



# Red Hat Ceph Storage 3

## 컨테이너 가이드

Deploying and Managing Red Hat Ceph Storage in Containers



# Red Hat Ceph Storage 3 컨테이너 가이드

---

Deploying and Managing Red Hat Ceph Storage in Containers

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

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Container\_Guide.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 초록

이 문서에서는 컨테이너에 Red Hat Ceph Storage를 배포하고 관리하는 방법을 설명합니다.

## 차례

<b>1장. 컨테이너에 RED HAT CEPH STORAGE 배포</b> .....	<b>4</b>
1.1. 사전 요구 사항	4
1.1.1. CDN 및 서브스크립션에 Red Hat Ceph Storage 노드 등록	4
사전 요구 사항	4
절차	5
추가 리소스	5
1.1.2. sudo 액세스를 사용하여 Ansible 사용자 생성	6
1.1.3. Ansible에 대해 암호 없는 SSH 활성화	7
사전 요구 사항	7
절차	8
추가 리소스	9
1.1.4. Red Hat Ceph Storage 방화벽 설정	9
1.1.5. HTTP 프록시 사용	13
사전 요구 사항	13
절차	13
1.2. 컨테이너에 RED HAT CEPH STORAGE 클러스터 설치	13
사전 요구 사항	13
절차	14
1.3. 모든 NVME 스토리지에 대한 OSD ANSIBLE 설정 구성	25
1.4. 컨테이너에 CEPH OBJECT GATEWAY 설치	27
1.5. 메타데이터 서버 설치	30
1.6. NFS-GANESHA 게이트웨이 설치	31
사전 요구 사항	31
절차	31
추가 리소스	33
1.7. 컨테이너에 CEPH ISCSI 게이트웨이 설치	33
1.7.1. 컨테이너에서 Ceph iSCSI 게이트웨이 구성	37
1.7.2. 컨테이너에서 Ceph iSCSI 게이트웨이 제거	43
1.7.3. iSCSI 대상의 성능 최적화	47
1.8. 제한 옵션 이해	49
1.9. 추가 리소스	49
<b>2장. 컨테이너화된 CEPH 데몬 공동 배치</b> .....	<b>50</b>
2.1. 공동 배치의 작동 방식 및 이점	50
Colocation Works	50
2.2. 협업된 데몬의 DEDICATED 리소스 설정	52
2.3. 추가 리소스	54
<b>3장. 컨테이너에서 실행되는 CEPH 클러스터 관리</b> .....	<b>55</b>
3.1. 컨테이너에서 실행되는 CEPH 데몬 시작, 중지 및 재시작	55
3.2. 컨테이너에서 실행되는 CEPH 데몬의 로그 파일 보기	56
3.3. 명령줄 인터페이스를 사용하여 CEPH OSD 추가	58
3.4. 명령줄 인터페이스를 사용하여 CEPH OSD 제거	64
3.5. OSD ID를 유지하면서 OSD 드라이브 교체	68
3.6. ANSIBLE로 배포된 클러스터 제거	70
<b>4장. 컨테이너 내에서 RED HAT CEPH STORAGE 업그레이드</b> .....	<b>72</b>
4.1. 사전 요구 사항	72
4.2. 컨테이너에서 실행되는 RED HAT CEPH STORAGE 클러스터 업그레이드	72
사전 요구 사항	74
4.3. 스토리지 클러스터 업그레이드	75
절차	75

4.4. RED HAT CEPH STORAGE 대시보드 업그레이드	80
<b>5장. RED HAT CEPH STORAGE 대시보드를 사용하여 컨테이너에서 실행 중인 CEPH 클러스터 모니터링</b> .....	<b>82</b>
사전 요구 사항	82
5.1. RED HAT CEPH STORAGE 대시보드	82
5.2. RED HAT CEPH STORAGE 대시보드 설치	87
5.3. RED HAT CEPH STORAGE 대시보드 액세스	90
사전 요구 사항	90
절차	91
추가 리소스	92
5.4. 기본 RED HAT CEPH STORAGE 대시보드 암호 변경	92
5.5. RED HAT CEPH STORAGE용 PROMETHEUS 플러그인	93
5.5.1. 사전 요구 사항	93
5.5.2. Prometheus 플러그인	93
5.5.3. Prometheus 환경 관리	94
5.5.4. Prometheus 데이터 및 쿼리 작업	96
5.5.5. Prometheus 표현식 브라우저 사용	98
5.5.6. 추가 리소스	99
5.6. RED HAT CEPH STORAGE 대시보드 경고	99
5.6.1. 사전 요구 사항	99
5.6.2. 경고 정보	99
5.6.3. 경고 상태 대시보드에 액세스	101
절차	101
5.6.4. 알림 대상 구성	101
절차	101
5.6.5. 기본 경고 및 새 항목 추가 변경	101
절차	101
추가 리소스	102
<b>부록 A. 버전 2와 3의 ANSIBLE 변수 변경</b> .....	<b>103</b>



# 1장. 컨테이너에 RED HAT CEPH STORAGE 배포

이 장에서는 컨테이너에 Ansible 애플리케이션을 **ceph-ansible** 플레이북과 함께 사용하여 Red Hat Ceph Storage 3을 배포하는 방법을 설명합니다.

- Red Hat Ceph Storage를 설치하려면 [1.2절. "컨테이너에 Red Hat Ceph Storage 클러스터 설치"](#) 를 참조하십시오.
- Ceph Object Gateway를 설치하려면 [1.4절. "컨테이너에 Ceph Object Gateway 설치"](#) 를 참조하십시오.
- 메타데이터 서버를 설치하려면 [1.5절. "메타데이터 서버 설치"](#) 를 참조하십시오.
- Ansible **--limit** 옵션에 대한 자세한 내용은 [1.8절. "제한 옵션 이해"](#) 을 참조하십시오.

## 1.1. 사전 요구 사항

- 유효한 고객 서브스크립션을 받으십시오.
- 클러스터 노드를 준비합니다. 각 노드에서 다음을 수행합니다.
  - [노드를 CDN\(Content Delivery Network\)에 등록합니다.](#)
  - **sudo** 액세스 권한을 사용하여 Ansible 사용자를 생성합니다.
  - [암호 없는 SSH 액세스를 활성화합니다.](#)
  - 선택 사항: [방화벽 구성.](#)
  - 선택 사항: [HTTP 프록시 사용.](#)

### 1.1.1. CDN 및 서브스크립션에 Red Hat Ceph Storage 노드 등록

각 RHCS(Red Hat Ceph Storage) 노드를 CDN(Content Delivery Network)에 등록하고 해당 서브스크립션을 연결하여 노드가 소프트웨어 리포지토리에 액세스할 수 있도록 합니다. 각 RHCS 노드는 전체 Red Hat Enterprise Linux 7 기본 콘텐츠와 추가 리포지토리 콘텐츠에 액세스할 수 있어야 합니다.

#### 사전 요구 사항

- 유효한 Red Hat 서브스크립션
- RHCS 노드는 인터넷에 연결할 수 있어야 합니다.
- 설치 중에 인터넷에 액세스할 수 없는 RHCS 노드의 경우 먼저 인터넷에 액세스할 수 있는 시스템에서 다음 단계를 수행해야 합니다.
  1. 로컬 Docker 레지스트리를 시작합니다.

```
# docker run -d -p 5000:5000 --restart=always --name registry registry:2
```

2. Red Hat 고객 포털에서 Red Hat Ceph Storage 3.x 이미지를 가져옵니다.

```
# docker pull registry.access.redhat.com/rhceph/rhceph-3-rhel7
```

3. 이미지에 태그를 지정하십시오.

■

```
# docker tag registry.access.redhat.com/rhceph/rhceph-3-rhel7 <local-host-fqdn>:5000/cephimageinlocalreg
```

<local-host-fqdn> 을 로컬 호스트 FQDN으로 바꿉니다.

4. 시작한 로컬 Docker 레지스트리에 이미지를 푸시합니다.

```
# docker push <local-host-fqdn>:5000/cephimageinlocalreg
```

<local-host-fqdn> 을 로컬 호스트 FQDN으로 바꿉니다.

## 절차

스토리지 클러스터의 모든 노드에서 **root** 사용자로 다음 단계를 수행합니다.

1. 노드를 등록합니다. 메시지가 표시되면 Red Hat Customer Portal 인증 정보를 입력합니다.

```
# subscription-manager register
```

2. CDN에서 최신 서브스크립션 데이터를 가져옵니다.

```
# subscription-manager refresh
```

3. Red Hat Ceph Storage에서 사용 가능한 모든 서브스크립션을 나열합니다.

```
# subscription-manager list --available --all --matches="*Ceph*"
```

적절한 서브스크립션을 식별하고 해당 풀 ID를 검색합니다.

4. 서브스크립션을 연결합니다.

```
# subscription-manager attach --pool=$POOL_ID
```

## replace

- 이전 단계에서 식별한 풀 ID가 있는 **\$POOL\_ID** 입니다.

5. 기본 소프트웨어 리포지토리를 비활성화합니다. 그런 다음 Red Hat Enterprise Linux 7 Server, Red Hat Enterprise Linux 7 Server Extras 및 RHCS 리포지토리를 활성화합니다.

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-extras-rpms
# subscription-manager repos --enable=rhel-7-server-rhceph-3-mon-els-rpms
# subscription-manager repos --enable=rhel-7-server-rhceph-3-osd-els-rpms
# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

6. 최신 패키지를 수신하도록 시스템을 업데이트합니다.

```
# yum update
```

## 추가 리소스

- Red Hat Enterprise Linux 7용 시스템 관리자 가이드의 시스템 및 서브스크립션 관리 장을 참조하십시오.

십시오.

### 1.1.2. sudo 액세스를 사용하여 Ansible 사용자 생성

Ansible은 모든 RHCS(Red Hat Ceph Storage) 노드에 **root** 권한이 있는 사용자로 로그인하고 암호를 요청하지 않고 구성 파일을 생성할 수 있어야 합니다. Ansible을 사용하여 Red Hat Ceph Storage 클러스터를 배포하고 구성할 때 스토리지 클러스터의 모든 노드에서 암호가 없는 루트 액세스 권한을 가진 Ansible 사용자를 생성해야 합니다.

#### 사전 요구 사항

- 스토리지 클러스터의 모든 노드에 **root** 또는 **sudo** 액세스 권한이 있어야 합니다.

#### 절차

1. Ceph 노드에 **root** 사용자로 로그인합니다.

```
ssh root@$HOST_NAME
```

#### replace

- **\$HOST\_NAME** - Ceph 노드의 호스트 이름이 사용됩니다.

#### 예제

```
# ssh root@mon01
```

메시지가 표시되면 루트 암호를 입력합니다.

