

***Red Hat Ceph Storage 7** 이상에서 **async** 는 기본 음성 유형입니다. **enger** 유형을 변경하려면 **Ceph** 구성 파일의 **[global]** 섹션에 **ms_type** 구성 설정을 지정합니다.*

참고

*비동기식의 경우 **Red Hat**은 **posix** 전송 유형을 지원하지만 현재 **rdma** 또는 **dpdk** 는 지원하지 않습니다. 기본적으로 **Red Hat Ceph Storage**의 **ms_type** 설정은 **async+posix** 를 반영합니다. 여기서 **async** 는 피터 유형이며 **posix** 은 전송 유형입니다.*

SimpleMessenger

***SimpleMessenger** 구현에서는 소켓당 두 개의 스레드가 있는 **TCP** 소켓을 사용합니다. **Ceph**는 각 논리 세션을 연결과 연결합니다. 파이프는 각 메시지의 입력 및 출력을 포함하여 연결을 처리합니다. **SimpleMessenger** 는 **posix** 전송 유형에 효과적이지만 **rdma** 또는 **dpdk** 와 같은 다른 운송 유형에는 효과가 없습니다.*

AsyncMessenger

*결과적으로 **AsyncMessenger** 는 **Red Hat Ceph Storage 7** 이상의 기본 전달자 유형입니다. **Red Hat Ceph Storage 7** 이상의 경우 **AsyncMessenger** 구현에서는 연결에 고정 크기 스레드 풀이 있는 **TCP** 소켓을 사용합니다. 이 소켓은 가장 많은 복제본 또는 삭제 코드 청크와 같아야 합니다. **CPU** 수가 부족하거나 서버당 **OSD** 수가 많은 경우 스레드 수를 더 낮은 값으로 설정할 수 있습니다.*

참고

***Red Hat**은 현재 **rdma** 또는 **dpdk** 와 같은 다른 운송 유형을 지원하지 않습니다.*

추가 리소스

- ***Red Hat Ceph Storage** 구성 가이드 의 **AsyncMessenger** 옵션, 특정 옵션 설명 및 사용은부록 **B** 를 참조하십시오.*

-

Ceph 전달자 버전 2 프로토콜과 함께 유선 암호화를 사용하는 방법에 대한 자세한 내용은 Red Hat Ceph Storage Architecture Guide 를 참조하십시오.

2.3. 공용 네트워크 구성

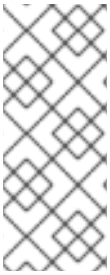
Ceph 네트워크를 구성하려면 cephadm 셸 내에서 config set 명령을 사용하십시오. 네트워크 구성에 설정한 IP 주소는 네트워크 클라이언트가 서비스에 액세스하는 데 사용할 수 있는 공용 방향 IP 주소와 다릅니다.

Ceph는 공용 네트워크에서만 완벽하게 작동합니다. 그러나 Ceph를 사용하면 공용 네트워크에 대한 여러 IP 네트워크를 포함하여 훨씬 더 구체적인 기준을 설정할 수 있습니다.

OSD 하트비트, 오브젝트 복제 및 복구 트래픽을 처리하기 위해 별도의 프라이빗 클러스터 네트워크를 설정할 수도 있습니다. 사설 네트워크에 대한 자세한 내용은 사설 네트워크 구성을 참조하십시오.

참고

Ceph는 서브넷에 CIDR 표기법을 사용합니다(예: 10.0.0.0/24). 일반적인 내부 IP 네트워크는 192.168.0.0/24 또는 10.0.0.0/24입니다.



참고

공용 또는 클러스터 네트워크에 대해 두 개 이상의 IP 주소를 지정하는 경우 네트워크 내의 서브넷이 서로 라우팅할 수 있어야 합니다. 또한 IP 테이블에 각 IP 주소를 포함하고 필요에 따라 포트를 열어야 합니다.

공용 네트워크 구성을 사용하면 공용 네트워크의 IP 주소 및 서브넷을 구체적으로 정의할 수 있습니다.

사전 요구 사항

- *Red Hat Ceph Storage 소프트웨어 설치.*

프로세스

1. ***cephadm** 쉘에 로그인합니다.*

예

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

2. *서브넷을 사용하여 공용 네트워크를 구성합니다.*

구문

```
ceph config set mon public_network IP_ADDRESS_WITH_SUBNET
```

예

```
[ceph: root@host01 /]# ceph config set mon public_network 192.168.0.0/24
```

3. *스토리지 클러스터에서 서비스 목록을 가져옵니다.*

예

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

4. *데몬을 다시 시작합니다. **Ceph** 데몬은 동적으로 바인딩되므로 특정 데몬의 네트워크 구성을 변경하는 경우 한 번에 전체 클러스터를 다시 시작할 필요가 없습니다.*

   예

   ```
   [ceph: root@host01 /]# ceph orch restart mon
   ```

5. *선택 사항: 클러스터를 다시 시작하려면 **root** 사용자로 **admin** 노드에서 **systemctl** 명령을 실행합니다.*

   구문

   ```
   systemctl restart ceph-FSID_OF_CLUSTER.target
   ```

   예제

   ```
   [root@host01 ~]# systemctl restart ceph-1ca9f6a8-d036-11ec-8263-fa163ee967ad.target
   ```

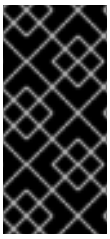
추가 리소스

- *특정 옵션 설명 및 사용은 **Red Hat Ceph Storage** 구성 가이드,부록 **B** 의 일반적인 옵션을 참조하십시오.*

2.4. 사설 네트워크 구성

네트워크 구성 설정은 필요하지 않습니다. Ceph는 프라이빗 네트워크라고도 하는 클러스터 네트워크를 구체적으로 구성하지 않는 한 모든 호스트가 작동하는 공용 네트워크를 가정합니다.

클러스터 네트워크를 생성하는 경우 OSD는 클러스터 네트워크를 통해 하트비트, 오브젝트 복제 및 복구 트래픽을 라우팅합니다. 이렇게 하면 단일 네트워크 사용과 비교하여 성능이 향상될 수 있습니다.



중요

보안을 강화하려면 공용 네트워크 또는 인터넷에서 클러스터 네트워크에 연결할 수 없어야 합니다.

클러스터 네트워크를 할당하려면 cephadm bootstrap 명령과 함께 --cluster-network 옵션을 사용합니다. 지정한 클러스터 네트워크는 CIDR 표기법에 서브넷을 정의해야 합니다(예: 10.90.90.0/24 또는 fe80::/64).

boostrap 후 cluster_network 를 구성할 수도 있습니다.

사전 요구 사항

- Ceph 소프트웨어 리포지토리에 액세스합니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

프로세스

- 스토리지 클러스터에서 모니터 노드로 사용할 초기 노드에서 cephadm bootstrap 명령을 실행합니다. 명령에 --cluster-network 옵션을 포함합니다.

  구문

```
cephadm bootstrap --mon-ip IP-ADDRESS --registry-url registry.redhat.io --registry-username USER_NAME --registry-password PASSWORD --cluster-network NETWORK-IP-ADDRESS
```

예

```
[root@host01 ~]# cephadm bootstrap --mon-ip 10.10.128.68 --registry-url registry.redhat.io --registry-username myuser1 --registry-password mypassword1 --cluster-network 10.10.0.0/24
```

- *부트스트랩 후 **cluster_network** 를 구성하려면 **config set** 명령을 실행하고 데몬을 재배포합니다.*

  1. ***cephadm** 쉘에 로그인합니다.*

     예

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

  2. *서브넷을 사용하여 클러스터 네트워크를 구성합니다.*

     구문

     ```
     ceph config set global cluster_network IP_ADDRESS_WITH_SUBNET
     ```

예

```
[ceph: root@host01 /]# ceph config set global cluster_network 10.10.0.0/24
```

3. *스토리지 클러스터에서 서비스 목록을 가져옵니다.*

예

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

4. *데몬을 다시 시작합니다. **Ceph** 데몬은 동적으로 바인딩되므로 특정 데몬의 네트워크 구성을 변경하는 경우 한 번에 전체 클러스터를 다시 시작할 필요가 없습니다.*

예

```
[ceph: root@host01 /]# ceph orch restart mon
```

5. *선택 사항: 클러스터를 다시 시작하려면 **root** 사용자로 **admin** 노드에서 **systemctl** 명령을 실행합니다.*

구문

```
systemctl restart ceph-FSID_OF_CLUSTER.target
```

예제

```
[root@host01 ~]# systemctl restart ceph-1ca9f6a8-d036-11ec-8263-fa163ee967ad.target
```

추가 리소스

- ***cephadm** 부트스트랩 호출에 대한 자세한 내용은 **Red Hat Ceph Storage** 설치 가이드의 새 스토리지 클러스터 부팅 섹션을 참조하십시오.*

2.5. 클러스터에 여러 공용 네트워크 구성

*사용자가 여러 네트워크 서브넷에 속하는 호스트에 **Ceph Monitor** 데몬을 배치하려면 클러스터에 여러 공용 네트워크를 구성해야 합니다. 사용 예로는 **OpenShift Data Foundation**용 **Metro DR**의 **ACS(Advanced Cluster Management)**에 사용되는 확장 클러스터 모드가 있습니다.*

부트 스트랩 중에 여러 공용 네트워크를 클러스터에 구성하고 부트스트랩이 완료되면 구성할 수 있습니다.

사전 요구 사항

- *호스트를 추가하기 전에 실행 중인 **Red Hat Ceph Storage** 클러스터가 있는지 확인하십시오.*

프로세스

1. *여러 공용 네트워크로 구성된 **Ceph** 클러스터를 부트스트랩합니다.*

   a. ***mon** 공용 네트워크 섹션이 포함된 **ceph.conf** 파일을 준비합니다.*

중요

제공된 공용 네트워크 중 하나 이상이 부트스트랩에 사용되는 현재 호스트에 구성되어야 합니다.

구문

```
[mon]
public_network = PUBLIC_NETWORK1, PUBLIC_NETWORK2
```

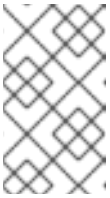
예

```
[mon]
public_network = 10.40.0.0/24, 10.41.0.0/24, 10.42.0.0/24
```

*이는 부트스트랩에 제공할 공용 네트워크 **3**개를 사용하는 예입니다.*

b.

***ceph.conf** 파일을 입력으로 제공하여 클러스터를 부트스트랩합니다.*

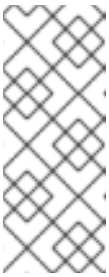


참고

부트스트랩 중에 제공할 다른 인수를 포함할 수 있습니다.

구문

```
cephadm --image IMAGE_URL bootstrap --mon-ip MONITOR_IP -c
PATH_TO_CEPH_CONF
```



참고

*또는 **IMAGE_ID** (예: **13ea90216d0be0be0d12d7869f72ad9de5cec9e54a27fd308e01e467c0a0a**) 는 **IMAGE_URL** 대신 사용할 수 있습니다.*

예

```
[root@host01 ~]# cephadm –image cp.icr.io/cp/ibm-ceph/ceph-5-rhel8:latest bootstrap –mon-ip 10.40.0.0/24 -c /etc/ceph/ceph.conf
```

2. *서브넷에 새 호스트를 추가합니다.*

참고

추가 중인 호스트는 활성 관리자가 실행 중인 호스트에서 연결할 수 있어야 합니다.

a. *새 호스트의 **root** 사용자의 **authorized_keys** 파일에 클러스터의 공개 **SSH** 키를 설치합니다.*

구문

```
ssh-copy-id -f -i /etc/ceph/ceph.pub root@NEW_HOST
```

예

```
[root@host01 ~]# ssh-copy-id -f -i /etc/ceph/ceph.pub root@host02
[root@host01 ~]# ssh-copy-id -f -i /etc/ceph/ceph.pub root@host03
```

b. ***cephadm** 쉘에 로그인합니다.*

예

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

c. *새 호스트를 **Ceph** 클러스터에 추가합니다.*

구문

```
ceph orch host add NEW_HOST IP [LABEL1 ...]
```

예

```
[root@host01 ~]# ceph orch host add host02 10.10.0.102 label1
[root@host01 ~]# ceph orch host add host03 10.10.0.103 label2
```

참고

- 호스트 **IP** 주소를 명시적으로 제공하는 것이 가장 좋습니다. **IP**를 제공하지 않으면 호스트 이름은 **DNS**를 통해 즉시 확인되고 해당 **IP**가 사용됩니다.

- 새 호스트에 즉시 레이블을 지정하도록 하나 이상의 레이블을 포함할 수도 있습니다. 예를 들어 기본적으로 **_admin** 레이블은 **cephadm** 에서 **ceph.conf** 파일의 사본을 유지하고 **/etc/ceph** 디렉터리에 **client.admin** 인증 키 파일을 유지합니다.

3. 공용 네트워크 매개 변수의 네트워크 구성을 실행 중인 클러스터에 추가합니다. 서브넷이 쉼표로 구분되고 서브넷이 서브넷/마스크 형식으로 나열되어 있는지 확인합니다.

구문

```
ceph config set mon public_network "SUBNET_1,SUBNET_2, ..."
```

예

```
[root@host01 ~]# ceph config set mon public_network "192.168.0.0/24, 10.42.0.0/24, ..."
```

필요한 경우 지정된 서브넷 내의 호스트에 **mon** 데몬을 배치하도록 **mon** 사양을 업데이트합니다.

추가 리소스

- **Red Hat Ceph Storage** 설치 가이드에서 호스트를 추가하는 방법에 대한 자세한 내용은 호스트 추가를 참조하십시오.

https://docs.redhat.com/en/documentation/red_hat_ceph_storage/8/html-single/installation_guide/#adding-hosts_install

- Red Hat Ceph Storage 관리 가이드의 확장 클러스터에 대한 자세한 내용은 Ceph 스토리지 용 Cryostat 클러스터를 참조하십시오.

2.6. 기본 CEPH 포트에 대한 방화벽 규칙 확인

기본적으로 Red Hat Ceph Storage 데몬은 TCP 포트 6800-7100을 사용하여 클러스터의 다른 호스트와 통신합니다. 호스트의 방화벽이 이러한 포트에서 연결을 허용하는지 확인할 수 있습니다.

참고

네트워크에 전용 방화벽이 있는 경우 다음 절차 외에 구성을 확인해야 할 수 있습니다. 자세한 내용은 방화벽 설명서를 참조하십시오.

자세한 내용은 방화벽 설명서를 참조하십시오.

사전 요구 사항

- 호스트에 대한 루트 수준 액세스.

프로세스

1. 호스트의 iptables 구성을 확인합니다.

   a. 활성 규칙을 나열합니다.

   ```
   [root@host1 ~]# iptables -L
   ```

   b. TCP 포트 6800-7100에서 연결을 제한하는 규칙이 없는지 확인합니다.

   예

```
REJECT all -- anywhere anywhere reject-with icmp-host-prohibited
```

2. *호스트의 **firewalld** 구성을 확인합니다.*

   a. *호스트에서 열려 있는 포트를 나열합니다.*

   구문

   ```
   firewall-cmd --zone ZONE --list-ports
   ```

   예

   ```
   [root@host1 ~]# firewall-cmd --zone default --list-ports
   ```

   b. *범위에 **TCP** 포트 **6800-7100**이 포함되어 있는지 확인합니다.*

2.7. CEPH MONITOR 노드의 방화벽 설정

*enger 버전 **2** 프로토콜을 도입하여 네트워크를 통한 모든 **Ceph** 트래픽에 대해 암호화를 활성화할 수 있습니다. **v2**의 보안 모드 설정은 **Ceph** 데몬과 **Ceph** 클라이언트 간의 통신을 암호화하여 엔드 투 엔드 암호화를 제공합니다.*

***Cryostat v2** 프로토콜*

Ceph의 on-wire 프로토콜의 두 번째 버전인 msgr2 에는 몇 가지 새로운 기능이 포함되어 있습니다.

- *보안 모드는 네트워크를 통해 이동하는 모든 데이터를 암호화합니다.*
- *인증 페이로드의 캡슐화 개선*
- *광고 및 협상에 대한 개선 사항*

Ceph 데몬은 여러 포트에 바인딩되므로 레거시 v1-호환 및 새로운 v2 호환 Ceph 클라이언트가 동일한 스토리지 클러스터에 연결할 수 있습니다. Ceph Monitor 데몬에 연결하는 Ceph 클라이언트 또는 기타 Ceph 데몬은 가능한 경우 v2 프로토콜을 먼저 사용하려고 하지만 그렇지 않은 경우 레거시 v1 프로토콜이 사용됩니다. 기본적으로 두 프로토콜 모두 v1 및 v2 가 활성화되어 있습니다. 새로운 v2 포트는 3300이며 레거시 v1 포트는 기본적으로 6789입니다.

사전 요구 사항

- *실행 중인 Red Hat Ceph Storage 클러스터.*
- *Ceph 소프트웨어 리포지토리에 액세스합니다.*
- *Ceph Monitor 노드에 대한 루트 수준 액세스.*

프로세스

1. *다음 예제를 사용하여 규칙을 추가합니다.*

```
[root@mon ~]# sudo iptables -A INPUT -i IFACE -p tcp -s IP-ADDRESS/NETMASK --dport
6789 -j ACCEPT
[root@mon ~]# sudo iptables -A INPUT -i IFACE -p tcp -s IP-ADDRESS/NETMASK --dport
3300 -j ACCEPT
```

   a. *IFACE 를 공용 네트워크 인터페이스(예: eth0,eth1 등)로 바꿉니다.*

b.

IP-ADDRESS 를 공용 네트워크의 *IP* 주소로 바꾸고 *NETMASK* 를 공용 네트워크의 넷 마스크로 바꿉니다.

2.

firewalld 데몬의 경우 다음 명령을 실행합니다.

```
[root@mon ~]# firewall-cmd --zone=public --add-port=6789/tcp
[root@mon ~]# firewall-cmd --zone=public --add-port=6789/tcp --permanent
[root@mon ~]# firewall-cmd --zone=public --add-port=3300/tcp
[root@mon ~]# firewall-cmd --zone=public --add-port=3300/tcp --permanent
```

추가 리소스

- 특정 옵션 설명 및 사용은 *Ceph* 네트워크 구성 옵션의 *Red Hat Ceph Storage*네트워크 구성 옵션을 참조하십시오.

- *Ceph* 전달자 버전 *2* 프로토콜에서 *Ceph* 온-와이어 암호화를 사용하는 방법에 대한 자세한 내용은 *Red Hat Ceph Storage Architecture Guide* 를 참조하십시오.

