Red Hat Ceph Storage 3

구성 가이드

Red Hat Ceph Storage의 구성 설정

Last Updated: 2023-03-07

Red Hat Ceph Storage 3 구성 가이드

Red Hat Ceph Storage의 구성 설정

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 부팅 시 및 런타임 시 Red Hat Ceph Storage를 설정하는 방법을 설명합니다. 구성 참조 정보도 제공합니다.

차례

1장. 설정 참조

모든 Ceph 클러스터에는 다음을 정의하는 구성이 있습니다.

- 클러스터 ID
- 인증 설정
- 클러스터의 Ceph 데몬 멤버십
- 네트워크 구성
- 호스트 이름 및 주소
- 인증 키에 대한 경로
- 데이터 경로(저널 포함)
- 기타 런타임 옵션

Red Hat Storage Console 또는 Ansible과 같은 배포 도구는 일반적으로 초기 Ceph 구성 파일을 생성합니다. 그러나 배포 도구를 사용하지 않고 클러스터를 부트스트랩하려는 경우 직접 생성할 수 있습니다.

편의를 위해 각 데몬에는 일련의 기본값, 즉, 많은 값이 **ceph/src/common/config_opts.h** 스크립트에서 설정됩니다. 모니터 **tell** 명령을 사용하거나 Ceph 노드의 데몬 소켓에 직접 연결하여 이러한 설정을 Ceph 구성 파일 또는 런타임에 재정의할 수 있습니다.

1.1. 일반 권장 사항

원하는 대로 Ceph 구성 파일을 유지할 수 있지만, Red Hat은 Ceph 구성 파일의 마스터 사본을 유지 관리하는 관리 노드를 사용하는 것이 좋습니다.

Ceph 구성 파일을 변경하면 일관성을 유지하기 위해 업데이트된 구성 파일을 Ceph 노드로 내보내는 것이 좋습니다.

1.2. 구성 파일 구조

Ceph 구성 파일은 시작 시 Ceph 데몬을 구성하여 기본값을 포함합니다. Ceph 구성 파일은 *ini* 스타일 구문을 사용합니다. pound 기호(#) 또는 Semi-colon(;)을 사용하여 주석 앞에 주석을 추가할 수 있습니다. 예를 들어 다음과 같습니다.

```
# <--A number (#) sign precedes a comment.
; A comment may be anything.
# Comments always follow a semi-colon (;) or a pound (#) on each line.
# The end of the line terminates a comment.
# We recommend that you provide comments in your configuration file(s).
```

구성 파일은 Ceph 스토리지 클러스터의 모든 Ceph 데몬 또는 시작 시 특정 유형의 모든 Ceph 데몬을 구성할 수 있습니다. 일련의 데몬을 구성하려면 다음과 같이 구성을 수신할 프로세스에 설정을 포함해야 합니다.

[global]

설명

[global] 아래의 설정은 Ceph Storage 클러스터의 모든 데몬에 영향을 미칩니다.

예제

인증 지원 = Gradle

[osd]

설명

[osd] 아래의 설정은 Ceph 스토리지 클러스터의 모든 **ceph-osd** 데몬에 영향을 미치고 **[global]** 에서 동일한 설정을 재정의합니다.

예제

OSD 저널 크기 = 1000

[mon]

설명

[mon] 아래의 설정은 Ceph 스토리지 클러스터의 모든 **ceph-mon** 데몬에 영향을 미치며 **[global]** 에서 동일한 설정을 재정의합니다.

예제

mon host = hostname1,hostname2,hostname3 mon addr = 10.0.0.101:6789

[클라이언트]

설명

[client] 아래의 설정은 모든 Ceph 클라이언트에 영향을 미칩니다(예: 마운트된 Ceph 블록 장치, Ceph 개체 게이트웨이 등).

예제

log file = /var/log/ceph/radosgw.log

글로벌 설정은 Ceph 스토리지 클러스터에 있는 모든 데몬의 모든 인스턴스에 영향을 미칩니다. Ceph 스토리지 클러스터의 모든 데몬에 공통된 값에 대해 **[global]** 설정을 사용합니다. 다음을 통해 각 **[global]** 설정을 덮어쓸 수 있습니다.

1. 특정 프로세스 유형의 설정 변경(예: **[osd]**, **[mon]**).
2. 특정 프로세스의 설정 변경(예: **[osd.1]**).

전역 설정을 재정의하면 특정 데몬에서 특별히 재정의되는 프로세스를 제외하고 모든 하위 프로세스에 영향을 미칩니다.

일반적인 글로벌 설정에는 인증을 활성화해야 합니다. 예를 들어 다음과 같습니다.

```
[global]
#Enable authentication between hosts within the cluster.
auth_cluster_required = cephx
auth_service_required = cephx
auth_client_required = cephx
```

특정 유형의 데몬에 적용되는 설정을 지정할 수 있습니다. 특정 인스턴스를 지정하지 않고 **[osd]** 또는 **[mon]** 아래에 설정을 지정하면 설정이 모든 OSD 또는 모니터 데몬에 각각 적용됩니다.

일반적인 데몬 전체 설정에는 저널 크기, 파일 저장소 설정 설정 등이 포함됩니다. 예를 들면 다음과 같습니다.

```
[osd]
osd_journal_size = 1000
```

데몬의 특정 인스턴스에 대한 설정을 지정할 수 있습니다. 유형을 입력하고 마침표(.)로 구분된 인스턴스 ID로 구분하여 인스턴스를 지정할 수 있습니다. Ceph OSD 데몬의 인스턴스 ID는 항상 숫자이지만 Ceph 모니터의 영숫자일 수 있습니다.

```
[osd.1]
# settings affect osd.1 only.

[mon.a]
# settings affect mon.a only.
```

기본 Ceph 구성 파일 위치는 순서대로 제공됩니다.

1. **$CEPH_octets** (**$CEPH_octets 환경 변수 다음에** 따르는 경로)
2. **-c path/path** (**-c** 명령줄 인수)
3. **/etc/ceph/ceph.conf**
4. **~/.ceph/config**
5. **./Ceph.conf** (현재 작업 디렉터리의 경우)

일반적인 Ceph 구성 파일에는 최소한 다음 설정이 있습니다.

```
[global]
fsid = {cluster-id}
mon_initial_members = {hostname}[, {hostname}]
mon_host = {ip-address}[, {ip-address}]

#All clusters have a front-side public network.
#If you have two NICs, you can configure a back side cluster
#network for OSD object replication, heart beats, backfilling,
#recovery, and so on
public_network = {network}[, {network}]
#cluster_network = {network}[, {network}]

#Clusters require authentication by default.
auth_cluster_required = cephx
auth_service_required = cephx
auth_client_required = cephx

#Choose reasonable numbers for your journals, number of replicas
#and placement groups.
osd_journal_size = {n}
osd_pool_default_size = {n}  # Write an object n times.
osd_pool_default_min_size = {n} # Allow writing n copy in a degraded state.
osd_pool_default_pg_num = {n}
osd_pool_default_pgp_num = {n}

#Choose a reasonable crush leaf type.
#0 for a 1-node cluster.
#1 for a multi node cluster in a single rack
```

```
#2 for a multi node, multi chassis cluster with multiple hosts in a chassis
#3 for a multi node cluster with hosts across racks, and so on
osd_crush_chooseleaf_type = {n}
```

1.3. 메타 변수

Meta capitals는 Ceph 스토리지 클러스터 구성을 급격하게 단순화합니다. metaPrice가 구성 값으로 설정되면 Ceph는 metaswitch를 구체적인 값으로 확장합니다.

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

Ceph는 다음과 같은 메타 변수를 지원합니다.

$cluster

설명

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

예제

/etc/ceph/$cluster.keyring

기본값

Ceph

$type

설명

인스턴트 데몬의 유형에 따라 **osd** 또는 **mon** 중 하나로 확장합니다.

예제

/var/lib/ceph/$type

$ID

설명

데몬 식별자로 확장합니다. **osd.0** 의 경우 **0** 이 됩니다.

예제

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

$host

설명

인스턴트 데몬의 호스트 이름으로 확장합니다.

$name

설명

$type.$id 로 확장합니다.

예제

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

1.4. CEPH 런타임 구성 보기

런타임 구성을 보려면 Ceph 노드에 로그인하고 다음을 실행합니다.

```
ceph daemon {daemon-type}.{id} config show
```

예를 들어 **osd.0** 에 대한 구성을 보려면 **osd.0** 을 포함하는 노드에 로그인하고 다음을 실행합니다.

```
ceph daemon osd.0 config show
```

추가 옵션의 경우 데몬과 **도움말** 을 지정합니다. 예를 들어 다음과 같습니다.

```
ceph daemon osd.0 help
```

1.5. 런타임 시 특정 구성 설정 가져오기

런타임 시 특정 구성 설정을 가져오려면 Ceph 노드에 로그인하고 다음을 실행합니다.

```
ceph daemon {daemon-type}.{id} config get {parameter}
```

예를 들어 **osd.0** 의 공용 주소를 검색하려면 다음을 실행합니다.

```
ceph daemon osd.0 config get public_addr
```

1.6. 런타임 시 특정 구성 설정 설정 설정

런타임 구성을 설정하는 두 가지 일반적인 방법은 다음과 같습니다.

- Ceph 모니터 사용
- 관리 소켓 사용

tell 및 **injectargs** 명령을 사용하여 모니터에 연결하여 Ceph 런타임 구성 설정을 설정할 수 있습니다. 이 방법을 사용하려면 수정하려는 모니터와 데몬이 실행 중이어야 합니다.

```
ceph tell {daemon-type}.{daemon id or *} injectargs --{name} {value} [--{name} {value}]
```

{daemon-type} 을 **osd** 또는 **mon** 중 하나로 바꿉니다. ** *를 사용하여 특정 유형의 모든 데몬에 런타임 설정을 적용하거나 특정 데몬의 ID(즉, 번호 또는 이름)를 지정할 수 있습니다. 예를 들어 **osd.0** 이라는 **ceph-osd** 데몬의 디버그 로깅을 **0/5** 로 변경하려면 다음 명령을 실행합니다.

```
ceph tell osd.0 injectargs '--debug-osd 0/5'
```

tell 명령은 여러 인수를 사용하므로 에 대한 각 인수가 단일 따옴표 내에 있어야 하며, 구성 앞에 두 개의 대시(**'-{config_opt} {opt-val} {opt-val} {opt-val}']**가 추가됩니다. 하나의 인수만 사용하므로 **데몬** 명령에 따옴표가 필요하지 않습니다.

ceph tell 명령은 모니터를 통과합니다. 모니터에 바인딩할 수 없는 경우 **ceph** 데몬을 사용하여 구성을 변경하려는 데몬 호스트에 로그인하여 계속 변경할 수 있습니다. 예를 들어 다음과 같습니다.

```
sudo ceph osd.0 config set debug_osd 0/5
```

1.7. 일반 구성 참조

일반적으로 일반 설정은 배포 툴에 의해 자동으로 설정됩니다.

fsid

설명

파일 시스템 ID입니다. 클러스터당 하나씩.

유형

UUID

필수 항목

아니요.

기본값

해당 없음 일반적으로 배포 툴에 의해 생성됩니다.

admin_socket

설명

Ceph 모니터가 쿼럼을 설정했는지 여부에 관계없이 데몬에서 관리 명령을 실행하기 위한 소켓입니다.

유형

문자열

필수 항목

없음

기본값

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

pid_file

설명

모니터 또는 OSD가 PID를 작성하는 파일입니다. 예를 들어 **/var/run/$cluster/$type.$id.pid** 는 **ceph** 클러스터에서 실행되는 ID와 함께 **mon** 에 대해 /var/run/$type.$id.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

기본값

없음

chdir

설명

Ceph 데몬이 시작 및 실행 중으로 변경되는 디렉터리입니다. 기본 / 디렉토리를 권장합니다.

유형

문자열

필수 항목

없음

기본값

/

max_open_files

설명

설정된 경우 Red Hat Ceph Storage 클러스터가 시작되면 Ceph는 OS 수준(즉, 파일 설명자의 최대 #)에 max_open_fds 를 설정합니다. Ceph OSD가 파일 설명자에서 실행되지 않도록 합니다.

유형

64비트 Integer

필수 항목

없음

기본값

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

기본값

true

1.8. OSD 메모리 대상

Bluestore는 osd_memory_target 구성 옵션을 사용하여 OSD 힙 메모리 사용량을 지정된 대상 크기로 유지합니다.

osd_memory_target 옵션은 시스템에서 사용 가능한 RAM에 따라 OSD 메모리를 설정합니다. 기본적으로 Anisble은 값을 4GB로 설정합니다. 데몬을 배포할 때 /usr/share/ceph-ansible/group_vars/all.yml 파일에서 바이트로 표시되는 값을 변경할 수 있습니다.

예: osd_memory_target 을 6000000000 바이트로 설정

```
ceph_conf_overrides:
  osd:
    osd_memory_target=6000000000
```

캐시 적중이 견고한 상태 드라이브보다 훨씬 높기 때문에 블록 장치의 속도가 기존 하드 드라이브와 같이 블록 장치의 속도가 느려지면 Ceph OSD 메모리 캐싱이 더 중요합니다. 그러나 이 작업은 HCI(하이퍼컨버지드 인프라) 또는 기타 애플리케이션과 같은 다른 서비스와 OSD를 함께 배치해야 합니다.

참고

osd_memory_target 은 기존 하드 드라이브 장치의 장치당 하나의 OSD와 NVMe SSD 장치의 장치당 OSD 2개입니다. osds_per_device 는 group_vars/osds.yml 파일에 정의되어 있습니다.

추가 리소스

- osd_memory_target 설정 OSD Memory Target

1.9. MDS 캐시 메모리 제한

MDS 서버는 해당 메타데이터를 cephfs_metadata 라는 별도의 스토리지 풀에 저장하고, Ceph OSD 사용자입니다. Ceph File Systems의 경우 MDS 서버는 스토리지 클러스터 내의 단일 스토리지 장치가 아닌 전체 Red Hat Ceph Storage 클러스터를 지원해야 하므로 특히 워크로드에 따라 메타데이터의 비율이 훨씬 높은 small-to-medium-size 파일로 구성된 경우 특히 메모리 요구 사항이 중요할 수 있습니다.

예: mds_cache_memory_limit 를 2000000000바이트로 설정

```
ceph_conf_overrides:
  osd:
    mds_cache_memory_limit=2000000000
```

참고

메타데이터 집약적인 대규모 **Red Hat Ceph Storage** 클러스터의 경우 다른 메모리 집약적 서비스와 동일한 노드에 **MDS** 서버를 배치하지 마십시오. 이렇게 하면 **100GB**보다 큰 크기의 **MDS**에 메모리를 더 많이 할당할 수 있는 옵션이 제공됩니다.

추가 리소스

- **MDS** 캐시 크기 제한 이해를참조하십시오.

2장. 네트워크 구성 참조

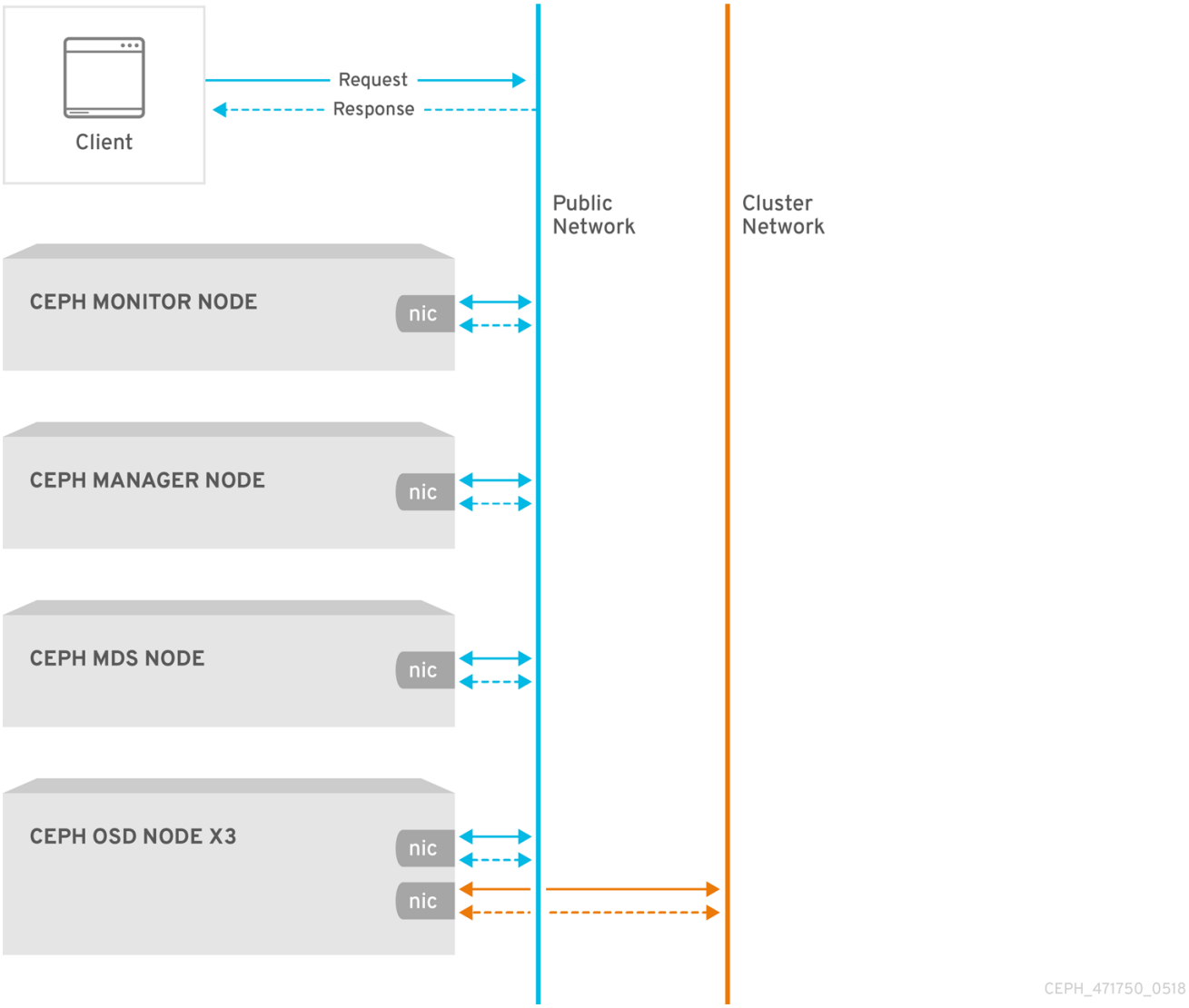
네트워크 구성은 고성능 **Red Hat Ceph Storage** 클러스터를 구축하는 데 중요합니다. **Ceph** 스토리지 클러스터는 **Ceph** 클라이언트를 대신하여 라우팅 또는 디스패치를 수행하지 않습니다. 대신 **Ceph** 클라이언트는 **Ceph OSD** 데몬에 직접 요청합니다. **Ceph OSD**는 **Ceph** 클라이언트를 대신하여 데이터 복제를 수행합니다. 즉, 복제 및 기타 요인으로 **Ceph** 스토리지 클러스터 네트워크에 추가 로드가 발생합니다.

모든 **Ceph** 클러스터에서는 공용 네트워크를 사용해야 합니다. 그러나 클러스터(내부) 네트워크를 지정하지 않으면 **Ceph**는 단일 공용 네트워크를 가정합니다. **Ceph**는 공용 네트워크로만 작동할 수 있지만 대규모 클러스터에서는 두 번째 "클러스터" 네트워크를 통해 성능이 크게 향상됩니다.

다음 두 개의 네트워크를 사용하여 **Ceph** 스토리지 클러스터를 실행하는 것이 좋습니다.

- 공용 네트워크
- 클러스터 네트워크입니다.

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



두 개의 별도 네트워크를 운영해야 하는 몇 가지 이유가 있습니다.

- 성능: Ceph OSD는 Ceph 클라이언트의 데이터 복제를 처리합니다. Ceph OSD가 데이터를 두 번 이상 복제하면 Ceph OSD 간 네트워크 로드가 Ceph 클라이언트와 Ceph 스토리지 클러스터 간의 네트워크 로드를 쉽게 분리할 수 있습니다. 이로 인해 대기 시간이 도입되고 성능 문제가 발생할 수 있습니다. 복구 및 재조정으로 공용 네트워크에 상당한 대기 시간이 발생할 수도 있습니다.

- 보안: 대부분의 사람들은 일반적으로 대중이지만 일부 행위자는 서비스 거부 (DoS) 공격으로 알려진 것에 참여할 것입니다. Ceph OSD 간 트래픽이 중단되면 피어링이 실패할 수 있으며 배치 그룹이 더 이상 활성 + 클린 상태가 반영되지 않을 수 있으므로 사용자가 데이터를 읽고 쓰는 것을 방지할 수 있습니다. 이러한 유형의 공격을 차단하는 가장 좋은 방법은 인터넷에 직접 연결되지 않는 완전히 분리된 클러스터 네트워크를 유지하는 것입니다.