2. 새 Ansible 사용자를 생성합니다.

```
adduser $USER_NAME
```

#### replace

- **\$USER\_NAME** - Ansible 사용자의 새 사용자 이름을 지정합니다.

#### 예제

```
# adduser admin
```



#### 중요

**ceph** 를 사용자 이름으로 사용하지 마십시오. **ceph** 사용자 이름은 Ceph 데몬용으로 예약되어 있습니다. 클러스터 전체의 균일한 사용자 이름은 사용 편의성을 향상시킬 수 있지만 침입자가 일반적으로 무차별 공격에 사용하기 때문에 명확한 사용자 이름을 사용하지 마십시오.

3. 이 사용자에 대한 새 암호를 설정합니다.

```
# passwd $USER_NAME
```

replace

- **\$USER\_NAME** - Ansible 사용자의 새 사용자 이름을 지정합니다.

예제

```
# passwd admin
```

메시지가 표시되면 새 암호를 두 번 입력합니다.

4. 새로 생성된 사용자에게 대해 **sudo** 액세스를 설정합니다.

```
cat << EOF >/etc/sudoers.d/$USER_NAME
$USER_NAME ALL = (root) NOPASSWD:ALL
EOF
```

replace

- **\$USER\_NAME** - Ansible 사용자의 새 사용자 이름을 지정합니다.

예제

```
# cat << EOF >/etc/sudoers.d/admin
admin ALL = (root) NOPASSWD:ALL
EOF
```

5. 올바른 파일 권한을 새 파일에 할당합니다.

```
chmod 0440 /etc/sudoers.d/$USER_NAME
```

replace

- **\$USER\_NAME** - Ansible 사용자의 새 사용자 이름을 지정합니다.

예제

```
# chmod 0440 /etc/sudoers.d/admin
```

추가 리소스

- [시스템 관리자 가이드의 새 사용자 추가](#) 섹션.

### 1.1.3. Ansible에 대해 암호 없는 SSH 활성화

Ansible 관리 노드에서 SSH 키 쌍을 생성하고, 스토리지 클러스터의 각 노드에 공개 키를 배포하여 Ansible이 암호를 확인하지 않고 노드에 액세스할 수 있도록 합니다.

사전 요구 사항

- **sudo** 액세스 권한을 사용하여 Ansible 사용자를 생성합니다 .

## 절차

Ansible 관리 노드 및 Ansible 사용자로 다음 단계를 수행합니다.

1. SSH 키 쌍을 생성하고 기본 파일 이름을 수락하고 암호를 비워 둡니다.

```
[user@admin ~]$ ssh-keygen
```

2. 스토리지 클러스터의 모든 노드에 공개 키를 복사합니다.

```
ssh-copy-id $USER_NAME@$HOST_NAME
```

### replace

- **\$USER\_NAME** - Ansible 사용자의 새 사용자 이름을 지정합니다.
- **\$HOST\_NAME** - Ceph 노드의 호스트 이름이 사용됩니다.

### 예제

```
[user@admin ~]$ ssh-copy-id admin@ceph-mon01
```

3. `~/.ssh/config` 파일을 만들고 편집합니다.



### 중요

`~/.ssh/config` 파일을 생성하고 편집하여 **ansible-playbook** 명령을 실행할 때마다 **-u \$USER\_NAME** 옵션을 지정할 필요가 없습니다.

- a. SSH 구성 파일을 생성합니다.

```
[user@admin ~]$ touch ~/.ssh/config
```

- b. 편집할 구성 파일을 엽니다. 스토리지 클러스터의 각 노드에 대한 **Hostname** 및 **User** 옵션을 설정합니다.

```
Host node1
  Hostname $HOST_NAME
  User $USER_NAME
Host node2
  Hostname $HOST_NAME
  User $USER_NAME
...
```

### replace

- **\$HOST\_NAME** - Ceph 노드의 호스트 이름이 사용됩니다.
- **\$USER\_NAME** - Ansible 사용자의 새 사용자 이름을 지정합니다.

### 예제

```
Host node1
```

```

Hostname monitor
User admin
Host node2
Hostname osd
User admin
Host node3
Hostname gateway
User admin

```

4. `~/.ssh/config` 파일에 대해 올바른 파일 권한을 설정합니다.

```
[admin@admin ~]$ chmod 600 ~/.ssh/config
```

#### 추가 리소스

- [ssh\\_config\(5\)](#) 매뉴얼 페이지
- Red Hat Enterprise Linux 7용 *시스템 관리자 가이드*의 [OpenSSH](#) 장

### 1.1.4. Red Hat Ceph Storage 방화벽 설정

RHCS(Red Hat Ceph Storage)는 **firewalld** 서비스를 사용합니다.

Monitor 데몬은 Ceph 스토리지 클러스터 내 통신에 포트 **6789** 를 사용합니다.

각 Ceph OSD 노드에서 OSD 데몬은 **6800-7300** 범위의 여러 포트를 사용합니다.

- 하나는 클라이언트와 통신하고 공용 네트워크를 통해 모니터링합니다.
- 사용 가능한 경우 클러스터 네트워크를 통해 다른 OSD로 데이터를 전송하는 방법
- 사용 가능한 경우 클러스터 네트워크에서 하트비트 패킷을 교환하는 방법

Ceph Manager(**ceph-mgr**) 데몬은 **6800-7300** 범위의 포트를 사용합니다. 동일한 노드에서 Ceph Monitor 를 사용하여 **ceph-mgr** 데몬을 공동 배치하는 것이 좋습니다.

Ceph Metadata Server 노드(**ceph-mds**)는 **6800-7300** 범위의 포트를 사용합니다.

Ceph Object Gateway 노드는 기본적으로 포트 **8080** 을 사용하도록 Ansible에서 구성합니다. 그러나 기본 포트(예: 포트 **80**)를 변경할 수 있습니다.

SSL/TLS 서비스를 사용하려면 포트 **443** 을 엽니다.

#### 사전 요구 사항

- 네트워크 하드웨어가 연결되어 있습니다.

#### 절차

**root** 사용자로 다음 명령을 실행합니다.

1. 모든 RHCS 노드에서 **firewalld** 서비스를 시작합니다. 부팅 시 실행되도록 활성화하고 실행 중인 지 확인합니다.

```
# systemctl enable firewalld
# systemctl start firewalld
# systemctl status firewalld
```

- 모든 모니터 노드에서 공용 네트워크에서 포트 **6789** 를 엽니다.

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

소스 주소를 기반으로 액세스를 제한하려면 다음을 수행합니다.

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="6789" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="6789" accept" --permanent
```

### replace

- Monitor 노드의 네트워크 주소가 있는 `ip_address` 입니다.
- CIDR 표기법으로 넷마스크가 있는 넷마스크 `_prefix`.

### 예제

```
[root@monitor ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.11/24" port protocol="tcp" \
port="6789" accept"
```

```
[root@monitor ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.11/24" port protocol="tcp" \
port="6789" accept" --permanent
```

- 모든 OSD 노드에서 공용 네트워크에서 포트 **6800-7300** 을 엽니다.

```
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

별도의 클러스터 네트워크가 있는 경우 해당 영역에서 명령을 반복합니다.

- 모든 Ceph Manager(**ceph-mgr**) 노드(일반적으로 Monitor와 동일한 노드)에서 공용 네트워크에서 포트 **6800-7300** 을 엽니다.

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

별도의 클러스터 네트워크가 있는 경우 해당 영역에서 명령을 반복합니다.

- 모든 Ceph 메타데이터 서버(**ceph-mds**) 노드에서 공용 네트워크에서 포트 **6800** 을 엽니다.

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800/tcp --permanent
```

별도의 클러스터 네트워크가 있는 경우 해당 영역에서 명령을 반복합니다.

6. 모든 Ceph Object Gateway 노드에서 공용 네트워크에서 관련 포트 또는 포트를 엽니다.

a. 기본 Ansible 구성된 **8080** 포트를 엽니다.

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

소스 주소를 기반으로 액세스를 제한하려면 다음을 수행합니다.

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="8080" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="8080" accept" --permanent
```

replace

- 오브젝트 게이트웨이 노드의 네트워크 주소가 있는 `ip_address`.
- CIDR 표기법으로 넷마스크가 있는 넷마스크 `_prefix`.

예제

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept" --permanent
```

b. 선택 사항: Ansible을 사용하여 Ceph Object Gateway를 설치하고 Ansible에서 **8080**에서 사용하도록 Ceph Object Gateway를 구성하는 기본 포트를 변경된 경우 포트 **80**으로 이 포트를 엽니다.

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
```

소스 주소를 기반으로 액세스를 제한하려면 다음 명령을 실행합니다.

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="80" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="80" accept" --permanent
```

#### replace

- 오브젝트 게이트웨이 노드의 네트워크 주소가 있는 ip\_address.
- CIDR 표기법으로 넷마스크가 있는 넷마스크\_prefix.

#### 예제

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept" --permanent
```

- c. 선택 사항: SSL/TLS를 사용하려면 포트 **443**을 엽니다.

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp --permanent
```

소스 주소를 기반으로 액세스를 제한하려면 다음 명령을 실행합니다.

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="443" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="443" accept" --permanent
```

#### replace

- 오브젝트 게이트웨이 노드의 네트워크 주소가 있는 ip\_address.
- CIDR 표기법으로 넷마스크가 있는 넷마스크\_prefix.

#### 예제

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept" --permanent
```

- 공용 및 클러스터 네트워크에 대한 자세한 내용은 [Red Hat Ceph Storage의 네트워크 구성 확인](#)을 참조하십시오.
- **firewalld**에 대한 자세한 내용은 Red Hat Enterprise Linux 7의 보안 가이드의 [방화벽 사용](#) 장을 참조하십시오.

### 1.1.5. HTTP 프록시 사용

Ceph 노드가 HTTP/HTTPS 프록시 뒤에 있는 경우 레지스트리의 이미지에 액세스하도록 docker를 구성해야 합니다. HTTP/HTTPS 프록시를 사용하여 Docker에 대한 액세스 권한을 구성하려면 다음 절차를 수행하십시오.

#### 사전 요구 사항

- 실행 중인 HTTP/HTTPS 프록시

#### 절차

1. **root**로서 docker 서비스에 대한 systemd 디렉터리를 생성합니다.

```
# mkdir /etc/systemd/system/docker.service.d/
```

2. 루트로서 HTTP/HTTPS 구성 파일을 만듭니다.

- a. HTTP의 경우 **/etc/systemd/system/docker.service.d/http-proxy.conf** 파일을 생성하고 파일에 다음 행을 추가합니다.

```
[Service]
Environment="HTTP_PROXY=http://proxy.example.com:80/"
```

- b. HTTPS의 경우 **/etc/systemd/system/docker.service.d/https-proxy.conf** 파일을 생성하고 파일에 다음 행을 추가합니다.