3장. CEPH MONITOR 구성

스토리지 관리자는 Ceph Monitor에 기본 구성 값을 사용하거나 의도한 워크로드에 따라 사용자 지정할 수 있습니다.

사전 요구 사항

- Red Hat Ceph Storage 소프트웨어 설치.

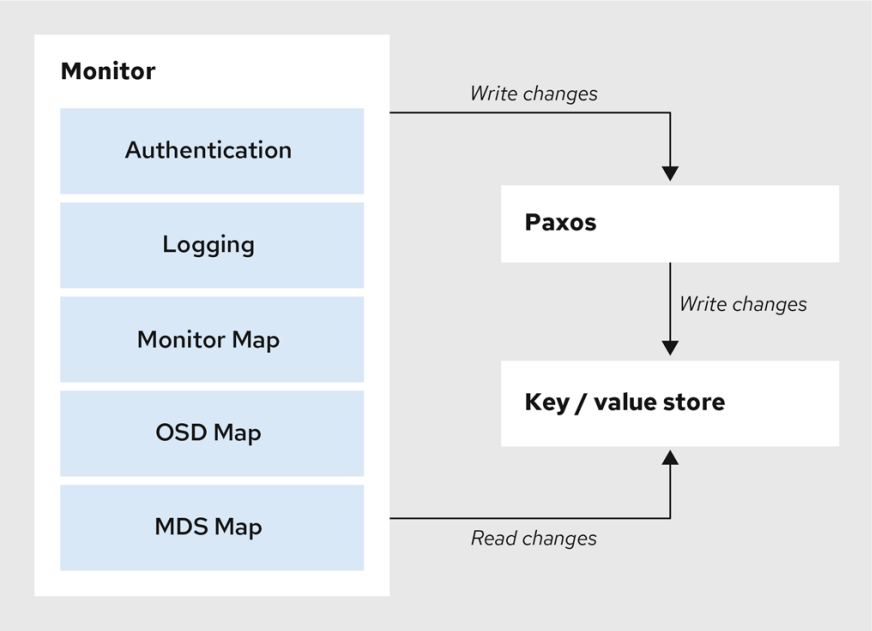
3.1. CEPH MONITOR 구성

Ceph Monitor를 구성하는 방법은 안정적인 Red Hat Ceph Storage 클러스터를 구축하는 데 중요한 부분입니다. 모든 스토리지 클러스터에는 하나 이상의 모니터가 있습니다. Ceph Monitor 구성은 일반적으로 상당히 일관되게 유지되지만 스토리지 클러스터에서 Ceph Monitor를 추가, 제거 또는 교체할 수 있습니다.

Ceph 모니터는 클러스터 맵의 "마스터 복사"를 유지 관리합니다. 즉, Ceph 클라이언트는 하나의 Ceph 모니터에 연결하고 현재 클러스터 맵을 검색하는 것만으로 모든 Ceph 모니터 및 Ceph OSD의 위치를 확인할 수 있습니다.

Ceph 클라이언트가 Ceph OSD에서 읽거나 쓸 수 있으려면 먼저 Ceph Monitor에 연결해야 합니다. 현재 클러스터 맵 및 CRUSH 알고리즘의 사본으로 Ceph 클라이언트는 모든 오브젝트의 위치를 계산할 수 있습니다. Ceph 클라이언트는 개체 위치를 계산하는 기능을 통해 Ceph OSD와 직접 통신할 수 있습니다. 이는 Ceph의 확장성 및 성능에 매우 중요한 측면입니다.

Ceph Monitor의 기본 역할은 클러스터 맵의 마스터 사본을 유지 관리하는 것입니다. Ceph 모니터는 인증 및 로깅 서비스도 제공합니다. Ceph Monitor는 모니터 서비스의 모든 변경 사항을 단일 Paxos 인스턴스에 작성하고 Paxos는 강력한 일관성을 위해 키-값 저장소에 변경 사항을 씁니다. Ceph 모니터는 동기화 작업 중에 최신 버전의 클러스터 맵을 쿼리할 수 있습니다. Ceph 모니터는 rocksdb 데이터베이스를 사용하여 키-값 저장소의 스냅샷과 Cryostat를 활용하여 저장소 전체 동기화를 수행합니다.



3.2. CEPH MONITOR 구성 데이터베이스 보기

구성 데이터베이스에서 **Ceph Monitor** 구성을 볼 수 있습니다.

참고

이전 **Red Hat Ceph Storage** 릴리스는 **/etc/ceph/ceph.conf** 에서 **Ceph Monitor** 구성을 중앙 집중화합니다. 이 구성 파일은 **Red Hat Ceph Storage 5**에서 더 이상 사용되지 않습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ceph Monitor** 호스트에 대한 루트 수준 액세스.

프로세스

1. **cephadm** 쉘에 로그인합니다.

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

2. **ceph config** 명령을 사용하여 구성 데이터베이스를 확인합니다.

예

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

추가 리소스

- **ceph config** 명령에 사용할 수 있는 옵션에 대한 자세한 내용은 **ceph config -h** 를 사용합니다.

3.3. CEPH 클러스터 맵

클러스터 맵은 모니터 맵, OSD 맵, 배치 그룹 맵을 포함한 복합 맵입니다. 클러스터 맵은 여러 중요한 이벤트를 추적합니다.

- **Red Hat Ceph Storage** 클러스터에 있는 프로세스는 무엇입니까.
- **Red Hat Ceph Storage** 클러스터에 있는 프로세스 중 실행 중이거나 중단되는 프로세스는 무엇입니까.
- 배치 그룹이 활성 상태이거나 비활성 상태인지, 정리 또는 기타 상태인지 여부입니다.
- 클러스터의 현재 상태를 반영하는 기타 세부 사항은 다음과 같습니다.
  - 총 저장 공간의 양 또는
  - 사용된 스토리지의 양입니다.

*클러스터 상태가 크게 변경되면 예를 들어 **Ceph OSD**가 중단되면 배치 그룹이 성능이 저하된 상태가 됩니다. 클러스터 맵이 업데이트되어 클러스터의 현재 상태가 반영됩니다. 또한 **Ceph** 모니터는 클러스터의 이전 상태 기록도 유지 관리합니다. 모니터 맵, **OSD** 맵, 배치 그룹 맵은 각각 맵 버전의 기록을 유지합니다. 각 버전을 **epoch** 라고 합니다.*

***Red Hat Ceph Storage** 클러스터를 작동할 때 이러한 상태를 추적하는 것이 클러스터 관리의 중요한 부분입니다.*

3.4. CEPH MONITOR 쿼럼

*클러스터는 단일 모니터로 충분히 실행됩니다. 그러나 단일 모니터는 단일 장애 지점입니다. 프로덕션 **Ceph** 스토리지 클러스터에서 고가용성을 보장하기 위해 여러 모니터가 있는 **Ceph**를 실행하여 단일 모니터가 실패하면 전체 스토리지 클러스터가 실패하지 않습니다.*

***Ceph** 스토리지 클러스터에서 고가용성을 위해 여러 **Ceph Monitor**를 실행하는 경우 **Ceph** 모니터는 **Paxos** 알고리즘을 사용하여 마스터 클러스터 맵에 대한 합의를 설정합니다. 합의를 위해서는 클러스터 맵에 대한 합의에 대한 쿼럼을 구축하기 위해 실행 중인 대부분의 모니터가 필요합니다. 예를 들어 **1; 3** 중 **2**개**; 5**개 중 **3**개**, 6**개 중 **4**개**, 6**개 중 **4**개 등*

***Red Hat**은 고가용성을 보장하기 위해 **3**개 이상의 **Ceph** 모니터를 사용하여 프로덕션 **Red Hat Ceph Storage** 클러스터를 실행하는 것이 좋습니다. 여러 모니터를 실행하는 경우 쿼럼을 설정하기 위해 스토리지 클러스터의 멤버여야 하는 초기 모니터를 지정할 수 있습니다. 이로 인해 스토리지 클러스터가 온라인 상태가 되는 데 걸리는 시간이 단축될 수 있습니다.*

```
[mon]
mon_initial_members = a,b,c
```

참고

*스토리지 클러스터의 대부분의 모니터는 쿼럼을 설정하기 위해 서로 연결할 수 있어야 합니다. 초기 모니터 수를 감소하여 **mon_initial_members** 옵션을 사용하여 쿼럼을 설정할 수 있습니다.*

3.5. CEPH MONITOR 일관성

***Ceph** 구성 파일에 모니터 설정을 추가할 때 **Ceph** 모니터의 아키텍처 측면을 알고 있어야 합니다. **Ceph**는 클러스터 내에서 다른 **Ceph Monitor**를 검색할 때 **Ceph Monitor**에 대한 엄격한 일관성 요구 사*

항을 적용합니다. Ceph 클라이언트 및 기타 Ceph 데몬은 Ceph 구성 파일을 사용하여 모니터를 검색하지만 모니터는 Ceph 구성 파일이 아닌 모니터 맵(monmap)을 사용하여 서로 검색합니다.

Red Hat Ceph Storage 클러스터에서 다른 Ceph 모니터를 검색할 때 Ceph Monitor는 항상 모니터 맵의 로컬 사본을 나타냅니다. Ceph 구성 파일 대신 모니터 맵을 사용하면 클러스터가 손상될 수 있는 오류가 발생하지 않습니다. 예를 들어 모니터 주소 또는 포트를 지정할 때 Ceph 구성 파일에 오타가 있습니다. 모니터는 검색에 모니터 맵을 사용하고 클라이언트 및 기타 Ceph 데몬과 모니터 맵을 공유하므로 모니터 맵은 모니터에서 합의가 유효함을 엄격하게 보장합니다.

모니터 맵에 업데이트를 적용할 때 엄격한 일관성

Ceph Monitor의 다른 업데이트와 마찬가지로 모니터 맵의 변경 사항은 항상 Paxos라는 분산 합의 알고리즘을 통해 실행됩니다. 쿼럼의 각 모니터에 동일한 버전의 모니터 맵이 있는지 확인하기 위해 Ceph 모니터는 Ceph Monitor 추가 또는 제거와 같은 모니터 맵의 각 업데이트에 동의해야 합니다. Ceph Monitor에 최신 합의 버전과 이전 버전 세트가 있도록 모니터 맵에 대한 업데이트가 증분됩니다.

기록 유지

기록을 유지 관리하면 이전 버전의 모니터 맵이 있는 Ceph Monitor가 Red Hat Ceph Storage 클러스터의 현재 상태를 파악할 수 있습니다.

Ceph 모니터가 모니터 맵 대신 Ceph 구성 파일을 통해 서로 발견되면 Ceph 구성 파일이 자동으로 업데이트 및 배포되지 않기 때문에 추가 위험이 발생합니다. Ceph 모니터는 이전 Ceph 구성 파일을 실수로 사용하거나 Ceph 모니터를 인식하지 못하거나 쿼럼을 대체하거나 Paxos가 시스템의 현재 상태를 정확하게 확인할 수 없는 상황을 개발할 수 있습니다.

3.6. CEPH MONITOR 부트스트랩

대부분의 구성 및 배포 사례에서 cephadm 과 같은 Ceph를 배포하는 툴은 모니터 맵을 생성하여 Ceph 모니터를 부트스트랩하는 데 도움이 될 수 있습니다.

Ceph 모니터에는 몇 가지 명시적 설정이 필요합니다.

- *파일 시스템 ID: fsid 는 오브젝트 저장소의 고유 식별자입니다. 동일한 하드웨어에서 여러 스토리지 클러스터를 실행할 수 있으므로 모니터를 부트스트랩할 때 오브젝트 저장소의 고유 ID를 지정해야 합니다. cephadm 과 같은 배포 도구를 사용하면 파일 시스템 식별자가 생성되지만 수동으로 fsid 를 지정할 수도 있습니다.*

- *모니터 ID: 모니터 ID는 클러스터 내의 각 모니터에 할당된 고유한 ID입니다. 규칙에 따라 ID는 모니터의 호스트 이름으로 설정됩니다. 이 옵션은 배포 도구, ceph 명령 또는 Ceph 구성 파일을 사용하여 설정할 수 있습니다. Ceph 구성 파일에서 섹션은 다음과 같이 구성됩니다.*

  예

  ```
  [mon.host1]

  [mon.host2]
  ```

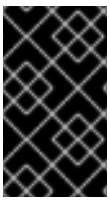
- *keys: 모니터에 시크릿 키가 있어야 합니다.*

추가 리소스

- *cephadm 및 Ceph 오케스트레이터에 대한 자세한 내용은 Red Hat Ceph Storage Operations Guide 를 참조하십시오.*

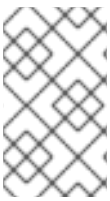
3.7. CEPH 모니터의 최소 구성

Ceph 구성 파일에서 Ceph Monitor의 베어 최소 모니터 설정에는 DNS 및 모니터 주소에 대해 구성되지 않은 경우 각 모니터의 호스트 이름이 포함되어 있습니다. Ceph 모니터는 기본적으로 포트 6789 및 3300 에서 실행됩니다.



중요

Ceph 구성 파일을 편집하지 마십시오.



참고

모니터의 최소 구성은 배포 도구가 fsid 및 mon. 키를 생성한다고 가정합니다.

다음 명령을 사용하여 스토리지 클러스터 구성 옵션을 설정하거나 읽을 수 있습니다.

- **Ceph** 구성 덤프 **-** 전체 스토리지 클러스터에 대해 전체 구성 데이터베이스를 덤프합니다.
- **Ceph config generate-minimal-conf -** 최소 **ceph.conf** 파일을 생성합니다.
- **Ceph config get Cryostat - Ceph Monitor**의 구성 데이터베이스에 저장된 대로 특정 데몬 또는 클라이언트의 구성을 덤프합니다.
- **Ceph config set Cryo stat OPTION VALUE - Ceph Monitor**의 구성 데이터베이스의 구성 옵션을 설정합니다.
- **Ceph config show kafka -** 실행 중인 데몬에 대해 보고된 실행 구성을 표시합니다.
- **Ceph config assimilate-conf -i INPUT_FILE -o OUTPUT_FILE -** 입력 파일에서 구성 파일을 부여하고 유효한 옵션을 **Ceph Monitor**의 구성 데이터베이스로 이동합니다.

여기에서 **Cryo stat** 매개 변수는 섹션의 이름 또는 **Ceph** 데몬일 수 있으며 **OPTION** 은 구성 파일이며 **VALUE** 는 **true** 또는 **false** 일 수 있습니다.

중요

구성 저장소에서 옵션을 가져오기 전에 **Ceph** 데몬에 **config** 옵션이 필요한 경우 다음 명령을 실행하여 구성을 설정할 수 있습니다.

```
ceph cephadm set-extra-ceph-conf
```

이 명령은 모든 데몬의 **ceph.conf** 파일에 텍스트를 추가합니다. 이는 해결 방법이며 권장되는 작업이 아닙니다.

3.8. CEPH의 고유 식별자

각 **Red Hat Ceph Storage** 클러스터에는 고유 식별자(**fsid**)가 있습니다. 지정하면 일반적으로 구성 파일의 **[global]** 섹션 아래에 표시됩니다. 배포 툴은 일반적으로 **fsid** 를 생성하고 이를 모니터 맵에 저장하므로 이 값은 구성 파일에 표시되지 않을 수 있습니다. **fsid** 를 사용하면 동일한 하드웨어에서 여러 클러스터에 대해 데몬을 실행할 수 있습니다.

참고

이를 위해 수행하는 배포 툴을 사용하는 경우 이 값을 설정하지 마십시오.

3.9. CEPH MONITOR 데이터 저장소

Ceph는 **Ceph** 모니터 데이터를 저장하는 기본 경로를 제공합니다.

중요

Red Hat은 프로덕션 **Red Hat Ceph Storage** 클러스터에서 최적의 성능을 위해 **Ceph OSD**의 별도의 드라이브에서 **Ceph** 모니터를 실행하는 것이 좋습니다.

참고

전용 **/var/lib/ceph** 파티션은 **50~100GB**의 크기가 있는 **MON** 데이터베이스에 사용해야 합니다.

Ceph 모니터는 **fsync()** 함수를 자주 호출하여 **Ceph OSD** 워크로드를 방해할 수 있습니다.

Ceph 모니터는 데이터를 키-값 쌍으로 저장합니다. 데이터 저장소를 사용하면 **Paxos**를 통해 손상된 버전에서 **Ceph** 모니터를 복구할 수 없으며, 다른 이점 중 하나의 단일 원자 배치로 여러 수정 작업을 수행할 수 있습니다.

중요

Red Hat은 기본 데이터 위치를 변경하지 않는 것이 좋습니다. 기본 위치를 수정하는 경우 구성 파일의 **[mon]** 섹션에서 설정하여 **Ceph** 모니터 전체에서 균일하게 설정합니다.

3.10. CEPH 스토리지 용량

Red Hat Ceph Storage 클러스터가 최대 용량(mon_osd_full_ratio 매개변수로 지정)에 근접하면 Ceph를 사용하면 데이터 손실을 방지하기 위해 Ceph OSD에 기록하거나 읽을 수 없습니다. 따라서 프로덕션 환경에서 Red Hat Ceph Storage 클러스터가 전체 비율에 접근하도록 하는 것은 고가용성을 저해하기 때문에 좋지 않습니다. 기본 전체 비율은 .95 또는 용량의 95%입니다. OSD 수가 적은 테스트 클러스터에 대해 매우 공격적인 설정입니다.

작은 정보

클러스터를 모니터링할 때 가까운 전체 비율과 관련된 경고에 유의하십시오. 즉, 일부 OSD가 실패하면 하나 이상의 OSD가 실패하면 일시적으로 서비스 중단이 발생할 수 있습니다. 스토리지 용량을 늘리려면 OSD를 추가하는 것이 좋습니다.

테스트 클러스터의 일반적인 시나리오에는 클러스터 재조정을 확인하기 위해 Red Hat Ceph Storage 클러스터에서 Ceph OSD를 제거하는 시스템 관리자가 포함됩니다. 그런 다음 Red Hat Ceph Storage 클러스터가 전체 비율 및 잠금에 도달할 때까지 다른 Ceph OSD를 제거합니다.

중요

Red Hat은 테스트 클러스터에서도 약간의 용량 계획을 권장합니다. 계획을 사용하면 고가용성을 유지하기 위해 필요한 예비 용량을 측정할 수 있습니다.