2.1. 네트워크 설정

네트워크 구성 설정이 필요하지 않습니다. Ceph 데몬을 실행하는 모든 호스트에 공용 네트워크가 구성

되어 있다고 가정하면 **Ceph**에서 공용 네트워크로 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 처리하기 위해 별도의 클러스터 네트워크를 설정할 수도 있습니다.

구성에 설정한 **IP** 주소를 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.

작은 정보

공용 또는 클러스터 네트워크에 대해 두 개 이상의 **IP** 주소와 서브넷 마스크를 지정하는 경우 네트워크 내의 서브넷이 서로 라우팅할 수 있어야 합니다. 필요한 경우 **IP** 테이블에 각 **IP** 주소/**subnet**을 포함하고 이를 위해 포트를 열어야 합니다.

참고

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

네트워크를 구성하면 클러스터를 다시 시작하거나 각 데몬을 다시 시작할 수 있습니다. **Ceph** 데몬이 동적으로 바인딩되므로 네트워크 구성을 변경하는 경우 전체 클러스터를 즉시 재시작할 필요가 없습니다.

2.1.1. 공용 네트워크

공용 네트워크를 구성하려면 **Ceph** 구성 파일의 **[global]** 섹션에 다음 옵션을 추가합니다.

```
[global]
    ...
    public_network = <public-network/netmask>
```

공용 네트워크 구성을 사용하면 공용 네트워크의 **IP** 주소 및 서브넷을 구체적으로 정의할 수 있습니다. 특정 데몬에 대한 공용 **addr** 설정을 사용하여 고정 **IP** 주소를 할당하거나 공용 네트워크 설정을 재정의할 수 있습니다.

public_network

설명

공용(**front-side**) 네트워크의 **IP** 주소 및 넷마스크(예:**journal**) 입니다. **[global]** 에 설정합니다. 쉼표로 구분된 서브넷을 지정할 수 있습니다.

유형

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

필수 항목

없음

기본값

해당 없음

public_addr

설명

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

유형

IP 주소

필수 항목

없음

기본값

해당 없음

2.1.2. 클러스터 네트워크

클러스터 네트워크를 선언하면 OSD는 클러스터 네트워크를 통해 하트비트, 오브젝트 복제 및 복구 트래픽을 라우팅합니다. 이는 단일 네트워크를 사용하는 것과 비교하여 성능을 향상시킬 수 있습니다. 클러스터 네트워크를 구성하려면 Ceph 구성 파일의 [global] 섹션에 다음 옵션을 추가합니다.

```
[global]
    ...
    cluster_network = <cluster-network/netmask>
```

보안을 강화하기 위해 공용 네트워크 또는 인터넷에서 클러스터 네트워크에 연결할 수 없는 것이 좋습니다.

클러스터 네트워크 구성을 사용하면 클러스터 네트워크를 선언하고 클러스터 네트워크의 IP 주소 및 서브넷을 구체적으로 정의할 수 있습니다. 특정 OSD 데몬의 클러스터 애드온 설정을 사용하여 고정 IP 주

소를 할당하거나 클러스터 네트워크 설정을 덮어쓸 수 있습니다.

cluster_network

설명

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

유형

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

필수 항목

없음

기본값

해당 없음

cluster_addr

설명

클러스터 네트워크의 IP 주소입니다. 각 데몬에 대해 설정됩니다.

유형

address

필수 항목

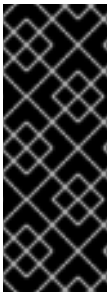
없음

기본값

해당 없음

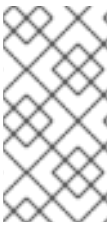
2.1.3. MTU 값 확인 및 구성

최대 전송 단위(MTU) 값은 링크 계층에서 전송되는 가장 큰 패킷의 크기(바이트)입니다. 기본 MTU 값은 1500바이트입니다. Red Hat은 Red Hat Ceph Storage 클러스터에서 MTU 값 9000바이트의 점보 프레임을 사용하는 것이 좋습니다.



중요

Red Hat Ceph Storage는 공용 네트워크와 클러스터 네트워크 모두에 대해 통신 경로의 모든 네트워킹 장치에 걸쳐 동일한 **MTU** 값이 필요합니다. 프로덕션에서 **Red Hat Ceph Storage** 클러스터를 사용하기 전에 **MTU** 값이 환경의 모든 노드 및 네트워킹 장비에서 같은지 확인합니다.



참고

본딩 네트워크 인터페이스를 함께 사용하는 경우 **MTU** 값은 결합된 인터페이스에서만 설정해야 합니다. 새로운 **MTU** 값은 본딩 장치에서 기본 네트워크 장치로 전파됩니다.

사전 요구 사항

- 노드에 대한 루트 수준 액세스.

절차

1. 현재 **MTU** 값을 확인합니다.

예제

```
[root@mon ~]# ip link list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp22s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
mode DEFAULT group default qlen 1000
```

이 예에서 네트워크 인터페이스는 **enp22s0f0** 이며 **MTU** 값은 **1500** 입니다.

2. **MTU** 값을 온라인으로 일시적으로 변경하려면 다음을 수행합니다.

구문

```
ip link set dev NET_INTERFACE mtu NEW_MTU_VALUE
```

예제

```
[root@mon ~]# ip link set dev enp22s0f0 mtu 9000
```

3. **MTU 값을 영구적으로 변경하려면 다음을 수행합니다.**

   a. **해당 특정 네트워크 인터페이스에 대한 네트워크 구성 파일을 편집하기 위해 을 엽니다.**

   구문

   ```
   vim /etc/sysconfig/network-scripts/ifcfg-NET_INTERFACE
   ```

   예제

   ```
   [root@mon ~]# vim /etc/sysconfig/network-scripts/ifcfg-enp22s0f0
   ```

   b. **새 줄에서 MTU=9000 옵션을 추가합니다.**

   예제

```
NAME="enp22s0f0"
DEVICE="enp22s0f0"
MTU=9000 (1)
ONBOOT=yes
NETBOOT=yes
UUID="a8c1f1e5-bd62-48ef-9f29-416a102581b2"
IPV6INIT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
```

c. **network** 서비스를 다시 시작하십시오.

예제

```
[root@mon ~]# systemctl restart network
```

추가 리소스

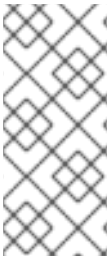
- 자세한 내용은 **Red Hat Enterprise Linux 7**의 *네트워킹 가이드* 를 참조하십시오.

2.1.4. messaging

Makefile은 **Ceph** 네트워크 계층 구현입니다. **Red Hat**은 두 가지 장애 조치 유형을 지원합니다.

- **simple**
- **async**

RHCS 2 및 이전 버전에서 **simple** 는 기본 래커 유형입니다. **RHCS 3**에서 **async** 는 기본 래지어 유형입니다. **messenger** 유형을 변경하려면 **Ceph** 구성 파일의 **[global]** 섹션에 **ms_type** 구성 설정을 지정합니다.



참고

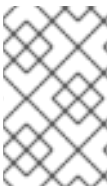
Red Hat은 비동기 선지의 경우 **posix** 전송 유형을 지원하지만 현재 **rdma** 또는 **dpdk** 를 지원하지 않습니다. 기본적으로 **RHCS 3**의 **ms_type** 설정은 **async+posix** 를 반영해야 합니다. 여기서 **async** 는 선지자 유형이며 **posix** 는 전송 유형입니다.

About 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 전송 유형에는 효과적이지 않습니다. 결과적으로 **AsyncMessenger** 는 **RHCS 3** 이상 릴리스의 기본 래커 유형입니다.

AsyncMessenger 정보

RHCS 3의 경우 **AsyncMessenger** 구현에서는 연결에 고정된 크기의 스레드 풀과 함께 **TCP** 소켓을 사용합니다. 이 풀은 가장 많은 복제본 또는 삭제 코드 청크 수와 같아야 합니다. **CPU** 수가 낮거나 서버당 **OSD** 수가 많아 성능이 저하되는 경우 스레드 수를 더 낮은 값으로 설정할 수 있습니다.



참고

Red Hat은 현재 **rdma** 또는 **dpdk** 와 같은 다른 전송 유형을 지원하지 않습니다.

Makefile 유형 설정

ms_type

설명

네트워크 전송 계층의 잘못된 유형. **Red Hat**은 **posix** 의미를 사용하여 간단하고 비동기 지저 유형을 지원합니다.

유형

문자열.

필수 항목

아니요.

기본값

async+posix

ms_public_type

설명

공용 네트워크의 네트워크 전송 계층을 위한 선지자 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 래커 유형을 사용할 수 있습니다.

유형

문자열.

필수 항목

아니요.

기본값

없음.

ms_cluster_type

설명

클러스터 네트워크의 네트워크 전송 계층에 대한 잘못된 전송 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 래커 유형을 사용할 수 있습니다.

유형

문자열.

필수 항목

아니요.

기본값

없음.

2.1.5. AsyncMessenger 설정

ms_async_transport_type

설명

AsyncMessenger 에서 사용하는 전송 유형입니다. **Red Hat**은 **posix** 설정을 지원하지만 현재 **dpdk** 또는 **rdma** 설정은 지원하지 않습니다. **POSIX**는 표준 **TCP/IP** 네트워킹을 사용하며 이는 기본값입니다. 다른 전송 유형은 실험적이며 지원되지 않습니다.

유형

문자열

필수 항목

없음

기본값

POSIX

ms_async_op_threads

설명

각 **AsyncMessenger** 인스턴스에서 사용하는 초기 작업자 스레드 수입니다. 이 구성 설정은 복제본 수 또는 삭제 코드 청크 수와 동일하지만 **CPU** 코어 수가 낮거나 단일 서버의 **OSD** 수가 높은 경우 더 낮게 설정할 수 있습니다.

유형

64비트 서명되지 않은 **Integer**

필수 항목

없음

기본값

3

ms_async_max_op_threads

설명

각 **AsyncMessenger** 인스턴스에서 사용하는 최대 작업자 스레드 수입니다. **OSD** 호스트에 **CPU** 수가 제한된 경우 더 낮은 값으로 설정하고 **Ceph**가 **CPU**를 사용하지 않는 경우 증가시킵니다.

유형

64비트 서명되지 않은 **Integer**

필수 항목

없음

기본값

5

ms_async_set_affinity

설명

AsyncMessenger 작업자를 특정 **CPU** 코어에 바인딩하려면 **true** 로 설정합니다.

유형

부울

필수 항목

없음

기본값

true

ms_async_affinity_cores

설명

ms_async_set_affinity 가 **true** 인 경우 이 문자열은 **AsyncMessenger** 작업자가 **CPU** 코어에 바인딩되는 방법을 지정합니다. 예를 들어 **0,2** 는 작업자 **#1** 및 **#2**를 **CPU** 코어 **#0** 및 **#2**에 각각 바인딩합니다. 참고: 유사성을 수동으로 설정할 때 물리적 **CPU** 코어보다 느리기 때문에 하이퍼스레딩 또는 유사한 기술의 효과로 생성된 가상 **CPU**에 작업자를 할당하지 않도록 해야 합니다.

유형

문자열

필수 항목

없음

기본값

(비어 있음)

ms_async_send_inline

설명

AsyncMessenger 스레드에서 대기 및 전송 대신 해당 스레드를 생성한 스레드에서 직접 메

시지를 보냅니다. 이 옵션은 많은 CPU 코어가 있는 시스템의 성능을 줄이는 것으로 알려져 있으므로 기본적으로 비활성화되어 있습니다.

유형

부울

필수 항목

없음

기본값

false

2.1.6. 바인딩

바인딩 설정은 Ceph OSD 데몬이 사용하는 기본 포트 범위를 설정합니다. 기본 범위는 6800:7100 입니다. 방화벽 구성이 구성된 포트 범위를 사용할 수 있는지 확인합니다.

Ceph 데몬을 활성화하여 IPv6 주소에 바인딩할 수도 있습니다.

ms_bind_port_min

설명

OSD 데몬이 바인딩할 최소 포트 번호입니다.

유형

32비트 정수

기본값

6800

필수 항목

없음

ms_bind_port_max

설명

OSD 데몬이 바인딩할 최대 포트 번호입니다.

유형

32비트 정수

기본값

7300

필수 항목

아니요.

ms_bind_ipv6

설명

Ceph 데몬이 IPv6 주소에 바인딩되도록 합니다.

유형

부울

기본값

false

필수 항목

없음

2.1.7. 호스트

Ceph 구성 파일에 하나 이상의 모니터가 있어야 하며, 각 모니터에는 mon addr 설정이 포함됩니다. Ceph 구성 파일에서 선언된 각 모니터, 메타데이터 서버, OSD 아래에 호스트 설정이 필요합니다.

mon_addr

설명

클라이언트가 Ceph 모니터에 연결하는 데 사용할 수 있는 < hostname>:<port > 항목 목록입니다. 설정하지 않는 경우 Ceph 검색 [mon.*] 섹션.

유형

문자열

필수 항목

없음

기본값

해당 없음

host

설명

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

유형

문자열

필수 항목

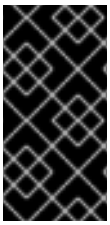
예. 데몬 인스턴스의 경우

기본값

localhost

작은 정보

localhost 를 사용하지 마십시오. 호스트 이름을 얻으려면 **hostname -s** 명령을 실행하고 정규화된 도메인 이름이 아닌 첫 번째 기간 동안 호스트 이름을 사용합니다.



중요

사용자를 위한 호스트 이름을 검색하는 타사 배포 시스템을 사용할 때 호스트의 값을 지정하지 마십시오.

2.1.8. TCP

Ceph는 기본적으로 TCP 버퍼링을 비활성화합니다.

ms_tcp_nodelay

설명

Ceph는 **ms_tcp_nodelay** 를 활성화하여 각 요청이 즉시 전송됩니다(기고 없음). **Nagle**의 알고리즘을 비활성화하면 네트워크 트래픽이 증가하여 혼잡을 초래할 수 있습니다. 많은 수의 작은 패킷이 발생하는 경우 **ms_tcp_nodelay** 를 비활성화하여 비활성화하면 일반적으로 대기 시간이 증가한다는 점에 유의하십시오.

유형

부울

필수 항목

없음

기본값

true

ms_tcp_rcvbuf

설명

네트워크 연결 수신 끝에 있는 소켓 버퍼의 크기입니다. 기본적으로 비활성화합니다.

유형

32비트 정수

필수 항목

없음

기본값

0

ms_tcp_read_timeout

설명

클라이언트 또는 데몬에서 다른 **Ceph** 데몬을 요청하고 사용되지 않는 연결을 삭제하지 않는 경우 **tcp** 읽기 타임아웃 은 지정된 수의 초 후에 연결을 유휴 상태로 정의합니다.

유형

서명되지 않은 64비트 정수

필수 항목

없음

기본값

900 15분.

2.1.9. 방화벽

기본적으로 데몬은 6800:7100 범위 내의 포트에 바인딩됩니다. 이 범위는 재량에 따라 구성할 수 있습니다. 방화벽을 구성하기 전에 기본 방화벽 구성을 확인합니다. 이 범위는 재량에 따라 구성할 수 있습니다.

```
sudo iptables -L
```

firewalld 데몬의 경우 **root** 로 다음 명령을 실행합니다.

```
# firewall-cmd --list-all-zones
```

일부 **Linux** 배포에는 모든 네트워크 인터페이스에서 **SSH**를 제외한 모든 인바운드 요청을 거부하는 규칙이 포함되어 있습니다. 예를 들어 다음과 같습니다.

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

2.1.9.1. 모니터 방화벽

Ceph 모니터는 기본적으로 포트 **6789** 에서 수신 대기합니다. 또한 **Ceph** 모니터는 항상 공용 네트워크에서 작동합니다. 아래 예제를 사용하여 규칙을 추가할 때 **< iface >**를 공용 네트워크 인터페이스(예: **eth0,eth1** 등)로 바꾸고 **< ip-address >**를 공용 네트워크의 **IP** 주소로, **< net mask >**를 공용 네트워크의 넷마스크로 바꿉니다.

```
sudo iptables -A INPUT -i <iface> -p tcp -s <ip-address>/<netmask> --dport 6789 -j ACCEPT
```

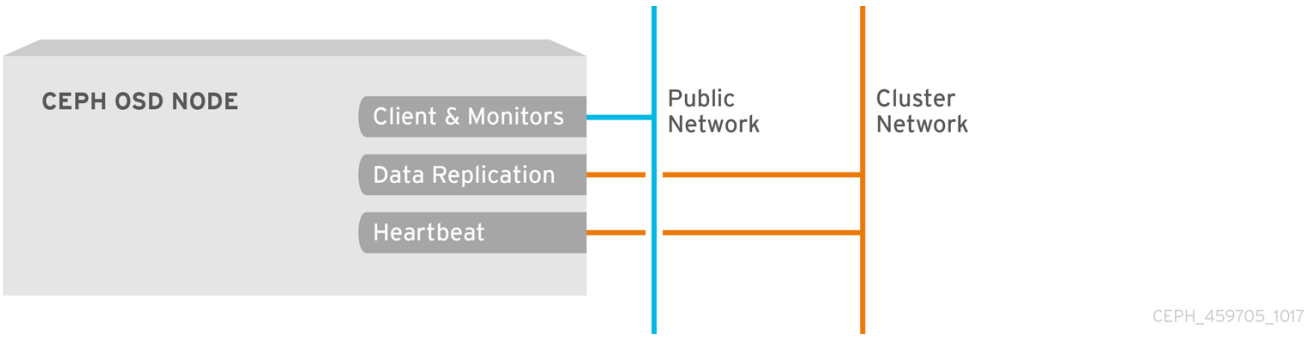
firewalld 데몬의 경우 **root** 로 다음 명령을 실행합니다.

```
# firewall-cmd --zone=public --add-port=6789/tcp
# firewall-cmd --zone=public --add-port=6789/tcp --permanent
```

2.1.9.2. OSD 방화벽

기본적으로 **Ceph OSD**는 포트 **6800**에서 시작되는 **Ceph** 노드에서 사용 가능한 첫 번째 포트에 바인딩됩니다. 호스트에서 실행되는 각 **OSD**마다 포트 **6800**에서 시작하는 포트 **3**개를 열어야 합니다.

1. 클라이언트 및 모니터(공용 네트워크)에 대해 이야기하는 것입니다.
2. 데이터를 다른 **OSD**(클러스터 네트워크)로 전송하기 위한 것입니다.
3. 하트비트 패킷(클러스터 네트워크)을 전송하는 것입니다.



포트는 노드별입니다. 그러나 프로세스가 다시 시작되고 바인딩된 포트가 릴리스되지 않는 경우 **Ceph** 노드에서 실행되는 **Ceph** 데몬에 필요한 포트 수보다 많은 포트를 열어야 할 수 있습니다. 다시 시작한 데몬이 새 포트에 바인딩되도록 포트를 해제하지 않고 데몬이 실패하는 경우 몇 가지 추가 포트를 여는 것이 좋습니다. 또한 각 **OSD** 호스트에서 **6800:7300** 의 포트 범위를 여는 것이 좋습니다.

별도의 공용 네트워크와 클러스터 네트워크를 설정하는 경우 클라이언트가 공용 네트워크를 사용하여 연결되고 기타 **Ceph OSD** 데몬이 클러스터 네트워크를 사용하여 연결되므로 공용 네트워크와 클러스터 네트워크에 대한 규칙을 추가해야 합니다.

아래 예제를 사용하여 규칙을 추가할 때 **< iface >**를 네트워크 인터페이스(예: **eth0** 또는 **eth1**), **'<ip-address >**를 **IP** 주소로, **< netmask >**을 공용 또는 클러스터 네트워크의 넷마스크로 바꿉니다. 예를 들어 다음과 같습니다.

```
sudo iptables -A INPUT -i <iface>  -m multiport -p tcp -s <ip-address>/<netmask> --dports 6800:6810
-j ACCEPT
```

firewalld 데몬의 경우 **root** 로 다음 명령을 실행합니다.

```
# firewall-cmd --zone=public --add-port=6800-6810/tcp
# firewall-cmd --zone=public --add-port=6800-6810/tcp --permanent
```

클러스터 네트워크를 다른 영역에 배치하면 해당 영역 내의 포트를 적절하게 엽니다.

2.2. CEPH 데몬

Ceph에는 모든 데몬에 적용되는 하나의 네트워크 구성 요구 사항이 있습니다. **Ceph** 구성 파일은 각 데몬의 호스트를 지정해야 합니다. **Ceph**에서 더 이상 **Ceph** 구성 파일에서 모니터 **IP** 주소와 해당 포트를 지정할 필요가 없습니다.

중요

일부 배포 유틸리티에서 구성 파일을 생성할 수 있습니다. 배포 유틸리티에서 이러한 값을 설정하지 않으면 설정하지 마십시오.