```
[Service]
Environment="HTTPS_PROXY=https://proxy.example.com:443/"
```

3. **root**로서 **ceph-ansible** 플레이북을 실행하기 전에 HTTP/HTTPS 구성 파일을 스토리지 클러스터의 모든 Ceph 노드에 복사합니다.

## 1.2. 컨테이너에 RED HAT CEPH STORAGE 클러스터 설치

**ceph-ansible** 플레이북과 함께 Ansible 애플리케이션을 사용하여 컨테이너에 Red Hat Ceph Storage 3을 설치합니다.

프로덕션에서 사용되는 Ceph 클러스터는 일반적으로 10개 이상의 노드로 구성됩니다. Red Hat Ceph Storage를 컨테이너 이미지로 배포하려면 3개 이상의 OSD 노드와 모니터 3개로 구성된 Ceph 클러스터를 사용하는 것이 좋습니다.



#### 중요

그러나 프로덕션 클러스터에서 고가용성을 보장하기 위해 Ceph를 하나의 모니터로 실행할 수 있지만, Red Hat은 최소 3개의 모니터 노드가 있는 배포만 지원합니다.

#### 사전 요구 사항

- Ansible 관리 노드에서 root 사용자 계정을 사용하여 Red Hat Ceph Storage 3 툴 리포지토리 및 Ansible 리포지토리를 활성화합니다.

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
--enable=rhel-7-server-ansible-2.6-rpms
```

- **ceph-ansible** 패키지를 설치합니다.

```
[root@admin ~]# yum install ceph-ansible
```

## 절차

달리 지침이 없는 경우 Ansible 관리 노드에서 다음 명령을 실행합니다.

1. Ansible 사용자로 Ansible에서 **ceph-ansible** 플레이북에서 생성한 임시 값을 저장하는 **ceph-ansible-keys** 디렉토리를 생성합니다.

```
[user@admin ~]$ mkdir ~/ceph-ansible-keys
```

2. root로 **/etc/ansible/** 디렉터리의 **/usr/share/ceph-ansible/group\_vars** 디렉터리에 대한 심볼릭 링크를 생성합니다.

```
[root@admin ~]# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

3. **/usr/share/ceph-ansible/** 디렉터리로 이동합니다.

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

4. **yml.sample** 파일의 새 복사본을 생성합니다.

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp site-docker.yml.sample site-docker.yml
```

5. 복사된 파일을 편집합니다.

- a. **group\_vars/all.yml** 파일을 편집합니다. 주석 처리를 위해 가장 일반적인 필수 매개 변수와 선택적 매개 변수는 아래 표를 참조하십시오. 테이블에는 모든 매개 변수가 포함되어 있지 않습니다.



### 중요

사용자 지정 클러스터 이름 사용은 지원되지 않으므로 **cluster: ceph** 매개 변수를 **ceph** 이외의 값으로 설정하지 마십시오.

표 1.1. 일반 Ansible 설정

옵션

값

필수 항목

참고

옵션	값	필수 항목	참고
<b>monitor_interface</b>	모니터 노드가 수신하는 인터페이스	<b>monitor_interface</b> , <b>monitor_address</b> 또는 <b>monitor_address_block</b> 이 필요합니다.	노드의 IP 주소를 알 수 없지만 서브넷을 알 수 없는 경우 사용합니다.
<b>monitor_address</b>	모니터 노드가 수신하는 주소입니다.		
<b>monitor_address_block</b>	Ceph 공용 네트워크의 서브넷		
<b>ip_version</b>	<b>ipv6</b>	예: IPv6 주소를 사용하는 경우	
<b>journal_size</b>	저널의 필요한 크기 (MB)	없음	
<b>public_network</b>	Ceph 공용 네트워크의 IP 주소 및 넷마스크	있음	Red Hat <i>Enterprise Linux</i> 용 설치 가이드의 <a href="#">Red Hat Ceph Storage의 네트워크 구성 확인</a> 섹션
<b>cluster_network</b>	Ceph 클러스터 네트워크의 IP 주소 및 넷마스크	없음	
<b>ceph_docker_image</b>	로컬 Docker 레지스트리를 사용하는 경우 <b>rhceph/rhceph-3-rhel7</b> 또는 <b>cephimageinlocal reg</b>	있음	
<b>containerized_deployment</b>	<b>true</b>	있음	
<b>ceph_docker_registry</b>	로컬 Docker 레지스트리를 사용하는 경우 <b>registry.access.redhat.com</b> 또는 <b>&lt;local-host-fqdn&gt;</b>	있음	

**all.yml** 파일의 예는 다음과 같습니다.

```
monitor_interface: eth0
journal_size: 5120
public_network: 192.168.0.0/24
```

```
ceph_docker_image: rhceph/rhceph-3-rhel7
containerized_deployment: true
ceph_docker_registry: registry.access.redhat.com
```

자세한 내용은 **all.yml** 파일을 참조하십시오.

- b. **group\_vars/osds.yml** 파일을 편집합니다. 주석 처리를 위해 가장 일반적인 필수 매개 변수와 선택적 매개 변수는 아래 표를 참조하십시오. 테이블에는 모든 매개 변수가 포함되어 있지 않습니다.



**중요**

다른 물리적 장치를 사용하여 운영 체제가 설치된 장치와 OSD를 설치합니다. 운영 체제와 OSD 간에 동일한 장치를 공유하면 성능에 문제가 발생합니다.

표 1.2. OSD Ansible 설정

옵션	값	필수 항목	참고
<b>osd_scenario</b>	쓰기 로깅 및 키/값 데이터(BlueStore) 또는 저널(FileStore) 및 OSD 데이터에 동일한 장치를 사용하도록 배치 됨  SSD 또는 NVMe 미디어와 같은 전용 장치를 사용하여 쓰기 로그 및 키/값 데이터(BlueStore) 또는 저널 데이터(FileStore)를 저장합니다.  <b>LVM</b> 에서 논리 볼륨 관리자를 사용하여 <b>OSD</b> 데이터를 저장합니다.	있음	<b>osd_scenario</b> 를 사용하는 경우, <b>ceph-ansible</b> 은 장치 및 <b>dedicated_device</b> s의 변수 수가 일치할 것으로 예상합니다. 예를 들어 <b>장치에서</b> 10 디스크를 지정하는 경우 <b>dedicated_device</b> s에 10개의 항목을 지정해야 합니다.
<b>osd_auto_discovery</b>	OSD를 자동으로 검색하려면 <b>true</b>	예: <b>osd_scenario: colocated</b>	장치 설정을 사용할 때는 사용할 수 없습니다.

옵션	값	필수 항목	참고
<b>devices</b>	<b>ceph</b> 데이터가 저장된 장치 목록	예: 장치 목록을 지정합니다.	<b>osd_auto_discovery</b> 설정이 사용되는 경우 사용할 수 없습니다. <b>lvm</b> 을 <b>osd_scenario</b> 로 사용하고 <b>devices</b> 옵션을 설정하면 <b>ceph-volume lvm</b> 배치 모드를 사용하면 최적화된 OSD 구성이 생성됩니다.
<b>dedicated_devices</b>	<b>ceph</b> 저널이 저장된 non-collocated OSD의 전용 장치 목록	<b>osd_scenario: non-collocated</b>	파티션되지 않은 장치여야 합니다.
<b>dmccrypt</b>	OSD 암호화	없음	기본값은 <b>false</b>
<b>lvm_volumes</b>	FileStore 또는 BlueStore 사전 목록	예: <b>osd_scenario</b> 를 사용하는 경우: <b>lvm</b> 및 스토리지 장치는 장치를 사용하여 정의되지 않습니다.	각 사전에는 데이터 <b>journal</b> 및 <b>data_vg</b> 키가 포함되어야 합니다. 논리 볼륨 또는 볼륨 그룹은 전체 경로가 아닌 이름이어야 합니다. 데이터, 및 저널 키는 논리 볼륨(LV) 또는 파티션이 될 수 있지만 여러 데이터 LV에는 저널을 사용하지 않습니다. <b>data_vg</b> 키는 <b>data</b> LV를 포함하는 볼륨 그룹이어야 합니다. 필요한 경우 <b>journal_vg</b> 키를 사용하여 저널 LV가 포함된 볼륨 그룹을 지정할 수 있습니다. 지원되는 다양한 구성은 아래 예제를 참조하십시오.
<b>osds_per_device</b>	장치당 생성할 OSD 수입니다.	없음	기본값은 <b>1</b> 입니다.

옵션	값	필수 항목	참고
<b>osd_objectstore</b>	OSD의 Ceph 오브젝트 저장소 유형입니다.	없음	기본값은 <b>bluestore</b> 입니다. 다른 옵션은 <b>filestore</b> 입니다. 업그레이드에 필요합니다.

다음은 세 개의 OSD 시나리오(, 결합되지 않은, , **lvm**)를 사용할 때 **osds.yml** 파일의 예입니다. 기본 **OSD** 개체 저장소 형식은 지정되지 않은 경우 **BlueStore**입니다.

### collocated

```
osd_objectstore: filestore
osd_scenario: collocated
devices:
- /dev/sda
- /dev/sdb
```

### 지원되지 않는 - BlueStore

```
osd_objectstore: bluestore
osd_scenario: non-collocated
devices:
- /dev/sda
- /dev/sdb
- /dev/sdc
- /dev/sdd
dedicated_devices:
- /dev/nvme0n1
- /dev/nvme0n1
- /dev/nvme1n1
- /dev/nvme1n1
```

이 지원되지 않는 예에서는 장치당 하나씩 4개의 **BlueStore OSD**를 생성합니다. 이 예에서 기존의 하드 드라이브(**sda,sdb,sdc,sdd**)는 오브젝트 데이터에 사용되며 솔리드 상태 드라이브(**SSD**) (**/dev/nvme0n1,/dev/nvme1n1**)는 **BlueStore** 데이터베이스 및 **write-ahead**

로그에 사용됩니다. 이 구성은 `/dev/sda` 및 `/dev/sdb` 장치를 `/dev/nvme0n1` 장치를 사용하여 연결하고 `/dev/sdc` 및 `/dev/sdd` 장치를 `/dev/nvme1n1` 장치와 쌍으로 연결합니다.

지원되지 않음 - 파일 저장소

```
osd_objectstore: filestore
osd_scenario: non-collocated
devices:
- /dev/sda
- /dev/sdb
- /dev/sdc
- /dev/sdd
dedicated_devices:
- /dev/nvme0n1
- /dev/nvme0n1
- /dev/nvme1n1
- /dev/nvme1n1
```

## LVM simple

```
osd_objectstore: bluestore
osd_scenario: lvm
devices:
- /dev/sda
- /dev/sdb
```

또는

```
osd_objectstore: bluestore
osd_scenario: lvm
devices:
- /dev/sda
- /dev/sdb
- /dev/nvme0n1
```

이러한 간단한 구성으로 `ceph-ansible` 은 배치 모드(`ceph-volume lvm batch`)를 사용하여 **OSD**를 생성합니다.