Ceph OSD를 즉시 교체하지 않고 클러스터가 활성 + 정리 상태로 복구할 수 있는 일련의 Ceph OSD 오류를 계획하는 것이 좋습니다. 활성 + 성능이 저하된 상태에서 클러스터를 실행할 수 있지만 정상적인 작동 상태에는 적합하지 않습니다.

다음 다이어그램에서는 호스트당 하나의 Ceph OSD가 있는 33 개의 Ceph 노드가 포함된 단순한 Red Hat Ceph Storage 클러스터를 보여줍니다. 각 Ceph OSD 데몬은 3TB 드라이브에서 읽고 씁니다. 따라서 이 예시적인 Red Hat Ceph Storage 클러스터는 최대 실제 용량이 99TB입니다. mon osd full ratio of 0.95 에서는 Red Hat Ceph Storage 클러스터가 5TB의 나머지 용량에 속하는 경우 Ceph 클라이언트가 데이터를 읽고 쓸 수 없습니다. 따라서 Red Hat Ceph Storage 클러스터의 운영 용량은 99TB가 아닌 95%TB입니다.

*하나 또는 두 개의 **OSD**가 실패하는 경우 클러스터에서 정상입니다. 랙의 라우터 또는 전원 공급이 덜 자주 발생하지만, 이로 인해 여러 **OSD**가 동시에 중단됩니다(예: **OSD 7-12**). 이러한 시나리오에서는 작동 상태를 유지하고 짧은 순서로 추가 **OSD**가 있는 몇 개의 호스트를 추가하는 것을 의미하더라도 활성 + 정리 상태를 달성할 수 있는 클러스터를 계속 사용해야 합니다. 용량 사용률이 너무 높으면 데이터를 손실할 수 없지만 클러스터의 용량 사용률이 전체 비율을 초과하면 장애 도메인 내에서 중단을 해결하면서 데이터 가용성을 저하시킬 수 있습니다. 이러한 이유로 **Red Hat**은 최소 일부의 대략적인 용량 계획을 권장합니다.*

클러스터의 두 번호를 확인합니다.

- ***OSD** 수*
- *클러스터의 총 용량*

*클러스터 내에서 **OSD**의 평균 용량을 확인하려면 클러스터의 총 용량을 클러스터의 **OSD** 수로 나눕니다. 정상적인 작업 중에 동시에 실패할 것으로 예상되는 **OSD** 수와 그 수를 곱하는 것이 좋습니다(상대적으로 적은 수). 마지막으로 최대 작동 용량에 도달하기 위해 클러스터의 용량을 전체 비율로 곱합니다. 그런 다음 적절한 전체 비율로 도달하지 못할 **OSD**에서 데이터 양을 제거합니다. **OSD** 오류가 많은 예상 프로세스(예: **OSD** 랙)를 반복하여 거의 전체 비율로 적절한 수에 도달합니다.*

3.11. CEPH 하트비트

Ceph 모니터는 각 OSD에서 보고해야 하고 인접 OSD의 상태에 대한 OSD에서 보고서를 수신하여 클러스터에 대해 알고 있습니다. Ceph는 모니터와 OSD 간의 상호 작용을 위해 적절한 기본 설정을 제공하지만 필요에 따라 수정할 수 있습니다.

3.12. CEPH MONITOR 동기화 역할

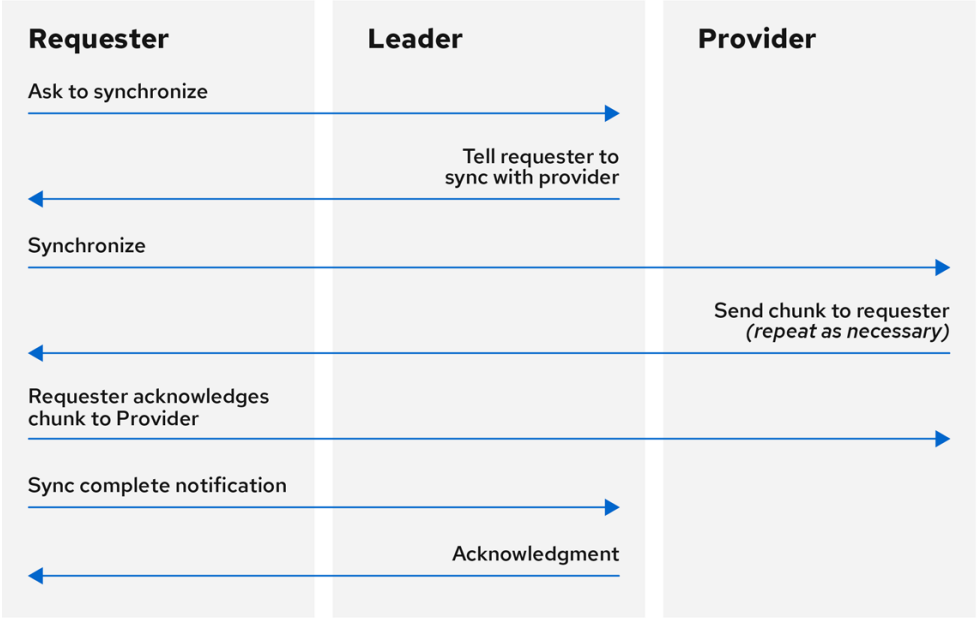
여러 모니터가 있는 프로덕션 클러스터를 실행할 때 각 모니터에서 주변 모니터에 더 최신 버전의 클러스터 맵이 있는지 확인합니다. 예를 들어, 인스턴트 모니터의 맵에서 가장 최신의 epoch보다 하나 이상의 epoch 번호를 갖는 인접 모니터의 맵. 주기적으로 클러스터의 한 모니터는 쿼럼을 떠나야 하는 시점으로 다른 모니터를 대체하고, 동기화를 통해 클러스터에 대한 최신 정보를 검색한 다음 쿼럼에 다시 참여할 수 있습니다.

동기화 역할

동기화를 위해 모니터는 다음 세 가지 역할 중 하나를 가정할 수 있습니다.

- *leader: 리더는 클러스터 맵의 최신 Paxos 버전을 달성하는 첫 번째 모니터입니다.*
- *공급자: 공급자는 클러스터 맵의 최신 버전이 있지만 가장 최신 버전을 달성하는 첫 번째 버전은 없는 모니터입니다.*
- *요청자: 요청자는 리더 뒤에 있는 모니터이며 쿼럼에 다시 참여하기 전에 클러스터에 대한 최신 정보를 검색하도록 동기화해야 합니다.*

이러한 역할을 통해 리더는 동기화 작업을 공급자에 위임할 수 있으므로 동기화 요청이 리더를 과부하하고 성능을 개선하는 것을 방지할 수 있습니다. 다음 다이어그램에서 요청자는 다른 모니터 뒤에 있다는 것을 알게 되었습니다. 요청자는 리더에게 동기화하도록 요청하고, 리더는 요청자에게 공급자와 동기화하도록 지시합니다.



동기화 모니터링

동기화는 새 모니터가 클러스터에 참여할 때 항상 발생합니다. 런타임 작업 중에 모니터는 다른 시간에 클러스터 맵에 대한 업데이트를 수신할 수 있습니다. 즉, 리더 및 공급자 역할은 한 모니터에서 다른 모니터로 마이그레이션할 수 있습니다. 예를 들어 공급자가 리더 뒤에 있으면 공급자는 요청자와 동기화를 종료할 수 있습니다.

*동기화가 완료되면 **Ceph**가 클러스터 전체에서 트리밍해야 합니다. 트리밍에서는 배치 그룹이 활성 + **clean**이어야 합니다.*

3.13. CEPH 시간 동기화

***Ceph** 데몬은 중요한 메시지를 서로 전달하며 데몬이 시간 초과 임계값에 도달하기 전에 처리해야 합니다. **Ceph** 모니터의 시계가 동기화되지 않으면 여러 가지 문제가 발생할 수 있습니다.*

예를 들면 다음과 같습니다.

- *데몬은 오래된 타임스탬프와 같은 수신된 메시지를 무시합니다.*
- *시간 초과는 메시지가 시간 내에 수신되지 않은 경우 너무 빨리 또는 늦었습니다.*

작은 정보

***Ceph** 모니터 호스트에 **NTP**를 설치하여 모니터 클러스터가 동기화된 클록으로 작동하는지 확인합니다.*

*불일치가 아직 유해하지는 않더라도 클럭 드리프트는 **NTP**에서 계속 사용할 수 있습니다. **NTP**가 적절한 수준의 동기화를 유지하더라도 **Ceph** 클럭 드리프트 및 클럭 스큐 경고가 트리거될 수 있습니다. 이러한 상황에서 클럭 드리프트를 늘리는 것은 허용될 수 있습니다. 그러나 워크로드, 네트워크 대기 시간, 기본 시간 초과에 대한 덮어쓰기 구성, 기타 동기화 옵션은 **Paxos** 보장을 손상시키지 않고 허용 가능한 클럭 드리프트 수준에 영향을 미칠 수 있습니다.*

추가 리소스

- *자세한 내용은 **Ceph** 시간 동기화 섹션을 참조하십시오.*
- *특정 옵션 설명 및 사용은 **Ceph Monitor** 구성 옵션의 모든 **Red Hat Ceph Storage Monitor** 구성 옵션을 참조하십시오.*

4장. CEPH 인증 구성

스토리지 관리자는 **Red Hat Ceph Storage** 클러스터 보안에 사용자 및 서비스를 인증하는 것이 중요합니다. **Red Hat Ceph Storage**에는 암호화 인증을 위한 **Cephx** 프로토콜과 스토리지 클러스터에서 인증을 관리하는 툴이 포함되어 있습니다.

Red Hat Ceph Storage에는 암호화 인증을 위한 **Cephx** 프로토콜과 스토리지 클러스터에서 인증을 관리하는 툴이 포함되어 있습니다.

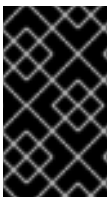
Ceph 인증 구성의 일부로 보안을 강화하기 위해 **Ceph** 및 게이트웨이 데몬의 키 교체를 고려하십시오. 키 교체는 **cephadm** 과 함께 명령줄을 통해 수행됩니다. 자세한 내용은 *키 교체 활성화를* 참조하십시오.

사전 요구 사항

- **Red Hat Ceph Storage** 소프트웨어 설치.

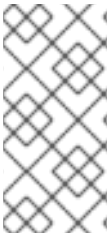
4.1. CEPHX 인증

cephx 프로토콜은 기본적으로 활성화되어 있습니다. 암호화 인증에는 약간의 컴퓨팅 비용이 있지만 일반적으로 매우 낮습니다. 클라이언트 및 호스트를 연결하는 네트워크 환경이 안전한 것으로 간주되고 인증 컴퓨팅 비용을 허용할 수 없는 경우 비활성화할 수 있습니다. **Ceph** 스토리지 클러스터를 배포할 때 배포 툴에서 **client.admin** 사용자 및 인증 키를 생성합니다.



중요

인증을 사용하는 것이 좋습니다.



참고

인증을 비활성화하면 메시지 가로채기(**man-in-the-middle**) 공격에서 클라이언트 및 서버 메시지를 변경할 위험이 있으므로 심각한 보안 문제가 발생할 수 있습니다.

Cephx 활성화 및 비활성화

Cephx를 활성화하려면 **Ceph Monitor** 및 **OSD**에 대한 키를 배포해야 합니다. **Cephx** 인증을 켜거나 끄는 경우 배포 절차를 반복할 필요가 없습니다.

4.2. CEPHX 활성화

cephx 가 활성화되면 **Ceph**는 기본 검색 경로에서 인증 키를 찾습니다. 여기에는 **/etc/ceph/$cluster.$name.keyring. Ceph** 구성 파일의 **[global]** 섹션에 인증 키 옵션을 추가하여 이 위치를 재정의할 수는 있지만 권장되지는 않습니다.

인증이 비활성화된 클러스터에서 **cephx** 를 활성화하려면 다음 절차를 실행합니다. 사용자 또는 배포 유틸리티에서 이미 키를 생성한 경우 키 생성과 관련된 단계를 건너뛸 수 있습니다.

사전 요구 사항

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

프로세스

1. **client.admin** 키를 생성하고 클라이언트 호스트에 대한 키 사본을 저장합니다.

```
[root@mon ~]# ceph auth get-or-create client.admin mon 'allow *' osd 'allow *' -o
/etc/ceph/ceph.client.admin.keyring
```

주의

그러면 기존 **/etc/ceph/client.admin.keyring** 파일의 내용이 지워집니다. 배포 툴에서 이미 수행한 경우 이 단계를 수행하지 마십시오.

2. 모니터 클러스터에 대한 인증 키를 생성하고 모니터 시크릿 키를 생성합니다.

```
[root@mon ~]# ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --
cap mon 'allow *'
```

3. *모니터 인증 키를 모든 모니터 **mon** 데이터 디렉터리의 **ceph.mon.keyring** 파일에 복사합니다. 예를 들어 클러스터 **ceph** 의 **mon.a** 에 복사하려면 다음을 사용합니다.*

```
[root@mon ~]# cp /tmp/ceph.mon.keyring /var/lib/ceph/mon/ceph-a/keyring
```

4. *모든 **OSD**에 대한 시크릿 키를 생성합니다. 여기서 **ID** 는 **OSD** 번호입니다.*

```
ceph auth get-or-create osd.ID mon 'allow rwx' osd 'allow *' -o
/var/lib/ceph/osd/ceph-ID/keyring
```

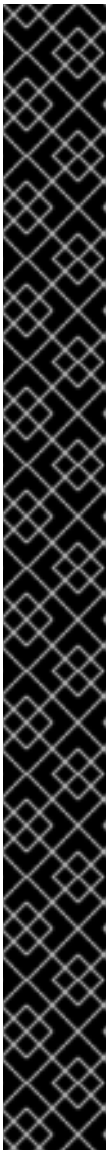
5. *기본적으로 **cephx** 인증 프로토콜은 활성화되어 있습니다.*

참고

*이전에 인증 옵션을 **none** 으로 설정하여 **cephx** 인증 프로토콜을 비활성화한 경우 **Ceph** 구성 파일(**/etc/ceph/ceph.conf**)의 **[global]** 섹션에서 다음 행을 제거하면 **cephx** 인증 프로토콜을 다시 활성화합니다.*

```
auth_cluster_required = none
auth_service_required = none
auth_client_required = none
```

6. ***Ceph** 스토리지 클러스터를 시작하거나 다시 시작합니다.*



중요

cephx 를 활성화하려면 클러스터를 완전히 다시 시작해야 하거나 클라이언트 I/O를 비활성화하는 동안 종료한 다음 시작해야 하므로 다운타임이 필요합니다.

스토리지 클러스터를 재시작하거나 종료하기 전에 다음 플래그를 설정해야 합니다.

```
[root@mon ~]# ceph osd set noout
[root@mon ~]# ceph osd set norecover
[root@mon ~]# ceph osd set norebalance
[root@mon ~]# ceph osd set nobackfill
[root@mon ~]# ceph osd set nodown
[root@mon ~]# ceph osd set pause
```

cephx 가 활성화되고 모든 **PG**가 활성 상태이고 정리되면 플래그를 설정 해제합니다.

```
[root@mon ~]# ceph osd unset noout
[root@mon ~]# ceph osd unset norecover
[root@mon ~]# ceph osd unset norebalance
[root@mon ~]# ceph osd unset nobackfill
[root@mon ~]# ceph osd unset nodown
[root@mon ~]# ceph osd unset pause
```

4.3. CEPHX 비활성화

다음 절차에서는 **Cephx**를 비활성화하는 방법을 설명합니다. 클러스터 환경이 비교적 안전한 경우 인증 실행 비용을 상쇄할 수 있습니다.

중요

인증을 활성화하는 것이 좋습니다.

그러나 인증을 일시적으로 비활성화하도록 설정 또는 문제 해결 중에 더 쉬워질 수 있습니다.

사전 요구 사항

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

프로세스

1. ***Ceph** 구성 파일의 **[global]** 섹션에서 다음 옵션을 설정하여 **cephx** 인증을 비활성화합니다.*

   예

```
auth_cluster_required = none
auth_service_required = none
auth_client_required = none
```

2. ***Ceph** 스토리지 클러스터를 시작하거나 다시 시작합니다.*

4.4. CEPHX 사용자 인증 키

*인증이 활성화된 **Ceph**를 실행하는 경우 **Ceph** 관리 명령 및 **Ceph** 클라이언트에 **Ceph** 스토리지 클러스터에 액세스하려면 인증 키가 필요합니다.*

*이러한 키를 **ceph** 관리 명령 및 클라이언트에 제공하는 가장 일반적인 방법은 **/etc/ceph/** 디렉터리에 **Ceph** 인증 키를 포함하는 것입니다. 파일 이름은 일반적으로 **ceph.client.admin.keyring** 또는 **$cluster.client.admin.keyring** 입니다. 인증 키를 **/etc/ceph/** 디렉터리에 포함하는 경우 **Ceph** 구성 파일에 인증 키 항목을 지정할 필요가 없습니다.*

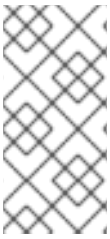
중요

***Red Hat**은 **client.admin** 키가 포함되어 있으므로 관리 명령을 실행할 노드에 **Red Hat Ceph Storage** 클러스터 인증 키를 복사하는 것이 좋습니다.*

이렇게 하려면 다음 명령을 실행합니다.

```
# scp USER@HOSTNAME:/etc/ceph/ceph.client.admin.keyring /etc/ceph/ceph.client.admin.keyring
```

USER 를 호스트에서 사용하는 사용자 이름으로 **client.admin** 키로, **HOSTNAME** 을 해당 호스트의 호스트 이름으로 바꿉니다.



참고

ceph.keyring 파일에 클라이언트 시스템에 적절한 권한이 설정되어 있는지 확인합니다.

권장되지 않는 키 설정 또는 **key file** 설정을 사용하여 키 파일의 경로를 **Ceph** 구성 파일에 지정할 수 있습니다.

4.5. CEPHX 데몬 인증 키

관리 사용자 또는 배포 툴은 사용자 인증 키를 생성하는 것과 동일한 방식으로 데몬 인증 키를 생성할 수 있습니다. 기본적으로 **Ceph**는 데이터 디렉터리에 데몬 인증 키를 저장합니다. 기본 키 링 위치와 데몬이 작동하는 데 필요한 기능입니다.