작은 정보

호스트 설정은 호스트의 짧은 이름입니다(즉, **FQDN**이 아님). **IP** 주소도 아닙니다. **hostname -s** 명령을 사용하여 호스트 이름을 검색합니다.

```
[mon.a]

    host = <hostname>
    mon addr = <ip-address>:6789

[osd.0]
    host = <hostname>
```

데몬의 호스트 **IP** 주소를 설정할 필요는 없습니다. 고정 **IP** 구성이 있고 공용 및 클러스터 네트워크가 모두 실행 중인 경우 **Ceph** 구성 파일에서 각 데몬의 호스트의 **IP** 주소를 지정할 수 있습니다. 데몬의 고정 **IP** 주소를 설정하려면 **Ceph** 구성 파일의 데몬 인스턴스 섹션에 다음 옵션이 표시되어야 합니다.

```
[osd.0]
    public_addr = <host-public-ip-address>
    cluster_addr = <host-cluster-ip-address>
```

두 네트워크 클러스터에서 하나의 **NIC OSD**

일반적으로 **Red Hat**은 두 개의 네트워크가 있는 클러스터에서 단일 **NIC**를 사용하여 **OSD** 호스트를 배포하는 것을 권장하지 않습니다. 그러나 **Ceph** 구성 파일의 **[osd.n]** 섹션에 공용 **addr** 항목을 추가하여 **OSD** 호스트가 공용 네트워크에서 작동하도록 강제 수행하여 이 작업을 수행합니다. 여기서 **n** 은 하나의 **NIC**가 있는 **OSD** 수를 나타냅니다. 또한 공용 네트워크와 클러스터 네트워크는 트래픽을 서로 라우팅할 수 있어야 합니다. 이 경우 보안상의 이유로 **Red Hat**은 권장되지 않습니다.

3장. 모니터 구성 참조

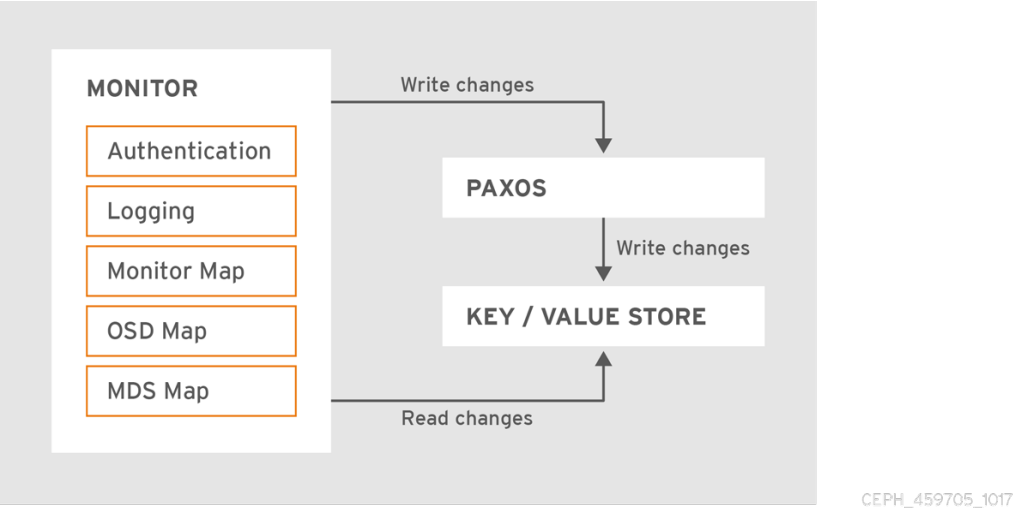
Ceph 모니터 구성 방법은 안정적인 Red Hat Ceph Storage 클러스터 구축에서 중요한 부분입니다. 모든 클러스터에는 모니터가 한 개 이상 있습니다. 모니터 구성은 일반적으로 매우 일관되게 유지되지만 클러스터에서 모니터를 추가, 제거 또는 교체할 수 있습니다.

3.1. 배경 정보

Ceph 모니터는 클러스터 맵의 "마스터 복사본"을 유지합니다. 따라서 Ceph 클라이언트는 하나의 Ceph 모니터에 연결하여 현재 클러스터 맵을 검색하여 모든 Ceph 모니터 및 Ceph OSD의 위치를 확인할 수 있습니다.

Ceph 클라이언트가 Ceph OSD에서 읽거나 쓸 수 있으려면 먼저 Ceph 모니터에 연결해야 합니다. 클러스터 맵의 현재 사본과 ArgoCD 알고리즘을 사용하면 Ceph 클라이언트가 모든 개체의 위치를 계산할 수 있습니다. Ceph 클라이언트는 오브젝트 위치를 계산하는 기능을 통해 Ceph OSD와 직접 통신할 수 있습니다. Ceph 높은 확장성과 성능이 매우 중요합니다.

Ceph 모니터의 기본 역할은 클러스터 맵의 마스터 사본을 유지하는 것입니다. Ceph 모니터는 인증 및 로깅 서비스도 제공합니다. Ceph 모니터는 모니터 서비스의 모든 변경 사항을 단일 Paxos 인스턴스에 쓰고 Paxos는 키-값 저장소에 변경 사항을 작성하여 강력한 일관성을 유지합니다. Ceph 모니터는 동기화 작업 중에 최신 버전의 클러스터 맵을 쿼리할 수 있습니다. Ceph 모니터는 키-값 저장소의 스냅샷과 이터레이터(leveldb 데이터베이스 사용)를 활용하여 저장소 전체 동기화를 수행합니다.



3.1.1. 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 는 무엇입니까.
- 배치 그룹이 활성 상태인지 또는 비활성 상태인지 여부, 정리 또는 기타 상태입니다.
- 다음과 같은 클러스터의 현재 상태를 반영하는 기타 세부 정보입니다.
  - 총 스토리지 공간 또는
  - 사용되는 스토리지 용량입니다.

예를 들어 클러스터 상태가 크게 변경되면 배치 그룹이 저하된 상태가 되고, 클러스터 맵은 클러스터의 현재 상태를 반영하도록 업데이트됩니다. 또한 **Ceph** 모니터는 클러스터의 이전 상태 기록도 유지합니다. 모니터 맵, **OSD** 맵, 배치 그룹은 각각 맵 버전의 기록을 유지합니다. 각 버전을 **epoch** 라고 합니다.

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

3.1.2. Quorum

클러스터는 단일 모니터로 충분히 실행됩니다. 그러나 단일 모니터는 단일 장애 지점입니다.**However, a single monitor is a single-point-of-failure.** 프로덕션 **Ceph** 스토리지 클러스터에서 고가용성을 보장하기 위해 여러 모니터가 있는 **Ceph**를 실행하여 단일 모니터 실패로 인해 전체 클러스터가 실패하지 않도록 합니다.

Ceph 스토리지 클러스터가 고가용성을 위해 여러 **Ceph** 모니터를 실행하는 경우 **Ceph** 모니터는 **Paxos** 알고리즘을 사용하여 마스터 클러스터 맵에 대한 합의를 설정합니다. 합의에 따라 클러스터 맵에 대해 쿼럼을 구축하기 위해 실행 중인 모니터의 대다수(예: 1개, 2개 중 3개), 5개 중 4개, 6개 중 4개 등)이 필요합니다.

mon_force_quorum_join

설명

이전에는 맵에서 제거된 경우에도 쿼럼에 참여하도록 강제 모니터링

유형

부울

기본값

False

3.1.3. 일관성

Ceph 구성 파일에 모니터 설정을 추가하는 경우 **Ceph** 모니터의 일부 아키텍처 측면을 알고 있어야 합니다. **Ceph**는 클러스터 내의 다른 **Ceph** 모니터를 검색할 때 **Ceph** 모니터에 엄격한 일관성 요구 사항을 적용합니다. **Ceph** 클라이언트 및 기타 **Ceph** 데몬은 **Ceph** 구성 파일을 사용하여 모니터를 검색하고, **Ceph** 구성 파일이 아닌 모니터 맵(**monmap**)을 사용하여 서로를 검색합니다.

Ceph 모니터는 **Red Hat Ceph Storage** 클러스터에서 다른 **Ceph** 모니터를 검색할 때 모니터 맵의 로컬 사본을 항상 나타냅니다. **Ceph** 구성 파일 대신 모니터 맵을 사용하면 클러스터가 손상될 수 있는 오류가 발생하지 않습니다(예: 모니터 주소 또는 포트를 지정할 때 **Ceph** 구성 파일에서 오타) 모니터는 검색을 위해 모니터 맵을 사용하고 클라이언트 및 기타 **Ceph** 데몬과 모니터 맵을 공유하므로 모니터 맵에서는 이러한 합의가 유효한지 엄격한 모니터링을 제공합니다.

엄격한 일관성은 모니터 맵에 업데이트에도 적용됩니다. **Ceph** 모니터의 다른 업데이트와 마찬가지로 모니터 맵의 변경 사항은 **Paxos**라는 분산된 합의 알고리즘을 통해 항상 실행됩니다. 쿼럼의 각 모니터에 동일한 버전의 모니터 맵이 있는지 확인하려면 **Ceph** 모니터 추가 또는 삭제와 같은 모니터 맵의 각 업데이트에 대해 **Ceph** 모니터에 동의해야 합니다. 모니터 맵의 업데이트는 증분되므로 **Ceph** 모니터가 버전 및 이전 버전 집합에 대해 최신 동의를 할 수 있습니다. 기록을 유지하면 이전 버전의 모니터 맵이 있는 **Ceph** 모니터를 통해 **Red Hat Ceph Storage** 클러스터의 현재 상태를 확인할 수 있습니다.

Ceph 모니터가 모니터 맵을 통해 대신 **Ceph** 구성 파일을 통해 서로 검색되면 **Ceph** 구성 파일이 자동으로 업데이트 및 배포되지 않기 때문에 추가 위험이 발생할 수 있습니다. **Ceph** 모니터는 실수로 이전 **Ceph** 구성 파일을 사용하거나 **Ceph** 모니터를 인식하지 못하거나 쿼럼이 떨어지거나 **Paxos**가 시스템의 현재 상태를 정확하게 확인할 수 없는 상황을 개발할 수 있습니다.

3.1.4. 부트스트랩 모니터

대부분의 구성 및 배포 사례에서 **Ceph**를 배포하는 툴은 **Red Hat Storage Console** 또는 **Ansible**과 같이 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 클러스터를 실행할 수 있으므로 모니터를 부트스트래핑할 때 오브젝트 저장소의 고유 **ID**를 지정해야 합니다. 예를 들어 **Red Hat Storage Console** 또는 **Ansible**을 사용하면 파일 시스템 식별자가 생성되지만 **fsid** 도 수동으로 지정할 수 있습니다.

- 모니터 **ID:** 모니터 **ID**는 클러스터 내의 각 모니터에 할당된 고유 **ID**입니다. 영숫자 값이며, 일반적으로 식별자는 알파벳 증가(예: **,b** 등) 를 따릅니다. 이는 **Ceph** 구성 파일(예: **[mon.a], [mon.b]),** 배포 툴을 통해 설정하거나 **ceph** 명령을 사용하여 설정할 수 있습니다.

- 키: 모니터에는 보안 키가 있어야 합니다.

3.2. 모니터 구성

전체 클러스터에 구성 설정을 적용하려면 **[global]** 섹션에 구성 설정을 입력합니다. 클러스터의 모든 모니터에 구성 설정을 적용하려면 **[mon]** 섹션에 구성 설정을 입력합니다. 특정 모니터에 구성 설정을 적용하려면 모니터 인스턴스를 지정합니다(예: **[mon.a]**). 규칙에 따라 인스턴스 이름을 모니터링하면 알파 표기법을 사용합니다.

```
[global]

[mon]

[mon.a]

[mon.b]

[mon.c]
```

3.2.1. 최소 구성

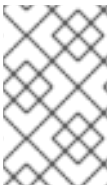
Ceph 구성 파일의 **Ceph** 모니터의 베어 최소 모니터 설정에는 **DNS** 및 모니터 주소가 구성되지 않은 경우 각 모니터의 호스트 이름이 포함됩니다. **[mon]** 아래에서 또는 특정 모니터에 대한 항목 아래에서 설정할 수 있습니다.

```
[mon]
mon_host = hostname1,hostname2,hostname3
mon_addr = 10.0.0.10:6789,10.0.0.11:6789,10.0.0.12:6789
```

또는

```
[mon.a]
host = hostname1
mon_addr = 10.0.0.10:6789
```

-



참고

모니터의 최소 구성에서는 배포 툴이 **fsid** 및 **mon.** 키를 생성하는 것으로 가정합니다.

중요

Ceph 클러스터를 배포한 후에는 모니터의 **IP** 주소를 변경하지 마십시오.

RHCS 2.4부터 클러스터가 **DNS** 서버를 통해 모니터를 조회하도록 구성된 경우 **Ceph**에는 **mon_host**가 필요하지 않습니다. **DNS** 조회를 위해 **Ceph** 클러스터를 구성하려면 **Ceph** 구성 파일에서 **mon_dns_srv_name** 설정을 설정합니다.

mon_dns_srv_name

설명

모니터 호스트/**addresses**에 대해 **DNS**를 쿼리하는 데 사용되는 서비스 이름입니다.

유형

문자열

기본값

ceph-mon

설정한 후 **DNS**를 구성합니다. **DNS** 영역의 모니터에 대한 **IPv4 (A)** 또는 **IPv6 (AAAA)** 레코드를 만듭니다. 예를 들어 다음과 같습니다.

```
#IPv4
mon1.example.com. A 192.168.0.1
mon2.example.com. A 192.168.0.2
mon3.example.com. A 192.168.0.3

#IPv6
mon1.example.com. AAAA 2001:db8::100
mon2.example.com. AAAA 2001:db8::200
mon3.example.com. AAAA 2001:db8::300
```

여기서 **example.com** 은 **DNS** 검색 도메인입니다.

그런 다음 세 개의 모니터를 가리키는 **mon_dns_srv_name** 구성 설정을 사용하여 **SRV TCP** 레코드를 만듭니다. 다음 예제에서는 기본값 **ceph-mon** 값을 사용합니다.

```
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 6789 mon1.example.com.
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 6789 mon2.example.com.
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 6789 mon3.example.com.
```

모니터는 기본적으로 포트 **6789** 에서 실행되며 우선 순위 및 가중치는 모두 각각 **10** 및 **60** 으로 설정되어 있습니다.

3.2.2. 클러스터 ID

각 **Red Hat Ceph Storage** 클러스터에는 고유한 식별자**(fsid)**가 있습니다. 지정된 경우 일반적으로 구성 파일의 **[global]** 섹션에 표시됩니다. 배포 도구는 일반적으로 **fsid** 를 생성하여 모니터 맵에 저장하므로 해당 값이 구성 파일에 표시되지 않을 수 있습니다. **fsid** 를 사용하면 동일한 하드웨어에서 여러 클러스터에 대한 데몬을 실행할 수 있습니다.

fsid

설명

클러스터 **ID**입니다. 클러스터당 하나씩.

유형

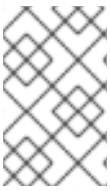
UUID

필수 항목

네, 필요합니다.

기본값

해당 없음 지정하지 않는 경우 배포 툴에 의해 생성될 수 있습니다.



참고

사용자를 위한 배포 툴을 사용하는 경우 이 값을 설정하지 마십시오.

3.2.3. 초기 멤버

고가용성을 보장하기 위해 **3**개 이상의 **Ceph** 모니터가 있는 프로덕션 **Red Hat Ceph Storage** 클러스

터를 실행하는 것이 좋습니다. 여러 모니터를 실행하는 경우 쿼럼을 구축하기 위해 클러스터의 멤버여야 하는 초기 모니터를 지정할 수 있습니다. 이로 인해 클러스터가 온라인 상태가 되는 데 걸리는 시간이 줄어들 수 있습니다.

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

mon_initial_members

설명

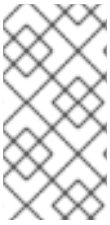
시작 중에 클러스터의 초기 모니터 ID입니다. 지정된 경우 Ceph에는 초기 쿼럼(예: 3)을 만들기 위해 홀수의 모니터가 필요합니다.

유형

문자열

기본값

없음



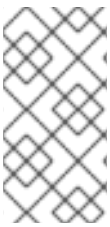
참고

쿼럼을 설정하려면 클러스터의 *대부분의* 모니터가 서로 연결할 수 있어야 합니다. 이 설정으로 쿼럼을 설정하는 초기 모니터 수를 줄일 수 있습니다.

3.2.4. data

Ceph는 Ceph가 데이터를 모니터링하는 기본 경로를 제공합니다. Red Hat Ceph Storage 프로덕션 환경에서 최적의 성능을 발휘하려면 별도의 호스트에서 Ceph 모니터를 실행하고 Ceph OSD와의 드라이브를 사용하는 것이 좋습니다. Ceph 모니터는 종종 fsync() 함수를 호출하여 Ceph OSD 워크로드를 방해할 수 있습니다.

Ceph 모니터는 데이터를 키-값 쌍으로 저장합니다. 데이터 저장소를 사용하면 Ceph 모니터가 Paxos를 통해 손상된 버전을 복구할 수 없으며 다른 이점 중에서 하나의 단일 원자 배치에서 여러 수정 작업을 수행할 수 있습니다.



참고

Red Hat은 기본 데이터 위치를 변경하는 것을 권장하지 않습니다. 기본 위치를 수정하는 경우 구성 파일의 [mon] 섹션에서 설정하여 Ceph 모니터 간에 균일하게 만듭니다.

mon_data

설명

모니터의 데이터 위치입니다.

유형

문자열

기본값

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

mon_data_size_warn

설명

모니터의 데이터 저장소가 이 임계값에 도달하면 **Ceph**는 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다. 기본값은 **15GB**입니다.

유형

정수

기본값

15*1024*1024*1024*`

mon_data_avail_warn

설명

모니터 데이터 저장소의 사용 가능한 디스크 공간이 이 백분율보다 작거나 같으면 **Ceph**에서 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다.

유형

정수

기본값

30

mon_data_avail_crit

설명

모니터 데이터 저장소의 사용 가능한 디스크 공간이 이 백분율보다 작거나 같으면 **Ceph**에

서 클러스터 로그에서 **HEALTH_ERR** 상태를 발행합니다.

유형

정수

기본값

5

mon_warn_on_cache_pools_without_hit_sets

설명

캐시 풀에 **hit_set_type paramater**가 설정되지 않은 경우 **Ceph**는 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다. 자세한 내용은 풀 값을 참조하십시오.

유형

부울

기본값

True

mon_warn_on_crush_straw_calc_version_zero

설명

ArgoCD의 **straw_calc_version** 이 **0**이면 **Ceph**에서 클러스터 로그에 **HEALTH_WARN** 상태를 발행합니다. 자세한 내용은 **autoscale** 튜닝 가능 항목을 참조하십시오.

유형

부울

기본값

True

mon_warn_on_legacy_crush_tunables

설명

nmap 튜닝 가능 항목이 너무 오래 된 경우 **Ceph**에서 클러스터 로그에 **HEALTH_WARN** 상태를 발행합니다(**mon_min_crush_required_version**미만).

유형

부울

기본값

True

mon_crush_min_required_version

설명

이 설정은 클러스터에 필요한 최소 조정 가능한 프로필 버전을 정의합니다. 자세한 내용은 autoscale 튜닝 가능 항목을 참조하십시오.

유형

문자열

기본값

firefly

mon_warn_on_osd_down_out_interval_zero

설명

mon_osd_down_out_interval 설정이 0이면 Ceph에서 HEALTH_WARN 상태를 발행합니다. noout 플래그가 설정될 때 Leader가 유사한 방식으로 동작하기 때문입니다. 관리자는 noout 플래그를 설정하여 클러스터의 문제를 더 쉽게 해결할 수 있습니다. Ceph에서 경고를 발행하여 관리자가 설정이 0인지 확인합니다.

유형

부울

기본값

True

mon_cache_target_full_warn_ratio

설명

Ceph는 cache_target_full 과 target_max_object 의 비율 간에 경고를 발행합니다.

유형

float

기본값

0.66

mon_health_data_update_interval

설명

쿼럼의 모니터가 해당 피어와 상태를 공유하는 빈도(초)입니다. 음수는 상태 업데이트를 비활성화합니다.

유형

float

기본값

60

mon_health_to_clog

설명

이 설정을 사용하면 Ceph가 정기적으로 상태 요약을 클러스터 로그에 보낼 수 있습니다.

유형

부울

기본값

True

mon_health_to_clog_tick_interval

설명

모니터가 클러스터 로그에 상태 요약을 보내는 빈도(초)입니다. 무의미한 수치는 이를 비활성화합니다. 현재 상태 요약이 마지막으로 비어 있거나 마지막으로 동일하면 모니터에서 상태를 클러스터 로그로 전송하지 않습니다.

유형

정수

기본값

3600

mon_health_to_clog_interval

설명

모니터가 클러스터 로그에 상태 요약을 보내는 빈도(초)입니다. 무의미한 수치는 이를 비활성화합니다. 모니터는 항상 요약을 클러스터 로그에 보냅니다.

유형

정수

기본값

60

3.2.5. 스토리지 용량

Red Hat Ceph Storage 클러스터가 최대 용량(**mon_osd_full_ratio** 매개 변수로 지정)에 가까운 경우, **Ceph**는 데이터 손실을 방지하기 위한 안전 조치로 **Ceph OSD**에 쓰거나 읽을 수 없습니다. 따라서 프로덕션 **Red Hat Ceph Storage** 클러스터가 전체 비율에 접근하도록 하는 것은 고가용성을 저하하기 때문에 좋지 않습니다. 기본 전체 비율은 **.95** 또는 용량의 **95%**입니다. 이는 소수의 **OSD**가 있는 테스트 클러스터에 대한 매우 공격적인 설정입니다.