첫 번째 시나리오에서는 장치가 기존 하드 드라이브 또는 **SSD**인 경우 장치당 하나의 **OSD**가 생성됩니다.

두 번째 시나리오에서는 기존 하드 드라이브와 **SSD**가 혼합되어 있는 경우 데이터는 기존 하드 드라이브(**sda,sdb**)에 배치되고 **BlueStore** 데이터베이스(**block.db**)는 **SSD(nvme0n1)**에서 최대한 크게 생성됩니다.

## LVM advance

```
osd_objectstore: filestore
osd_scenario: lvm
lvm_volumes:
- data: data-lv1
  data_vg: vg1
  journal: journal-lv1
  journal_vg: vg2
- data: data-lv2
  journal: /dev/sda
  data_vg: vg1
```

또는

```
osd_objectstore: bluestore
osd_scenario: lvm
lvm_volumes:
- data: data-lv1
  data_vg: data-vg1
  db: db-lv1
  db_vg: db-vg1
  wal: wal-lv1
  wal_vg: wal-vg1
- data: data-lv2
  data_vg: data-vg2
  db: db-lv2
  db_vg: db-vg2
  wal: wal-lv2
  wal_vg: wal-vg2
```

이러한 고급 시나리오 예제를 사용하면 볼륨 그룹과 논리 볼륨을 사전에 생성해야 합니다. **ceph-ansible** 에서 생성되지 않습니다.



## 참고

모든 NVMe SSD를 사용하는 경우 `osd_scenario: lvm` 및 `osds_per_device`를 설정합니다. 4 옵션. 자세한 내용은 [Red Hat Ceph Storage Container 가이드의 모든 NVMe 스토리지에 대한 OSD Ansible 설정 구성](#) 섹션을 참조하십시오.

자세한 내용은 `osds.yml` 파일의 주석을 참조하십시오.

6.

`/etc/ansible/hosts` 에 기본적으로 있는 **Ansible** 인벤토리 파일을 편집합니다. 예제 호스트를 주석 처리하십시오.

a.

**[mons]** 섹션 아래에 모니터 노드를 추가합니다.

```
[mons]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

b.

**[osds]** 섹션 아래에 **OSD** 노드를 추가합니다. 노드의 이름이 순차적인 경우 범위를 사용하는 것이 좋습니다.

```
[osds]
<osd-host-name[1:10]>
```



## 참고

새 설치의 **OSD**의 경우 기본 오브젝트 저장소 형식은 **BlueStore**입니다.

또는 **[mons]** 및 **[osds]** 섹션에 동일한 노드를 추가하여 한 노드에서 **OSD** 데몬과 함께 모니터를 배치할 수 있습니다. 자세한 내용은 [2장. 컨테이너화된 Ceph 데몬 공동 배치](#) 을 참조하십시오.

7.

필요한 경우 모든 배포, 베어 메탈 또는 컨테이너 의 경우 **ansible- playbook** 을 사용하여 사용자 지정 **CRUSH** 계층을 생성할 수 있습니다.

a.

**Ansible** 인벤토리 파일을 설정합니다. `osd_crush_location` 매개변수를 사용하여 **OSD**

호스트가 **CRUSH** 맵 계층 구조에 있을 위치를 지정합니다. **OSD** 위치를 지정하려면 최소 2개의 **CRUSH** 버킷 유형을 지정해야 하며 하나의 버킷 유형이 호스트여야 합니다. 기본적으로 루트,데이터 센터,방,행,**pod**, **du** ,랙,새시 및 호스트가 포함됩니다.

구문

```
[osds]
CEPH_OSD_NAME osd_crush_location="{ 'root': ROOT_BUCKET_', 'rack':
'RACK_BUCKET', 'pod': 'POD_BUCKET', 'host': 'CEPH_HOST_NAME }"
```

예제

```
[osds]
ceph-osd-01 osd_crush_location="{ 'root': 'default', 'rack': 'rack1', 'pod': 'monpod', 'host':
'ceph-osd-01' }"
```

b.

**crush\_rule\_config** 및 **create\_crush\_tree** 매개변수를 **True** 로 설정하고 기본 **CRUSH** 규칙을 사용하지 않으려면 최소 하나의 **CRUSH** 규칙을 만듭니다. 예를 들어 **HDD** 장치를 사용하는 경우 다음과 같이 매개 변수를 편집합니다.

```
crush_rule_config: True
crush_rule_hdd:
  name: replicated_hdd_rule
  root: root-hdd
  type: host
  class: hdd
  default: True
crush_rules:
  - "{{ crush_rule_hdd }}"
create_crush_tree: True
```

**SSD** 장치를 사용하는 경우 다음과 같이 매개 변수를 편집합니다.

```
crush_rule_config: True
crush_rule_ssd:
  name: replicated_ssd_rule
```

```

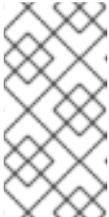
root: root-ssd
type: host
class: ssd
default: True
crush_rules:
- "{{ crush_rule_ssd }}"
create_crush_tree: True

```



참고

기본 규칙에는 이제 정의해야 하는 **class** 매개 변수가 포함되어 있으므로 **ssd** 및 **hdd** OSD 둘 다 배포되지 않으면 기본 **CRUSH** 규칙이 실패합니다.



참고

아래 단계에 설명된 대로 **host\_vars** 디렉터리의 **OSD** 파일에 사용자 지정 **nmap** 계층 구조를 추가하여 이 구성 작업을 수행합니다.

c.

**group\_vars/clients.yml** 파일에서 생성된 **crush\_rules** 를 사용하여 풀 생성.

예제

```

copy_admin_key: True
user_config: True
pool1:
  name: "pool1"
  pg_num: 128
  pgp_num: 128
  rule_name: "HDD"
  type: "replicated"
  device_class: "hdd"
pools:
- "{{ pool1 }}"

```

d.

트리를 봅니다.

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

e.

풀을 검증합니다.

```
# for i in $(rados lspools);do echo "pool: $i"; ceph osd pool get $i crush_rule;done

pool: pool1
crush_rule: HDD
```

8.

모든 배포, 베어 메탈 또는 컨테이너 의 경우 기본적으로 **/etc/ansible/hosts** 파일을 통해 **Ansible** 인벤토리 파일을 편집하기 위해 를 엽니다. 예제 호스트를 주석 처리합니다.

a.

**[mgrs]** 섹션에 **Ceph Manager(ceph-mgr)** 노드를 추가합니다. **Ceph Manager** 데몬을 모니터 노드와 분리합니다.

```
[mgrs]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

9.

**Ansible** 사용자로 **Ansible**이 **Ceph** 호스트에 연결할 수 있는지 확인합니다.

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

10.

**root** 로서 **/var/log/ansible/** 디렉터리를 생성하고 **ansible** 사용자에게 적절한 권한을 할당합니다.

```
[root@admin ~]# mkdir /var/log/ansible
[root@admin ~]# chown ansible:ansible /var/log/ansible
[root@admin ~]# chmod 755 /var/log/ansible
```

a.

**/usr/share/ceph-ansible/ansible.cfg** 파일을 편집하여 다음과 같이 **log\_path** 값을 업데이트합니다.

```
log_path = /var/log/ansible/ansible.log
```

11.

**Ansible** 사용자로 **/usr/share/ceph-ansible/** 디렉터리로 변경합니다.

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

12.

**ceph-ansible** 플레이북을 실행합니다.

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml
```



참고

**Red Hat Ceph Storage**를 **Red Hat Enterprise Linux Atomic Host** 호스트에 배포하는 경우 **--skip-tags=with\_pkg** 옵션을 사용합니다.

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --skip-tags=with_pkg
```



참고

배포 속도를 높이려면 **--forks** 옵션을 **ansible-playbook** 에 사용합니다. 기본적으로 **ceph-ansible** 은 포크를 **20** 으로 설정합니다. 이 설정을 사용하면 최대 **20** 개의 노드가 동시에 설치됩니다. 한 번에 최대 **30**개의 노드를 설치하려면 **ansible-playbook --forks 30 PLAYBOOK FILE**을 실행합니다. 사용하지 않도록 관리 노드의 리소스를 모니터링해야 합니다. 이 값이 있는 경우 **--forks** 에 전달된 수를 줄입니다.

13.

**Monitor** 노드에서 **root** 계정을 사용하여 **Ceph** 클러스터의 상태를 확인합니다.

```
docker exec ceph-<mon|mgr>-<id> ceph health
```

교체:



**Monitor** 노드의 호스트 이름이 있는 **<id>**:

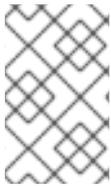
예를 들면 다음과 같습니다.

```
[root@monitor ~]# docker exec ceph-mon-mon0 ceph health
HEALTH_OK
```

### 1.3. 모든 NVME 스토리지에 대한 OSD ANSIBLE 설정 구성

스토리지에 NVMe(Non-volatile Memory express) 장치만 사용할 때 성능을 최적화하려면 각 NVMe

장치에서 4개의 OSD를 구성합니다. 일반적으로 하나의 OSD만 장치별로 구성되며 NVMe 장치의 처리량이 낮습니다.



참고

SSD와 HDD를 혼합하는 경우 OSD가 아닌 저널 또는 `block.db` 에 SSD가 사용됩니다.



참고

테스트에서는 각 NVMe 장치에서 OSD 4개를 구성하는 것이 최적의 성능을 제공하는 것으로 확인되었습니다. `osds_per_device`를 설정하는 것이 좋습니다. 4 하지만 필수는 아닙니다. 다른 값은 사용자 환경에서 더 나은 성능을 제공할 수 있습니다.

### 사전 요구 사항

- Ceph 클러스터에 대한 모든 소프트웨어 및 하드웨어 요구 사항을 충족합니다.

### 절차

1. `osd_scenario: lvm` 및 `osds_per_device`를 설정합니다. `4 group_vars/osds.yml`:

```
osd_scenario: lvm
osds_per_device: 4
```

2. 장치에서 NVMe 장치 나열:

```
devices:
- /dev/nvme0n1
- /dev/nvme1n1
- /dev/nvme2n1
- /dev/nvme3n1
```

3. `group_vars/osds.yml` 의 설정은 다음 예와 유사합니다.

```
osd_scenario: lvm
osds_per_device: 4
devices:
- /dev/nvme0n1
- /dev/nvme1n1
- /dev/nvme2n1
- /dev/nvme3n1
```



## 참고

**lvm\_volumes** 가 아닌 이 구성으로 장치를 사용해야 합니다. 이는 **lvm\_volumes** 가 일반적으로 미리 생성된 논리 볼륨과 함께 사용되고 **osds\_per\_device** 는 Ceph에서 자동 논리 볼륨을 생성하기 때문입니다.