참고

모니터 인증 키링에는 키가 포함되어 있지만 기능은 없으며 **Ceph** 스토리지 클러스터 인증 데이터베이스의 일부가 아닙니다.

데몬 데이터 디렉터리 위치는 기본적으로 양식의 디렉터리로 설정됩니다.

```
/var/lib/ceph/$type/CLUSTER-ID
```

예

```
/var/lib/ceph/osd/ceph-12
```

이러한 위치를 재정의할 수는 있지만 권장되지는 않습니다.

4.6. CEPHX 메시지 서명

***Ceph**는 세분화된 제어를 제공하므로 클라이언트와 **Ceph** 간의 서비스 메시지의 서명을 활성화하거나 비활성화할 수 있습니다. **Ceph** 데몬 간 메시지의 서명을 활성화하거나 비활성화할 수 있습니다.*

중요

***Red Hat**은 초기 인증에 설정된 세션 키를 사용하여 엔터티 간 모든 진행 중인 메시지를 인증하는 것이 좋습니다.*

참고

***Ceph** 커널 모듈은 아직 서명을 지원하지 않습니다.*

5장. 풀, 배치 그룹 및 CRUSH 구성

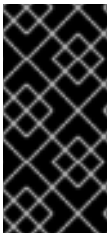
스토리지 관리자는 풀, 배치 그룹 및 CRUSH 알고리즘에 대해 Red Hat Ceph Storage 기본 옵션을 사용하거나 의도한 워크로드에 맞게 사용자 지정할 수 있습니다.

사전 요구 사항

- Red Hat Ceph Storage 소프트웨어 설치.

5.1. 풀 배치 그룹 및 CRUSH

풀을 생성하고 풀에 대한 배치 그룹 수를 설정할 때 기본값을 구체적으로 재정의하지 않는 경우 Ceph는 기본값을 재정의하지 않는 경우 기본값을 사용합니다.



중요

Red Hat은 일부 기본값을 재정의하는 것이 좋습니다. 특히 풀의 복제본 크기를 설정하고 기본 배치 그룹 수를 재정의합니다.

pool 명령을 실행할 때 이러한 값을 설정할 수 있습니다.

기본적으로 Ceph는 3개의 오브젝트 복제본을 만듭니다. 오브젝트의 복사본 4개를 기본값, 기본 복사 및 3개의 복제본 복사본으로 설정하려면 osd_pool_default_size 에 표시된 대로 기본값을 재설정합니다. Ceph에서 성능이 저하된 상태의 복사본 수를 더 적게 작성하도록 허용하려면 osd_pool_default_min_size 값을 osd_pool_default_size 값보다 작은 숫자로 설정합니다.

예

```
[ceph: root@host01 /]# ceph config set global osd_pool_default_size 4  # Write an object 4 times.
[ceph: root@host01 /]# ceph config set global osd_pool_default_min_size 1  # Allow writing one copy in a degraded state.
```

*비현실적인 수의 배치 그룹이 있는지 확인하십시오. OSD당 약 100개를 권장합니다. 예를 들어 총 OSD 수를 복제본 수를 100으로 나눕니다(즉, osd_pool_default_size). 10개의 OSD 및 osd_pool_default_size = 4의 경우 약 (100 * 10) / 4 = 250를 권장합니다.*

예

```
[ceph: root@host01 /]# ceph config set global osd_pool_default_pg_num 250
[ceph: root@host01 /]# ceph config set global osd_pool_default_pgp_num 250
```

추가 리소스

- *특정 옵션 설명 및 사용은 I부록 E 의 모든 Red Hat Ceph Storage 풀, 배치 그룹 및 CRUSH 구성 옵션을 참조하십시오.*

6장. CEPH OSD(오브젝트 스토리지 데몬) 구성

스토리지 관리자는 의도한 워크로드에 따라 중복되고 최적화되도록 Ceph OSD(오브젝트 스토리지 데몬)를 구성할 수 있습니다.

사전 요구 사항

- Red Hat Ceph Storage 소프트웨어 설치.

6.1. CEPH OSD 구성

모든 Ceph 클러스터에는 다음과 같은 구성이 있습니다.

- 클러스터 ID
- 인증 설정
- 클러스터의 Ceph 데몬 멤버십
- 네트워크 구성
- 호스트 이름 및 주소
- 인증 키 경로
- OSD 로그 파일의 경로
- 기타 런타임 옵션

cephadm 과 같은 배포 툴은 일반적으로 초기 Ceph 구성 파일을 생성합니다. 그러나 배포 도구를 사용

하지 않고 클러스터를 부트스트랩하려는 경우 클러스터를 직접 생성할 수 있습니다.

편의를 위해 각 데몬에는 일련의 기본값이 있습니다. 많은 사용자가 **ceph/src/common/config_opts.h** 스크립트에서 설정합니다. **monitor tell** 명령을 사용하거나 **Ceph** 노드의 데몬 소켓에 직접 연결하여 **Ceph** 구성 파일 또는 런타임에 이러한 설정을 재정의할 수 있습니다.

중요

Red Hat은 나중에 **Ceph**의 문제를 해결하기가 더 어렵기 때문에 기본 경로를 변경하지 않는 것이 좋습니다.

추가 리소스

- **cephadm** 및 **Ceph** 오케스트레이터에 대한 자세한 내용은 **Red Hat Ceph Storage Operations Guide** 를 참조하십시오.

6.2. OSD 스크럽

Ceph는 여러 오브젝트 복사본을 만드는 것 외에도 배치 그룹을 스크럽하여 데이터 무결성을 보장합니다. **Ceph** 스크럽은 오브젝트 스토리지 계층의 **fsck** 명령과 유사합니다.

Ceph는 각 배치 그룹에 대해 모든 오브젝트의 카탈로그를 생성하고 각 기본 오브젝트와 해당 복제본을 비교하여 오브젝트가 누락되거나 일치하지 않도록 합니다.

Light scrubbing (daily)은 오브젝트 크기와 특성을 확인합니다. **Deep scrubbing (weekly)**은 데이터를 읽고 체크섬을 사용하여 데이터 무결성을 보장합니다.

스크럽은 데이터 무결성을 유지하는 데 중요하지만 성능을 줄일 수 있습니다. 스크럽 작업을 늘리거나 줄이기 위해 다음 설정을 조정합니다.

추가 리소스

- 자세한 내용은 **Red Hat Ceph Storage** 구성 가이드의 **Ceph**스크럽 옵션에서 참조하십시오.

6.3. OSD 백필

Ceph OSD를 클러스터에 추가하거나 클러스터에서 제거하는 경우 CRUSH 알고리즘은 배치 그룹을 Ceph OSD로 이동하거나 Ceph OSD에서 복원하여 클러스터를 재조정합니다. 배치 그룹 및 포함된 오브젝트를 마이그레이션하는 프로세스는 클러스터 운영 성능을 크게 줄일 수 있습니다. Ceph는 운영 성능을 유지하기 위해 'backfill' 프로세스를 사용하여 이 마이그레이션을 수행하므로 Ceph에서 데이터를 읽거나 쓸 요청보다 우선 순위가 낮은 우선 순위로 설정할 수 있습니다.

6.4. OSD 복구

클러스터가 시작되거나 Ceph OSD가 예기치 않게 종료되면 쓰기 작업이 발생하기 전에 OSD가 다른 Ceph OSD와 피어링되기 시작합니다.

Ceph OSD가 충돌하여 다시 온라인 상태가 되면 일반적으로 배치 그룹에 최신 버전의 오브젝트가 포함된 다른 Ceph OSD와 동기화되지 않습니다. 이 경우 Ceph OSD가 복구 모드로 전환되고 최신 데이터 사본을 가져오고 해당 맵을 최신 상태로 되돌립니다. Ceph OSD가 중단된 기간에 따라 OSD의 오브젝트 및 배치 그룹이 최신 상태가 될 수 있습니다. 또한 장애 도메인이 다운된 경우(예: 랙) 두 개 이상의 Ceph OSD가 동시에 온라인 상태가 될 수 있습니다. 이로 인해 복구 프로세스에 시간이 많이 소비되고 리소스가 많이 소요될 수 있습니다.

Ceph는 운영 성능을 유지하기 위해 복구 요청, 스레드 및 오브젝트 청크 크기 제한으로 복구를 수행합니다. 그러면 Ceph가 성능이 저하된 상태에서 제대로 수행할 수 있습니다.

추가 리소스

- 특정 옵션 설명 및 사용은 OSD 개체 데몬 스토리지 구성 옵션의 모든 Red Hat Ceph Storage Ceph OSD 구성 옵션을 참조하십시오.

7장. CEPH MONITOR 및 OSD 상호 작용 구성

스토리지 관리자는 안정적인 작동 환경을 보장하기 위해 Ceph 모니터와 OSD 간의 상호 작용을 올바르게 구성해야 합니다.

사전 요구 사항

- Red Hat Ceph Storage 소프트웨어 설치.

7.1. CEPH MONITOR 및 OSD 상호 작용

초기 Ceph 구성을 완료한 후에는 Ceph를 배포하고 실행할 수 있습니다. ceph 상태 또는 ceph -s 와 같은 명령을 실행하면 Ceph Monitor가 Ceph 스토리지 클러스터의 현재 상태에 대해 보고합니다. Ceph Monitor는 각 Ceph OSD 데몬에서 보고해야 하고, Ceph OSD 데몬에서 Ceph OSD 데몬의 상태에 대한 보고서를 수신하여 Ceph 스토리지 클러스터에 대해 알고 있습니다. Ceph Monitor가 보고서를 수신하지 않거나 Ceph 스토리지 클러스터의 변경 사항 보고가 수신되는 경우 Ceph 모니터는 Ceph 클러스터 맵의 상태를 업데이트합니다.

Ceph는 Ceph Monitor 및 OSD 상호 작용에 적절한 기본 설정을 제공합니다. 그러나 기본값을 재정의할 수 있습니다. 다음 섹션에서는 Ceph 스토리지 클러스터를 모니터링하기 위해 Ceph Monitor 및 Ceph OSD 데몬이 상호 작용하는 방법을 설명합니다.

7.2. OSD 하트비트

각 Ceph OSD 데몬은 6초마다 다른 Ceph OSD 데몬의 하트비트를 확인합니다. 하트비트 간격을 변경하려면 런타임 시 값을 변경합니다.

구문

```
ceph config set osd osd_heartbeat_interval TIME_IN_SECONDS
```

예

```
[ceph: root@host01 /]# ceph config set osd osd_heartbeat_interval 60
```

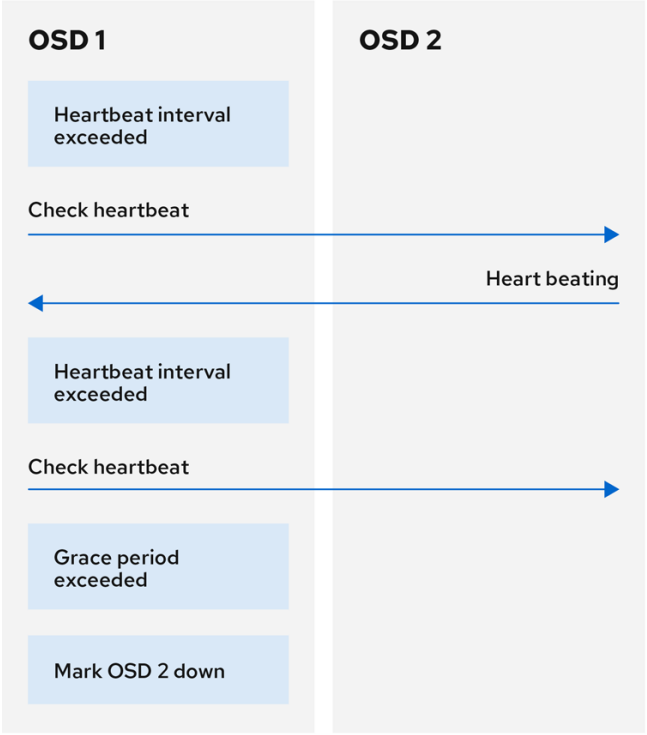
주변 **Ceph OSD** 데몬이 **20**초 유예 기간 내에 하트비트 패킷을 보내지 않는 경우 **Ceph OSD** 데몬에서 **Ceph OSD** 데몬을 아래로 간주할 수 있습니다. **Ceph** 클러스터 맵을 업데이트하는 **Ceph Monitor**로 다시 보고할 수 있습니다. 유예 기간을 변경하려면 런타임 시 값을 설정합니다.

구문

```
ceph config set osd osd_heartbeat_grace TIME_IN_SECONDS
```

예

```
[ceph: root@host01 /]# ceph config set osd osd_heartbeat_grace 30
```



7.3. OSD를 DOWN으로 보고

기본적으로 다른 호스트의 두 Ceph OSD 데몬은 Ceph 모니터에서 보고된 Ceph OSD 데몬이 다운된 것을 확인하기 전에 다른 Ceph OSD 데몬이 다운 되었음을 Ceph 모니터에 보고해야 합니다.

그러나 오류를 보고하는 모든 OSD가 랙의 다른 호스트에 있어 OSD 간의 연결 문제가 발생할 수 있습니다.

"false 알람"을 방지하기 위해 Ceph는 오류를 지연과 유사한 "subcluster"의 프록시로 보고하는 피어를 고려합니다. 이는 항상 그런 것은 아니지만 관리자가 제대로 작동하지 않는 시스템의 하위 집합에 대한 유예 수정 사항을 현지화하는 데 도움이 될 수 있습니다.

Ceph는 mon_osd_reporter_subtree_level 설정을 사용하여 CRUSH 맵의 공통 ancestor 유형으로 피어를 "subcluster"로 그룹화합니다.

기본적으로 다른 하위 트리의 두 보고서 만 다른 Ceph OSD 데몬 을 보고해야 합니다. 관리자는 런타임 시 mon_osd_min_ down _reporters 및 mon_osd_reporter_subtree_level 값을 설정하여 Ceph OSD 데몬을 Ceph Monitor로 보고하는 데 필요한 고유한 하위 트리 및 일반적인 ancestor 유형에서 보고자 수를 변경할 수 있습니다.

구문

```
ceph config set mon mon_osd_min_down_reporters NUMBER
```

예

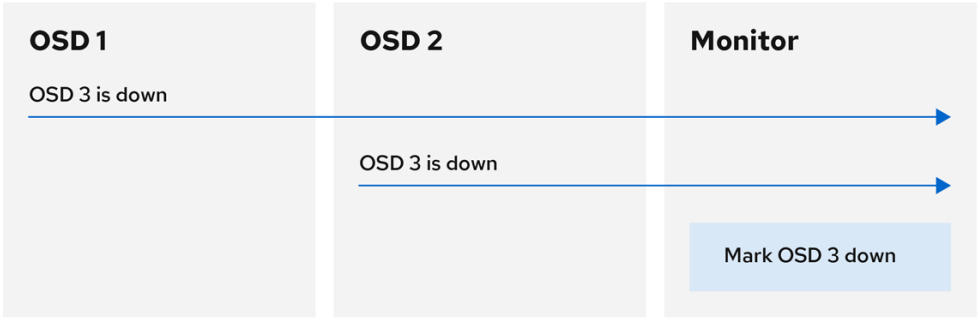
```
[ceph: root@host01 /]# ceph config set mon mon_osd_min_down_reporters 4
```

구문

```
ceph config set mon mon_osd_reporter_subtree_level CRUSH_ITEM
```

예

```
[ceph: root@host01 /]# ceph config set mon mon_osd_reporter_subtree_level host
[ceph: root@host01 /]# ceph config set mon mon_osd_reporter_subtree_level rack
[ceph: root@host01 /]# ceph config set mon mon_osd_reporter_subtree_level osd
```



7.4. 피어링 실패 보고

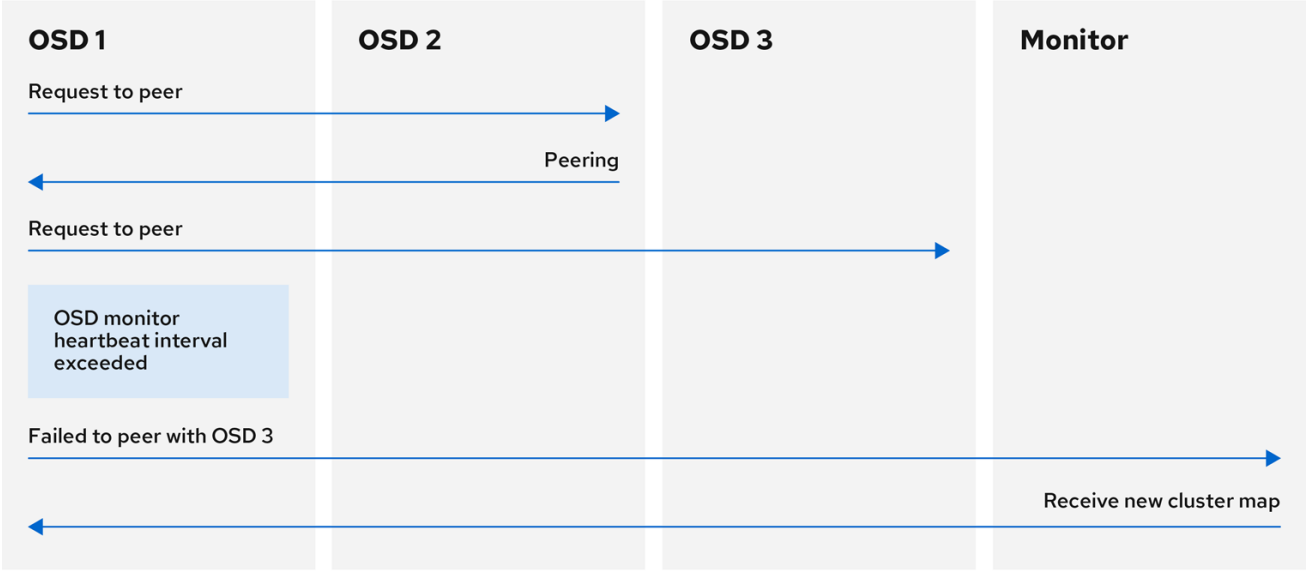
***Ceph OSD** 데몬이 **Ceph** 구성 파일 또는 클러스터 맵에 정의된 **Ceph OSD** 데몬을 피어링할 수 없는 경우 **30**초마다 클러스터 맵의 최신 사본에 대해 **Ceph Monitor**를 **ping**합니다. 런타임 시 값을 설정하여 **Ceph Monitor** 하트비트 간격을 변경할 수 있습니다.*

구문

```
ceph config set osd osd_mon_heartbeat_interval TIME_IN_SECONDS
```

예

```
[ceph: root@host01 /]# ceph config set osd osd_mon_heartbeat_interval 60
```



7.5. OSD 보고 상태

Ceph OSD 데몬에서 Ceph Monitor에 보고하지 않으면 mon_osd_report_timeout 다음에 Ceph OSD 데몬을 아래로 표시합니다. 이는 900초입니다. Ceph OSD 데몬은 장애, 배치 그룹 통계 변경, up_thru 의 변경 또는 5초 내에 부팅되는 경우 보고 가능한 이벤트가 있는 경우 Ceph OSD 데몬으로 보고서를 Ceph Monitor로 보냅니다.