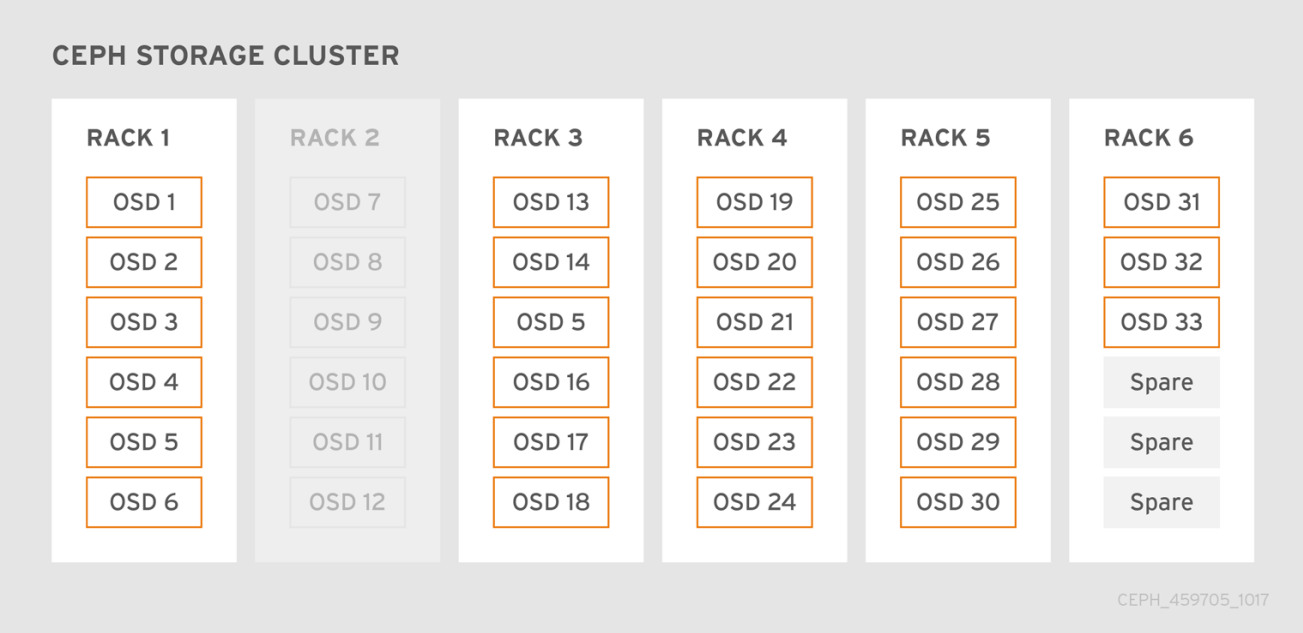
작은 정보

클러스터를 모니터링할 때 거의 전체 비율과 관련된 경고를 경고하십시오. 즉, 하나 이상의 **OSD**에 오류가 발생하면 일부 **OSD**에서 일시적인 서비스 중단이 발생할 수 있습니다. 스토리지 용량을 늘리려면 **OSD**를 더 추가하는 것이 좋습니다.

테스트 클러스터에 대한 일반적인 시나리오에는 시스템 관리자가 **Red Hat Ceph Storage** 클러스터에서 **Ceph OSD**를 제거하여 클러스터 재조정이 포함됩니다. 그런 다음 다른 **Ceph OSD**를 제거하여 **Red Hat Ceph Storage** 클러스터가 결국 전체 비율 및 잠금에 도달할 때까지 이러한 **OSD**를 제거합니다.

Red Hat은 테스트 클러스터가 있는 경우에도 약간의 용량 계획을 수립할 것을 권장합니다. 계획을 사용하면 고가용성을 유지하기 위해 필요한 예비 용량의 용량을 측정할 수 있습니다. 이러한 **Ceph OSD**를 즉시 교체하지 않고 클러스터가 활성 **+** 클린 상태로 복구할 수 있는 일련의 **Ceph OSD** 실패를 계획하는 것이 좋습니다. 활성 **+** 성능 저하 상태에서 클러스터를 실행할 수 있지만 정상적인 운영 조건에 적합하지 않습니다.

다음 다이어그램은 호스트당 하나의 Ceph OSD가 있는 33개의 Ceph 노드, 각 Ceph OSD 데몬에서 읽고 3TB 드라이브에 쓰는 간단한 Red Hat Ceph Storage 클러스터를 보여줍니다. 따라서 Red Hat Ceph Storage 클러스터에는 최대 99TB의 용량을 사용할 수 있습니다. mon osd 전체 비율 0.95 의 경우 Red Hat Ceph Storage 클러스터가 남은 용량의 5TB에 속하는 경우 클러스터는 Ceph 클라이언트가 데이터를 읽고 쓸 수 없습니다. 따라서 Red Hat Ceph Storage 클러스터의 운영 용량은 99TB가 아닌 95% 입니다.



이러한 클러스터에서는 하나 또는 두 개의 OSD가 실패하는 것이 일반적입니다. 덜 빈번하지만 합리적인 시나리오에는 랙의 라우터 또는 전원 공급이 실패하여 여러 OSD가 동시에 중단됩니다(예: OSD 7-12). 이러한 시나리오에서는 추가 OSD가 있는 호스트를 짧은 순서로 추가하는 경우에도 작동 상태를 유지하고 활성 + 클린 상태를 유지할 수 있는 클러스터를 계속 수행해야 합니다. 용량 사용률이 너무 높으면 데이터가 손실되지 않을 수 있지만 클러스터의 용량 사용률이 전체 비율을 초과하면 장애 도메인 내의 중단을 해결하는 동안 데이터 가용성을 저하시킬 수 있습니다. 이러한 이유로 Red Hat은 적어도 일부 대략적인 용량 계획을 권장합니다.

클러스터의 두 가지 번호를 식별합니다.

- OSD 수
- 클러스터의 총 용량

클러스터 내에서 OSD의 평균 용량을 확인하려면 클러스터의 총 용량을 클러스터의 OSD 수로 나눕니다. 일반 작업 중에 동시에 실패할 것으로 예상되는 OSD 수(비교적 적은 수)로 이 수를 곱하는 것이 좋습니다. 마지막으로, 최대 운영 용량에 도달하기 위해 클러스터의 용량을 전체 비율로 곱합니다. 그런 다음, 적절한 전체 비율로 도달하지 못할 것으로 예상되는 OSD의 데이터 양을 뺀 것입니다. OSD 실패 수(예: OSD 랙)와 함께 foregoing 프로세스를 반복하여 거의 완전한 비율을 위해 적절한 번호로 도달합니다.

```
[global]
...
mon_osd_full_ratio = .80
mon_osd_nearfull_ratio = .70
```

mon_osd_full_ratio

설명

OSD 전에 사용된 디스크 공간의 백분율은 전체로 간주됩니다.

유형

float:

기본값

.95

mon_osd_nearfull_ratio

설명

OSD 이전에 사용된 디스크 공간의 백분율은 거의 완전한 것으로 간주됩니다.

유형

float

기본값

.85

작은 정보

일부 OSD가 가득 차 있지만 용량이 충분한 경우 거의 완전한 OSD의 NetNamespace 가중치에 문제가 있을 수 있습니다.

3.2.6. 하트비트

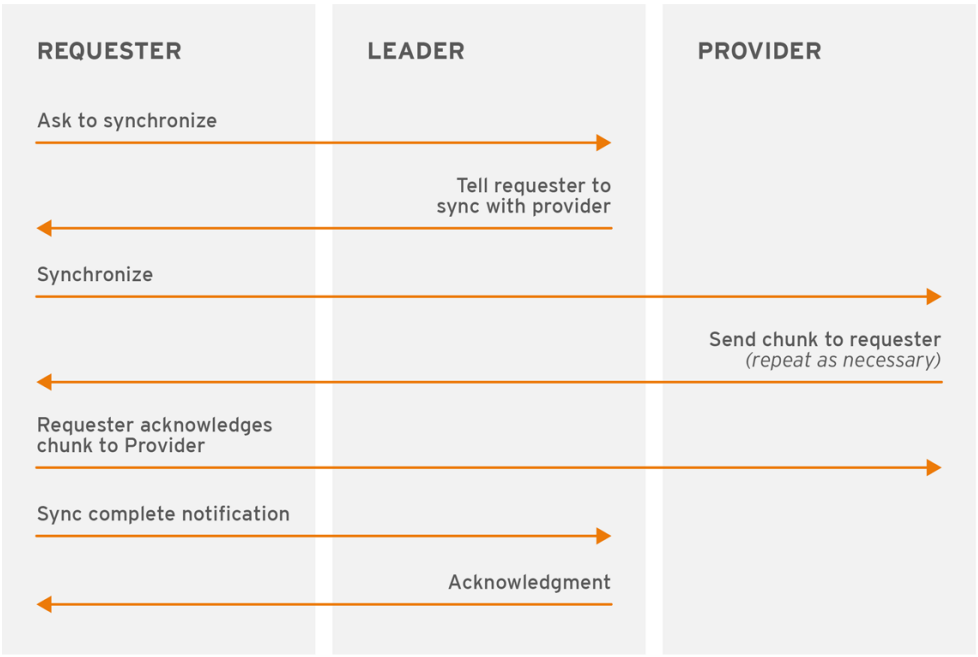
Ceph 모니터는 각 OSD의 보고서가 필요하고, OSD에서 인접한 OSD 상태에 대한 보고서를 수신하는 방식으로 클러스터에 대해 알고 있습니다. Ceph는 모니터와 OSD 간의 상호 작용을 위해 적절한 기본 설정을 제공하지만 필요에 따라 수정할 수 있습니다.

3.2.7. 저장소 동기화 모니터링

권장되는 여러 모니터가 있는 프로덕션 클러스터를 실행하면 각 모니터에서 인접한 모니터의 최신 버전의 클러스터 맵이 있는지 확인합니다. 예를 들어, 주변 모니터의 맵은 인스턴트 모니터 맵에서 가장 최신의 에포치보다 높은 하나 이상의 에포치 모니터와 함께, **For example, a map in a neighboring monitor on one or more epoch numbers higher than the most current epoch in the instant monitor** 클러스터의 한 모니터는 쿼럼을 떠나야 하는 시점까지 다른 모니터 뒤에 떨어질 수 있으며 클러스터의 최신 정보를 검색한 다음 쿼럼에 다시 참여하도록 동기화할 수 있습니다. 동기화를 위해 모니터는 다음 세 가지 역할 중 하나를 가정할 수 있습니다.

- 리더: 리더는 클러스터 맵의 최신 **Paxos** 버전을 달성하는 첫 번째 모니터입니다.
- 공급자: 공급자는 클러스터 맵의 최신 버전이 있지만 최신 버전을 처음으로 달성하지 않은 모니터입니다.
- 요청자는 리더 뒤에 떨어진 모니터이며 쿼럼에 다시 참여하기 전에 클러스터에 대한 최신 정보를 검색해야 합니다.

이러한 역할을 통해 리더는 동기화 작업을 공급업체에 위임할 수 있으므로 동기화 요청이 리더 과부하와 성능을 개선할 수 있습니다. 다음 다이어그램에서 요청자는 다른 모니터 뒤에 떨어졌습니다. 요청자는 리더가 동기화하도록 요청하고, 리더는 공급자와 동기화하도록 요청자에게 지시합니다.



동기화는 새 모니터가 클러스터에 참여할 때 항상 수행됩니다. 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. 동기화 중에 이러한 상황이 발생하면(예: 공급자가 리더 뒤에 있는 경우) 공급자는 요청자를 사용하여 동기화를 종료할 수 있습니다.

동기화가 완료되면 **Ceph**에는 클러스터 전체에서 트리밍이 필요합니다. 트리밍에서는 배치 그룹이 활성 **+** 클린이어야 합니다.

mon_sync_trim_timeout

설명, 유형

double

기본값

30.0

mon_sync_heartbeat_timeout

설명, 유형

double

기본값

30.0

mon_sync_heartbeat_interval

설명, 유형

double

기본값

5.0

mon_sync_backoff_timeout

설명, 유형

double

기본값

30.0

mon_sync_timeout

설명

모니터가 동기화 공급자에서 다음 업데이트 메시지를 대기할 때까지 대기하는 시간(초)은 다

시 가동 및 부트 스트랩을 포기합니다.

유형

double

기본값

30.0

mon_sync_max_retries

설명, 유형

정수

기본값

5

mon_sync_max_payload_size

설명

동기화 페이로드의 최대 크기(바이트 단위)입니다.

유형

32비트 정수

기본값

1045676

paxos_max_join_drift

설명

먼저 모니터 데이터 저장소를 동기화하기 전에 최대 **Paxos** 반복 횟수입니다. 모니터가 피어가 너무 멀리 떨어져 있음을 발견하면 이동 전에 데이터 저장소와 먼저 동기화됩니다.

유형

정수

기본값

10

paxos_stash_full_interval

설명

PaxosService 상태의 전체 사본을 배치하는 빈도(커밋)입니다. 현재 이 설정은 mds,mon,auth 및 mgr PaxosServices에만 영향을 미칩니다.

유형

정수

기본값

25

paxos_propose_interval

설명

맵 업데이트를 제공하기 전에 이 시간 간격에 대한 업데이트를 수집합니다.

유형

double

기본값

1.0

paxos_min

설명

주변을 유지할 수 있는 최소 paxos 상태 수

유형

정수

기본값

500

paxos_min_wait

설명

비활성 기간 후에 업데이트를 수집하는 최소 시간입니다.

유형

double

기본값

0.05

paxos_trim_min

설명

트리밍하기 전에 허용되는 추가 제안 수

유형

정수

기본값

250

paxos_trim_max

설명

한 번에 트리밍하기 위한 추가 제안의 최대 수

유형

정수

기본값

500

paxos_service_trim_min

설명

트리밍을 트리거할 최소 버전 수(0이 사용 안 함)

유형

정수

기본값

250

paxos_service_trim_max

설명

단일 제안 중에 트리밍할 최대 버전 수(0사용 안 함)

유형

정수

기본값

500

mon_max_log_epochs

설명

단일 제안 중에 트리밍할 최대 로그 양입니다.

유형

정수

기본값

500

mon_max_pgmap_epochs

설명

단일 제안 중에 트리밍할 pgmap epochs의 최대 양

유형

정수

기본값

500

mon_mds_force_trim_to

설명

force monitor to trimmaps to this point (0 이 시점으로 강제 모니터링(0으로 비활성화)

유형

정수

기본값

0

mon_osd_force_trim_to

설명

특정 epoch에서 정리되지 않은 PG가 있어도 이 시점으로 트리밍 맵을 강제 모니터링(0으로 비활성화)

유형

정수

기본값

0

mon_osd_cache_size

설명

기본 저장소의 캐시를 사용하지 않는 osdmaps 캐시의 크기

유형

정수

기본값

10

mon_election_timeout

설명

선거에서 몇 초 안에 모든 **ACK**에 대한 대기 시간을 최대로 설정합니다.

유형

float

기본값

5

mon_lease

설명

모니터 버전에서 리스의 길이(초)입니다.

유형

float

기본값

5

mon_lease_renew_interval_factor

설명

Mon lease * mon lease renew interval factor 는 리더가 다른 모니터 임대를 갱신할 수 있는 간격이 됩니다. 인수는 **1.0** 보다 작아야 합니다.

유형

float

기본값

0.6

mon_lease_ack_timeout_factor

설명

리더는 공급자가 임대 확장을 승인할 수 있도록 **mon lease * mon lease ack timeout factor** 를 기다립니다.

유형

float

기본값

2.0

mon_accept_timeout_factor

설명

리더는 요청자가 **Paxos** 업데이트를 수락하기 위해 **mon lease * mon**은 시간 초과 요인 을 허용합니다. 또한 비슷한 목적으로 **Paxos** 복구 단계에서 사용됩니다.

유형

float

기본값

2.0

mon_min_osdmap_epochs

설명

항상 유지할 최소 **OSD** 맵의 수입니다.

유형

32비트 정수

기본값

500

mon_max_pgmap_epochs

설명

모니터가 유지해야 하는 최대 **PG** 맵 수입니다.

유형

32비트 정수

기본값

500

mon_max_log_epochs

설명

모니터가 유지해야 하는 최대 로그 epoch 수입니다.

유형

32비트 정수

기본값

500

3.2.8. clock

Ceph 데몬은 중요한 메시지를 서로 전달합니다. 이 메시지는 데몬이 시간 제한 임계값에 도달하기 전에 처리해야 합니다. Ceph 모니터의 시계가 동기화되지 않으면 여러 문제가 발생할 수 있습니다. 예를 들어 다음과 같습니다.

- 수신한 메시지를 무시하는 데몬(예: 타임스탬프 오래된).
- 시간 초과 메시지가 시간 내에 수신되지 않은 경우 너무 빨리 또는 늦게 트리거됩니다.

자세한 내용은 스토어 동기화 모니터링 을 참조하십시오.

작은 정보

Ceph 모니터 호스트에 NTP를 설치하여 모니터 클러스터가 동기화 시계로 작동하는지 확인합니다.

불일치가 아직 해롭지 않은 경우에도 NTP에서 클럭 드리프트를 볼 수 있습니다. NTP가 적절한 수준의 동기화를 유지하더라도 Ceph 클럭 드리프트 및 클럭 스큐w 경고가 트리거될 수 있습니다. 시계 드리프트를 늘리는 것은 이러한 상황에서 용감될 수 있습니다. 그러나 워크로드, 네트워크 대기 시간, 기본 시간 초과로 재정의를 구성하는 여러 요인이 Paxos 보장을 손상시키지 않고 허용되는 클럭 드리프트 수준에 영향을 미칠 수 있습니다. ???

Ceph는 허용 가능한 값을 찾을 수 있도록 다음과 같은 조정 가능한 옵션을 제공합니다.

clock_offset

설명

시스템 시계를 오프셋하는 방법. 자세한 내용은 Clock.cc 를 참조하십시오.

유형

double

기본값

0

mon_tick_interval

설명

모니터의 틱 간격(초)입니다.

유형

32비트 정수

기본값

5

mon_clock_drift_allowed

설명

모니터 간에 허용되는 클럭 드리프트(초)입니다.

유형

float

기본값

.050

mon_clock_drift_warn_backoff

설명

클럭 드리프트 경고에 대한 지수 백오프입니다.

유형

float

기본값

5

mon_timecheck_interval

설명

리더의 시간 점검 간격(초)입니다.

유형

float

기본값

300.0

mon_timecheck_skew_interval

설명

리더의 경우 시간 점검 간격(초)(초)입니다(초)입니다.

유형

float

기본값

30.0

3.2.9. 클라이언트

mon_client_hunt_interval

설명

클라이언트는 연결을 설정할 때까지 N 초마다 새 모니터를 시도합니다.

유형

double

기본값

3.0

mon_client_ping_interval

설명

클라이언트는 N 초마다 모니터를 ping합니다.

유형

double

기본값

10.0

mon_client_max_log_entries_per_message

설명

모니터에서 클라이언트 메시지별로 생성되는 최대 로그 항목 수입니다.

유형

정수

기본값

1000

mon_client_bytes

설명

메모리에서 허용되는 클라이언트 메시지 데이터의 양(바이트 단위)입니다.

유형

64비트 Integer 서명되지 않음

기본값

100UL< 20

3.3. 기타

mon_max_osd

설명

클러스터에 허용되는 최대 OSD 수입니다.

유형

32비트 정수

기본값

10000

mon_globalid_prealloc

설명

클러스터의 클라이언트 및 데몬에 대한 사전 할당하기 위한 글로벌 ID 수입니다.

유형

32비트 정수

기본값

100

mon_sync_fs_threshold

설명

지정된 수의 오브젝트를 작성할 때 파일 시스템과 동기화합니다. 이를 비활성화하려면 0 으로 설정합니다.

유형

32비트 정수

기본값

5

mon_subscribe_interval

설명

서브스크립션의 새로 고침 간격(초)입니다. 서브스크립션 메커니즘을 사용하면 클러스터 맵 및 로그 정보를 가져올 수 있습니다.

유형

double

기본값

300

mon_stat_smooth_intervals

설명

Ceph는 마지막 N PG 맵에 대한 통계를 원활하게 수행할 수 있습니다.

유형

정수

기본값

2

mon_probe_timeout

설명

부트 스트랩하기 전에 모니터가 피어를 찾기 위해 대기하는 시간(초)입니다.

유형

double

기본값

2.0

mon_daemon_bytes

설명

메타데이터 서버 및 OSD 메시지의 메시지 메모리 용량(바이트 단위)입니다.

유형

64비트 Integer 서명되지 않음

기본값

400UL < 20

mon_max_log_entries_per_event

설명

이벤트당 최대 로그 항목 수입니다.

유형

정수

기본값

4096

mon_osd_prime_pg_temp

설명

OSD가 클러스터로 돌아 오면 이전 OSD를 사용하여 PGMap을 활성화하거나 비활성화합니다. true 설정을 사용하면 클라이언트가 해당 PG 피어로서 OSD에서 새로 추가될 때까지 이전 OSD를 계속 사용합니다.

유형

부울

기본값

true

mon_osd_prime_pg_temp_max_time

설명

모니터에서 OSD가 클러스터에 다시 돌아올 때 PGMap을 차용하는 데 소비해야 하는 시간(초)입니다.