## 추가 리소스

- [Red Hat Enterprise Linux에 Red Hat Ceph Storage 클러스터 설치](#)
- [Ubuntu에 Red Hat Ceph Storage 클러스터 설치](#)

## 1.4. 컨테이너에 CEPH OBJECT GATEWAY 설치

**ceph-ansible** 플레이북과 함께 **Ansible** 애플리케이션을 사용하여 컨테이너에 **Ceph Object Gateway** 를 설치합니다.

## 사전 요구 사항

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

## 절차

별도로 지정하지 않는 한 **Ansible** 관리 노드에서 다음 명령을 실행합니다.

1. **root** 사용자로 **/usr/share/ceph-ansible/** 디렉터리로 이동합니다.

```
[root@admin ~]# cd /usr/share/ceph-ansible/
```

2. **group\_vars/all.yml** 파일의 **radosgw\_interface** 매개변수의 주석을 제거합니다.

```
radosgw_interface: interface
```

**Ceph Object Gateway** 노드가 수신 대기하는 인터페이스로 인터페이스를 바꿉니다.

3.

선택 사항: 기본 변수를 변경합니다.

a.

**group\_vars** 디렉터리에 있는 **rgws.yml.sample** 파일의 새 사본을 생성합니다.

```
[root@admin ceph-ansible]# cp group_vars/rgws.yml.sample group_vars/rgws.yml
```

b.

**group\_vars/rgws.yml** 파일을 편집합니다. 자세한 내용은 **rgws.yml** 파일을 참조하십시오.

4.

기본적으로 **/etc/ansible/hosts** 에 있는 **Ansible** 인벤토리 파일의 **[rgws]** 섹션에 **Ceph Object Gateway** 노드의 호스트 이름을 추가합니다.

```
[rgws]
gateway01
```

또는 **[osds]** 및 **[rgws]** 섹션에 동일한 노드를 추가하여 하나의 노드에서 **OSD** 데몬과 **Ceph Object Gateway**를 배치할 수 있습니다. 자세한 내용은 컨테이너화된 **Ceph** 데몬 공동 배치를 참조하십시오.

5.

**Ansible** 사용자로 **ceph-ansible** 플레이북을 실행합니다.

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit rgws
```



참고

**Red Hat Ceph Storage**를 **Red Hat Enterprise Linux Atomic Host** 호스트에 배포하는 경우 **--skip-tags=with\_pkg** 옵션을 사용합니다.

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --skip-tags=with_pkg
```

6.

**Ceph Object Gateway** 노드가 성공적으로 배포되었는지 확인합니다.

a.

**root** 사용자로 모니터 노드에 연결합니다.

```
ssh hostname
```

`hostname` 을 **Monitor** 노드의 호스트 이름으로 교체합니다. 예를 들면 다음과 같습니다.

```
[user@admin ~]$ ssh root@monitor
```

b.

**Ceph Object Gateway** 플러그인이 올바르게 생성되었는지 확인합니다.

```
[root@monitor ~]# docker exec ceph-mon-mon1 rados lspools
rbd
cephfs_data
cephfs_metadata
.rgw.root
default.rgw.control
default.rgw.data.root
default.rgw.gc
default.rgw.log
default.rgw.users.uid
```

c.

**Ceph** 클러스터와 동일한 네트워크의 모든 클라이언트에서(예: **Monitor** 노드)에서 **Ceph Object Gateway** 호스트의 IP 주소를 사용하여 포트 8080에 HTTP 요청을 보냅니다.

```
curl http://IP-address:8080
```

`IP-address` 를 **Ceph Object Gateway** 노드의 IP 주소로 바꿉니다. **Ceph Object Gateway** 호스트의 IP 주소를 확인하려면 `ifconfig` 또는 `ip` 명령을 사용합니다.

```
[root@client ~]# curl http://192.168.122.199:8080
<?xml version="1.0" encoding="UTF-8"?><ListAllMyBucketsResult
xmlns="http://s3.amazonaws.com/doc/2006-03-01/"><Owner><ID>anonymous</ID>
<DisplayName></DisplayName></Owner><Buckets></Buckets>
</ListAllMyBucketsResult>
```

d.

버킷 나열:

```
[root@monitor ~]# docker exec ceph-mon-mon1 radosgw-admin bucket list
```

추가 리소스

- [Red Hat Enterprise Linux용 Red Hat Ceph Storage 3 Ceph Object Gateway 가이드](#)
- [제한 옵션 이해](#)

## 1.5. 메타데이터 서버 설치

**Ansible** 자동화 애플리케이션을 사용하여 **Ceph Metadata Server(MDS)**를 설치합니다. **Ceph** 파일 시스템을 배포하려면 **metadata** 서버 데몬이 필요합니다.

### 사전 요구 사항

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

### 절차

**Ansible** 관리 노드에서 다음 단계를 수행합니다.

1. **/etc/ansible/hosts** 파일에 새 섹션 **[mdss]** 를 추가합니다.

```
[mdss]
hostname
hostname
hostname
```

**hostname** 을 **Ceph** 메타데이터 서버를 설치하려는 노드의 호스트 이름으로 바꿉니다.

또는 **[osds]** 및 **[mds]** 섹션에 동일한 노드를 추가하여 한 노드에서 메타데이터 서버를 **OSD** 데몬과 배치할 수 있습니다. 자세한 내용은 [컨테이너화된 Ceph 데몬 공동 배치](#)를 참조하십시오.

2. **/usr/share/ceph-ansible** 디렉토리로 이동합니다.

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. 선택 사항: 기본 변수를 변경합니다.

- a. **mdss.yml** 이라는 **group\_vars/mdss.yml.sample** 파일의 사본을 생성합니다.

```
[root@admin ceph-ansible]# cp group_vars/mdss.yml.sample group_vars/mdss.yml
```

- b. 선택적으로 **mdss.yml** 의 매개변수를 편집합니다. 자세한 내용은 **mdss.yml** 을 참조하

십시오.

4. **Ansible** 사용자로 **Ansible** 플레이북을 실행합니다.

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit mdss
```

5. 메타데이터 서버를 설치한 후 구성합니다. 자세한 내용은 **Red Hat Ceph Storage 3용 Ceph 파일 시스템 가이드의 메타데이터 서버 데몬 구성** 장을 참조하십시오.

#### 추가 리소스

- **Red Hat Ceph Storage 3용 Ceph 파일 시스템 가이드**
- [제한 옵션 이해](#)

### 1.6. NFS-GANESHA 게이트웨이 설치

**Ceph NFS Ganesha** 게이트웨이는 파일 시스템 내의 파일을 **Ceph Object Storage**로 마이그레이션하기 위한 **POSIX** 파일 시스템 인터페이스를 **Ceph Object Gateway**에 제공하기 위해 **Ceph Object Gateway** 상단에 구축된 **NFS** 인터페이스입니다.

#### 사전 요구 사항

- 실행 중인 **Ceph** 스토리지 클러스터(특히 활성 + 정리 상태)입니다.
- **Ceph Object Gateway**를 실행하는 하나 이상의 노드.
- [시작하기 전에](#) 절차를 수행하십시오.

#### 절차

**Ansible** 관리 노드에서 다음 작업을 수행합니다.

1. 샘플 파일에서 **nfss** 파일을 생성합니다.

```
[root@ansible ~]# cd /usr/share/ceph-ansible/group_vars
[root@ansible ~]# cp nfss.yml.sample nfss.yml
```

2.

**[nfss]** 그룹 아래의 **/etc/ansible/hosts** 파일에 게이트웨이 호스트를 추가하여 **Ansible**에 대한 그룹 멤버십을 식별합니다. 호스트에 순차적 이름 지정이 있는 경우 범위를 사용합니다. 예를 들면 다음과 같습니다.

```
[nfss]
<nfs_host_name_1>
<nfs_host_name_2>
<nfs_host_name[3..10]>
```

3.

**Ansible** 구성 디렉터리 **/etc/ansible/**로 이동합니다.

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

4.

관리자 키를 **Ceph Object Gateway** 노드에 복사하려면 **/usr/share/ceph-ansible/group\_vars/nfss.yml** 파일의 **copy\_admin\_key** 설정의 주석 처리를 해제합니다.

```
copy_admin_key: true
```

5.

**/usr/share/ceph-ansible/group\_vars/nfss.yml** 파일의 **FSAL (File System Abstraction Layer)** 섹션을 구성합니다. **ID**, **S3** 사용자 **ID**, **S3** 액세스 키 및 시크릿을 제공합니다. **NFSv4**의 경우 다음과 같이 표시됩니다.

```
#####
# FSAL RGW Config #
#####
#ceph_nfs_rgw_export_id: <replace-w-numeric-export-id>
#ceph_nfs_rgw_pseudo_path: "/"
#ceph_nfs_rgw_protocols: "3,4"
#ceph_nfs_rgw_access_type: "RW"
#ceph_nfs_rgw_user: "cephnfs"
# Note: keys are optional and can be generated, but not on containerized, where
# they must be configured.
#ceph_nfs_rgw_access_key: "<replace-w-access-key>"
#ceph_nfs_rgw_secret_key: "<replace-w-secret-key>"
```



주의

액세스 및 시크릿 키는 선택 사항이며 생성할 수 있습니다.

6.

**Ansible Playbook**을 실행합니다.

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit nfss
```

추가 리소스

- [1.8절. “제한 옵션 이해”](#)
- [Red Hat Enterprise Linux용 오브젝트 게이트웨이 가이드](#)

## 1.7. 컨테이너에 CEPH ISCSI 게이트웨이 설치

**Ansible** 배포 애플리케이션은 컨테이너에서 **Ceph iSCSI** 게이트웨이를 구성하는 데 필요한 데몬 및 툴을 설치합니다.

사전 요구 사항

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

절차

1. **root** 사용자로 **/etc/ansible/hosts** 파일을 열고 편집합니다. **iSCSI** 게이트웨이 그룹에 노드 이름을 항목을 추가합니다.

예제

```
[iscsigws]
ceph-igw-1
ceph-igw-2
```

2. `/usr/share/ceph-ansible` 디렉토리로 이동합니다.

```
[root@admin ~]# cd /usr/share/ceph-ansible/
```

3. `iscsigws.yml.sample` 파일의 사본을 만들고 이름을 `iscsigws.yml` 로 지정합니다.

```
[root@admin ceph-ansible]# cp group_vars/iscsigws.yml.sample group_vars/iscsigws.yml
```

#### 중요

새 파일 이름(`iscsigws.yml`)과 새 섹션 제목(`[iscsigws]`)은 Red Hat Ceph Storage 3.1 이상에만 적용됩니다. 이전 버전의 Red Hat Ceph Storage에서 3.1로 업그레이드하면 이전 파일 이름(`iscsi-gws.yml`)과 이전 섹션 제목(`[iscsi-gws]`)이 계속 사용됩니다.