런타임 시 osd_mon_report_interval 값을 설정하여 Ceph OSD Daemon 최소 보고서 간격을 변경할 수 있습니다.

구문

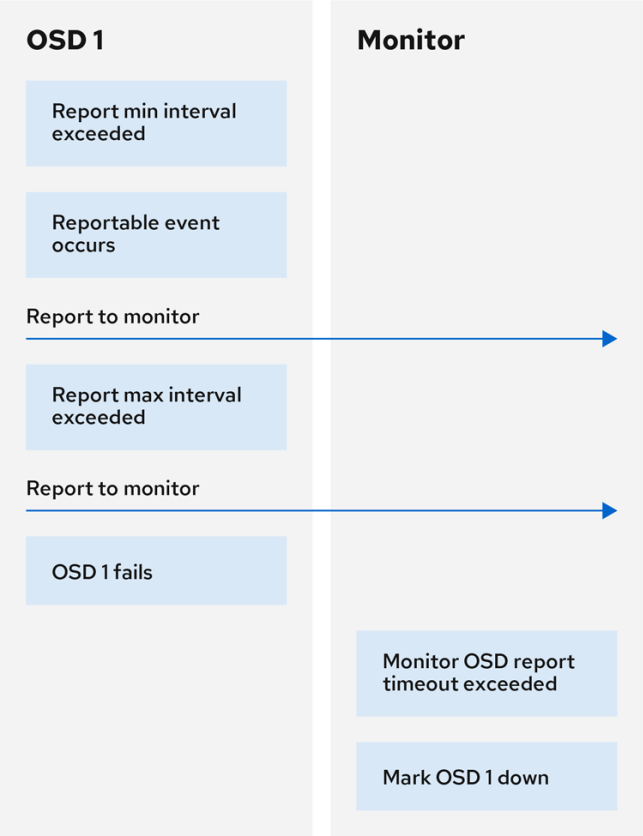
```
ceph config set osd osd_mon_report_interval TIME_IN_SECONDS
```

다음 예제를 사용하여 구성을 설정하고 확인합니다.

예

```
[ceph: root@host01 /]# ceph config get osd osd_mon_report_interval
5
[ceph: root@host01 /]# ceph config set osd osd_mon_report_interval 20
[ceph: root@host01 /]# ceph config dump | grep osd

global          advanced  osd_pool_default_crush_rule          -1
  osd           basic     osd_memory_target                    4294967296
  osd           advanced  osd_mon_report_interval              20
```



추가 리소스

- *특정 옵션 설명 및 사용은 **Ceph Monitor** 및 **OSD** 구성 옵션의 모든 **Red Hat** Ceph Storage Ceph Monitor 및 OSD 구성 옵션을 참조하십시오.*

8장. CEPH 디버깅 및 로깅 구성

스토리지 관리자는 cephadm 의 디버깅 및 로깅 정보를 늘릴 수 있으므로 Red Hat Ceph Storage 문제 진단에 도움이 됩니다.

사전 요구 사항

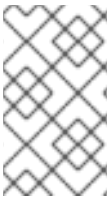
- Red Hat Ceph Storage 소프트웨어가 설치되어 있습니다.

추가 리소스

- 특정 옵션 설명 및 사용은 Ceph 디버깅의 모든 Red Hat Ceph Storage Ceph 디버깅 및 로깅 구성 옵션을 참조하십시오.
- cephadm 문제 해결에 대한 자세한 내용은 Red Hat Ceph Storage 관리 가이드의 Cephadm 문제 해결을 참조하십시오.
- cephadm 로깅에 대한 자세한 내용은 Red Hat Ceph Storage 관리 가이드의 Cephadm 작업을 참조하십시오.

부록 A. 일반 구성 옵션

이는 Ceph의 일반적인 구성 옵션입니다.



참고

일반적으로 cephadm 과 같은 배포 툴을 통해 자동으로 설정됩니다.

fsid

설명

파일 시스템 ID입니다. 클러스터당 하나씩.

유형

UUID

필수 항목

아니요.

기본

해당 없음. 일반적으로 배포 툴을 통해 생성됩니다.

admin_socket

설명

Ceph 모니터에 쿼럼을 설정했는지와 관계없이 데몬에서 관리 명령을 실행하는 소켓입니다.

유형

문자열

필수 항목

없음

기본

/var/run/ceph/$cluster-$name.asok

pid_file

설명

모니터 또는 OSD가 PID를 쓰는 파일입니다. 예를 들어, /var/run/$cluster/$type.$id.pid 는 ceph 클러스터에서 실행 중인 ID를 사용하여 mon에 대해 /var/run/ ceph / mon. a.pid를 생성합니다. 데몬이 정상적으로 중지되면 pid 파일이 제거됩니다. 프로세스가 데몬화되지 않은 경우(-f 또는 -d 옵션으로 실행됨) pid 파일이 생성되지 않습니다.

유형

문자열

필수 항목

없음

기본

없음

chdir

설명

디렉터리 Ceph 데몬이 가동되어 실행되면 변경됩니다. 기본 / 디렉터리가 권장됩니다.

유형

문자열

필수 항목

없음

기본

/

max_open_files

설명

Red Hat Ceph Storage 클러스터가 시작되면 **Ceph**는 **OS** 수준에서 **max_open_fds** 를 설정합니다(즉, 최대 #의 파일 설명자임). **Ceph OSD**가 파일 설명자가 실행되지 않도록 하는 데 도움이 됩니다.

유형

64비트 정수

필수 항목

없음

기본

0

fatal_signal_handlers

설명

설정된 경우 **SEGV, ABRT, BUS, ILL, FPE, XCPU, XFSZ, SYS** 신호에 대한 신호 처리기를 설치하여 유용한 로그 메시지를 생성합니다.

유형

부울

기본

true

부록 B. CEPH 네트워크 구성 옵션

이는 Ceph의 일반적인 네트워크 구성 옵션입니다.

public_network

설명

공용(front-side) 네트워크의 IP 주소 및 넷마스크(예: 192.168.0.0/24)입니다. [global] 로 설정합니다. 쉼표로 구분된 서브넷을 지정할 수 있습니다.

유형

<ip-address>/<netmask> [, <ip-address>/<netmask>]

필수 항목

없음

기본

해당 없음

public_addr

설명

공용(전면) 네트워크의 IP 주소입니다. 각 데몬에 대해 설정합니다.

유형

IP 주소

필수 항목

없음

기본

해당 없음

cluster_network

설명

*클러스터 네트워크의 **IP** 주소 및 넷마스크(예: **10.0.0.0/24**). **[global]** 로 설정합니다. 쉼표로 구분된 서브넷을 지정할 수 있습니다.*

유형

<ip-address>/<netmask> [, <ip-address>/<netmask>]

필수 항목

없음

기본

해당 없음

cluster_addr

설명

*클러스터 네트워크의 **IP** 주소입니다. 각 데몬에 대해 설정합니다.*

유형

address

필수 항목

없음

기본

해당 없음

ms_type

설명

*네트워크 전송 계층의 **Messageenger** 유형입니다. **Red Hat**은 **posix** 의미 체계를 사용하여 간단하고 동기화된 워커 유형을 지원합니다.*

유형

문자열.

필수 항목

아니요.

기본

async+posix

ms_public_type

설명

*공용 네트워크의 네트워크 전송 계층에 대한 메시징 유형입니다. **ms_type** 과 동일하게 작동하지만 공용 또는 전면 네트워크에만 적용됩니다. 이 설정을 사용하면 **Ceph**에서 공용 또는 프런트 엔드 및 클러스터 또는 백 측 네트워크에 다른 음성 유형을 사용할 수 있습니다.*

유형

문자열.

필수 항목

아니요.

기본

없음.

ms_cluster_type

설명

*클러스터 네트워크의 네트워크 전송 계층에 대한 수신자 유형입니다. **ms_type** 과 동일하게 작동하지만 클러스터 또는 백엔드 네트워크에만 적용됩니다. 이 설정을 사용하면 **Ceph**에서 공용 또는 프런트 엔드 및 클러스터 또는 백 측 네트워크에 다른 음성 유형을 사용할 수 있습니다.*

유형

문자열.

필수 항목

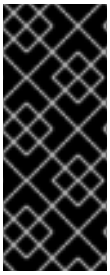
아니요.

기본

없음.

호스트 옵션

선언된 각 모니터 아래에 **mon addr** 설정을 사용하여 **Ceph** 구성 파일에서 하나 이상의 **Ceph Monitor** 를 선언해야 합니다. **Ceph**는 **Ceph** 구성 파일에서 선언된 각 모니터, 메타데이터 서버 및 **OSD** 아래에 호스트 설정이 필요합니다.



중요

localhost 를 사용하지 마십시오. **FQDN**(정규화된 도메인 이름)이 아닌 노드의 짧은 이름을 사용합니다. 노드 이름을 검색하는 타사 배포 시스템을 사용할 때 호스트에 대한 값을 지정하지 마십시오.

mon_addr

설명

클라이언트가 **Ceph** 모니터에 연결하는 데 사용할 수 있는 **< hostname>:<port >** 항목 목록입니다. 설정되지 않은 경우 **Ceph**는 **[mon.*]** 섹션을 검색합니다.

유형

문자열

필수 항목

없음

기본

해당 없음

host

설명

호스트 이름입니다. 특정 데몬 인스턴스에 이 설정을 사용합니다(예: **[osd.0]**).

유형

문자열

필수 항목

예, 데몬 인스턴스의 경우입니다.

기본

localhost

TCP 옵션

Ceph는 기본적으로 **TCP** 버퍼링을 비활성화합니다.

ms_tcp_nodelay

설명

Ceph는 **ms_tcp_nodelay** 를 활성화하여 각 요청이 즉시 전송됩니다(**Buffering** 없음). 나글의 알고리즘을 비활성화하면 네트워크 트래픽이 증가하여 혼잡이 발생할 수 있습니다. 다수의 작은 패킷이 발생하는 경우 **ms_tcp_nodelay** 를 비활성화하려고 시도할 수 있지만 비활성화하면 일반적으로 대기 시간이 증가합니다.

유형

부울

필수 항목

없음

기본

true

ms_tcp_rcvbuf

설명

네트워크 연결 수신 끝에 있는 소켓 버퍼의 크기입니다. 기본적으로 비활성되어 있습니다.

유형

32비트 정수

필수 항목

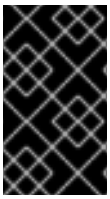
없음

기본

0

바인딩 옵션

bind 옵션은 Ceph OSD 데몬의 기본 포트 범위를 구성합니다. 기본 범위는 6800:7100 입니다. Ceph 데몬이 IPv6 주소에 바인딩하도록 활성화할 수도 있습니다.



중요

방화벽 구성에서 구성된 포트 범위를 사용할 수 있는지 확인합니다.

ms_bind_port_min

설명

OSD 데몬이 바인딩할 최소 포트 번호입니다.

유형

32비트 정수

기본

6800

필수 항목

없음

ms_bind_ipv6

설명

Ceph 데몬이 **IPv6** 주소에 바인딩할 수 있습니다.

유형

부울

기본

false

필수 항목

없음

비동기 메시징 옵션

Ceph Messageenger 옵션은 **AsyncMessenger** 의 동작을 구성합니다.

ms_async_op_threads

설명

각 **AsyncMessenger** 인스턴스에서 사용하는 초기 작업자 스레드 수입니다. 이 구성 설정 은 복제본 수 또는 삭제 코드 청크 수와 같아야 하지만 **CPU** 코어 수가 낮거나 단일 서버의 **OSD** 수가 높은 경우 더 낮을 수 있습니다.

유형

64비트 서명되지 않은 정수

필수 항목

없음

기본

3

연결 모드 구성 옵션

대부분의 연결의 경우 암호화 및 압축에 사용되는 모드를 제어하는 옵션이 있습니다.

ms_cluster_mode

설명

Ceph *데몬 간 클러스터 내 통신에 사용되는 연결 모드입니다. 여러 모드가 나열된 경우 먼저 나열된 모드가 우선합니다.*

유형

문자열

기본

CRC *보안*

ms_service_mode

설명

스토리지 클러스터에 연결할 때 클라이언트가 사용할 수 있는 모드 목록입니다.

유형

문자열

기본

CRC *보안*

ms_client_mode

설명

Ceph *클러스터와 상호 작용할 때 클라이언트가 사용할 연결 모드의 기본 설정 목록입니다.*

유형

문자열

기본

CRC *보안*

ms_mon_cluster_mode

설명

Ceph 모니터 간에 사용할 연결 모드입니다.

유형

문자열

기본

Secure crc

ms_mon_service_mode

설명

모니터에 연결할 때 사용할 클라이언트 또는 기타 **Ceph** 데몬에 허용된 모드 목록입니다.

유형

문자열

기본

Secure crc

ms_mon_client_mode

설명

Ceph 모니터에 연결할 때 사용할 클라이언트 또는 비모니터 데몬의 우선 순위로 연결 모드 목록입니다.

유형

문자열

기본

Secure crc

압축 모드 구성 옵션

Messageenger v2 프로토콜을 사용하면 압축 모드에 대한 구성 옵션을 사용할 수 있습니다.

ms_compress_secure

설명

*암호화를 압축과 결합하면 피어 간 메시지의 보안 수준이 줄어듭니다. 암호화와 압축이 모두 활성화되어 있는 경우 압축 설정이 무시되고 메시지는 압축되지 않습니다. 이 설정을 옵션으로 재정의합니다. **AsyncMessenger** 스레드에서 메시지를 큐에 보내고 보내는 대신 해당 스레드에서 메시지를 직접 보냅니다. 이 옵션은 많은 **CPU** 코어가 있는 시스템의 성능을 저하시켜서 기본적으로 비활성화되어 있습니다.*

유형

부울

기본

false

ms_osd_compress_mode

설명

***Ceph OSD**와의 통신에 사용할 압축 정책입니다.*

유형

문자열

기본

none

유효한 선택 사항

none** 또는 **force

ms_osd_compress_min_size

설명

유선 압축이 가능한 최소 메시지 크기입니다.

유형

정수

기본

1 Ki

ms_osd_compression_algorithm

설명

*기본 설정으로 **OSD**를 사용하는 연결의 압축 알고리즘*

유형

문자열

기본

snappy

유효한 선택 사항

snappy,zstd,zlib** 또는 **lz4

부록 C. CEPH MONITOR 구성 옵션

다음은 배포 중에 설정할 수 있는 Ceph 모니터 구성 옵션입니다.

ceph config set mon CONFIGURATION_OPTION VALUE 명령을 사용하여 이러한 구성 옵션을 설정할 수 있습니다.

구성 옵션	설명	유형	기본
mon_force_quorum_join	이전에 맵에서 제거된 경우에도 모니터가 쿼럼에 참여하도록 합니다.	부울	**False**
mon_dns_srv_name	모니터 호스트/호스트의 DNS를 쿼리하는 데 사용되는 서비스 이름입니다.	문자열	**ceph-mon**
fsid	클러스터 ID입니다. 클러스터당 하나씩.	UUID	해당 없음. 지정하지 않는 경우 배포 툴에 의해 생성될 수 있습니다.
mon_data_size_warn	모니터의 데이터 저장소가 이 임계값에 도달하면 Ceph에서 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다. 기본값은 15GB입니다.	정수	**15*1024*1024*1024***
mon_data_avail_warn	모니터 데이터 저장소의 사용 가능한 디스크 공간이 이 백분율보다 작거나 같으면 Ceph에서 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다.	정수	**30**
mon_data_avail_crit	모니터 데이터 저장소의 사용 가능한 디스크 공간이 이 백분율보다 낮거나 같을 때 Ceph에서 클러스터 로그에서 **HEALTH_ERR** 상태를 발행합니다.	정수	**5**
mon_warn_on_cache_pools_without_hit_sets	캐시 풀에 **hit_set_type** 매개변수가 설정되지 않은 경우 Ceph에서 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다.	부울	True
mon_warn_on_crush_straw_calc_version_zero	CRUSH의 **straw_calc_version** 이 0이면 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다. 자세한 내용은 CRUSH 튜닝 가능 항목을 참조하십시오.	부울	True

구성 옵션	설명	유형	기본
mon_warn_on_legacy_crush_tunables	CRUSH 튜닝 가능 항목이 너무 오래된 경우(**mon_min_crush_required_version**) Ceph에서 **HEALTH_WARN** 상태를 발행합니다.	부울	True
mon_crush_min_required_version	이 설정은 클러스터에 필요한 최소 튜닝 가능 프로필 버전을 정의합니다.	문자열	**Hammer**
mon_warn_on_osd_down_out_interval_zero	**noout** 플래그가 설정될 때와 유사한 방식으로 동작하기 때문에, **mon_osd_down_out_interval** 설정이 0인 경우 Ceph는 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다. 관리자는 noout 플래그를 설정하여 클러스터의 문제를 보다 쉽게 해결할 수 있습니다. Ceph에서 관리자가 설정이 0임을 알 수 있도록 경고를 발행합니다.	부울	True
mon_health_to_clog	이 설정을 사용하면 Ceph에서 정기적으로 상태 요약을 클러스터 로그에 보낼 수 있습니다.	부울	True
mon_health_detail_to_clog	이 설정을 사용하면 Ceph에서 상태 세부 정보를 주기적으로 클러스터 로그에 보낼 수 있습니다.	부울	True
mon_op_complaint_time	업데이트없이 Ceph Monitor 작업이 차단된 것으로 간주되는 시간(초)입니다.	정수	**30**
mon_health_to_clog_tick_interval	모니터에서 클러스터 로그에 상태 요약을 보내는 빈도(초)입니다. 양수가 아닌 숫자가 비활성화됩니다. 현재 상태 요약이 비어 있거나 마지막으로 동일한 경우 모니터는 클러스터 로그로 상태를 보내지 않습니다.	플로트	**60.000000**
mon_health_to_clog_interval	모니터에서 클러스터 로그에 상태 요약을 보내는 빈도(초)입니다. 양수가 아닌 숫자가 비활성화됩니다. 모니터는 항상 요약을 클러스터 로그에 보냅니다.	정수	**600**
mon_sync_timeout	모니터에서 동기화 공급자의 다음 업데이트 메시지를 포기하고 부트스트랩하기 전에 대기하는 시간(초)입니다.	double	**60.000000**