유형

float

기본값

0.5

mon_osd_prime_pg_temp_max_time_estimate

설명

모든 PG를 병렬로 소장하기 전에 각 PG에서 보낸 최대 추정 시간을 추정합니다.

유형

float

기본값

0.25

mon_osd_allow_primary_affinity

설명

osdmap에 primary_affinity 를 설정할 수 있습니다.

유형

부울

기본값

False

mon_osd_pool_ec_fast_read

설명

풀에서 빠르게 읽을 수 있는지 여부입니다. fast_read 가 생성 시 지정되지 않은 경우 새로 생성된 삭제 풀의 기본 설정으로 사용됩니다.

유형

부울

기본값

False

mon_mds_skip_sanity

설명

FSMap에 대한 안전 어설션을 건너뜁니다(한 번 계속하려는 버그의 경우). monitor는 FSMap Sanity 검사에 실패하는 경우 종료되지만 이 옵션을 활성화하여 비활성화할 수 있습니다.

유형

부울

기본값

False

mon_max_mdsmap_epochs

설명

단일 제안 중에 트리밍할 최대 **mdsmap epochs**입니다.

유형

정수

기본값

500

mon_config_key_max_entry_size

설명

config-key 항목의 최대 크기(바이트 단위)

유형

정수

기본값

4096

mon_scrub_interval

설명

저장된 체크섬을 모든 저장된 키의 계산된 키와 비교하여 모니터가 해당 저장소의 저장소를 얼마나 자주 (초)입니까.

유형

정수

기본값

3600*24

mon_scrub_max_keys

설명

매번 스크럽할 최대 키 수입니다.

유형

정수

기본값

100

mon_compact_on_start

설명

ceph-mon start에서 **Ceph Monitor** 저장소로 사용되는 데이터베이스를 압축합니다. 수동 압축을 사용하면 모니터 데이터베이스를 축소하고 일반 압축이 작동하지 않는 경우 성능이 향상됩니다.

유형

부울

기본값

False

mon_compact_on_bootstrap

설명

부트스트랩의 에 **Ceph Monitor** 저장소로 사용된 데이터베이스를 압축합니다. 모니터는 부

트스트랩 후 쿼럼을 생성하기 위해 서로 프로빙하기 시작합니다. 쿼럼에 참여하기 전에 시간이 초과되면 다시 시작되고 자체 부트스트랩이 다시 시작됩니다.

유형

부울

기본값

False

mon_compact_on_trim

설명

이전 상태를 트리밍할 때 특정 접두사(**paxos** 포함)를 압축합니다.

유형

부울

기본값

True

mon_cpu_threads

설명

모니터에서 **CPU** 집약적인 작업을 수행하는 스레드 수입니다.

유형

부울

기본값

True

mon_osd_mapping_pgs_per_chunk

설명

배치 그룹에서 청크로 **OSD**로의 매핑을 계산합니다. 이 옵션은 청크당 배치 그룹 수를 지정합니다.

유형

정수

기본값

4096

mon_osd_max_split_count

설명

"결합된" OSD당 최대 PG 수를 분할할 수 있습니다. 풀의 pg_num 을 늘리면 해당 풀을 제공하는 모든 OSD에서 배치 그룹이 분할됩니다. PG 분할에서 극단적인 멀티플라이어를 방지하려고 합니다.

유형

정수

기본값

300

mon_session_timeout

설명

모니터는 이 시간 동안 비활성 세션이 유휴 상태로 유지됩니다.

유형

정수

기본값

300

rados_mon_op_timeout

설명

RADOS 작업에서 오류를 반환하기 전에 RADOS에서 Ceph 모니터의 응답을 대기하는 시간(초)입니다. 값이 0이면 제한이 없음을 의미합니다.

유형

double

기본값

0

4장. RUNTIMECLASS 구성 참조

Gradle 프로토콜은 기본적으로 활성화되어 있습니다. 암호화 인증은 일반적으로 매우 낮지만 일부 계산 비용이 많이 듭니다. 클라이언트 및 서버 호스트를 연결하는 네트워크 환경이 매우 안전하고 인증을 허용할 수 없는 경우 이를 비활성화할 수 있습니다. 그러나 **Red Hat**은 인증을 사용하는 것이 좋습니다.

참고

인증을 비활성화하면 메시지 가로채기(**man-in-the-middle**) 공격으로 인한 클라이언트 및 서버 메시지가 변경될 위험이 있으므로 심각한 보안 문제가 발생할 수 있습니다.

4.1. 수동

수동으로 클러스터를 배포할 때 모니터를 수동으로 부트스트랩하고 **client.admin** 사용자 및 인증 키를 생성해야 합니다. **Ceph**를 수동으로 배포하려면 지식베이스 문서 를 참조하십시오. 모니터 부트 스트랩은 **Chef, Puppet, Juju** 등과 같은 타사 배포 툴을 사용할 때 수행해야 하는 논리적 단계입니다.

4.2. CEPHX 활성화 및 비활성화

Cephx를 활성화하려면 모니터 및 **OSD**에 키를 배포해야 합니다. 단순히 / **off**에서 **Cephx**를 모으려면 부트스트랩 절차를 반복할 필요가 없습니다.

4.2.1. Cephx 활성화

RuntimeClass가 활성화되면 **Ceph**는 기본 검색 경로에서 인증 키를 찾습니다. 여기에는 **/etc/ceph/$cluster.$name.keyring** 이 포함됩니다. **Ceph** 구성 파일의 **[global]** 섹션에 인증 키 옵션을 추가하여 이 위치를 덮어쓸 수 있지만 이는 권장되지 않습니다.

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

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

```
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 콘텐츠가 지워집니다. 배포 툴이 이미 이를 수행한 경우 이 단계를 수행하지 마십시오.

2. 모니터 클러스터의 인증 키를 생성하고 모니터 시크릿 키를 생성합니다.

```
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.

```
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
```

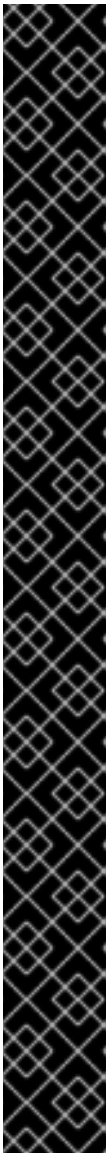
5. 기본적으로 **Gradle** 인증 프로토콜이 활성화됩니다.

참고

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

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

6. **Ceph** 클러스터를 시작하거나 다시 시작합니다.



중요

Runtime Class 를 활성화하려면 클러스터를 완전히 다시 시작해야 하거나 클라이언트 **I/O**를 비활성화하는 동안 종료한 다음 시작해야 하므로 다운타임이 필요합니다.

스토리지 클러스터를 재시작하거나 종료하기 전에 다음 플래그를 설정해야 합니다.

```
# ceph osd set noout
# ceph osd set norecover
# ceph osd set norebalance
# ceph osd set nobackfill
# ceph osd set nodown
# ceph osd set pause
```

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

```
# ceph osd unset noout
# ceph osd unset norecover
# ceph osd unset norebalance
# ceph osd unset nobackfill
# ceph osd unset nodown
# ceph osd unset pause
```

4.2.2. Cephx 비활성화

다음 절차에서는 **Cephx**를 비활성화하는 방법을 설명합니다. 클러스터 환경이 상대적으로 안전한 경우 실행 중인 인증의 계산 비용을 오프셋할 수 있습니다. **Red Hat**은 인증을 활성화하는 것이 좋습니다. 그러나 인증을 일시적으로 비활성화하려면 설정 또는 문제 해결 중에 더 쉬울 수 있습니다.

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

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

2. **Ceph** 클러스터를 시작하거나 다시 시작합니다.

4.3. 구성 설정

4.3.1. Enable

auth_cluster_required

설명

활성화된 경우 **Red Hat Ceph Storage** 클러스터 데몬(즉, **ceph-mon** 및 **ceph-osd**)은 서로 인증해야 합니다. 유효한 설정은 **Gradle** 또는 **none** 입니다.

유형

문자열

필수 항목

없음

기본값

RuntimeClass.

auth_service_required

설명

활성화된 경우 **Red Hat Ceph Storage** 클러스터 데몬을 사용하려면 **Ceph** 클라이언트가 **Ceph** 서비스에 액세스하기 위해 **Red Hat Ceph Storage** 클러스터를 인증해야 합니다. 유효한 설정은 **Gradle** 또는 **none** 입니다.

유형

문자열

필수 항목

없음

기본값

RuntimeClass.

auth_client_required

설명

활성화된 경우 **Ceph** 클라이언트에 **Ceph** 클라이언트를 인증하려면 **Red Hat Ceph Storage** 클러스터가 필요합니다. 유효한 설정은 **Gradle** 또는 **none** 입니다.

유형

문자열

필수 항목

없음

기본값

RuntimeClass.

4.3.2. 키

인증이 활성화된 **Ceph**를 실행하면 **ceph** 관리 명령 및 **Ceph** 클라이언트에 **Ceph** 스토리지 클러스터에 액세스하기 위해 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. **/etc/ceph/** 디렉터리에 인증 키를 포함하는 경우 **Ceph** 구성 파일에 인증 키 항목을 지정할 필요가 없습니다.

Red Hat은 **client.admin** 키가 포함되어 있으므로 관리 명령을 실행할 노드에 **Red Hat Ceph Storage** 클러스터 인증 키 파일을 복사하는 것이 좋습니다. 이렇게 하려면 **root** 로 다음 명령을 실행하십시오.

```
# scp <user>@<hostname>:/etc/ceph/ceph.client.admin.keyring /etc/ceph/ceph.client.admin.keyring
```

& lt;user >를 호스트에서 사용되는 사용자 이름으로 바꾸고 **< hostname >**을 해당 호스트의 호스트 이름으로 바꿉니다.

참고

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

권장 키 설정을 사용하여 **Ceph** 구성 파일에 키 자체를 지정하거나 **key file** 설정을 사용하는 키 파일의 경로를 지정할 수 있습니다.

인증 키

설명

인증 키 파일의 경로입니다.

유형

문자열

필수 항목

없음

기본값

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

keyfile

설명

키 파일(즉, 키만 포함하는 파일)의 경로입니다.

유형

문자열

필수 항목

없음

기본값

없음

key

설명

키 자체의 텍스트 문자열입니다. 권장되지 않음.

유형

문자열

필수 항목

없음

기본값

없음

4.3.3. 데몬 키 링

관리 사용자 또는 배포 툴은 사용자 인증 키를 생성하는 것과 동일한 방식으로 데몬 인증 키를 생성할 수 있습니다. 기본적으로 **Ceph**는 데이터 디렉터리 내에 데몬 인증 키를 저장합니다. 기본 인증 키 위치 및 데몬이 작동하는 데 필요한 기능은 다음과 같습니다.

ceph-mon

위치

$mon_data/keyring

capabilities

mon 'allow *'

ceph-osd

위치

$osd_data/keyring

capabilities

Mon 'allow profile osd' osd 'allow *'

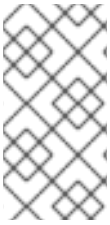
radosgw

위치

$rgw_data/keyring

capabilities

Mon 'rwx' osd 'rwx'



참고

모니터 인증 키(즉, **mon**)에는 키가 있지만 기능은 없으며 클러스터 **auth** 데이터베이스의 일부가 아닙니다.

데몬 데이터 디렉터리는 기본적으로 폼의 디렉터리에 위치합니다.

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

예를 들어 **osd.12** 는 다음과 같습니다.

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

이러한 위치는 재정의할 수 있지만 권장되지는 않습니다.

4.3.4. 서명

Red Hat은 **Ceph**가 초기 인증에 대해 설정된 세션 키를 사용하여 엔티티 간에 지속적인 모든 메시지를 인증할 것을 권장합니다.

Ceph 인증의 다른 부분과 마찬가지로 **Ceph**는 세분화된 제어를 제공하므로 클라이언트와 **Ceph** 간의 서비스 메시지의 서명을 활성화하거나 비활성화할 수 있으며 **Ceph** 데몬 간의 메시지에 대한 서명을 활성화 또는 비활성화할 수 있습니다.

cephx_require_signatures

설명

true 로 설정하면 **Ceph** 클라이언트와 **Red Hat Ceph Storage** 클러스터 간의 모든 메시지 트래픽과 **Red Hat Ceph Storage** 클러스터를 구성하는 데몬 간에 서명이 필요합니다.

유형

부울

필수 항목

없음

기본값

false

cephx_cluster_require_signatures

설명

true 로 설정된 경우 **Ceph**는 **Red Hat Ceph Storage** 클러스터를 구성하는 **Ceph** 데몬 간의 모든 메시지 트래픽에 서명이 필요합니다.

유형

부울

필수 항목

없음

기본값

false

cephx_service_require_signatures

설명

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

유형

부울

필수 항목

없음

기본값

false

cephx_sign_messages

설명

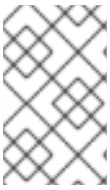
Ceph 버전이 메시지 서명을 지원하는 경우 **Ceph**는 모든 메시지에 서명하여 스푸핑될 수 없도록 합니다.

유형

부울

기본값

true



참고

Ceph 커널 모듈은 아직 서명을 지원하지 않습니다.

4.3.5. 라이브 시간

auth_service_ticket_ttl

설명

Red Hat Ceph Storage 클러스터에서 Ceph 클라이언트에 인증 티켓을 보내면 클러스터는 티켓을 실시간으로 할당합니다.

유형

double

기본값

60*60

5장. 풀, PG 및 NETNAMESPACE 구성 참조

풀을 생성하고 풀의 배치 그룹 수를 설정하면 **Ceph**는 기본값을 구체적으로 재정의하지 않으면 기본값을 기본값으로 사용합니다. **Red Hat**은 일부 기본값을 재정의하는 것이 좋습니다. 특히 풀의 복제본 크기를 설정하고 기본 배치 그룹 수를 재정의합니다. **pool** 명령을 실행할 때 이러한 값을 설정할 수 있습니다. **Ceph** 구성 파일의 **[global]** 섹션에 새 기본값을 추가하여 기본값을 덮어쓸 수도 있습니다.

```
[global]

# By default, Ceph makes 3 replicas of objects. If you want to set 4
# copies of an object as the default value--a primary copy and three replica
# copies--reset the default values as shown in 'osd pool default size'.
# If you want to allow Ceph to write a lesser number of copies in a degraded
# state, set 'osd pool default min size' to a number less than the
# 'osd pool default size' value.

osd_pool_default_size = 4  # Write an object 4 times.
osd_pool_default_min_size = 1 # Allow writing one copy in a degraded state.

# Ensure you have a realistic number of placement groups. We recommend
# approximately 100 per OSD. E.g., total number of OSDs multiplied by 100
# divided by the number of replicas (i.e., osd pool default size). So for
# 10 OSDs and osd pool default size = 4, we'd recommend approximately
# (100 * 10) / 4 = 250.

osd_pool_default_pg_num = 250
osd_pool_default_pgp_num = 250
```

5.1. 설정

mon_allow_pool_delete

설명

모니터에서 풀을 삭제할 수 있습니다. **RHCS 3** 이상 릴리스에서는 데이터 보호를 위한 추가 조치로 모니터가 기본적으로 풀을 삭제할 수 없습니다.

유형

부울

기본값

false

mon_max_pool_pg_num

설명

풀당 최대 배치 그룹 수입니다.

유형

정수

기본값

65536

mon_pg_create_interval

설명

동일한 **Ceph OSD** 데몬에서 **PG** 생성 사이의 시간(초)입니다.

유형

float

기본값

30.0

mon_pg_stuck_threshold

설명

PG가 정지된 것으로 간주될 수 있는 시간(초)입니다.

유형

32비트 정수

기본값

300

mon_pg_min_inactive

설명

mon_pg_stuck_threshold 보다 더 오래 남아 있는 **PG** 수가 이 설정을 초과하는 경우 **Ceph** 에서 클러스터 로그에서 **HEALTH_ERR** 상태를 발행합니다. 기본 설정은 하나의 **PG**입니다. 양수가 아닌 수는 이 설정을 비활성화합니다.

유형

정수

기본값

1

mon_pg_warn_min_per_osd

설명

클러스터 로그에서 **OSD**당 평균 **PG** 수가 이 설정보다 작으면 **Ceph**에서 **HEALTH_WARN** 상태를 발행합니다. 양수가 아닌 수는 이 설정을 비활성화합니다.

유형

정수

기본값

30

mon_pg_warn_max_per_osd

설명

클러스터의 **OSD**당 평균 **PG** 수가 이 설정보다 크면 **Ceph**에서 클러스터 로그에서 **HEALTH_WARN** 상태를 발행합니다. 양수가 아닌 수는 이 설정을 비활성화합니다.

유형

정수

기본값

300

mon_pg_warn_min_objects

설명

클러스터의 총 오브젝트 수가 이 수 아래에 있는지 경고하지 마십시오.

유형

정수

기본값

1000

mon_pg_warn_min_pool_objects

설명

오브젝트 번호가 이 번호 아래에 있는 풀에 경고하지 마십시오.

유형

정수

기본값

1000

mon_pg_check_down_all_threshold

설명

Ceph가 모든 PG를 확인하여 고정되거나 오래되지 않도록 모든 PG를 확인한 후 down OSD의 임계값을 백분율로 설정합니다.

유형

float

기본값

0.5

mon_pg_warn_max_object_skew

설명

풀의 평균 오브젝트 수가 mon pg이라고 경고하는 경우 Ceph에서 클러스터 로그에서 HEALTH_WARN 상태를 발행합니다. 최대 오브젝트 수가 모든 풀의 평균 오브젝트 수입니다. 양수가 아닌 수는 이 설정을 비활성화합니다.

유형

float

기본값

10

mon_delta_reset_interval

설명

Ceph가 PGECDSA를 0으로 재설정하기 전 비활성 시간(초)입니다. Ceph는 관리자가 복구 및 성능 진행 상황을 평가하는 데 도움이 되도록 각 풀에 사용된 공간의 전구를 추적합니다.

유형

정수

기본값

10

mon_osd_max_op_age

설명

HEALTH_WARN 상태를 발행하기 전에 작업이 완료될 수 있는 최대 시간(초)입니다.

유형

float

기본값

32.0

osd_pg_bits

설명

Ceph OSD 데몬당 배치 그룹 비트

유형

32비트 정수

기본값

6

osd_pgp_bits

설명

PG(배치 그룹)를 위한 Ceph OSD Daemon당 비트 수입니다.

유형

32비트 정수

기본값

6

osd_crush_chooseleaf_type

설명

nmap 규칙에서 leaf를 선택하는 데 사용할 버킷 유형입니다. 이름이 아닌 직선 순위를 사용합니다.

유형

32비트 정수

기본값

1. 일반적으로 하나 이상의 Ceph OSD 데몬을 포함하는 호스트입니다.

osd_pool_default_crush_replicated_ruleset

설명

복제된 풀을 만들 때 사용할 기본 NetNamespace 규칙 세트입니다.

유형

8비트 정수

기본값

0

osd_pool_erasure_code_stripe_unit

설명

코딩된 풀의 개체 스트라이프 청크의 기본 크기(바이트)를 설정합니다. 크기 S의 모든 오브젝트는 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 스트라이프_unit 설정에 의해 재정의될 수 있습니다.

유형

서명되지 않은 32비트 정수

기본값

4096

osd_pool_default_size

설명

풀에 있는 오브젝트의 복제본 수를 설정합니다. 기본값은 ceph osd pool set {pool-name} 크기 {size} 와 동일합니다.

유형

32비트 정수

기본값

3

osd_pool_default_min_size

설명

클라이언트에 쓰기 작업을 승인하기 위해 풀의 개체에 대해 기록된 최소 복제본 수를 설정합니다. 최소가 충족되지 않으면 Ceph에서 클라이언트에 대한 쓰기를 인식하지 못합니다. 이 설정을 사용하면 성능 저하 모드에서 작동할 때 최소 복제본 수가 보장됩니다.

유형

32비트 정수

기본값

0. 이는 특정 최소값이 없음을 의미합니다. 0 인 경우 최소 크기는 (크기 / 2) 입니다.

osd_pool_default_pg_num

설명

풀의 기본 배치 그룹 수입니다. 기본값은 pg_num 과 mkpool 와 동일합니다.

유형

32비트 정수

기본값

8

osd_pool_default_pgp_num

설명

풀에 배치할 기본 배치 그룹 수입니다. 기본값은 pgp_num 과 mkpool 와 동일합니다. PG 및 PGP는 동등해야 합니다(현재는).

유형

32비트 정수

기본값

8

osd_pool_default_flags

설명

새 풀의 기본 플래그입니다.

유형

32비트 정수

기본값

0

osd_max_pgls

설명

나열할 최대 배치 그룹 수입니다. 많은 숫자를 요청하는 클라이언트는 Ceph OSD 데몬을 연결할 수 있습니다.

유형

서명되지 않은 64비트 정수

기본값

1024

참고

기본값은 fine여야 합니다.

osd_min_pg_log_entries

설명

로그 파일을 트리밍할 때 유지 관리하는 최소 배치 그룹 로그 수입니다.