#### 중요

현재 Red Hat은 컨테이너 기반 배포 시 `ceph-ansible`을 사용하여 설치할 수 있는 다음 옵션을 지원하지 않습니다.

- `gateway_iqn`
- `rbd_devices`
- `client_connections`

이러한 옵션을 수동으로 구성하는 방법은 [컨테이너의 Ceph iSCSI 게이트웨이 구성](#) 섹션을 참조하십시오.

4. 편집할 `iscsigws.yml` 파일을 엽니다.

- 5.

**IPv4 또는 IPv6 주소를 사용하여 iSCSI 게이트웨이 IP 주소를 추가하여 gateway\_ip\_list 옵션을 구성합니다.**

예제

```
gateway_ip_list: 192.168.1.1,192.168.1.2
```



중요

**IPv4 및 IPv6 주소를 혼합하여 사용할 수 없습니다.**

6.

선택적으로 **trusted\_ip\_list** 옵션의 주석을 제거하고 **SSL**을 사용하려면 **IPv4 또는 IPv6** 주소를 적절하게 추가합니다. **SSL**을 구성하려면 **iSCSI 게이트웨이 컨테이너에 대한 루트 액세스 권한이 필요합니다. SSL**을 구성하려면 다음 단계를 수행합니다.

a.

필요한 경우 모든 **iSCSI 게이트웨이 컨테이너** 내에 **openssl** 패키지를 설치합니다.

b.

기본 **iSCSI 게이트웨이 컨테이너**에서 **SSL 키**를 저장할 디렉토리를 만듭니다.

```
# mkdir ~/ssl-keys
# cd ~/ssl-keys
```

c.

기본 **iSCSI 게이트웨이 컨테이너**에서 **인증서 및 키 파일**을 만듭니다.

```
# openssl req -newkey rsa:2048 -nodes -keyout iscsi-gateway.key -x509 -days 365 -out iscsi-gateway.crt
```



참고

환경 정보를 입력하라는 메시지가 표시됩니다.

d.

기본 **iSCSI 게이트웨이 컨테이너**에서 **PEM 파일**을 만듭니다.

```
# cat iscsi-gateway.crt iscsi-gateway.key > iscsi-gateway.pem
```

e.

기본 **iSCSI 게이트웨이 컨테이너**에서 공개 키를 만듭니다.

```
# openssl x509 -inform pem -in iscsi-gateway.pem -pubkey -noout > iscsi-gateway-pub.key
```

f.

기본 **iSCSI 게이트웨이 컨테이너**에서 **iscsi-gateway.crt, iscsi-gateway.pem, iscsi-gateway-pub.key, iscsi-gateway.key** 파일을 다른 **iSCSI 게이트웨이 컨테이너**의 **/etc/ceph/** 디렉터리에 복사합니다.

7.

필요에 따라 다음 **iSCSI 대상 API 서비스 옵션**을 검토하고 주석 처리를 해제합니다.

```
#api_user: admin
#api_password: admin
#api_port: 5000
#api_secure: false
#loop_delay: 1
#trusted_ip_list: 192.168.122.1,192.168.122.2
```

8.

선택적으로 워크로드 요구 사항에 따라 다음 리소스 옵션을 검토하고 주석 처리를 해제합니다.

```
# TCMU_RUNNER resource limitation
#ceph_tcmu_runner_docker_memory_limit: 1g
#ceph_tcmu_runner_docker_cpu_limit: 1

# RBD_TARGET_GW resource limitation
#ceph_rbd_target_gw_docker_memory_limit: 1g
#ceph_rbd_target_gw_docker_cpu_limit: 1

# RBD_TARGET_API resource limitation
#ceph_rbd_target_api_docker_memory_limit: 1g
#ceph_rbd_target_api_docker_cpu_limit: 1
```

9.

**Ansible** 사용자로 **Ansible** 플레이북을 실행합니다.

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit iscsigws
```

**Red Hat Enterprise Linux Atomic**의 경우 **--skip-tags=with\_pkg** 옵션을 추가합니다.

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit iscsigws --skip-tags=with_pkg
```

10.

**Ansible** 플레이북이 완료되면 **TCP 포트 3260** 을 열고 **trusted\_ip\_list** 옵션에 나열된 각 노드의 **iscsigws.yml** 파일에 지정된 **api\_port** 를 엽니다.



참고

**api\_port** 옵션을 지정하지 않으면 기본 포트는 **5000** 입니다.

추가 리소스

- 컨테이너에 **Red Hat Ceph Storage**를 설치하는 방법에 대한 자세한 내용은 컨테이너의 **Red Hat Ceph Storage 클러스터 설치** 섹션을 참조하십시오.
- **Ceph의 iSCSI 게이트웨이** 옵션에 대한 자세한 내용은 **Red Hat Ceph Storage Block Device 가이드의 Table 8.1** 을 참조하십시오.
- **iSCSI 대상 API** 옵션에 대한 자세한 내용은 **Red Hat Ceph Storage Block Device Guide의 Table 8.2** 를 참조하십시오.
- **iscsigws.yml** 파일의 예는 **부록 A Red Hat Ceph Storage Block Device Guide**를 참조하십시오.

### 1.7.1. 컨테이너에서 Ceph iSCSI 게이트웨이 구성

**Ceph iSCSI 게이트웨이** 구성은 **iSCSI 대상, LUN(Logical Unit Number)** 및 **ACL(액세스 제어 목록)** 을 생성하고 관리하기 위한 **gwcli** 명령줄 유틸리티로 수행됩니다.

사전 요구 사항

- 작동 중인 **Red Hat Ceph Storage** 클러스터.
- **iSCSI 게이트웨이 소프트웨어** 설치.

절차

1. **root** 사용자로 **iSCSI** 게이트웨이 명령줄 인터페이스를 시작합니다.

```
# docker exec -it rbd-target-api gwcli
```

2. **IPv4** 또는 **IPv6** 주소를 사용하여 **iSCSI** 게이트웨이를 생성합니다.

구문

```
>/iscsi-target create iqn.2003-01.com.redhat.iscsi-gw:$TARGET_NAME
> goto gateways
> create $ISCSI_GW_NAME $ISCSI_GW_IP_ADDR
> create $ISCSI_GW_NAME $ISCSI_GW_IP_ADDR
```

예제

```
>/iscsi-target create iqn.2003-01.com.redhat.iscsi-gw:ceph-igw
> goto gateways
> create ceph-gw-1 10.172.19.21
> create ceph-gw-2 10.172.19.22
```



중요

**IPv4** 및 **IPv6** 주소를 혼합하여 사용할 수 없습니다.

3. **RADOS** 블록 장치(**RBD**)를 추가합니다.

구문

```
> cd /disks  
>/disks/ create $POOL_NAME image=$IMAGE_NAME size=$IMAGE_SIZE[m|g|t]  
max_data_area_mb=$BUFFER_SIZE
```

### 예제

```
> cd /disks  
>/disks/ create rbd image=disk_1 size=50g max_data_area_mb=32
```



### 중요

풀 이름 또는 이미지 이름에 마침표(.)가 있을 수 없습니다.



주의

Red Hat 지원팀이 이에 지시하지 않는 한 `max_data_area_mb` 옵션을 조정하지 마십시오.

`max_data_area_mb` 옵션은 각 이미지가 **iSCSI** 대상과 **Ceph** 클러스터 간에 **SCSI** 명령 데이터를 전달하는 데 사용할 수 있는 메가바이트 단위로 메모리 양을 제어합니다. 이 값이 너무 작으면 과도한 대기열 전체 재시도 횟수가 발생하여 성능에 영향을 미칠 수 있습니다. 값이 너무 크면 한 디스크가 너무 많은 시스템 메모리를 사용하므로 다른 하위 시스템에 대한 할당 오류가 발생할 수 있습니다. 기본값은 **8**입니다.

이 값을 `reconfigure` 명령을 사용하여 변경할 수 있습니다. 이 명령을 적용하려면 **iSCSI** 이니시에이터에서 이미지를 사용할 수 없습니다.

구문

```
>/disks/ reconfigure max_data_area_mb $NEW_BUFFER_SIZE
```

예제

```
>/disks/ reconfigure max_data_area_mb 64
```

- 4. 클라이언트를 생성합니다.

구문

■

```
> goto hosts  
> create iqn.1994-05.com.redhat:$CLIENT_NAME  
> auth chap=$USER_NAME/$PASSWORD
```

### 예제

```
> goto hosts  
> create iqn.1994-05.com.redhat:rh7-client  
> auth chap=iscsiuser1/temp12345678
```



중요

**CHAP 비활성화는 Red Hat Ceph Storage 3.1 이상에서만 지원됩니다. Red Hat은 클라이언트 혼합, 일부 CHAP 활성화 및 일부 CHAP 비활성화를 지원하지 않습니다. 모든 클라이언트에는 CHAP이 활성화되어 있거나 CHAP이 비활성화되어 있어야 합니다. 기본 동작은 이니시에이터 이름으로만 이니시에이터를 인증하는 것입니다.**

이니시에이터가 대상에 로그인하지 못하면 **CHAP** 인증이 일부 이니시에이터에 대해 잘못 구성되어 있을 수 있습니다.

예제

```
o- hosts ..... [Hosts: 2: Auth: MISCONFIG]
```

모든 **CHAP** 인증을 재설정하려면 호스트 수준에서 다음 명령을 수행합니다.

```
/> goto hosts
/iscsi-target...csi-igw/hosts> auth nochap
ok
ok
/iscsi-target...csi-igw/hosts> ls
o- hosts ..... [Hosts: 2: Auth: None]
o- iqn.2005-03.com.ceph:esx ..... [Auth: None, Disks: 4(310G)]
o- iqn.1994-05.com.redhat:rh7-client .. [Auth: None, Disks: 0(0.00Y)]
```

- 5. 클라이언트에 디스크 추가:

구문

```
>/iscsi-target..eph-igw/hosts> cd iqn.1994-05.com.redhat:$CLIENT_NAME
> disk add $POOL_NAME.$IMAGE_NAME
```

## 예제

```
>/iscsi-target..eph-igw/hosts> cd iqn.1994-05.com.redhat:rh7-client
> disk add rbd.disk_1
```

6.

다음 명령을 실행하여 **iSCSI** 게이트웨이 구성을 확인합니다.

```
> ls
```

7.