구성 옵션	설명	유형	기본
mon_sync_max_payload_size	동기화 페이로드의 최대 크기(바이트)입니다.	32비트 정수	**1045676**
paxos_max_join_drift	모니터 데이터 저장소를 먼저 동기화하기 전에 최대 Paxos 반복입니다. 모니터가 피어가 너무 앞서 있음을 발견하면 계속하기 전에 먼저 데이터 저장소와 동기화됩니다.	정수	**10**
paxos_stash_full_interval	PaxosService 상태의 전체 사본을 중단하는 빈도(커밋 중)입니다. 현재 이 설정은 **mds**,**mon**,**auth** 및 **mgr** PaxosServices에만 영향을 미칩니다.	정수	**25**
paxos_propose_interval	맵 업데이트를 제안하기 전에 이 시간 간격에 대한 업데이트를 수집합니다.	double	**1.0**
paxos_min	유지할 최소 paxos 상태 수	정수	**500**
paxos_min_wait	일정이 비활성화된 후 업데이트를 수집하는 최소 시간입니다.	double	**0.05**
paxos_trim_min	트리밍 전에 허용되는 추가 제안 수	정수	**250**
paxos_trim_max	한 번에 트리밍할 최대 추가 제안 수	정수	**500**
paxos_service_trim_min	트리플을 트리거할 최소 버전 양(0이 비활성화)	정수	250
paxos_service_trim_max	단일 제안 중에 트리밍할 최대 버전 양(0이 비활성화)	정수	500
mon_mds_force_trim_to	모니터가 mdsmaps를 이 시점으로 트리밍하도록 강제 적용합니다(0이 비활성화됨). 위험, 주의와 함께 사용)	정수	0
mon_osd_force_trim_to	지정된 epoch에서 정리되지 않은 PG가 있는 경우에도 이 시점에 osdmaps를 트리밍하도록 합니다(0이 비활성화됨. 위험, 주의와 함께 사용 가능)	정수	0
mon_osd_cache_size	기본 저장소 캐시를 사용하지 않는 osdmaps 캐시의 크기	정수	500
mon_election_timeout	선택 제안에서 초 단위로 모든 ACK에 대한 최대 대기 시간(초)입니다.	플로트	**5**

구성 옵션	설명	유형	기본
mon_lease	모니터 버전의 리스 길이(초)입니다.	플로트	**5**
mon_lease_renew_interval_factor	**Mon lease * 리스 갱신 간격 요소는** 리더가 다른 모니터의 리스를 갱신하는 간격이 됩니다. 인수는 **1.0** 보다 작아야 합니다.	플로트	**0.6**
mon_lease_ack_timeout_factor	리더는 공급자가 리스 확장을 승인할 수 있도록 **mon lease * mon lease ack 시간 초과 요소를** 기다립니다.	플로트	**2.0**
mon_min_osdmap_epochs	항상 유지할 최소 OSD 맵 수입니다.	32비트 정수	**500**
mon_max_log_epochs	모니터에서 유지해야 하는 최대 로그 수입니다.	32비트 정수	**500**
mon_tick_interval	모니터의 눈금 간격(초)입니다.	32비트 정수	**5**
mon_clock_drift_allowed	모니터 간에 허용되는 클럭 드리프트(초)입니다.	플로트	**.050**
mon_clock_drift_warn_backoff	클럭 드리프트 경고에 대한 기하급수적 백오프입니다.	플로트	**5**
mon_timecheck_interval	리더의 시간 점검 간격(clock 드리프트 검사)입니다.	플로트	**300.0**
mon_timecheck_skew_interval	리더에게 스큐가 있는 경우 시간 검사 간격(clock 드리프트 검사)(초)입니다.	플로트	**30.0**
mon_max_osd	클러스터에서 허용되는 최대 OSD 수입니다.	32비트 정수	**10000**
mon_globalid_prealloc	클러스터의 클라이언트 및 데몬에 대해 사전 할당 가능한 글로벌 ID 수입니다.	32비트 정수	**10000**
mon_subscribe_interval	서브스크립션의 새로 고침 간격(초)입니다. 서브스크립션 메커니즘을 사용하면 클러스터 맵 및 로그 정보를 가져올 수 있습니다.	double	**86400.000000**
mon_stat_smooth_intervals	Ceph는 마지막 **N** PG 맵에 대한 원활한 통계를 제공합니다.	정수	**6**

구성 옵션	설명	유형	기본
mon_probe_timeout	모니터가 부트 스트랩하기 전에 피어를 찾을 때까지 대기하는 시간(초)입니다.	double	**2.0**
mon_daemon_bytes	메타데이터 서버 및 OSD 메시지의 메시지 메모리 제한(바이트)입니다.	64비트 정수 서명되지 않음	**400UL << 20**
mon_max_log_entries_per_event	이벤트당 최대 로그 항목 수입니다.	정수	**4096**
mon_osd_prime_pg_temp	OSD가 클러스터로 다시 돌아올 때 이전 OSD를 사용하여 PGMap의 우선 순위를 활성화하거나 비활성화합니다. **실제** 설정을 사용하면 클라이언트는 해당 PG가 피어링된 OSD에서 새로 표시될 때까지 이전 OSD를 계속 사용합니다.	부울	**true**
mon_osd_prime_pg_temp_max_time	OSD가 클러스터로 다시 돌아올 때 모니터가 PGMap의 우선 순위를 정하는 데 걸리는 시간(초)입니다.	플로트	**0.5**
mon_lease_ack_timeout_factor	리더는 공급자가 리스 확장을 승인할 수 있도록 **mon lease * mon lease ack 시간 초과 요소를** 기다립니다.	플로트	**2.0**
mon_accept_timeout_factor	리더는 요청자가 Paxos 업데이트를 수락할 때까지 **mon lease** * **mon accept timeout factor** 를 기다립니다. 또한 비슷한 목적으로 Paxos 복구 단계에서도 사용됩니다.	플로트	**2.0**
mon_min_osdmap_epochs	항상 유지할 최소 OSD 맵 수입니다.	32비트 정수	**500**
mon_max_pgmap_epochs	모니터에서 유지해야 하는 최대 PG map 수입니다.	32비트 정수	**500**
mon_max_log_epochs	모니터에서 유지해야 하는 최대 로그 수입니다.	32비트 정수	**500**
clock_offset	시스템 시계를 얼마나 오프셋할 수 있습니까. 자세한 내용은 **Clock.cc** 를 참조하십시오.	double	**0**

구성 옵션	설명	유형	기본
mon_tick_interval	모니터의 눈금 간격(초)입니다.	32비트 정수	**5**
mon_clock_drift_allowed	모니터 간에 허용되는 클럭 드리프트(초)입니다.	플로트	**.050**
mon_clock_drift_warn_backoff	클럭 드리프트 경고에 대한 기하급수적 백오프입니다.	플로트	**5**
mon_timecheck_interval	리더의 시간 점검 간격(clock 드리프트 검사)입니다.	플로트	**300.0**
mon_timecheck_skew_interval	리더에게 스큐가 있는 경우 시간 검사 간격(clock 드리프트 검사)(초)입니다.	플로트	**30.0**
mon_max_osd	클러스터에서 허용되는 최대 OSD 수입니다.	32비트 정수	**10000**
mon_globalid_prealloc	클러스터의 클라이언트 및 데몬에 대해 사전 할당 가능한 글로벌 ID 수입니다.	32비트 정수	**10000**
mon_sync_fs_threshold	지정된 수의 오브젝트를 작성할 때 파일 시스템과 동기화합니다. 이를 비활성화하려면 **0** 으로 설정합니다.	32비트 정수	**5**
mon_subscribe_interval	서브스크립션의 새로 고침 간격(초)입니다. 서브스크립션 메커니즘을 사용하면 클러스터 맵 및 로그 정보를 가져올 수 있습니다.	double	**86400.000000**
mon_stat_smooth_intervals	Ceph는 마지막 **N** PG 맵에 대한 원활한 통계를 제공합니다.	정수	**6**
mon_probe_timeout	모니터가 부트 스트랩하기 전에 피어를 찾을 때까지 대기하는 시간(초)입니다.	double	**2.0**
mon_daemon_bytes	메타데이터 서버 및 OSD 메시지의 메시지 메모리 제한(바이트)입니다.	64비트 정수 서명되지 않음	**400UL << 20**
mon_max_log_entries_per_event	이벤트당 최대 로그 항목 수입니다.	정수	**4096**

구성 옵션	설명	유형	기본
mon_osd_prime_pg_temp	OSD가 클러스터로 다시 돌아올 때 이전 OSD를 사용하여 PGMap의 우선 순위를 활성화하거나 비활성화합니다. **실제** 설정을 사용하면 클라이언트는 해당 PG가 피어링된 OSD에서 새로 표시될 때까지 이전 OSD를 계속 사용합니다.	부울	**true**
mon_osd_prime_pg_temp_max_time	OSD가 클러스터로 다시 돌아올 때 모니터가 PGMap의 우선 순위를 정하는 데 걸리는 시간(초)입니다.	플로트	**0.5**
mon_mds_skip_sanity	FSMap에 대한 안전 어설션을 건너뛰십시오. 우리가 어쨌든 계속하려는 버그의 경우. FSMap sanity 검사가 실패하면 모니터가 종료되지만 이 옵션을 활성화하여 비활성화할 수 있습니다.	부울	False
mon_max_mdsmap_epochs	단일 제안 중에 트리밍할 mdsmap epoch의 최대 양입니다.	정수	500
mon_config_key_max_entry_size	config-key 항목의 최대 크기(바이트)입니다.	정수	65536
mon_warn_pg_not_scrubbed_ratio	경고할 scrub max 간격 이후의 scrub max 간격의 백분율입니다.	플로트	0.5
mon_warn_pg_not_deep_scrubbed_ratio	경고할 깊은 스크러브 간격의 백분율입니다.	플로트	0.75
mon_scrub_interval	저장된 체크섬과 저장된 체크섬을 저장된 모든 키의 계산된 체크섬과 비교하여 모니터가 저장소를 스크럽하는 빈도(초)입니다.	정수	3600*24
mon_scrub_timeout	mon 쿼럼 참가자의 scrub를 다시 시작하는 시간 초과는 최신 청크에 응답하지 않습니다.	정수	5분
mon_scrub_max_keys	매번 스크럽할 수 있는 최대 키 수입니다.	정수	100
mon_scrub_inject_missing_keys	mon scrub에 누락된 키를 삽입할 확률입니다.	플로트	0

구성 옵션	설명	유형	기본
mon_compact_on_start	**ceph-mon** start에서 Ceph Monitor 저장소로 사용되는 데이터베이스를 압축합니다. 수동 압축은 정기적인 압축이 작동하지 않는 경우 모니터 데이터베이스를 축소하고 성능을 개선하는 데 도움이 됩니다.	부울	False
mon_compact_on_bootstrap	부트스트랩에서 Ceph Monitor 저장소로 사용되는 데이터베이스를 압축합니다. 모니터는 부트스트랩 후 쿼럼을 생성하기 위해 서로 검사를 시작합니다. 쿼럼에 가입하기 전에 시간이 초과되면 처음부터 다시 시작하고 다시 부트스트랩합니다.	부울	False
mon_compact_on_trim	이전 상태를 트리링할 때 특정 접두사(Paxos 포함)를 압축합니다.	부울	True
mon_osd_mapping_pgs_per_chunk	청크에서 배치 그룹에서 OSD로의 매핑을 계산합니다. 이 옵션은 청크당 배치 그룹 수를 지정합니다.	정수	4096
rados_mon_op_timeout	rados 작업에서 오류를 반환하기 전에 모니터에서 응답을 대기하는 시간(초)입니다. 0은 제한 시 또는 대기 시간이 없음을 의미합니다.	double	0

추가 리소스

- *풀 값*
- ***CRUSH** 튜닝 가능 항목*

부록 D. CEPHX 구성 옵션

다음은 배포 중에 설정할 수 있는 Cephx 구성 옵션입니다.

auth_cluster_required

설명

활성화된 경우 Red Hat Ceph Storage 클러스터 데몬, ceph-mon 및 ceph-osd 가 활성화된 경우 서로 인증해야 합니다. 유효한 설정은 cephx 또는 none 입니다.

유형

문자열

필수 항목

없음

기본

cephx.

auth_service_required

설명

활성화하면 Red Hat Ceph Storage 클러스터 데몬에서 Ceph 서비스에 액세스하려면 Ceph 클라이언트가 Red Hat Ceph Storage 클러스터로 인증해야 합니다. 유효한 설정은 cephx 또는 none 입니다.

유형

문자열

필수 항목

없음

기본

cephx.

auth_client_required

설명

*활성화하면 **Ceph** 클라이언트에 **Ceph** 클라이언트를 인증하기 위해 **Red Hat Ceph Storage** 클러스터가 필요합니다. 유효한 설정은 **cephx** 또는 **none** 입니다.*

유형

문자열

필수 항목

없음

기본

cephx.

키 링

설명

키 링 파일의 경로입니다.

유형

문자열

필수 항목

없음

기본

/etc/ceph/$cluster.$name.keyring,/etc/ceph/$cluster.keyring,/etc/ceph/keyring,/etc/ceph/keyring.bin

keyfile

설명

키 파일의 경로(즉, 키만 포함된 파일).

유형

문자열

필수 항목

없음

기본

없음

key

설명

키(즉, 키 자체의 텍스트 문자열)입니다. 권장되지 않음.

유형

문자열

필수 항목

없음

기본

없음

ceph-mon

위치

$mon_data/keyring

capabilities

*'허용 *'*

ceph-osd

위치

$osd_data/keyring

capabilities

Mon 'allow profile osd' osd 'allow *'

radosgw

위치

$rgw_data/keyring

capabilities

Mon 'rwx 허용' osd 'allow rwx'

cephx_require_signatures

설명

true *로 설정하면* **Ceph**가 **Ceph** *클라이언트와* **Red Hat Ceph Storage** *클러스터 간의 모든 메시지 트래픽에 서명하고* **Red Hat Ceph Storage** *클러스터를 구성하는 데몬 간 서명이 필요합니다.*

유형

부울

필수 항목

없음

기본

false

cephx_cluster_require_signatures

설명

true *로 설정하면* **Ceph**가 **Red Hat Ceph Storage** *클러스터를 구성하는* **Ceph** *데몬 간의 모든 메시지 트래픽에 서명해야 합니다.*

유형

부울

필수 항목

없음

기본

false

cephx_service_require_signatures

설명

***true** 로 설정하면 **Ceph** 클라이언트와 **Red Hat Ceph Storage** 클러스터 간의 모든 메시지 트래픽에 서명이 필요합니다.*

유형

부울

필수 항목

없음

기본

false

cephx_sign_messages

설명

***Ceph** 버전이 메시지 서명을 지원하는 경우 **Ceph**는 모든 메시지를 서명하므로 스푸핑할 수 없습니다.*

유형

부울

기본

true

auth_service_ticket_ttl

설명

***Red Hat Ceph Storage** 클러스터가 **Ceph** 클라이언트에 인증 티켓을 보내면 클러스터에서*

실시간 티켓을 할당합니다.

유형

double

기본

60*60

부록 E. 풀, 배치 그룹 및 CRUSH 구성 옵션

풀, 배치 그룹 및 CRUSH 알고리즘을 관리하는 Ceph 옵션입니다.