유형

서명되지 않은 32bit Int

기본값

1000

osd_default_data_pool_replay_window

설명

OSD에서 클라이언트가 요청을 재생할 때까지 대기하는 시간(초)입니다.

유형

32비트 정수

기본값

45

6장. OSD 구성 참조

Ceph 구성 파일에서 Ceph OSD를 구성할 수 있지만 Ceph OSD는 기본값과 매우 최소한의 구성을 사용할 수 있습니다. 최소 Ceph OSD 구성은 osd 저널 크기와 osd 호스트 옵션을 설정하고 다른 거의 모든 항목에 대해 기본값을 사용합니다.

Ceph OSD는 다음 규칙을 사용하여 0 부터 증분 방식으로 숫자로 식별됩니다.

```
osd.0
osd.1
osd.2
```

구성 파일에서 구성 설정을 구성 파일의 [osd] 섹션에 추가하여 클러스터의 모든 Ceph OSD에 대한 설정을 지정할 수 있습니다. 특정 Ceph OSD(예: osd 호스트)에 직접 설정을 추가하려면 Ceph 구성 파일에서 해당 OSD에만 해당 OSD를 입력합니다. 예를 들어 다음과 같습니다.

```
[osd]
osd journal size = 1024

[osd.0]
osd host = osd-host-a

[osd.1]
osd host = osd-host-b
```

6.1. 일반 설정

다음 설정은 Ceph OSD의 ID를 제공하고 데이터 및 저널 경로를 결정합니다. 일반적으로 Ceph 배포 스크립트는 UUID를 자동으로 생성합니다.

중요

Red Hat은 나중에 Ceph 문제를 해결하는 데 문제가 있으므로 데이터 또는 저널의 기본 경로를 변경하는 것을 권장하지 않습니다.

저널 크기는 filestore 최대 동기화 간격 옵션의 값을 곱한 예상 드라이브 속도의 제품 두 배 이상이어야 합니다. 그러나 가장 일반적인 방법은 저널 드라이브(단일 SSD)를 분할하고 Ceph가 저널에 전체 파티션을 사용하도록 마운트하는 것입니다.

osd_uuid

설명

Ceph OSD의 범용 고유 식별자(UUID)입니다.

유형

UUID

기본값

UUID입니다.

참고

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

osd_data

설명

OSD 데이터의 경로입니다. Ceph를 배포할 때 디렉터리를 만들어야 합니다. 이 마운트 지점에 OSD 데이터의 드라이브를 마운트합니다. Red Hat은 기본값을 변경하는 것을 권장하지 않습니다.

유형

문자열

기본값

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

osd_max_write_size

설명

쓰기 크기(MB)입니다.

유형

32비트 정수

기본값

90

osd_client_message_size_cap

설명

메모리에 허용되는 가장 큰 클라이언트 데이터 메시지입니다.

유형

64비트 Integer 서명되지 않음

기본값

500MB의 기본값. 500*1024L*1024L

osd_class_dir

설명

RADOS 클래스 플러그인의 클래스 경로입니다.

유형

문자열

기본값

$libdir/rados-classes

6.2. 저널 설정

기본적으로 Ceph는 다음 경로를 사용하여 Ceph OSD의 저널을 저장할 것으로 예상합니다.

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

성능 최적화가 없으면 Ceph OSD의 데이터와 동일한 디스크에 저널을 저장합니다. 성능에 최적화된 Ceph OSD는 별도의 디스크를 사용하여 저널 데이터를 저장할 수 있습니다(예: 고성능 저널링을 제공하는 솔리드 상태 드라이브).

저널 크기는 filestore 최대 동기화 간격 과 예상 처리량의 제품을 찾고 제품을 2개의(2)로 곱해야 합니다.

```
osd journal size = <2 * (expected throughput * filestore max sync interval)>
```

예상되는 처리량 수에는 예상 디스크 처리량(즉, 유지된 데이터 전송 속도) 및 네트워크 처리량이 포함되어야 합니다. 예를 들어, 7200개의 RPM 디스크에는 약 100MB/s가 있을 수 있습니다. 디스크 및 네트워

크 처리량의 **min()** 을 사용하면 예상 처리량을 적절히 제공해야 합니다. 일부 사용자는 **10GB** 저널 크기로 시작합니다. 예를 들어 다음과 같습니다.

```
osd journal size = 10000
```

주의

OSD의 올바른 크기 조정이 중요합니다. 소규모 저널을 사용하면 **OSD** 오류가 발생할 경우 복구 속도가 느려집니다. 저널에 대한 부담을 허용 가능한 수준으로 유지하여 안정적인 복구를 수행하기 위해 복구 스레드 수를 줄여야 합니다. 또한 파일 저장소로 트랜잭션을 커밋하면 대기 중인 트랜잭션 크기가 저널 크기보다 큰 경우 파일 저장소가 중단될 수 있습니다.

osd_journal

설명

OSD의 저널 경로입니다. 파일 또는 블록 장치(예: **SSD**의 파티션)의 경로일 수 있습니다. 파일이 파일인 경우 파일을 포함할 디렉터리를 만들어야 합니다. **osd** 데이터 드라이브와 별도로 드라이브를 사용하는 것이 좋습니다.

유형

문자열

기본값

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

osd_journal_size

설명

저널의 크기(**MB**)입니다. 이 값이 **0**이고 저널이 블록 장치이면 전체 블록 장치가 사용됩니다. 저널이 블록 장치이고 전체 블록 장치가 사용되는 경우 무시됩니다.

유형

32비트 정수

기본값

5120

권장

1GB로 시작하십시오. 예상 속도의 두 배 이상 filestore 최대 동기화 간격을 곱해야 합니다.

6.3. SCRUBBING

Ceph는 여러 오브젝트 복사본을 생성하는 것 외에도 배치 그룹을 스크럽하여 데이터 무결성을 보장합니다. Ceph 스크러빙은 오브젝트 스토리지 계층에서 fsck 명령과 유사합니다.

각 배치 그룹에 대해 Ceph는 모든 개체의 카탈로그를 생성하고 각 기본 오브젝트와 복제본을 비교하여 개체가 누락되거나 일치하지 않도록 합니다.

light scrubbing (daily)은 오브젝트 크기 및 특성을 확인합니다. 딥 스크럽(주)은 데이터를 읽고 체크섬을 사용하여 데이터 무결성을 보장합니다.

데이터 무결성을 유지하는 데는 스크럽이 중요하지만 성능을 저하시킬 수 있습니다. 다음 설정을 조정하여 스크럽 작업을 늘리거나 줄입니다.

osd_max_scrubs

설명

Ceph OSD의 최대 동시 스크럽 작업 수입니다.

유형

32비트 Int

기본값

1

osd_scrub_thread_timeout

설명

scrub 스레드를 제한하기 전 최대 시간(초)입니다.

유형

32비트 정수

기본값

60

osd_scrub_finalize_thread_timeout

설명

scrub 종료 스레드를 시간 초과하기 전의 최대 시간(초)입니다.

유형

32비트 정수

기본값

60*10

osd_scrub_begin_hour

설명

가장 빨리 또는 깊은 스크럽을 시작할 수 있습니다. osd scrub end hour 매개변수와 함께 사용하여 스크럽 시간 창을 정의하고 교육 스크럽을 사용하지 않는 시간으로 제한할 수 있습니다. 설정은 정수를 사용하여 24시간 동안 시간을 지정합니다. 여기서 0 은 12:01 a.m. ~ 1:00 a.m., 13은 1:01 p.m. ~ 2:00 p.m.까지의 시간을 나타냅니다.

유형

32비트 정수

기본값

0:01 ~ 1:00.m.

osd_scrub_end_hour

설명

가장 최근의 시간 동안 또는 깊은 스크럽을 시작할 수 있습니다. osd scrub start hour 매개변수와 함께 사용하여 스크럽 시간 창을 정의하고 교육 스크럽을 사용하지 않는 시간으로 구성할 수 있습니다. 설정은 정수를 사용하여 24시간 동안 시간을 지정합니다. 여기서 0 은 12:01 a.m. ~ 1:00 a.m., 13은 1:01 p.m. ~ 2:00 p.m.까지의 시간을 나타냅니다. 종료 시간은 시작 시간보다 커야 합니다.

유형

32비트 정수

기본값

24:01 p.m. ~ 12:00 a.m.

osd_scrub_load_threshold

설명

최대 로드입니다. 시스템 로드(**getloadavg()** 함수에 정의된 대로)가 이 수보다 크면 **Ceph**가 스크럽되지 않습니다. 기본값은 **0.5** 입니다.

유형

float

기본값

0.5

osd_scrub_min_interval

설명

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

유형

float

기본값

하루에 한 번. 60*60*24

osd_scrub_max_interval

설명

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

유형

float

기본값

일주일에 한 번. 7*60*60*24

osd_scrub_interval_randomize_ratio

설명

비율을 사용하고 **osd scrub min interval** 과 **osd scrub max** 간격 사이에 예약된 **scrub**를 무작위로 설정합니다.

유형

float

기본값

0.5.

mon_warn_not_scrubbed

설명

osd_scrub_interval 이 스크럽되지 않은 모든 **PG**에 대해 경고하라는 시간(초)입니다.

유형

정수

기본값

0 (경고 없음).

osd_scrub_chunk_min

설명

오브젝트 저장소는 해시 경계로 끝나는 청크로 분할됩니다. 청크 스크럽의 경우 **Ceph scrubs** 개체는 해당 청크에 대해 차단된 쓰기가 한 번에 하나씩 청크입니다. **osd scrub chunk min** 설정은 **scrub**에 대한 최소 청크 수를 나타냅니다.

유형

32비트 정수

기본값

5

osd_scrub_chunk_max

설명

scrub에 대한 최대 청크 수입니다.

유형

32비트 정수

기본값

25

osd_scrub_sleep

설명

딥 스크럽 작업 간에 잠자는 시간입니다.

유형

float

기본값

0 (또는 끄기).

osd_scrub_during_recovery

설명

복구 중에 스크럽을 허용합니다.

유형

bool

기본값

false

osd_scrub_invalid_stats

설명

추가 스크럽이 잘못된 것으로 표시된 통계를 수정하도록 강제 적용합니다.

유형

bool

기본값

true

osd_scrub_priority

설명

scrub 작업의 대기열 우선 순위와 클라이언트 I/O를 제어합니다.

유형

서명되지 않은 32비트 정수

기본값

5

osd_scrub_cost

설명

큐 스케줄링 목적으로 스크럽 작업 수가 메가바이트입니다.

유형

서명되지 않은 32비트 정수

기본값

50 << 20

osd_deep_scrub_interval

설명

모든 데이터를 완전히 읽는 딥 스크럽을 위한 간격입니다. osd scrub 로드 임계값 매개변수는 이 설정에 영향을 미치지 않습니다.

유형

float

기본값

일주일에 한 번. 60*60*24*7

osd_deep_scrub_stride

설명

딥 스크럽을 수행할 때 크기를 읽습니다.

유형

32비트 정수

기본값

512KB. 524288

mon_warn_not_deep_scrubbed

설명

osd_deep_scrub_interval 이 스크럽되지 않은 모든 PG에 대해 경고하기 위해 osd_deep_scrub_interval 이후의 초입니다.

유형

정수

기본값

0 (경고 없음).

osd_deep_scrub_randomize_ratio

설명

스크럽이 무작위로 딥 스크럽이 되는 속도가 (osd_deep_scrub_interval 이 지난 경우에도) 입니다.

유형

float

기본값

0.15 또는 15 %.

osd_deep_scrub_update_digest_min_age

설명

scrub에서 전체 오브젝트 다이제스트를 업데이트하기 전에 몇 초의 이전 오브젝트가 있어야 합니다.

유형

정수

기본값

120 시간(시간).

6.4. 작업

작업 설정을 사용하면 요청 서비스의 스레드 수를 구성할 수 있습니다.

기본적으로 **Ceph**는 **30**초의 시간 제한이 있는 두 개의 스레드와 해당 시간 매개 변수 내에서 작업이 완료되지 않으면 **30**초의 불만 시간을 사용합니다. 복구 중에 최적의 성능을 보장하기 위해 클라이언트 작업과 복구 작업 간의 가중치를 설정합니다.

osd_op_num_shards

설명

클라이언트 작업의 **shard** 수입니다.

유형

32비트 정수

기본값

0

osd_op_num_threads_per_shard

설명

클라이언트 작업을 위한 **shard**당 스레드 수입니다.

유형

32비트 정수

기본값

0

osd_op_num_shards_hdd

설명

foo 작업용 shard 수입니다.

유형

32비트 정수

기본값

5

osd_op_num_threads_per_shard_hdd

설명

HD 작업용 shard당 스레드 수입니다.

유형

32비트 정수

기본값

1

osd_op_num_shards_ssd

설명

SSD 작업의 shard 수입니다.

유형

32비트 정수

기본값

8

osd_op_num_threads_per_shard_ssd

설명

SSD 작업용 shard당 스레드 수입니다.

유형

32비트 정수

기본값

2

osd_client_op_priority

설명

클라이언트 작업에 설정된 우선순위입니다. 이는 **osd recovery op priority** 와 관련이 있습니다.

유형

32비트 정수

기본값

63

유효한 범위

1-63

osd_recovery_op_priority

설명

복구 작업에 대해 설정된 우선 순위입니다. 이는 **osd** 클라이언트 **op priority** 와 관련이 있습니다.

유형

32비트 정수

기본값

3

유효한 범위

1-63

osd_op_thread_timeout

설명

Ceph OSD 작업 스레드 타임아웃(초)입니다.

유형

32비트 정수

기본값

30

osd_op_complaint_time

설명

지정된 시간(초)이 경과한 후 작업 문제가 발생할 수 있습니다.

유형

float

기본값

30

osd_disk_threads

설명

백그라운드 디스크 집약적 OSD 작업을 수행하는 데 사용되는 디스크 스레드 수(scrubbing 및 snap trimming)입니다.

유형

32비트 정수

기본값

1

osd_disk_thread_ioprio_class

설명

디스크 스레드의 ioprio_set(2) I/O 스케줄링 클래스 를 설정합니다. 허용 가능한 값은 다음과 같습니다.

- idle
- be
- rt

idle 클래스는 디스크 스레드가 OSD의 다른 스레드보다 우선 순위가 더 낮다는 것을 의미합니다. 이 명령은 클라이언트 작업을 처리하는 데 사용하는 OSD에서 스크럽을 늦추는 데 유용합니다.

be 클래스는 기본값이며 OSD의 다른 모든 스레드와 동일한 우선 순위입니다.

rt 클래스는 디스크 스레드가 OSD의 다른 모든 스레드보다 우선함을 의미합니다. 이는 스크럽이 많이 필요하며 클라이언트 작업을 대신 진행해야 하는 경우에 유용합니다.

유형

문자열

기본값

빈 문자열

osd_disk_thread_ioprio_priority

설명

디스크 스레드의 ioprio_set(2) I/O 스케줄링 우선 순위를 0(최고)에서 7(낮음)으로 설정합니다. 지정된 호스트의 모든 OSD가 유휴 상태에 있고 컨트롤러 정체로 인해 I/O용으로 경쟁하면 한 OSD의 디스크 스레드 우선 순위를 7로 낮추어 우선 순위가 0인 다른 OSD가 잠재적으로 더 빠르게 스크럽될 수 있습니다.

유형

사용할 수 없는 경우 0에서 7 사이의 범위 또는 -1의 정수입니다.An integer in the range of 0 to 7 or -1 if not to be used.

기본값

-1

중요

osd 디스크 스레드 **ioprio** 클래스와 **osd** 디스크 스레드 **ioprio** 우선 순위 옵션은 둘 다 기본값이 아닌 값으로 설정된 경우에만 사용됩니다. 또한 **Linux Kernel CFQ** 스케줄러에서만 작동합니다.

osd_op_history_size

설명

추적할 최대 작업 수입니다.

유형

32비트 서명되지 않은 Integer

기본값

20

osd_op_history_duration

설명

추적하는 가장 오래된 완료 작업입니다.

유형

32비트 서명되지 않은 Integer

기본값

600

osd_op_log_threshold

설명

한 번에 표시할 작업 로그 수입니다.

유형

32비트 정수

기본값

5

osd_op_timeout

설명

OSD 작업을 실행하는 시간(초)입니다.

유형

정수

기본값

0

중요

클라이언트가 결과를 처리할 수 없는 경우 **osd op timeout** 옵션을 설정하지 마십시오. 예를 들어 가상 머신에서 가상 머신에서 실행 중인 클라이언트에 이 매개변수를 설정하면 이 시간 초과가 하드웨어 오류로 해석되므로 데이터 손상이 발생할 수 있습니다.

6.5. BACKFILLING

Ceph OSD를 클러스터에 추가하거나 클러스터에서 제거하면 배치 그룹을 Ceph OSD로 이동하거나 균형을 복원하여 클러스터의 균형을 다시 조정합니다. 배치 그룹을 마이그레이션하는 프로세스와 포함된 오브젝트를 사용하면 클러스터 운영 성능이 크게 저하될 수 있습니다. 운영 성능을 유지하기 위해 Ceph는 **'backfill'** 프로세스로 이 마이그레이션을 수행하여 Ceph에서 데이터를 읽거나 쓰는 요청보다 우선 순위가 낮은 우선 순위로 백필 작업을 설정할 수 있습니다.

osd_max_backfills

설명

OSD에서 또는 단일 OSD에서 수행할 수 있는 최대 백필 작업 수입니다.

유형

64비트 서명되지 않은 Integer

기본값

1

osd_backfill_scan_min

설명

백필 검사당 최소 오브젝트 수입니다.

유형

32비트 정수

기본값

64

osd_backfill_scan_max

설명

백필 검사당 최대 오브젝트 수입니다.

유형

32비트 정수

기본값

512

osd_backfillfull_ratio

설명

Ceph OSD의 전체 비율이 이 값보다 클 때 백필 요청을 수락하지 않습니다.

유형

float

기본값

0.85

osd_backfill_retry_interval

설명

백필 요청을 다시 시도하기 전에 대기하는 시간(초)입니다.

유형

double

기본값

10.0

6.6. OSD 맵

OSD 맵은 클러스터에서 작동하는 OSD 데몬을 반영합니다. 시간이 지남에 따라 맵 epochs의 수가 증가합니다. Ceph는 Ceph가 작동하고 OSD 맵이 크게 확장되도록 하기 위해 다음 설정을 제공합니다.

osd_map_dedup

설명

OSD 맵에서 중복 제거를 활성화합니다.

유형

부울

기본값

true

osd_map_cache_size

설명

OSD 맵 캐시의 크기(MB)입니다.

유형

32비트 정수

기본값

50

osd_map_cache_bl_size

설명

OSD 데몬의 메모리 내 OSD 맵 캐시의 크기입니다.

유형

32비트 정수

기본값

50

osd_map_cache_bl_inc_size

설명

OSD 데몬에서 메모리 내 OSD 맵 캐시의 크기가 증분됩니다.

유형

32비트 정수

기본값

100

osd_map_message_max

설명

MOSDMap 메시지당 허용된 최대 맵 항목입니다.

유형

32비트 정수

기본값

40

6.7. 복구

클러스터가 시작되거나 Ceph OSD가 예기치 않게 종료되고 재시작되면 쓰기 작업이 발생하기 전에 OSD가 다른 Ceph OSD와 피어링하기 시작합니다.

Ceph OSD가 충돌하고 다시 온라인 상태가 되면 일반적으로 배치 그룹에 있는 최신 버전의 오브젝트가 포함된 다른 Ceph OSD와 동기화되지 않습니다. 이 경우 Ceph OSD는 복구 모드로 전환되고 데이터의 최신 사본을 가져오고 맵을 최신 상태로 유지합니다. Ceph OSD가 다운된 기간에 따라 OSD의 오브젝트 및 배치 그룹이 오래 걸릴 수 있습니다. 또한 실패 도메인이 다운된 경우(예: 랙) 둘 이상의 Ceph OSD가 동시에 다시 온라인 상태가 될 수 있습니다. 이렇게 하면 복구 프로세스 시간과 리소스가 많이 소비될 수 있습니다.

운영 성능을 유지하기 위해 Ceph는 숫자 복구 요청, 스레드 및 개체 청크 크기를 제한하여 복구하여 Ceph가 성능이 저하된 상태로 잘 수행할 수 있습니다.

osd_recovery_delay_start

설명

피어링이 완료되면 Ceph가 오브젝트 복구를 시작하기 전에 지정된 시간 동안 지연됩니다.

유형

float

기본값

0

osd_recovery_max_active

설명

한 번에 OSD당 활성 복구 요청 수입니다. 요청 수가 증가하면 복구 속도가 빨라지지만 요청 시 클러스터에 로드가 증가합니다.

유형

32비트 정수

기본값

3

osd_recovery_max_chunk

설명

내보낼 복구된 데이터 청크의 최대 크기입니다.

유형

64비트 Integer 서명되지 않음

기본값

8 << 20

osd_recovery_threads

설명

데이터를 복구하기 위한 스레드 수입니다.

유형

32비트 정수

기본값

1

osd_recovery_thread_timeout

설명

복구 스레드를 제한하기 전 최대 시간(초)입니다.