선택적으로 **API가 SSL**을 올바르게 사용하고 있는지 확인합니다. 예를 들면 **https**의 `/var/log/rbd-target-api.log` 파일을 찾습니다.

```
Aug 01 17:27:42 test-node.example.com python[1879]: * Running on https://0.0.0.0:5000/
```

8.

다음 단계는 **iSCSI** 이니시에이터를 구성하는 것입니다.

## 추가 리소스

- 컨테이너에 **Red Hat Ceph Storage**를 설치하는 방법에 대한 자세한 내용은 컨테이너의 **Red Hat Ceph Storage 클러스터 설치** 섹션을 참조하십시오.
- 컨테이너에 **iSCSI** 게이트웨이 소프트웨어 설치에 대한 자세한 내용은 컨테이너의 **Ceph iSCSI 게이트웨이 설치** 섹션을 참조하십시오.
- **iSCSI** 이니시에이터 연결 방법에 대한 자세한 내용은 **Red Hat Ceph Storage Block Device Guide**의 **iSCSI Initiator 구성** 섹션을 참조하십시오.

### 1.7.2. 컨테이너에서 Ceph iSCSI 게이트웨이 제거

**Ansible**을 사용하여 **Ceph iSCSI** 게이트웨이 구성을 제거할 수 있습니다.

## 사전 요구 사항

- 작동 중인 **Red Hat Ceph Storage** 클러스터.
- **iSCSI 게이트웨이** 소프트웨어 설치.
- 내보낸 **RBD** 이미지.
- **Red Hat Ceph Storage** 클러스터에 대한 루트 수준 액세스.
- **iSCSI** 이니시에이터에 대한 루트 수준 액세스.
- **Ansible** 관리 노드에 대한 액세스.

## 절차

1.

**iSCSI** 게이트웨이 구성을 제거하기 전에 모든 **iSCSI** 이니시에이터의 연결을 끊습니다. 적절한 운영 체제에 대해 아래 단계를 따르십시오.

a.

**Red Hat Enterprise Linux** 이니시에이터:

**root** 사용자로 다음 명령을 실행합니다.

## 구문

```
iscsiadm -m node -T TARGET_NAME --logout
```

**TARGET\_NAME** 을 구성된 **iSCSI** 대상 이름으로 교체합니다.

## 예제

```
# iscsiadm -m node -T iqn.2003-01.com.redhat.iscsi-gw:ceph-igw --logout
Logging out of session [sid: 1, target: iqn.2003-01.com.redhat.iscsi-gw:iscsi-igw, portal:
10.172.19.21,3260]
Logging out of session [sid: 2, target: iqn.2003-01.com.redhat.iscsi-gw:iscsi-igw, portal:
10.172.19.22,3260]
Logout of [sid: 1, target: iqn.2003-01.com.redhat.iscsi-gw:iscsi-igw, portal:
10.172.19.21,3260] successful.
Logout of [sid: 2, target: iqn.2003-01.com.redhat.iscsi-gw:iscsi-igw, portal:
10.172.19.22,3260] successful.
```

b.

**Windows 이니시에이터:**

자세한 내용은 [Microsoft](#) 문서를 참조하십시오.

c.

**VMware ESXi 이니시에이터:**

자세한 내용은 [VMware 설명서](#) 를 참조하십시오.

2.

**root** 사용자로 **iSCSI 게이트웨이 명령줄 유틸리티**를 실행합니다.

```
# gwcli
```

3.

호스트를 제거합니다.

구문

```
/> cd /iscsi-target/iqn.2003-01.com.redhat.iscsi-gw:_TARGET_NAME_/hosts
/> /iscsi-target...TARGET_NAME/hosts> delete CLIENT_NAME
```

**TARGET\_NAME** 을 구성된 **iSCSI** 대상 이름으로 바꾸고 **CLIENT\_NAME** 을 **iSCSI** 이니시에이터 이름으로 교체합니다.

**예제**

```

/> cd /iscsi-target/iqn.2003-01.com.redhat.iscsi-gw:ceph-igw/hosts
/> /iscsi-target...eph-igw/hosts> delete iqn.1994-05.com.redhat:rh7-client

```

4.

**디스크를 제거합니다.****구문**

```

/> cd /disks/
/disks> delete POOL_NAME.IMAGE_NAME

```

**POOL\_NAME** 을 풀 이름으로 바꾸고 **IMAGE\_NAME** 을 이미지 이름으로 교체합니다.**예제**

```

/> cd /disks/
/disks> delete rbd.disk_1

```

5.

**iSCSI 대상 및 게이트웨이 구성을 제거합니다.**

```

/> cd /iscsi-target/
/iscsi-target> clearconfig confirm=true

```

6.

**Ceph Monitor** 또는 **Client** 노드에서 **root** 사용자로 **iSCSI 게이트웨이 구성 개체 (gateway.conf)**를 제거합니다.

■

```
[root@mon ~]# rados rm -p pool gateway.conf
```

7.

선택적으로 내보낸 **Ceph RADOS Block Device(RBD)**가 더 이상 필요하지 않은 경우 **RBD** 이미지를 제거합니다. **Ceph Monitor** 또는 **Client** 노드에서 **root** 사용자로 다음 명령을 실행합니다.

구문

```
rbid rm IMAGE_NAME
```

예제

```
[root@mon ~]# rbd rm rbd01
```

추가 리소스

- 컨테이너에 **Red Hat Ceph Storage**를 설치하는 방법에 대한 자세한 내용은 컨테이너의 **Red Hat Ceph Storage 클러스터 설치** 섹션을 참조하십시오.
- 컨테이너에 **iSCSI 게이트웨이 소프트웨어** 설치에 대한 자세한 내용은 컨테이너의 **Ceph iSCSI 게이트웨이 설치** 섹션을 참조하십시오.

### 1.7.3. iSCSI 대상의 성능 최적화

**iSCSI Target**이 네트워크를 통해 데이터를 전송하는 방법을 제어하는 많은 설정이 있습니다. 이러한 설정을 사용하여 **iSCSI 게이트웨이**의 성능을 최적화할 수 있습니다.



주의

**Red Hat 지원 또는 이 문서에 명시된 대로 해당 설정을 변경합니다.**

### **gwcli reconfigure** 하위 명령

**gwcli reconfigure** 하위 명령은 **iSCSI** 게이트웨이의 성능을 최적화하는 데 사용되는 설정을 제어합니다.

### **iSCSI** 대상의 성능에 영향을 주는 설정

- **max\_data\_area\_mb**
- **cmds\_n\_depth**
- **immediate\_data**
- **initial\_r2t**
- **max\_outstanding\_r2t**
- **first\_burst\_length**
- **max\_burst\_length**
- **max\_recv\_data\_segment\_length**
- **max\_xmit\_data\_segment\_length**

추가 리소스

- **gwcli** 재구성 방법을 보여주는 예제를 포함하여 **max\_data\_area\_mb**에 대한 정보는 블록 장치 가이드의 **명령줄 인터페이스를 사용하여 iSCSI 대상 구성** 섹션에 있으며 컨테이너 가이드의 컨테이너에서 **Ceph iSCSI 게이트웨이 구성** 섹션에 있습니다.

## 1.8. 제한 옵션 이해

이 섹션에는 **Ansible --limit** 옵션에 대한 정보가 포함되어 있습니다.

**Ansible**은 인벤토리 파일의 특정 섹션에 사이트 및 **site-docker Ansible** 플레이북을 사용할 수 있는 **-limit** 옵션을 지원합니다.

```
$ ansible-playbook site.yml|site-docker.yml --limit osds|rgws|clients|mdss|nfss|iscsigws
```

예를 들어 컨테이너에서 **OSD**만 재배포하려면 **Ansible** 사용자로 다음 명령을 실행합니다.

```
$ ansible-playbook /usr/share/ceph-ansible/site-docker.yml --limit osds
```

## 1.9. 추가 리소스

- **Red Hat Enterprise Linux Atomic Host**용 **컨테이너 시작하기 가이드**

## 2장. 컨테이너화된 CEPH 데몬 공동 배치

이 섹션에서는 다음을 설명합니다.

- [공동 배치의 작동 방식 및 이점](#)
- [colocated 데몬을 위한 전용 리소스 설정 방법](#)

### 2.1. 공동 배치의 작동 방식 및 이점

컨테이너화된 Ceph 데몬을 동일한 노드에서 배치할 수 있습니다. 다음은 일부 Ceph 서비스를 조정하는 이점이 있습니다.

- 소규모로 ownership 비용 (ownership cost of ownership)이 크게 향상되었습니다.
- 최소 구성을 위해 6개의 노드에서 3으로 감소
- 간편한 업그레이드
- 리소스 격리 개선

### Colocation Works

Ansible 인벤토리 파일의 해당 섹션에 동일한 노드를 추가하여 다음 목록에서 하나의 데몬을 OSD 데몬과 함께 배치할 수 있습니다.

- **Ceph Object Gateway(radosgw)**
- **메타데이터 서버(MDS)**
- **RBD 미러(rbd-mirror)**

- 모니터링 및 **Ceph Manager 데몬(ceph-mgr)**
- **NFS Ganesha**

다음 예제에서는 **colocated** 데몬이 있는 인벤토리 파일이 다음과 같은 방법을 보여줍니다.