구성 옵션	설명	유형	기본
mon_allow_pool_delete	모니터에서 풀을 삭제할 수 있습니다. RHCS 3 이상 릴리스에서는 데이터를 보호하기 위해 기본적으로 모니터에서 풀을 삭제할 수 없습니다.	부울	**false**
mon_max_pool_pg_num	풀당 최대 배치 그룹 수입니다.	정수	**65536**
mon_pg_create_interval	동일한 Ceph OSD 데몬에서 PG 생성 사이의 시간(초)입니다.	플로트	**30.0**
mon_pg_stuck_threshold	PG가 중단된 것으로 간주되는 시간(초)입니다.	32비트 정수	**300**
mon_pg_min_inactive	**mon_pg_stuck_threshold** 보다 비활성 상태인 PG 수가 이 설정을 초과하면 클러스터 로그에서 **HEALTH_ERR** 상태를 발행합니다. 기본 설정은 하나의 PG입니다. 양수가 아닌 경우 이 설정이 비활성화됩니다.	정수	**1**
mon_pg_warn_max_per_osd	클러스터에서 OSD당 평균 PG 수가 이 설정보다 크면 Ceph에서 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다. 양수가 아닌 경우 이 설정이 비활성화됩니다.	정수	**300**
mon_pg_warn_min_per_osd	클러스터에서 OSD당 평균 PG 수가 이 설정보다 작으면 Ceph에서 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다. 양수가 아닌 경우 이 설정이 비활성화됩니다.	정수	**30**
mon_pg_warn_min_objects	클러스터의 총 오브젝트 수가 이 수 미만인 경우 경고하지 마십시오.	정수	**1000**
mon_pg_warn_min_pool_objects	이 번호 아래의 오브젝트 번호가 있는 풀에 대해 경고하지 마십시오.	정수	**1000**

구성 옵션	설명	유형	기본
mon_pg_check_down_all_threshold	Ceph에서 모든 PG를 확인하여 문제가 없거나 오래되지 않았는지 확인하는 OSD의 임계값을 백분율로 설정합니다.	플로트	**0.5**
mon_pg_warn_max_object_skew	풀의 평균 오브젝트 수가 **mon pg warn max 오브젝트 skew** 보다 크면 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다. 양수가 아닌 경우 이 설정이 비활성화됩니다.	플로트	**10**
mon_delta_reset_interval	Ceph가 PG delta를 0으로 재설정하기 전의 비활성 시간(초)입니다. Ceph는 각 풀에 사용된 공간을 추적하여 관리자가 복구 및 성능 진행 상황을 평가할 수 있도록 지원합니다.	정수	**10**
mon_osd_max_op_age	**HEALTH_WARN** 상태를 발행하기 전에 작업이 완료될 수 있는 최대 기간(초)입니다.	플로트	**32.0**
osd_pg_bits	Ceph OSD 데몬당 배치 그룹 비트.	32비트 정수	**6**
osd_pgp_bits	배치 용도(PGP)를 위한 배치 그룹에 대한 Ceph OSD 데몬당 비트 수입니다.	32비트 정수	**6**
osd_crush_chooseleaf_type	CRUSH 규칙에서 **chooseleaf** 에 사용할 버킷 유형입니다. 이름이 아닌 ordinal 순위를 사용합니다.	32비트 정수	**1**. 일반적으로 하나 이상의 Ceph OSD 데몬을 포함하는 호스트입니다.
osd_pool_default_crush_replicated_ruleset	복제된 풀을 생성할 때 사용할 기본 CRUSH 규칙 세트입니다.	8비트 정수	**0**
osd_pool_erasure_code_stripe_unit	코딩된 풀의 오브젝트 스트라이프 청크의 기본 크기(바이트)를 설정합니다. 크기 S의 모든 개체는 N 스트라이프로 저장되며 각 데이터 청크는 **스트라이프 단위** 바이트를 수신합니다. **N * 스트라이프 단위 바이트의 각 스트라이프** 는 개별적으로 인코딩/디코딩됩니다. 이 옵션은 삭제 코드 프로필의 **stripe_unit** 설정으로 재정의할 수 있습니다.	서명되지 않은 32비트 정수	**4096**

구성 옵션	설명	유형	기본
osd_pool_default_size	풀에 있는 오브젝트의 복제본 수를 설정합니다. 기본값은 **ceph osd pool set {pool-name} size {size}** 와 동일합니다.	32비트 정수	**3**
osd_pool_default_min_size	클라이언트에 대한 쓰기 작업을 승인하기 위해 풀에 있는 오브젝트에 대한 최소 쓰기 복제본 수를 설정합니다. 최소가 충족되지 않으면 Ceph에서 클라이언트에 대한 쓰기를 승인하지 않습니다. 이 설정은 **degraded** 모드에서 작동할 때 최소 복제본 수를 보장합니다.	32비트 정수	**0** 은 특정 최소값이 없음을 의미합니다. **0** 인 경우 최소**크기는 -(크기 / 2)** 입니다.
osd_pool_default_pg_num	풀의 기본 배치 그룹 수입니다. 기본값은 **mkpool** 인 **pg_num** 과 동일합니다.	32비트 정수	**32**
osd_pool_default_pgp_num	풀에 배치를 위한 기본 배치 그룹 수입니다. 기본값은 **mkpool** 인 **pgp_num** 과 동일합니다. PG와 PGP는 동일해야 합니다.	32비트 정수	**0**
osd_pool_default_flags	새 풀의 기본 플래그입니다.	32비트 정수	**0**
osd_max_pgls	나열할 최대 배치 그룹 수입니다. 많은 수를 요청하는 클라이언트는 Ceph OSD 데몬을 연결할 수 있습니다.	서명되지 않은 64비트 정수	**1024**
osd_min_pg_log_entries	로그 파일을 트리밍할 때 유지 관리할 최소 배치 그룹 로그 수입니다.	32비트 Int 서명되지 않음	**250**
osd_default_data_pool_replay_window	OSD에서 클라이언트가 요청을 재생할 때까지 대기하는 시간(초)입니다.	32비트 정수	**45**

부록 F. OSD(오브젝트 스토리지 데몬) 구성 옵션

다음은 배포 중에 설정할 수 있는 Ceph OSD(오브젝트 스토리지 데몬) 구성 옵션입니다.

ceph config set osd CONFIGURATION_OPTION VALUE 명령을 사용하여 이러한 구성 옵션을 설정할 수 있습니다.

osd_uuid

설명

Ceph OSD의 UUID(Universally unique identifier)입니다.

유형

UUID

기본

UUID입니다.

참고

osd uuid 는 단일 Ceph OSD에 적용됩니다. fsid 는 전체 클러스터에 적용됩니다.

osd_data

설명

OSD 데이터의 경로입니다. Ceph를 배포할 때 디렉터리를 생성해야 합니다. 이 마운트 지점에 OSD 데이터의 드라이브를 마운트합니다.

> **IMPORTANT:** Red Hat does not recommend changing the default.

유형

문자열

기본

/var/lib/ceph/osd/$cluster-$id

osd_max_write_size

설명

쓰기 크기(MB)의 최대 크기입니다.

유형

32비트 정수

기본

90

osd_client_message_size_cap

설명

메모리에서 허용되는 가장 큰 클라이언트 데이터 메시지입니다.

유형

64비트 정수 서명되지 않음

기본

*500MB 기본값. 500*1024L*1024L*

osd_class_dir

설명

RADOS 클래스 플러그인의 클래스 경로입니다.

유형

문자열

기본

$libdir/rados-classes

osd_max_scrubs

설명

Ceph OSD의 최대 동시 scrub 작업 수입니다.

유형

32비트 Int

기본

1

osd_scrub_thread_timeout

설명

scrub 스레드를 타이밍하기 전의 최대 시간(초)입니다.

유형

32비트 정수

기본

60

osd_scrub_finalize_thread_timeout

설명

scrub가 스레드를 종료하기 전의 최대 시간(초)입니다.

유형

32비트 정수

기본

*60*10*

osd_scrub_begin_hour

설명

이렇게 하면 스크럽이 하루 또는 그 이후 시간으로 제한됩니다. osd_scrub_begin_hour = 0 및 osd_scrub_end_hour = 0 을 사용하여 하루 전체의 스크럽을 허용합니다.
osd_scrub_end_hour 과 함께 스크러블이 발생할 수 있는 시간 창을 정의합니다. 그러나 배치 그

*룹의 **scrub** 간격이 **osd_scrub_max_interval** 을 초과하는 한 시간 창 허용 여부에 관계없이 **scrub**가 수행됩니다.*

유형

정수

기본

0

허용되는 범위

[0,23]

osd_scrub_end_hour

설명

*이렇게 하면 스크러블링이 이 시간 이전 시간으로 제한됩니다. **osd_scrub_begin_hour = 0** 및 **osd_scrub_end_hour = 0** 을 사용하여 하루 동안 스크럽을 허용합니다.*
***osd_scrub_begin_hour** 과 함께 스크럽이 발생할 수 있는 시간 창을 정의합니다. 그러나 배치 그룹의 **scrub** 간격이 **osd_scrub_max_interval** 을 초과하는 한 시간 창 허용 여부에 관계없이 **scrub**가 수행됩니다.*

유형

정수

기본

0

허용되는 범위

[0,23]

osd_scrub_load_threshold

설명

*최대 로드입니다. 시스템 로드(**getloadavg()** 함수에 의해 정의됨)가 이 수보다 높으면 **Ceph** 는 스크럽하지 않습니다. 기본값은 **0.5** 입니다.*

유형

플로트

기본

0.5

osd_scrub_min_interval

설명

***Red Hat Ceph Storage** 클러스터 로드가 낮은 경우 **Ceph OSD**를 스크럽하는 최소 간격(초)입니다.*

유형

플로트

기본

*하루에 한 번. **60*60*24***

osd_scrub_max_interval

설명

*클러스터 로드와 관계없이 **Ceph OSD**를 스크럽하는 최대 간격(초)입니다.*

유형

플로트

기본

*일주일에 한 번. **7*60*60*24***

osd_scrub_interval_randomize_ratio

설명

***osd scrub min** 간격과 **osd scrub max** 간격 사이의 예약된 **srub** 비율을 무작위로 사용합니다.*

유형

플로트

기본

0.5.

mon_warn_not_scrubbed

설명

***osd_scrub_interval** 에서 스크럽되지 않은 **PG**에 대해 경고하는 시간(초)입니다.*

유형

정수

기본

***0** (경고 없음).*

osd_scrub_chunk_min

설명

*오브젝트 저장소는 해시 경계에서 끝나는 청크로 분할됩니다. 청크의 경우 **Ceph**는 해당 청크에 대해 쓰기가 차단된 상태에서 한 번에 하나의 청크로 오브젝트를 스크럽합니다. **osd scrub chunk min** 설정은 **scrub**할 최소 청크 수를 나타냅니다.*

유형

***32**비트 정수*

기본

5

osd_scrub_chunk_max

설명

***scrub**를 최대 청크 수입니다.*

유형

32비트 정수

기본

25

osd_scrub_sleep

설명

딥 스크럽 작업 사이에 잠드는 시간.

유형

플로트

기본

0 (또는 off).

osd_scrub_during_recovery

설명

복구 중에 스크럽을 허용합니다.

유형

bool

기본

false

osd_scrub_invalid_stats

설명

추가 스크러브가 유효하지 않은 것으로 표시된 통계를 수정하도록 강제 시행합니다.

유형

bool

기본

true

osd_scrub_priority

설명

scrub 작업과 클라이언트 I/O의 대기열 우선 순위를 제어합니다.

유형

서명되지 않은 32비트 정수

기본

5

osd_requested_scrub_priority

설명

작업 대기열에서 사용자가 요청한 scrub의 우선순위 세트입니다. 이 값이 osd_client_op_priority 보다 작으면 scrub가 클라이언트 작업을 차단하는 경우 osd_client_op_priority 값을 높일 수 있습니다.

유형

서명되지 않은 32비트 정수

기본

120

osd_scrub_cost

설명

큐 스케줄링을 위해 스크럽 작업 비용(MB)입니다.

유형

서명되지 않은 32비트 정수

기본

52428800

osd_deep_scrub_interval

설명

모든 데이터를 완전히 읽고 있는 깊은 스크럽 간격입니다. ***osd scrub load threshold*** *매개변수는 이 설정에 영향을 미치지 않습니다.*

유형

플로트

기본

일주일에 한 번. ***60*60*24*7***

osd_deep_scrub_stride

설명

깊은 스크러브를 할 때 크기를 읽으십시오.

유형

32*비트 정수*

기본

512KB. 524288

mon_warn_not_deep_scrubbed

설명

osd_deep_scrub_interval *이후의 모든* ***PG****는 스크럽되지 않은* ***PG****에 대해 경고하는 시간(초)입니다.*

유형

정수

기본

0 (경고 없음)

osd_deep_scrub_randomize_ratio

설명

scrubs가 무작위로 깊은 스크러브가 되는 비율 (osd_deep_scrub_interval 이 통과되기 전에도)

유형

플로트

기본

0.15 또는 15 %

osd_deep_scrub_update_digest_min_age

설명

scrub가 전체 오브젝트 다이제스트를 업데이트하기 전에 몇 초 동안 오래된 오브젝트를 사용해야 합니까.

유형

정수

기본

7200 (120시간)

osd_deep_scrub_large_omap_object_key_threshold

설명

이 보다 더 많은 OMAP 키가 있는 오브젝트에 경고.

유형

정수

기본

200000

osd_deep_scrub_large_omap_object_value_sum_threshold

설명

이 보다 더 많은 OMAP 키 바이트가 있는 오브젝트에 경고합니다.

유형

정수

기본

1 G

osd_delete_sleep

설명

다음 제거 트랜잭션 전에 유휴 시간(초)입니다. 이 설정은 배치 그룹 삭제 프로세스를 제한합니다.

유형

플로트

기본

0.0

osd_delete_sleep_hdd

설명

HDD의 다음 제거 트랜잭션 전에 유휴 시간(초)입니다.

유형

플로트

기본

5.0

osd_delete_sleep_ssd

설명

***SSD**의 다음 제거 트랜잭션 전에 유휴 시간(초)입니다.*

유형

플로트

기본

1.0

osd_delete_sleep_hybrid

설명

Ceph OSD 데이터가 HDD 및 OSD 저널 또는 WAL 및 DB가 SSD에 있을 때 다음 제거 트랜잭션 전에 유휴 시간(초)입니다.

유형

플로트

기본

1.0

osd_op_num_shards

설명

*클라이언트 작업의 **shard** 수입니다.*

유형

***32**비트 정수*

기본

0

osd_op_num_threads_per_shard

설명

*클라이언트 작업의 **shard**당 스레드 수입니다.*

유형

32비트 정수

기본

0

osd_op_num_shards_hdd

설명

***HDD** 작업의 **shard** 수입니다.*

유형

32비트 정수

기본

5

osd_op_num_threads_per_shard_hdd

설명

***HDD** 작업의 **shard**당 스레드 수입니다.*

유형

32비트 정수

기본

1

osd_op_num_shards_ssd

설명

***SSD** 작업의 **shard** 수입니다.*

유형

32비트 정수

기본

8

osd_op_num_threads_per_shard_ssd

설명

***SSD** 작업의 **shard**당 스레드 수입니다.*

유형

32비트 정수

기본

2

osd_op_queue

설명

***Ceph OSD** 내에서 작업의 우선 순위를 지정하는 데 사용할 대기열 유형을 설정합니다. **OSD** 데몬을 다시 시작해야 합니다.*

유형

문자열

기본

wpq

유효한 선택 사항

wpq, mclock_scheduler, debug_random

중요

mClock OSD 스케줄러는 기술 프리뷰 기능 전용입니다. 기술 프리뷰 기능은 **Red Hat** 프로덕션 서비스 수준 계약(**SLA**)에서 지원되지 않으며 기능적으로 완전하지 않을 수 있으며 **Red Hat**은 해당 기능을 프로덕션용으로 사용하지 않는 것이 좋습니다. 이러한 기능을 사용하면 향후 제품 기능을 조기에 이용할 수 있어 개발 과정에서 고객이 기능을 테스트하고 피드백을 제공할 수 있습니다. 자세한 내용은 **Red Hat** 기술 프리뷰 기능에 대한 지원 범위를 참조하십시오.

osd_op_queue_cut_off

설명

엄격한 큐로 전송되고 일반 큐로 전송되는 우선 순위 작업을 선택합니다. **OSD** 데몬을 다시 시작해야 합니다.

낮은 설정은 모든 복제와 더 높은 작업을 엄격한 큐로 보내는 반면, 높은 옵션은 복제 확인 작업만 엄격한 큐에 보냅니다.

높은 설정은 클러스터의 일부 **Ceph OSD**가 사용량이 많은 경우 특히 **osd_op_queue** 설정에서 **wpq** 옵션과 결합할 때 유용합니다. 복제 트래픽을 매우 많이 처리하는 **Ceph OSD**는 이러한 **OSD**에서 기본 클라이언트 트래픽을 소모할 수 있습니다.

유형

문자열

기본

높음

유효한 선택 사항

low, high, debug_random

osd_client_op_priority

설명

클라이언트 작업에 대해 설정된 우선순위입니다. 이는 **osd** 복구 **op** 우선순위 를 기준으로 합니다.

유형

32비트 정수

기본

63

유효한 범위

1-63

osd_recovery_op_priority

설명

*복구 작업에 설정된 우선순위입니다. **osd** 클라이언트 **op priority** 를 기준으로 합니다.*

유형

32비트 정수

기본

3

유효한 범위

1-63

osd_op_thread_timeout

설명

Ceph OSD 작업 스레드 시간(초)입니다.

유형

32비트 정수

기본

15

osd_op_complaint_time

설명

지정된 시간(초)이 경과한 후 작업이 불만을 제기합니다.

유형

플로트

기본

30

osd_disk_threads

설명

*백그라운드 디스크 집약적 **OSD** 작업을 수행하는 데 사용되는 디스크 스레드 수(예: 스크러블링 및 스냅 트림)입니다.*

유형

***32**비트 정수*

기본

1

osd_op_history_size

설명

추적할 완료된 작업의 최대 수입니다.

유형

***32**비트 서명되지 않은 정수*

기본

20

osd_op_history_duration

설명

추적하기 위한 가장 오래된 완료된 작업입니다.

유형

32비트 서명되지 않은 정수

기본

600

osd_op_log_threshold

설명

한 번에 표시할 작업 로그 수입니다.

유형

32비트 정수

기본

5

osd_op_timeout

설명

OSD 작업을 실행하는 시간(초)입니다.

유형

정수

기본

0

중요

*클라이언트가 결과를 처리할 수 없는 한 **osd op** 시간 초과 옵션을 설정하지 마십시오. 예를 들어 가상 머신이 가상 머신에서 실행되는 클라이언트에 이 매개변수를 설정하면 가상 머신이 이 시간 초과를 하드웨어 오류로 해석하기 때문에 데이터가 손상될 수 있습니다.*

osd_max_backfills

설명

*단일 **OSD**에 또는 단일 **OSD**에서 허용되는 최대 백필 작업 수입니다.*

유형

***64**비트 서명되지 않은 정수*

기본

1

osd_backfill_scan_min

설명

백필 스캔당 최소 오브젝트 수입니다.

유형

***32**비트 정수*

기본

64

osd_backfill_scan_max

설명

백필 스캔당 최대 오브젝트 수입니다.

유형

***32**비트 정수*

기본

512

osd_backfill_full_ratio

설명

Ceph OSD의 전체 비율이 이 값보다 높은 경우 백필 요청을 거부합니다.

유형

플로트

기본

0.85

osd_backfill_retry_interval

설명

백필 요청을 다시 시도하기 전에 대기하는 시간(초)입니다.

유형

double

기본

30.000000

osd_map_dedup

설명

OSD 맵에서 중복 제거를 활성화합니다.

유형

부울

기본

true

osd_map_cache_size

설명

OSD 맵 캐시의 크기(MB)입니다.

유형

32비트 정수

기본

50

osd_map_cache_bl_size

설명

OSD 데몬의 메모리 내 OSD 맵 캐시의 크기입니다.

유형

32비트 정수

기본

50

osd_map_cache_bl_inc_size

설명

OSD 데몬에서 메모리 내 OSD 맵 캐시의 크기가 증가합니다.

유형

32비트 정수

기본

100

osd_map_message_max

설명

MOSDMap 메시지당 허용되는 최대 맵 항목입니다.

유형

32비트 정수

기본

40

osd_snap_trim_thread_timeout

설명

snap trim 스레드를 타이밍하기 전의 최대 시간(초)입니다.

유형

32비트 정수

기본

*60*60*1*

osd_pg_max_concurrent_snap_trims

설명

최대 병렬 스냅 트리/PG 수입니다. 이는 한 번에 트리밍할 PG당 오브젝트 수를 제어합니다.

유형

32비트 정수

기본

2

osd_snap_trim_sleep

설명

PG 문제를 처리하는 모든 트리 사이에 절전을 삽입합니다.