유형

32비트 정수

기본값

30

osd_recover_clone_overlap

설명

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

유형

부울

기본값

true

6.8. 기타

osd_snap_trim_thread_timeout

설명

스냅 트리 스레드를 시간 초과하기 전 최대 시간(초)입니다.

유형

32비트 정수

기본값

60*60*1

osd_pg_max_concurrent_snap_trims

설명

병렬 snap trims/PG의 최대 수입니다. 이는 한 번에 트리밍할 PG당 오브젝트 수를 제어합니다.

유형

32비트 정수

기본값

2

osd_snap_trim_sleep

설명

PG 문제를 실행할 때마다 트리밍 사이에 수면을 삽입합니다.

유형

32비트 정수

기본값

0

osd_max_trimming_pgs

설명

최대 트리밍 **PG** 수

유형

32비트 정수

기본값

2

osd_backlog_thread_timeout

설명

백로그 스레드를 시간 초과하기 전의 시간(초)입니다.

유형

32비트 정수

기본값

60*60*1

osd_default_notify_timeout

설명

OSD 기본 알림 시간 초과(초)입니다.

유형

서명되지 않은 **32**비트 정수

기본값

30

osd_check_for_log_corruption

설명

로그 파일에서 손상을 확인합니다. 컴퓨팅적으로 비용이 많이 들 수 있습니다.

유형

부울

기본값

false

osd_remove_thread_timeout

설명

OSD 스레드 제거를 시간 초과하기 전 최대 시간(초)입니다.

유형

32비트 정수

기본값

60*60

osd_command_thread_timeout

설명

명령 스레드를 제한하기 전 최대 시간(초)입니다.

유형

32비트 정수

기본값

10*60

osd_command_max_records

설명

반환할 손실된 오브젝트 수를 제한합니다.

유형

32비트 정수

기본값

256

osd_auto_upgrade_tmap

설명

이전 오브젝트에서 **Omap**에 **t map** 을 사용합니다.

유형

부울

기본값

true

osd_tmapput_sets_users_tmap

설명

디버깅에만 **tmap** 을 사용합니다.

유형

부울

기본값

false

osd_preserve_trimmed_log

설명

트리밍된 로그 파일을 보존하지만 디스크 공간을 더 많이 사용합니다.

유형

부울

기본값

false

rados_osd_op_timeout

설명

RADOS 작업에서 오류를 반환하기 전에 RADOS에서 OSD의 응답을 대기하는 시간(초)입니다. 값이 0이면 제한이 없음을 의미합니다.

유형

double

기본값

0

7장. 모니터 및 OSD 상호 작용 구성

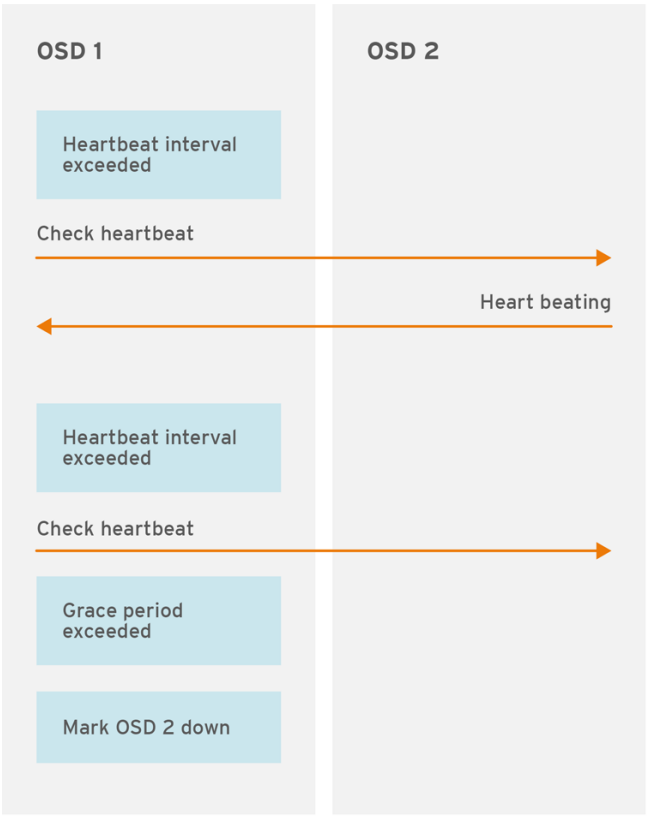
초기 Ceph 구성을 완료한 후에는 Ceph를 배포하고 실행할 수 있습니다. ceph 상태 또는 ceph -s 와 같은 명령을 실행하면 Ceph Storage 클러스터의 현재 상태에 대해 Ceph Monitor가 보고됩니다. Ceph 모니터는 각 Ceph OSD 데몬의 보고서가 필요하여 Ceph Storage 클러스터에 대해 알고 있으며 인접한 Ceph OSD 데몬의 상태에 대한 Ceph OSD 데몬의 보고서를 수신하는 것입니다. Ceph Monitor에서 보고서를 받지 못하거나 Ceph Storage Cluster의 변경 보고서를 수신하는 경우 Ceph Monitor는 Ceph Cluster Map의 상태를 업데이트합니다.

Ceph 모니터 및 Ceph OSD 데몬 상호 작용에 대한 적절한 기본 설정을 제공합니다. 그러나 기본값을 재정의할 수 있습니다. 다음 섹션에서는 Ceph 모니터 및 Ceph OSD 데몬이 Ceph Storage 클러스터 모니터링을 위해 상호 작용하는 방법을 설명합니다.

7.1. OSD 확인 HEARTBEATS

각 Ceph OSD Daemon은 6초마다 다른 Ceph OSD 데몬의 하트비트를 확인합니다. 하트비트 간격을 변경하려면 Ceph 구성 파일의 [osd] 섹션에 osd 하트비트 간격 설정을 추가하거나 런타임 시 값을 변경합니다.

인접한 Ceph OSD 데몬에서 20초의 유예 기간 내에 하트비트 패킷을 보내지 않는 경우 Ceph OSD Daemon은 인접한 Ceph OSD 데몬을 중단하고 Ceph Monitor로 다시 보고할 수 있습니다. 그러면 Ceph 클러스터 맵이 업데이트됩니다. 이 유예 기간을 변경하려면 Ceph 구성 파일의 [osd] 섹션에 osd 하트 하트비트 유예 설정을 추가하거나 런타임 시 값을 설정합니다.



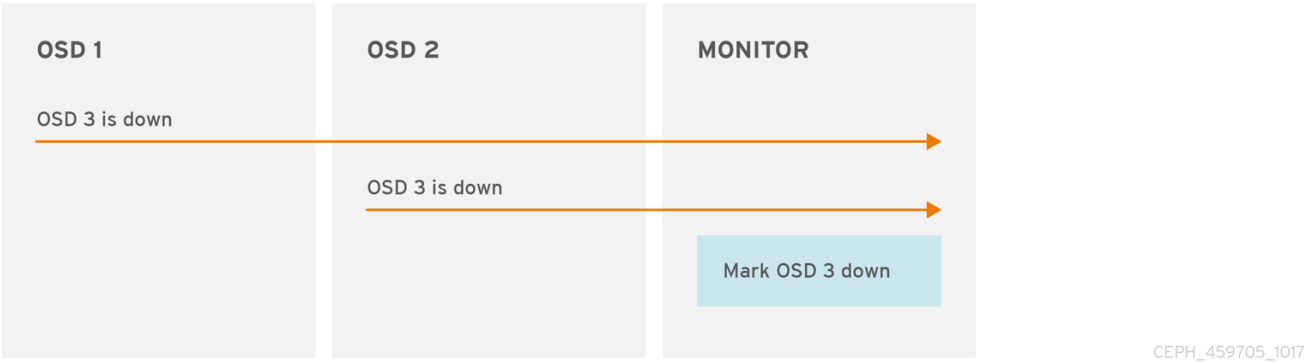
7.2. OSD REPORT DOWN OSD

기본적으로 다른 호스트의 두 개의 Ceph OSD 데몬을 Ceph Monitor에 보고해야 Ceph 모니터에서 다른 Ceph OSD 데몬이 다운 되었음을 보고해야 Ceph OSD Daemon이 보고된 Ceph OSD Daemon이 중단되 었음을 확인할 수 있습니다.

그러나 오류를 보고하는 모든 OSD가 랙의 다른 호스트에 배치되어 OSD 간 연결 문제가 발생할 가능성이 있습니다.

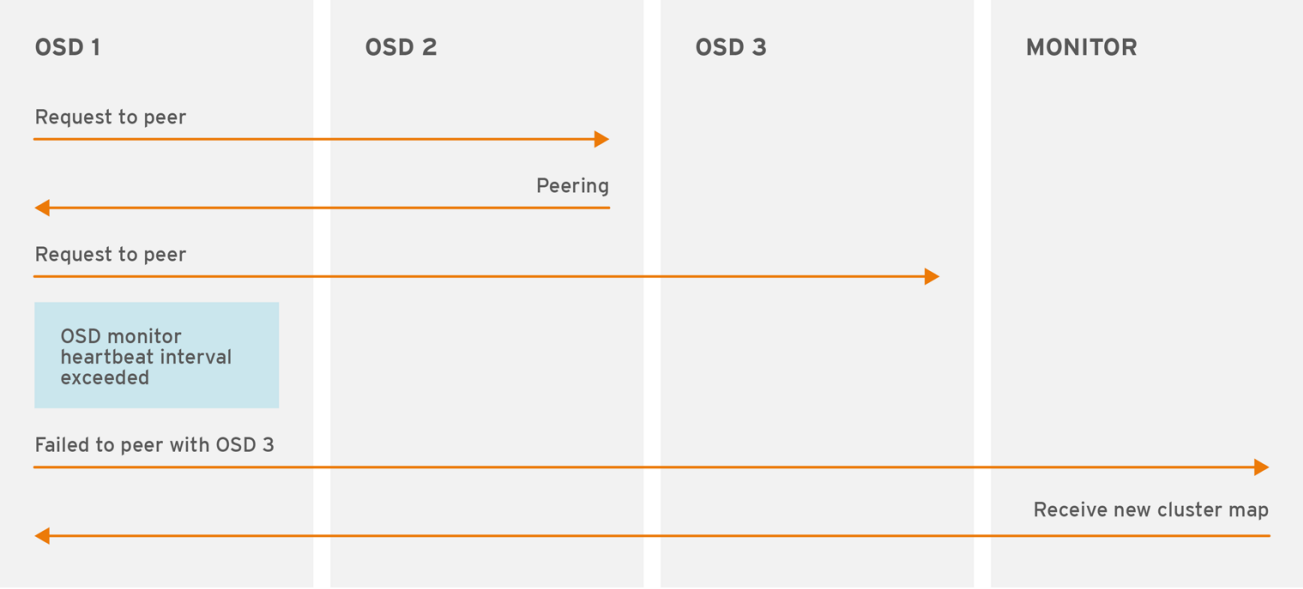
"false 알람"을 방지하기 위해 Ceph는 피어가 이와 유사하게 지연되는 "subcluster"의 대상으로 오류를 보고하는 것을 고려합니다. 항상 그렇지는 않지만 관리자가 제대로 수행되어 있는 시스템의 하위 집합에 유예 수정을 현지화하는 데 도움이 될 수 있습니다.

Ceph는 mon_osd_reporter_subtree_level 설정을 사용하여 NetNamespace 맵의 공통ions 유형에서 피어를 "subcluster"로 그룹화합니다. 기본적으로 다른 하위 트리의 보고서 두 개만 다른 Ceph OSD 데몬 을 보고해야 합니다. 관리자는 고유 하위 트리에서 보고자 수와 Ceph OSD 데몬을 Ceph 모니터에 보고하는 데 필요한 공통 NetNamespace 유형을 mon_osd_min_ down _reporters 및 mon_osd_subtree_level 설정의 [mon_osd_reporter_subtree_level 설정은 Ceph 구성 파일의 [mon] 섹션에 설정하여 변경할 수 있습니다.



7.3. OSD REPORT PEERING FAILURE

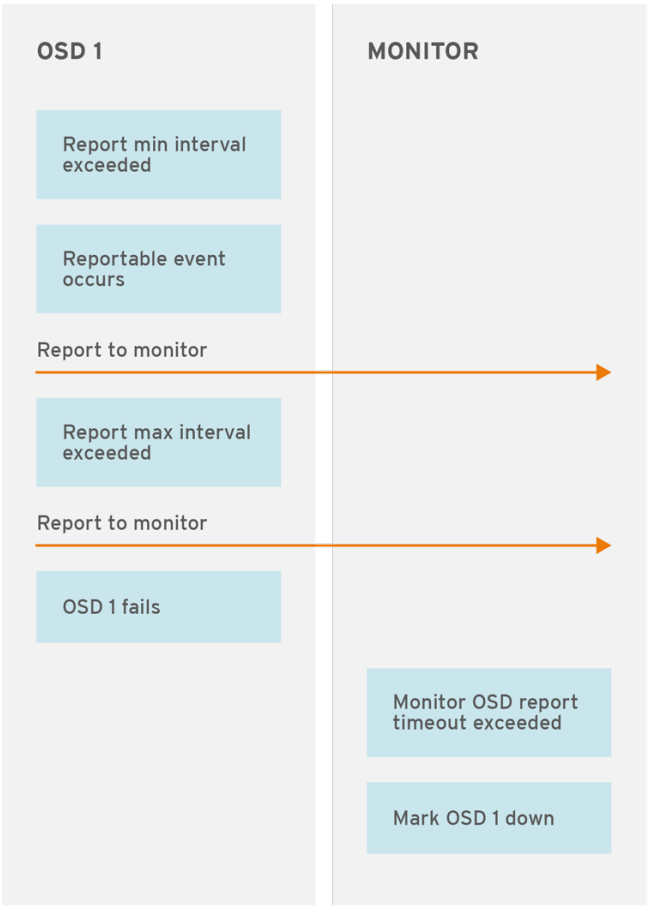
Ceph OSD Daemon이 Ceph 구성 파일 또는 클러스터 맵에 정의된 Ceph OSD 데몬과 피어링할 수 없는 경우 Ceph Monitor에서 30초마다 클러스터 맵의 최신 사본을 ping합니다. Ceph 구성 파일의 [osd] 섹션에 osd mon 하트 간격을 추가하거나 런타임에 값을 설정하여 Ceph 모니터 하트비트 간격을 변경할 수 있습니다.



7.4. OSD에서 해당 상태 확인

Ceph OSD Daemon이 **Ceph Monitor**에 보고되지 않으면 **mon osd** 보고서 시간 초과 시간이 지나면 **Ceph OSD** 데몬을 아래로 고려합니다. **Ceph OSD Daemon**은 실패, 배치 그룹 통계 변경, **up_thru** 변경 또는 **5**초 이내에 부팅 가능한 이벤트의 경우 **Ceph Monitor**로 보고서를 보냅니다. **Ceph** 구성 파일의 **[osd]** 섹션에 **osd mon report interval min** 설정을 추가하거나 런타임에 값을 설정하여 **Ceph OSD Daemon** 최소 보고서 간격을 변경할 수 있습니다.

Ceph OSD Daemon은 주요 변경 사항이 발생하는지 여부에 관계없이 **120**초마다 **Ceph** 모니터에 보고서를 보냅니다. **Ceph** 구성 파일의 **[osd]** 섹션에 **osd mon report interval max** 설정을 추가하거나 런타임 시 값을 설정하여 **Ceph Monitor** 보고서 간격을 변경할 수 있습니다.



7.5. 구성 설정

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

7.5.1. 모니터 설정

mon_osd_min_up_ratio

설명

Ceph가 **Ceph OSD** 데몬 앞에 있는 최대 **Ceph OSD** 데몬의 최소 비율은 **Ceph OSD** 데몬을 아래로 표시합니다.

유형

double

기본값

.3

mon_osd_min_in_ratio

설명

Ceph가 Ceph OSD 데몬보다 먼저 Ceph OSD 데몬에서의 최소 비율은 Ceph OSD 데몬 을 끕니다.

유형

double

기본값

.3

mon_osd_laggy_halflife

설명

몇 초 지연 추정치 가 감소합니다.

유형

정수

기본값

60*60

mon_osd_laggy_weight

설명

La ggy estimation decay의 새로운 샘플 의 가중치입니다.

유형

double

기본값

0.3

mon_osd_laggy_max_interval

설명

지연 추정에 있는 laggy_interval 의 최대 값(초)입니다. 모니터는 적응형 접근 방식을 사용하여 특정 OSD의 laggy_interval 을 평가합니다. 이 값은 해당 OSD의 유예 시간을 계산하는 데 사용됩니다.

유형

정수

기본값

300

mon_osd_adjust_heartbeat_grace

설명

true 로 설정하면 Ceph는 지연 추정치에 따라 확장 됩니다.

유형

부울

기본값

true

mon_osd_adjust_down_out_interval

설명

true 로 설정하면 Ceph가 지연 추정에 따라 확장 됩니다.

유형

부울

기본값

true

mon_osd_auto_mark_in

설명

Ceph는 모든 부팅 Ceph OSD 데몬을 Ceph Storage 클러스터에서 로 표시합니다.

유형

부울

기본값

false

mon_osd_auto_mark_auto_out_in

설명

Ceph는 클러스터에서와 같이 **Ceph Storage** 클러스터에서 자동으로 표시된 **Ceph OSD** 데몬 부팅을 표시합니다.

유형

부울

기본값

true

mon_osd_auto_mark_new_in

설명

Ceph는 새 **Ceph OSD** 데몬 부팅을 **Ceph Storage** 클러스터에서와 같이 표시합니다.

유형

부울

기본값

true

mon_osd_down_out_interval

설명

Ceph OSD 데몬을 아래로 표시하고 응답하지 않는 경우 **Ceph**가 대기하는 시간(초)입니다.

유형

32비트 정수

기본값

600

mon_osd_downout_subtree_limit

설명

Ceph가 자동으로 표시할 가장 큰dpdk 장치 유형입니다.

유형

문자열

기본값

rack

mon_osd_reporter_subtree_level

설명

이 설정은 보고 OSD에 대한 상위 NetNamespace 장치 유형을 정의합니다. OSD는 응답하지 않는 피어를 찾으면 실패 보고서를 모니터에 보냅니다. 모니터는 보고된 OSD를 종료한 다음 유예 기간 후에 표시할 수 있습니다.

유형

문자열

기본값

host

mon_osd_report_timeout

설명

응답하지 않는 Ceph OSD 데몬을 삭제하기 전에 유예 기간(초)입니다.

유형

32비트 정수

기본값

900

mon_osd_min_down_reporters

설명

다운 된 Ceph OSD 데몬을 보고하는 데 필요한 최소 Ceph OSD 데몬 수입니다.

유형

32비트 정수

기본값

2

7.5.2. OSD 설정

osd_heartbeat_address

설명

하트비트용 Ceph OSD 데몬의 네트워크 주소입니다.

유형

address

기본값

호스트 주소입니다.

osd_heartbeat_interval

설명

Ceph OSD Daemon이 피어(초)를 ping하는 빈도입니다.

유형

32비트 정수

기본값

6

osd_heartbeat_grace

설명

Ceph OSD 데몬에 Ceph Storage 클러스터가 이를 중단 한다고 간주하는 하트비트가 표시되지 않는 시간이 경과했습니다.

유형

32비트 정수

기본값

20

osd_mon_heartbeat_interval

설명

Ceph OSD 데몬에 Ceph OSD 데몬 피어가 없는 경우 Ceph Monitor를 ping하는 빈도입니다.

유형

32비트 정수

기본값

30

osd_mon_report_interval_max

설명

Ceph OSD 데몬에서 대기할 수 있는 최대 시간(초)은 Ceph Monitor에 보고해야 합니다.

유형

32비트 정수

기본값

120

osd_mon_report_interval_min

설명

Ceph OSD Daemon은 Ceph Monitor에 보고하기 전에 시작 또는 다른 보고 가능 이벤트에서 대기하는 최소 시간(초)입니다.

유형

32비트 정수

기본값

5

유효한 범위

osd mon 보고서 간격 **max**보다 작아야 합니다.

osd_mon_ack_timeout

설명

Ceph 모니터가 통계 요청을 승인할 때까지 대기하는 시간(초)입니다.

유형

32비트 정수

기본값

30

8장. 파일 저장소 구성 참조

8.1. 확장 속성

확장된 속성(XATTR)은 CephFS 구성의 중요한 측면입니다. 일부 파일 시스템에는 확장 속성에 저장된 바이트 수가 제한됩니다. 또한 경우에 따라 파일 시스템이 확장 속성을 저장하는 대체 방법만큼 빠르지 않을 수 있습니다. 다음 설정은 기본 파일 시스템에 extrinsic인 확장 속성을 저장하는 방법을 사용하여 CephFS 성능을 향상시킵니다.

Ceph 확장 속성은 크기 제한을 적용하지 않는 경우 기본 파일 시스템에서 제공하는 확장된 속성을 사용하여 인라인 xattr 로 저장됩니다. 크기 제한이 있는 경우(예: ext4에 총 4KB) 일부 Ceph 확장 속성은 filestore max inline xattr 크기 또는 filestore max inline xattrs 임계값에 도달할 때 omap 이라는 키-값 데이터베이스에 저장됩니다.

filestore_xattr_use_omap

설명

XATTRS에 개체 맵을 사용합니다. ext4 파일 시스템의 경우 true 로 설정합니다.

유형

부울

필수 항목

없음

기본값

false

filestore_omap_header_cache_size

설명

개체 omap 헤더를 캐시하는 데 사용되는 LRU의 크기를 결정합니다. 더 큰 값은 메모리를 더 사용하지만 omap 에서 조회를 줄일 수 있습니다. (시험만 해당).

유형

정수

기본값

1024

filestore_omap_backend

설명

omap 에 사용되는 백엔드를 결정하는 데 사용됩니다. **leveldb** 또는 **basdb** 로 설정할 수 있습니다. (시험만 가능합니다. **basdb** 는 실험적입니다.)

유형

문자열

기본값

leveldb

filestore_debug_omap_check

설명

동기화에서 디버깅 검사. 비용이 많이 듭니다. 디버깅 전용입니다.**For debugging only.**

유형

부울

필수 항목

없음

기본값

0

filestore_max_inline_xattr_size

설명

오브젝트당 파일 시스템(즉, **XFS, btrfs, ext4** 등)에 저장된 확장 속성의 최대 크기입니다. 파일 시스템에서 처리할 수 있는 것보다 크지 않아야 합니다.

유형

서명되지 않은 **32**비트 정수

필수 항목

없음

기본값

512

filestore_max_inline_xattrs

설명

오브젝트당 파일 시스템에 저장된 확장 속성의 최대 수입니다.

유형

32비트 정수

필수 항목

없음

기본값

2

filestore_max_inline_xattr_size_xfs

설명

오브젝트당 XFS 파일 시스템의 파일 시스템에 저장된 확장 속성의 최대 크기입니다. 파일 시스템에서 처리할 수 있는 것보다 크지 않아야 합니다.

유형

서명되지 않은 32비트 정수

기본값

65536

filestore_max_inline_xattr_size_btrfs

설명

오브젝트당 btrfs의 파일 시스템에 저장된 확장 속성의 최대 크기입니다. 파일 시스템에서 처리할 수 있는 것보다 크지 않아야 합니다.

유형

서명되지 않은 32비트 정수

기본값

2048

filestore_max_inline_xattr_size_other

설명

오브젝트당 btrfs 또는 XFS 이외의 파일 시스템의 경우 파일 시스템에 저장된 확장 속성의 최대 크기입니다. 파일 시스템에서 처리할 수 있는 것보다 크지 않아야 합니다.

유형

서명되지 않은 32비트 정수

기본값

512

filestore_max_inline_xattrs

설명

오브젝트당 파일 시스템에 저장된 확장 속성의 최대 수입니다. 세분화된 설정을 재정의합니다.

유형

서명되지 않은 32비트 정수

기본값

0

filestore_max_inline_xattrs_xfs

설명

오브젝트당 XFS 파일 시스템에 저장된 최대 확장 속성 수입니다.

유형

서명되지 않은 32비트 정수

기본값

10

filestore_max_inline_xattrs_btrfs

설명

오브젝트당 **btrfs** 파일 시스템에 저장된 확장 속성의 최대 수입니다.

유형

서명되지 않은 **32**비트 정수

기본값

10

filestore_max_inline_xattrs_other

설명

오브젝트당 **btrfs** 또는 **XFS** 이외의 파일 시스템에 저장된 확장 속성의 최대 수입니다.

유형

서명되지 않은 **32**비트 정수

기본값

2

8.2. 동기화 간격

정기적으로 파일 저장소는 쓰기 작업을 정지하고 파일 시스템을 동기화하여 일관된 커밋 지점을 만들어야 합니다. 그런 다음 커밋 지점까지 저널 항목을 해제할 수 있습니다. 동기화는 동기화를 수행하는 데 필요한 시간을 줄이고 저널에 남아 있어야 하는 데이터의 양을 줄이는 경향이 있습니다. 덜 자주 동기화를 사용하면 백업 파일 시스템이 작은 쓰기 및 메타데이터 업데이트를 최적으로 업데이트함으로써 동기화 효율성을 높일 수 있습니다.

filestore_max_sync_interval

설명

파일 저장소를 동기화하는 최대 간격(초)입니다.

유형

double

필수 항목

없음

기본값

5

filestore_min_sync_interval

설명

파일 저장소를 동기화하는 최소 간격(초)입니다.

유형

double

필수 항목

없음

기본값

.01

8.3. FLUSHER

파일 저장소 플러시기는 최종 동기화 비용을 줄이기 위해 동기화 파일 범위 옵션을 사용하여 대규모 쓰기 작업의 데이터를 강제로 기록합니다. 실제로 파일 저장소 플러시를 비활성화하면 일부 경우에는 성능이 향상되는 것처럼 보입니다.

filestore_flusher

설명

파일 저장소 플러시를 활성화합니다.

유형

부울

필수 항목

없음

기본값

false

filestore_flusher_max_fds

설명

플러시자의 최대 파일 설명자 수를 설정합니다.

유형

정수

필수 항목

없음

기본값

512

filestore_sync_flush

설명

동기화 플러시기를 활성화합니다.

유형

부울

필수 항목

없음

기본값

false

filestore_fsync_flushes_journal_data

설명

파일 시스템 동기화 중에 저널 데이터를 플러시합니다.

유형

부울

필수 항목

없음

기본값

false

8.4. QUEUE

다음 설정은 파일 저장소 대기열의 크기에 대한 제한을 제공합니다.

filestore_queue_max_ops

설명

새 작업을 삭제하기 전에 파일 저장소에서 허용하는 최대 작업 수를 진행 중인 최대 작업 수를 정의합니다.

유형

정수

필수 항목

아니요. 성능에 최소한의 영향

기본값

500

filestore_queue_max_bytes

설명

작업의 최대 바이트 수입니다.

유형

정수

필수 항목

없음

기본값

100 << 20

filestore_queue_committing_max_ops

설명

파일 저장소에서 커밋할 수 있는 최대 작업 수입니다.

유형

정수

필수 항목

없음

기본값

500

filestore_queue_committing_max_bytes

설명

파일 저장소에서 커밋할 수 있는 최대 바이트 수입니다.

유형

정수

필수 항목

없음

기본값

100 << 20

8.5. WRITEBACK THROTTLE

페이지 캐시가 더티 데이터 라운드가 너무 길기 때문에 Ceph에서 일부 나중 쓰기 동작을 복제합니다.

filestore_wbthrottle_enable

설명

파일 저장소 쓰기 기능을 활성화합니다. file store write-back throttle는 많은 양의 커밋되지 않은 데이터가 각 파일 저장소 동기화 전에 빌드되지 않도록 하는 데 사용됩니다. (시험만 해당).

유형

부울

기본값

true

filestore_wbthrottle_btrfs_bytes_start_flusher

설명

Ceph가 btrfs 파일 시스템에 대한 백그라운드 플러시를 시작하는 더티 바이트 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

41943040

filestore_wbthrottle_btrfs_bytes_hard_limit

설명

플러시자가 btrfs를 가져올 때까지 Ceph가 I/O를 제한하는 더티 바이트 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

419430400

filestore_wbthrottle_btrfs_ios_start_flusher

설명

Ceph가 btrfs에 대한 백그라운드 플러시를 시작하는 더티 I/Os 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

500

filestore_wbthrottle_btrfs_ios_hard_limit

설명

flusher가 btrfs를 가져올 때까지 Ceph가 IO를 제한하기 시작하는 더티 I/Os 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

5000

filestore_wbthrottle_btrfs_inodes_start_flusher

설명

Ceph가 btrfs에 대한 백그라운드 플러시를 시작하는 더티 inode 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

500

filestore_wbthrottle_btrfs_inodes_hard_limit

설명

flusher가 btrfs에 대해 catch할 때까지 Ceph가 IO를 제한하기 시작합니다. fd 제한보다 작아야 합니다.

유형

64비트 서명되지 않은 Integer

기본값

5000

filestore_wbthrottle_xfs_bytes_start_flusher

설명

Ceph가 XFS 파일 시스템의 백그라운드 플러시를 시작하는 더티 바이트 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

41943040

filestore_wbthrottle_xfs_bytes_hard_limit

설명

Ceph가 XFS를 위해 플러시될 때까지 IO를 제한하기 시작하는 더티 바이트 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

419430400

filestore_wbthrottle_xfs_ios_start_flusher

설명

Ceph가 XFS의 백그라운드 플러시를 시작하는 더티 I/O 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

500

filestore_wbthrottle_xfs_ios_hard_limit

설명

Ceph가 XFS를 위해 플러싱할 때까지 IO를 제한하기 시작하는 더티 I/Os 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

5000

filestore_wbthrottle_xfs_inodes_start_flusher

설명

Ceph가 XFS의 백그라운드 플러시를 시작하는 더티한 inode 임계값입니다.

유형

64비트 서명되지 않은 Integer

기본값

500

filestore_wbthrottle_xfs_inodes_hard_limit

설명

Ceph가 XFS에 대해 플러싱할 때까지 IO를 제한하기 시작합니다. fd 제한보다 작아야 합니다.

유형

64비트 서명되지 않은 Integer

기본값

5000

8.6. TIMEOUTS

filestore_op_threads

설명

병렬로 실행되는 파일 시스템 작업 스레드 수입니다.

유형

정수

필수 항목

없음

기본값

2

filestore_op_thread_timeout

설명

파일 시스템 작업 스레드(초)의 시간 제한입니다.

유형

정수

필수 항목

없음

기본값

60

filestore_op_thread_suicide_timeout

설명

커밋(초)을 취소하기 전에 커밋 작업의 타임아웃입니다.

유형

정수

필수 항목

없음

기본값

180

8.7. B-TREE 파일 시스템

filestore_btrfs_snap

설명

btrfs 파일 저장소에 대한 스냅샷을 활성화합니다.

유형

부울

필수 항목

아니요. btrfs에만 사용됩니다.

기본값

true

filestore_btrfs_clone_range

설명

btrfs 파일 저장소에 대한 복제 범위를 활성화합니다.

유형

부울

필수 항목

아니요. btrfs에만 사용됩니다.

기본값

true

8.8. 저널

filestore_journal_parallel

설명

btrfs에 대해 기본적으로 병렬 저널링을 활성화합니다.

유형

부울

필수 항목

없음

기본값

false

filestore_journal_writeahead

설명

XFS에 대해 미리 쓰기(Write-ahead) 저널링을 활성화합니다.

유형

부울

필수 항목

없음

기본값

false

filestore_journal_trailing

설명

더 이상 사용되지 않으며 절대 사용하지 않습니다.

유형

부울

필수 항목

없음

기본값

false

8.9. 기타

filestore_merge_threshold

설명

상위 참고에 병합하기 전에 하위 디렉터리의 최소 파일 수: 음수 값은 하위 디렉터리 병합을 비활성화하는 수단입니다.

유형

정수

필수 항목

없음

기본값

10

filestore_split_multiple

설명

Filestore_split_multiple * abs(filestore_merge_threshold) * 16 은 하위 디렉터리로 분할되기 전에 하위 디렉터리의 최대 파일 수입니다.

유형

정수

필수 항목

없음

기본값

2

filestore_update_to

설명

파일 저장소 자동 업그레이드를 지정된 버전으로 제한합니다.

유형

정수

필수 항목

없음

기본값

1000

filestore_blackhole

설명

새로운 트랜잭션을 바닥에 배치하십시오.

유형

부울

필수 항목

없음

기본값

false

filestore_dump_file

설명

트랜잭션 덤프를 저장하는 파일입니다.

유형

부울

필수 항목

없음

기본값

false

filestore_kill_at

설명

실패를 n번째 기회에 주입합니다.

유형

문자열

필수 항목

없음

기본값

false

filestore_fail_eio

설명

EIO에서 실패하거나 예기치 않게 종료합니다.

유형

부울

필수 항목

없음

기본값

true

9장. 저널리어 구성 참조

Ceph OSD는 다음과 같은 이유로 저널을 사용합니다.

속도

저널을 사용하면 **Ceph OSD** 데몬에서 작은 쓰기 작업을 신속하게 수행할 수 있습니다. **Ceph**는 작은 임의 **I/O**를 저널에 순차적으로 작성하므로 백업 파일 시스템이 쓰기 작업을 더 많이 수행할 수 있도록 하여 버스트된 워크로드의 속도를 높일 수 있습니다. 그러나 **Ceph OSD Daemon**의 저널은 파일 시스템이 저널을 차지할 때 쓰기 진행 단계 없이 단기 고속 쓰기가 짧은 기간과 함께 급격한 성능을 초래할 수 있습니다.

일관성

Ceph OSD 데몬에는 원자성 복합 작업을 보장하는 파일 시스템 인터페이스가 필요합니다. **Ceph OSD** 데몬은 작업에 대한 설명을 저널에 작성하고 작업을 파일 시스템에 적용합니다. 이렇게 하면 개체(예: 배치 그룹 메타데이터)에 대한 원자성 업데이트가 가능합니다. **Ceph OSD**는 몇 초 간격으로 **filestore** 최대 동기화 간격 과 **filestore** 최소 동기화 간격 설정을 분리합니다. **Ceph OSD**는 쓰기 작업을 중지하고 파일 시스템과 저널을 동기화하여 **Ceph OSD**에서 저널을 트리밍하고 공간을 재사용할 수 있습니다. 실패 시 **Ceph OSD**는 마지막 동기화 작업 후부터 저널을 재생합니다.

9.1. 설정

Ceph OSD 데몬에서는 다음 저널 설정을 지원합니다.

journal_dio

설명

저널에 직접 **I/O**를 활성화합니다. **journal block align** 옵션을 **true** 로 설정해야 합니다.

유형

부울

필수 항목

Aio 를 사용할 때

기본값

true

journal_aio

설명

libaio 를 사용하여 저널에 대한 비동기 쓰기를 활성화합니다. **journal dio** 옵션을 **true** 로 설정해야 합니다.

유형

부울

필수 항목

아니요.

기본값

True.

journal_block_align

설명

Block은 쓰기 작업을 조정합니다. **dio** 및 **aio** 에 필요합니다.

유형

부울

필수 항목

예, **dio** 및 **aio** 를 사용할 때

기본값

true

journal_max_write_bytes

설명

저널이 한 번에 쓸 최대 바이트 수입니다.

유형

정수

필수 항목

없음

기본값

10 << 20

journal_max_write_entries

설명

저널이 한 번에 쓸 최대 항목 수입니다.

유형

정수

필수 항목

없음

기본값

100

journal_queue_max_ops

설명

한 번에 큐에 허용되는 최대 작업 수입니다.

유형

정수

필수 항목

없음

기본값

500

journal_queue_max_bytes

설명

한 번에 큐에 허용되는 최대 바이트 수입니다.

유형

정수

필수 항목

없음

기본값

10 << 20

journal_align_min_size

설명

지정된 최소값보다 큰 데이터 페이로드를 정렬합니다.

유형

정수

필수 항목

없음

기본값

64 << 10

journal_zero_on_create

설명

파일 저장소가 ' **mkfs** 중에 전체 저널을 **0**'으로 덮어 쓰도록 합니다.

유형

부울

필수 항목

없음

기본값

false

10장. 로깅 구성 참조

Ceph 구성 파일에는 로깅 및 디버깅 설정이 필요하지 않지만 필요에 따라 기본 설정을 재정의할 수 있습니다.

옵션은 채널에 관계없이 모든 데몬의 기본값으로 간주되는 단일 항목을 사용합니다. 예를 들어 "info"를 지정하는 것은 "default=info"로 해석됩니다. 그러나 옵션은 키/값 쌍을 사용할 수도 있습니다. 예를 들어 "default=daemon audit=local0"은 "default all to 'daemon'으로 해석되며 'audit'을 'local0'으로 재정의합니다.

Ceph는 다음 설정을 지원합니다.

log_file

설명

클러스터의 로깅 파일의 위치입니다.

유형

문자열

필수 항목

없음

기본값

/var/log/ceph/$cluster-$name.log

mon_cluster_log_file

설명

모니터 클러스터 로그 파일의 위치입니다.

유형

문자열

필수 항목

없음

기본값

/var/log/ceph/$cluster.log

log_max_new

설명

새 로그 파일의 최대 수입니다.

유형

정수

필수 항목

없음

기본값

1000

log_max_recent

설명

로그 파일에 포함할 최근 이벤트의 최대 수입니다.

유형

정수

필수 항목

없음

기본값

1000000

log_flush_on_exit

설명

종료 후 Ceph가 로그 파일을 플러시하는지 결정합니다.

유형

부울

필수 항목

없음

기본값

true

mon_cluster_log_file_level

설명

모니터 클러스터의 파일 로깅 수준입니다. 유효한 설정에는 "debug", "info", "sec", "warn", "error"가 포함됩니다.

유형

문자열

기본값

"info"

log_to_stderr

설명

stderr 에 로깅 메시지가 표시되는지 여부를 결정합니다.

유형

부울

필수 항목

없음

기본값

true

err_to_stderr

설명

stderr 에 오류 메시지가 표시되는지 여부를 결정합니다.

유형

부울

필수 항목

없음

기본값

true

log_to_syslog

설명

syslog 에 로깅 메시지가 표시되는지 여부를 결정합니다.

유형

부울

필수 항목

없음

기본값

false

err_to_syslog

설명

syslog 에 오류 메시지가 표시되는지 여부를 확인합니다.

유형

부울

필수 항목

없음

기본값

false

clog_to_syslog

설명

clog 메시지가 syslog 로 전송되는지 여부를 결정합니다.

유형

부울

필수 항목

없음

기본값

false

mon_cluster_log_to_syslog

설명

클러스터 로그가 syslog 로 출력되는지 여부를 결정합니다.

유형

부울

필수 항목

없음

기본값

false

mon_cluster_log_to_syslog_level

설명

모니터 클러스터에 대한 syslog 로깅 수준입니다. 유효한 설정에는 "debug", "info", "sec", "warn", "error"가 포함됩니다.

유형

문자열

기본값

"info"

mon_cluster_log_to_syslog_facility

설명

syslog 출력을 생성하는 기능입니다. 일반적으로 Ceph 데몬의 경우 "daemon"으로 설정됩니다.

유형

문자열

기본값

"daemon"

clog_to_monitors

설명

clog 메시지가 모니터 로 전송되는지 여부를 확인합니다.

유형

부울

필수 항목

없음

기본값

true

mon_cluster_log_to_graylog

설명

클러스터가 로그 메시지를 회색log로 출력하는지 결정합니다.

유형

문자열

기본값

"false"

mon_cluster_log_to_graylog_host

설명

graylog 호스트의 IP 주소입니다. 회색log 호스트가 모니터 호스트와 다른 경우 이 설정을 적절한 IP 주소로 재정의합니다.

유형

문자열

기본값

"127.0.0.1"

mon_cluster_log_to_graylog_port

설명

Graylog 로그가 이 포트로 전송됩니다. 포트 에서 데이터 수신을 위해 열려 있는지 확인합니다.

유형

문자열

기본값

"12201"

10.1. OSD

osd_preserve_trimmed_log

설명

트리밍 후 트리밍 로그를 유지합니다.

유형

부울

필수 항목

없음

기본값

false

osd_tmapput_sets_uses_tmap

설명

Tmap을 사용합니다. 디버그용만 해당됩니다.

유형

부울

필수 항목

없음

기본값

false

osd_min_pg_log_entries

설명

배치 그룹에 대한 최소 로그 항목 수입니다.

유형

32비트 서명되지 않은 Integer

필수 항목

없음

기본값

1000

osd_op_log_threshold

설명

한 패스에 표시될 **op** 로그 메시지의 수입니다.

유형

정수

필수 항목

없음

기본값

5

10.2. 파일 저장소

filestore_debug_omap_check

설명

동기화에서 디버깅 검사. 이는 비용이 많이 드는 작업입니다.

유형

부울

필수 항목

없음

기본값

0

10.3. CEPH OBJECT GATEWAY

rgw_log_nonexistent_bucket

설명

존재하지 않는 버킷을 기록합니다.

유형

부울

필수 항목

없음

기본값

false

rgw_log_object_name

설명

오브젝트의 이름을 기록합니다.

유형

문자열

필수 항목

없음

기본값

%Y-%m-%d-%H-%i-%n

rgw_log_object_name_utc

설명

오브젝트 로그 이름에는 **UTC**가 포함되어 있습니다.

유형

부울

필수 항목

없음

기본값

false

rgw_enable_ops_log

설명

모든 **RGW** 작동을 로깅할 수 있습니다.

유형

부울

필수 항목

없음

기본값

true

rgw_enable_usage_log

설명

RGW의 대역폭 사용량 로깅을 활성화합니다.

유형

부울

필수 항목

없음

기본값

true

rgw_usage_log_flush_threshold

설명

보류 중인 로그 데이터를 플러시하는 임계값입니다.

유형

정수

필수 항목

없음

기본값

1024

rgw_usage_log_tick_interval

설명

보류 중인 로그 데이터를 1초마다 플러시 합니다.

유형

정수

필수 항목

없음

기본값

30

rgw_intent_log_object_name

설명, 유형

문자열

필수 항목

없음

기본값

%Y-%m-%d-%i-%n

rgw_intent_log_object_name utc

설명

의도 로그 오브젝트 이름에 UTC 타임스탬프를 포함합니다.

유형

부울

필수 항목

없음

기본값

false