유형

32비트 정수

기본

0

osd_snap_trim_sleep_hdd

설명

HDD에 대한 다음 스냅샷 트리밍 전에 유휴 시간(초)입니다.

유형

플로트

기본

5.0

osd_snap_trim_sleep_ssd

설명

NVMe를 포함하여 SSD OSD의 다음 스냅샷 트리밍 작업 전에 유휴 시간(초)입니다.

유형

플로트

기본

0.0

osd_snap_trim_sleep_hybrid

설명

OSD 데이터가 HDD에 있고 OSD 저널 또는 WAL 및 DB가 SSD에 있을 때 다음 스냅샷 트리밍 작업 전에 유휴 시간(초)입니다.

유형

플로트

기본

2.0

osd_max_trimming_pgs

설명

최대 트리밍 PG 수

유형

32비트 정수

기본

2

osd_backlog_thread_timeout

설명

백로그 스레드를 타이밍하기 전의 최대 시간(초)입니다.

유형

32비트 정수

기본

*60*60*1*

osd_default_notify_timeout

설명

OSD 기본 알림 제한 시간(초)입니다.

유형

32비트 정수 서명되지 않음

기본

30

osd_check_for_log_corruption

설명

로그 파일에 손상이 있는지 확인합니다. 계산적으로 비용이 많이 들 수 있습니다.

유형

부울

기본

false

osd_remove_thread_timeout

설명

***OSD** 스레드를 삭제하는 데 걸리는 최대 시간(초)입니다.*

유형

***32**비트 정수*

기본

60*60

osd_command_thread_timeout

설명

명령 스레드를 타이밍하기 전의 최대 시간(초)입니다.

유형

***32**비트 정수*

기본

10*60

osd_command_max_records

설명

반환될 손실된 오브젝트 수를 제한합니다.

유형

32비트 정수

기본

256

osd_auto_upgrade_tmap

설명

이전 개체의 ***omap*** *에* ***tmap*** *을 사용합니다.*

유형

부울

기본

true

osd_tmapput_sets_users_tmap

설명

디버깅에만 ***tmap*** *을 사용합니다.*

유형

부울

기본

false

osd_preserve_trimmed_log

설명

트리밍된 로그 파일을 유지하지만 더 많은 디스크 공간을 사용합니다.

유형

부울

기본

false

osd_recovery_delay_start

설명

피어링이 완료되면 오브젝트 복구를 시작하기 전에 지정된 시간(초) 동안 **Ceph**가 지연됩니다.

유형

플로트

기본

0

osd_recovery_max_active

설명

한 번에 **OSD**당 활성 복구 요청 수입니다. 더 많은 요청으로 인해 복구 속도가 빨라지지만 요청이 클러스터에서 로드를 늘리게 됩니다.

유형

32비트 정수

기본

0

osd_recovery_max_active_hdd

설명

*기본 장치가 **HDD**인 경우 한 번에 **Ceph OSD**당 활성 복구 요청 수입니다.*

유형

정수

기본

3

osd_recovery_max_active_ssd

설명

*기본 장치가 **SSD**인 경우 한 번에 **Ceph OSD**당 활성 복구 요청 수입니다.*

유형

정수

기본

10

osd_recovery_sleep

설명

다음 복구 또는 백필 작업 전에 유휴 시간(초)입니다. 이 값을 늘리면 클라이언트 작업이 덜 영향을 미치는 동안 복구 작업이 느려집니다.

유형

플로트

기본

0.0

osd_recovery_sleep_hdd

설명

***HDD**의 다음 복구 또는 백필 작업 전에 유휴 시간(초)입니다.*

유형

플로트

기본

0.1

osd_recovery_sleep_ssd

설명

***SSD**의 다음 복구 또는 백필 작업 전에 유휴 시간(초)입니다.*

유형

플로트

기본

0.0

osd_recovery_sleep_hybrid

설명

***Ceph OSD** 데이터가 **HDD** 및 **OSD** 저널 또는 **WAL** 및 **DB**가 **SSD**에 있을 때 다음 복구 또는 백필 작업 전에 유휴 시간(초)입니다.*

유형

플로트

기본

0.025

osd_recovery_max_chunk

설명

푸시할 데이터 청크의 최대 크기입니다.

유형

64비트 정수 서명되지 않음

기본

8388608

osd_recovery_threads

설명

데이터 복구를 위한 스레드 수입니다.

유형

32비트 정수

기본

1

osd_recovery_thread_timeout

설명

복구 스레드를 타이밍하기 전의 최대 시간(초)입니다.

유형

32비트 정수

기본

30

osd_recover_clone_overlap

설명

*복구 중에 복제 중복을 유지합니다. 항상 **true** 로 설정해야 합니다.*

유형

부울

기본

true

rados_osd_op_timeout

설명

RADOS가 **RADOS** 작업에서 오류를 반환하기 전에 **OSD**의 응답을 기다리는 시간(초)입니다. 값이 **0**이면 제한이 없음을 의미합니다.

유형

double

기본

0

부록 G. CEPH MONITOR 및 OSD 구성 옵션

하트비트 설정을 수정할 때 Ceph 구성 파일의 [global] 섹션에 포함합니다.

mon_osd_min_up_ratio

설명

Ceph가 Ceph OSD 데몬 다운을 표시하기 전의 Ceph OSD 데몬의 최소 비율.

유형

double

기본

.3

mon_osd_min_in_ratio

설명

Ceph 의 Ceph OSD 데몬에서 Ceph OSD 데몬의 최소 비율 .

유형

double

기본

0.750000

mon_osd_laggy_halflife

설명

지연 시간(초) 이 감소합니다.

유형

정수

기본

60*60

mon_osd_laggy_weight

설명

지연 횟수 추정의 새 샘플의 가중치입니다.

유형

double

기본

0.3

mon_osd_laggy_max_interval

설명

지연 추정의 **delaygy_interval** 의 최대 값(초)입니다. 모니터는 **Adaptive** 접근 방식을 사용하여 특정 **OSD**의 **delaygy_interval** 을 평가합니다. 이 값은 해당 **OSD**의 유예 시간을 계산하는 데 사용됩니다.

유형

정수

기본

300

mon_osd_adjust_heartbeat_grace

설명

true 로 설정하면 **Ceph**는 지연 추정에 따라 크기가 조정 됩니다.

유형

부울

기본

true

mon_osd_adjust_down_out_interval

설명

true 로 설정하면 지연 추정에 따라 Ceph 가 확장됩니다.

유형

부울

기본

true

mon_osd_auto_mark_in

설명

Ceph는 부팅 Ceph OSD 데몬을 Ceph Storage 클러스터에서 로 표시합니다.

유형

부울

기본

false

mon_osd_auto_mark_auto_out_in

설명

Ceph는 클러스터에서 Ceph Storage Cluster 가 자동으로 표시되지 않는 Ceph OSD 데몬 부팅을 표시합니다.

유형

부울

기본

true

mon_osd_auto_mark_new_in

설명

Ceph는 새 Ceph OSD 데몬 부팅을 Ceph Storage 클러스터에서 로 표시합니다.

유형

부울

기본

true

mon_osd_down_out_interval

설명

Ceph가 응답하지 않는 경우 Ceph OSD 데몬을 중단하고 로그아웃 하기 전에 대기하는 시간(초)입니다.

유형

32비트 정수

기본

600

mon_osd_downout_subtree_limit

설명

Ceph가 자동으로 로그아웃 하는 가장 큰 CRUSH 단위 유형입니다.

유형

문자열

기본

rack

mon_osd_reporter_subtree_level

설명

*이 설정은 보고 **OSD**의 상위 **CRUSH** 단위 유형을 정의합니다. **OSD**는 응답하지 않는 피어를 찾는 경우 오류 보고서를 모니터에 보냅니다. 모니터는 보고된 **OSD**를 중단한 다음 유예 기간이 지나면 꺼질 수 있습니다.*

유형

문자열

기본

host

mon_osd_report_timeout

설명

*응답하지 않는 **Ceph OSD** 데몬을 선언하기 전의 유예 기간(초)입니다.*

유형

***32**비트 정수*

기본

900

mon_osd_min_down_reporters

설명

***Ceph OSD** 데몬을 보고하는 데 필요한 최소 **Ceph OSD** 데몬 수입니다.*

유형

***32**비트 정수*

기본

2

osd_heartbeat_address

설명

하트비트용 **Ceph OSD** 데몬의 네트워크 주소입니다.

유형

address

기본

호스트 주소입니다.

osd_heartbeat_interval

설명

Ceph OSD 데몬은 피어를 **ping**하는 빈도(초)입니다.

유형

32비트 정수

기본

6

osd_heartbeat_grace

설명

Ceph OSD 데몬에 **Ceph Storage** 클러스터에서 이를 중단 한다고 간주하는 하트비트가 표시되지 않은 경우의 경과 시간입니다.

유형

32비트 정수

기본

20

osd_mon_heartbeat_interval

설명

Ceph OSD 데몬 피어가 없는 경우 **Ceph OSD** 데몬 데몬이 **Ceph Monitor**를 **ping**하는 빈도입니다.

유형

32비트 정수

기본

30

osd_mon_report_interval_max

설명

Ceph OSD 데몬에서 Ceph 모니터에 보고하기 전에 대기할 수 있는 최대 시간(초)입니다.

유형

32비트 정수

기본

120

osd_mon_report_interval_min

설명

Ceph OSD 데몬은 시작 또는 보고 가능한 다른 이벤트에서 Ceph Monitor로 보고할 수 있는 최소 시간(초)입니다.

유형

32비트 정수

기본

5

유효한 범위

osd mon 보고서 간격 max보다 작아야 합니다.

osd_mon_ack_timeout

설명

***Ceph Monitor**가 통계 요청을 승인할 때까지 대기하는 시간(초)입니다.*

유형

***32**비트 정수*

기본

30

부록 H. CEPH 스크럽 옵션

Ceph는 배치 그룹을 스크럽하여 데이터 무결성을 보장합니다. 다음은 스크럽 작업을 늘리거나 줄이기 위해 조정할 수 있는 Ceph 스크럽 옵션입니다.

ceph config set global CONFIGURATION_OPTION VALUE 명령을 사용하여 이러한 구성 옵션을 설정할 수 있습니다.

mds_max_scrub_ops_in_progress

설명

병렬로 수행되는 최대 scrub 작업 수입니다. ceph config set mds_max_scrub_ops_in_progress VALUE 명령을 사용하여 이 값을 설정할 수 있습니다.

유형

integer

기본

5

osd_max_scrubs

설명

Ceph OSD 데몬의 최대 동시 스크러 작업 수입니다.

유형

integer

기본

1

osd_scrub_begin_hour

설명

스크럽이 시작되는 특정 시간입니다. osd_scrub_end_hour 과 함께 스크러브가 발생할 수 있는 시간 창을 정의할 수 있습니다. osd_scrub_begin_hour = 0 및 osd_scrub_end_hour = 0 을

사용하여 하루 전체의 스크럽을 허용합니다.

유형

integer

기본

0

허용되는 범위

[0, 23]

osd_scrub_end_hour

설명

스크럽이 종료되는 특정 시간입니다. ***osd_scrub_begin_hour*** *과 함께 스크럽이 발생할 수 있는 시간 창을 정의할 수 있습니다.* ***osd_scrub_begin_hour = 0*** *및* ***osd_scrub_end_hour = 0*** *을 사용하여 하루 동안 스크럽을 허용합니다.*

유형

integer

기본

0

허용되는 범위

[0, 23]

osd_scrub_begin_week_day

설명

스크럽이 시작되는 특정 날입니다. ***0 =*** *일요일,* ***1 =*** *월요일, 등.* ***"osd_scrub_end_week_day"****와 함께* ***scrubs****가 발생할 수 있는 시간 창을 정의할 수 있습니다.* ***osd_scrub_begin_week_day = 0*** *및* ***osd_scrub_end_week_day = 0*** *을 사용하여 주 전체의 스크럽을 허용합니다.*

유형

integer

기본

0

허용되는 범위

[0, 6]

osd_scrub_end_week_day

설명

*이는 스크럽이 종료되는 날을 정의합니다. **0** = 일요일, **1** = 월요일, 등.* ***osd_scrub_begin_week_day*** *와 함께 스크러블이 발생할 수 있는 시간 창을 정의합니다.* ***osd_scrub_begin_week_day = 0*** *및* ***osd_scrub_end_week_day = 0*** *을 사용하여 주 전체의 스크럽을 허용합니다.*

유형

integer

기본

0

허용되는 범위

[0, 6]

osd_scrub_during_recovery

설명

복구 중에 ***scrub****를 허용하십시오. 이 값을* ***false*** *로 설정하면 활성 복구가 있는 동안 새* ***scrub*** *및* ***deep-scrub*** *예약이 비활성화됩니다. 이미 실행 중인* ***scrubs****는 계속 실행되므로 사용 중인 스토리지 클러스터의 부하를 줄이는 데 유용합니다.*

유형

boolean

기본

false

osd_scrub_load_threshold

설명

*정규화된 최대 로드입니다. **getloadavg()**/ 온라인 **CPU** 수에 정의된 대로 시스템 로드가 이 정의된 수보다 높으면 스크러블링이 발생하지 않습니다.*

유형

플로트

기본

0.5

osd_scrub_min_interval

설명

Ceph 스토리지 클러스터 로드가 낮은 경우 Ceph OSD 데몬을 스크럽하는 최소 간격(초)입니다.

유형

플로트

기본

1일

osd_scrub_max_interval

설명

클러스터 로드와 관계없이 Ceph OSD 데몬을 스크럽하는 최대 간격(초)입니다.

유형

플로트

기본

7일

osd_scrub_chunk_min

설명

단일 작업 중에 스크러브할 최소 오브젝트 저장소 청크 수입니다. Ceph 블록은 스크러브 중에 단일 청크에 씁니다.

type

integer

기본

5

osd_scrub_chunk_max

설명

단일 작업 중에 스크러브할 최대 오브젝트 저장소 청크 수입니다.

type

integer

기본

25

osd_scrub_sleep

설명

다음 청크 그룹을 스크럽하기 전에 잠자는 시간입니다. 이 값을 늘리면 클라이언트 작업의 영향을 줄일 수 있도록 전체 스크럽 속도가 느려집니다.

type

플로트

기본

0.0

osd_scrub_extended_sleep

설명

스크럽 시간 또는 초를 스크럽하는 동안 지연을 삽입하는 기간입니다.

type

플로트

기본

0.0

osd_scrub_backoff_ratio

설명

*스케줄링 **scrubs**에 대한 백오프 비율입니다. 이것은 스크럽을 예약하지 않는 틱의 백분율이며 **66%**는 틱 **3**개 중 **1**개가 스크러브를 예약한다는 것을 의미합니다.*

type

플로트

기본

0.66

osd_deep_scrub_interval

설명

*깊은 스크럽을 위한 간격이 모든 데이터를 완전히 읽습니다. **osd_scrub_load_threshold** 는 이 설정에 영향을 미치지 않습니다.*

type

플로트

기본

7일

osd_debug_deep_scrub_sleep

설명

딥 스러브 IO 중에 비용이 많이 드는 수면을 삽입하여 선점을 더 쉽게 유도할 수 있도록 합니다.

type

플로트

기본

0

osd_scrub_interval_randomize_ratio

설명

*배치 그룹에 대한 다음 scrub 작업을 예약할 때 osd_scrub_min_interval 에 임의의 지연을 추가합니다. 지연은 osd_scrub_min_interval * osd_scrub_interval_randomized_ratio 보다 작은 임의의 값입니다. 기본 설정은 [1, 1.5] * osd_scrub_min_interval 의 허용된 시간 전체에 스큐브를 분배합니다.*

type

플로트

기본

0.5

osd_deep_scrub_stride

설명

깊은 스크러브를 할 때 크기를 읽으십시오.

type

size

기본

512KB

osd_scrub_auto_repair_num_errors

설명

이 많은 오류가 발견되면 자동 복구가 발생하지 않습니다.

type

integer

기본

5

osd_scrub_auto_repair

설명

*이 값을 **true** 로 설정하면 **scrubs** 또는 **deep-scrubs**에서 오류를 찾을 때 **PG**(자동 배치 그룹) 복구를 사용할 수 있습니다. 그러나 **osd_scrub_auto_repair_num_errors** 오류가 발견되면 복구가 수행되지 않습니다.*

type

boolean

기본

false

osd_scrub_max_preemptions

설명

*클라이언트 **IO**를 차단하여 **scrub**를 완료하기 전에 클라이언트 작업으로 인해 딥 스크럽을 선점해야 하는 최대 횟수를 설정합니다.*

type

integer

기본

5

osd_deep_scrub_keys

설명

딥 스크러블 중 한 번에 오브젝트에서 읽을 키 수입니다.

type

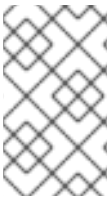
integer

기본

1024

부록 I. BLUESTORE 구성 옵션

다음은 배포 중에 구성할 수 있는 Ceph BlueStore 구성 옵션입니다.



참고

이 목록은 완전하지 않습니다.

rocksdb_cache_size

설명

RocksDB 캐시의 크기(MB)입니다.

유형

32비트 정수

기본

512

bluestore_throttle_bytes

설명

사용자가 입력 또는 출력(I/O) 제출을 제한하기 전에 사용 가능한 최대 바이트 수입니다.

유형

크기

기본

64MB

bluestore_throttle_deferred_bytes

설명

사용자가 I/O 제출을 제한하기 전에 지연된 쓰기의 최대 바이트 수입니다.

유형

크기

기본

128MB

bluestore_throttle_cost_per_io

설명

각 I/O에 대한 트랜잭션 비용(바이트)에 오버헤드가 추가되었습니다.

유형

크기

기본

0 B

bluestore_throttle_cost_per_io_hdd

설명

HDD의 기본 bluestore_throttle_cost_per_io 값입니다.

유형

서명되지 않은 정수

기본

67 000

bluestore_throttle_cost_per_io_ssd

설명

SSD의 기본 bluestore_throttle_cost_per_io 값입니다.

유형

서명되지 않은 정수

기본

4 000

bluestore_debug_enforce_settings

설명

***HDD** 는 회전 드라이브 위의 **BlueStore** 용 설정을 적용합니다. **SSD** 는 솔리드 드라이브 위의 **BlueStore** 용 설정 적용*

유형

*기본값,**hdd,ssd***

기본

default

참고

***bluestore_debug_enforce_settings** 옵션을 변경한 후 **OSD**를 다시 시작합니다.*