**예 2.1. 배치된 데몬이 포함된 Ansible 인벤토리 파일**

```
[mons]
<hostname1>
<hostname2>
<hostname3>

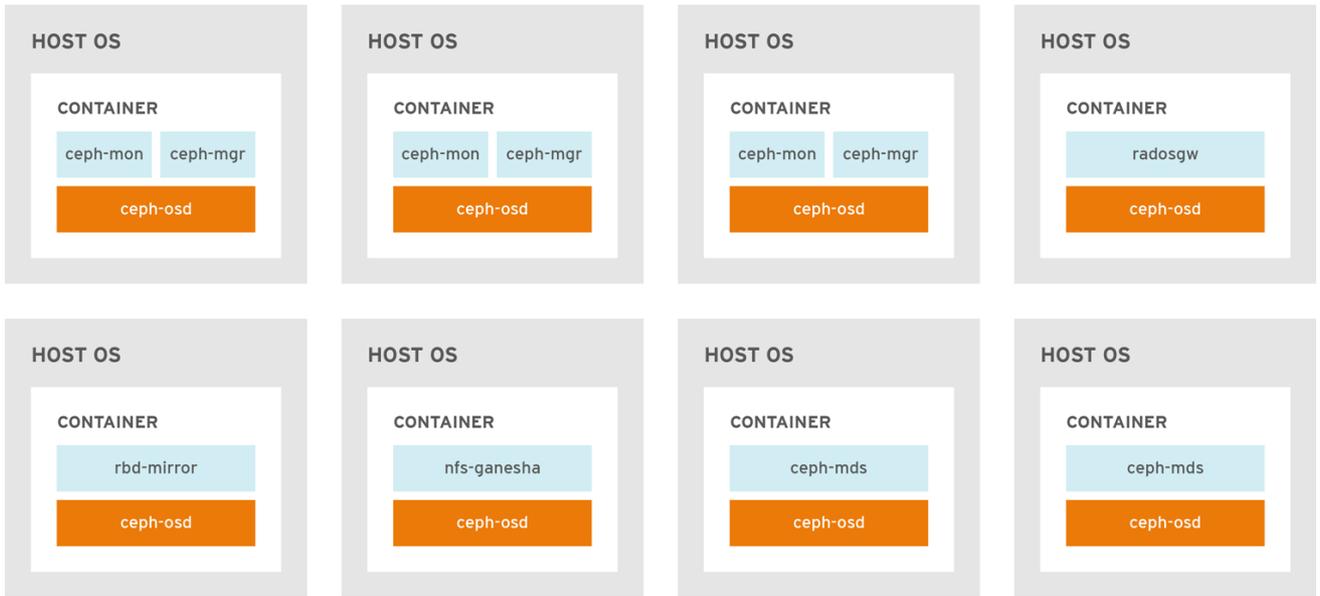
[mgrs]
<hostname1>
<hostname2>
<hostname3>

[osds]
<hostname4>
<hostname5>
<hostname6>

[rgws]
<hostname4>
<hostname5>
```

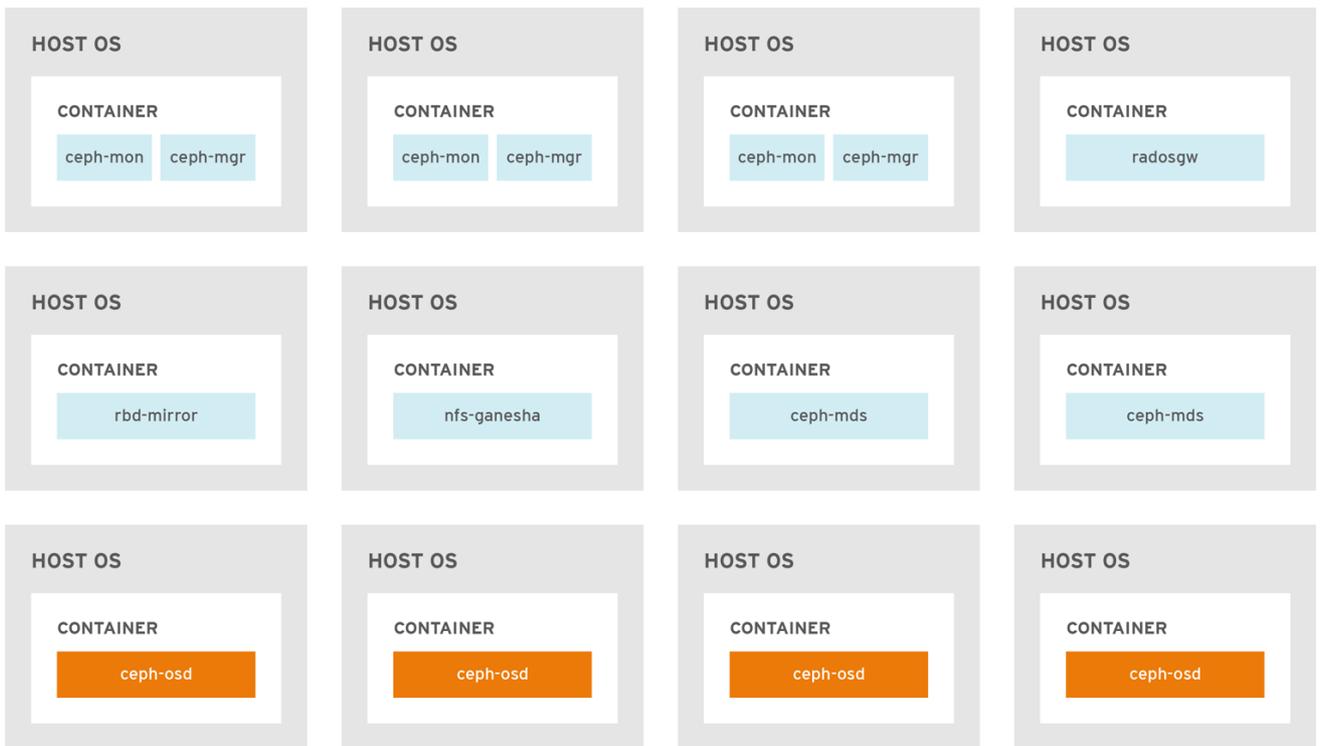
**그림 2.1. “배치된 데몬”** 및 **그림 2.2. “인식되지 않는 데몬”** 이미지는 공동 배치된 데몬과 인식되지 않는 클러스터의 차이점을 보여줍니다.

그림 2.1. 배치된 데몬



CEPH\_459072\_1017

그림 2.2. 인식되지 않는 데몬



CEPH\_459072\_1017

동일한 노드에서 컨테이너화된 두 개의 Ceph 데몬을 배치하면 **ceph-ansible** 플레이북은 각각 전용 CPU 및 RAM 리소스를 예약합니다. 기본적으로 **ceph-ansible** 은 **Red Hat Ceph Storage Hardware Selection Guide 3**의 권장 최소 하드웨어 장에 나열된 값을 사용합니다. 기본값을 변경하는 방법을 알아보려면 **협업 데몬에 대한 전용 리소스 설정** 섹션을 참조하십시오.

2.2. 협업된 데몬의 DEDICATED 리소스 설정

동일한 노드에서 두 개의 Ceph 데몬을 공동 배치하면 **ceph-ansible** 플레이북은 각 데몬에 대해 CPU 및 RAM 리소스를 예약합니다. **ceph-ansible** 에서 사용하는 기본값은 **Red Hat Ceph Storage Hardware Selection 가이드의 권장 최소** 하드웨어 장에 나열되어 있습니다. 기본값을 변경하려면 Ceph 데몬을 배포할 때 필요한 매개변수를 설정합니다.

## 절차

1.

데몬의 기본 CPU 제한을 변경하려면 데몬을 배포할 때 적절한 **.yml** 구성 파일에서 **ceph\_daemon-type\_docker\_cpu\_limit** 매개변수를 설정합니다. 자세한 내용은 다음 표를 참조하십시오.

데몬	매개변수	설정 파일
OSD	<b>ceph_osd_docker_cpu_limit</b>	<b>osds.yml</b>
MDS	<b>ceph_mds_docker_cpu_limit</b>	<b>mdss.yml</b>
RGW	<b>ceph_rgw_docker_cpu_limit</b>	<b>rgws.yml</b>

예를 들어 **Ceph Object Gateway**의 기본 CPU 제한을 2로 변경하려면 다음과 같이 **/usr/share/ceph-ansible/group\_vars/rgws.yml** 파일을 편집합니다.

```
ceph_rgw_docker_cpu_limit: 2
```

2.

**OSD** 데몬의 기본 RAM을 변경하려면 데몬을 배포할 때 **/usr/share/ceph-ansible/group\_vars/all.yml** 파일에 **osd\_memory\_target** 을 설정합니다. 예를 들어 **OSD RAM** 을 **6GB**로 제한하려면 다음을 수행합니다.

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



중요

**osd\_memory\_target** 매개변수를 사용하여 하이퍼컨버지드 인프라(HCI) 구성에서 OSD 메모리를 제한하는 것이 좋습니다.  
**ceph\_osd\_docker\_memory\_limit** 매개변수가 필요하지 않지만 사용하려면 **ceph\_osd\_docker\_memory\_limit** 를 **osd\_memory\_target** 보다 50% 더 높으므로 CGroup 제한이 HCI 구성에 기본적으로 있는 것보다 더 많은 **constraining**이 됩니다. 예를 들어 **osd\_memory\_target** 이 6GB로 설정된 경우 **ceph\_osd\_docker\_memory\_limit** 를 9GB로 설정합니다.

```
ceph_osd_docker_memory_limit: 9g
```



주의

**ceph\_osd\_docker\_memory\_limit** 매개변수는 하드 제한을 설정합니다. 값이 초과되면 OSD가 실행 중지될 수 있습니다. **osd\_memory\_target** 매개변수는 소프트웨어 제한을 설정하므로 값이 초과되면 컨테이너가 실행을 중지하고 서비스가 중단되지 않습니다.

추가 리소스

- [/usr/share/ceph-ansible/group\\_vars/ 디렉터리의 샘플 구성 파일](#)

2.3. 추가 리소스

- [컨테이너에 Red Hat Ceph Storage 배포](#)
- [Red Hat Ceph Storage Hardware Selection Guide](#)

### 3장. 컨테이너에서 실행되는 CEPH 클러스터 관리

이 장에서는 다음과 같이 컨테이너에서 실행되는 Ceph 클러스터에서 수행해야 하는 기본 관리 작업을 설명합니다.

- **3.1절. “컨테이너에서 실행되는 Ceph 데몬 시작, 중지 및 재시작”**
- **3.2절. “컨테이너에서 실행되는 Ceph 데몬의 로그 파일 보기”**
- **3.6절. “Ansible로 배포된 클러스터 제거”**
- **4.2절. “컨테이너에서 실행되는 Red Hat Ceph Storage 클러스터 업그레이드”**

#### 3.1. 컨테이너에서 실행되는 CEPH 데몬 시작, 중지 및 재시작

컨테이너에서 실행되는 Ceph 데몬을 시작, 중지 또는 다시 시작합니다.

##### 절차

1. 컨테이너에서 실행되는 Ceph 데몬을 시작, 중지 또는 다시 시작하려면 다음 형식으로 구성된 root 로 `systemctl` 명령을 실행합니다.

```
systemctl action ceph-daemon@ID
```

다음과 같습니다.

- 작업은 수행할 작업입니다. 시작, 중지 또는 다시 시작
- `daemon` 은 데몬이며 `osd,mon,mds` 또는 `rgw`
- ID 중 하나

○

`ceph-mon`, `ceph-mds` 또는 `ceph-rgw` 데몬이 실행되는 경우 `ceph` 클러스터 이름입니다.

**ceph-mon,ceph-mus** 또는 **ceph-rgw** 데몬이 실행 중인 짧은 오스드 이름입니다.

- **osd\_scenario** 매개변수를 **lvm**으로 설정한 경우 **ceph-osd** 데몬의 ID
- **ceph-osd** 데몬이 **collocated** 또는 **non-collocated**로 설정된 **osd\_scenario** 매개변수를 사용하여 배포된 경우 사용하는 장치 이름입니다.

예를 들어 ID가 **osd01** 인 **ceph-osd** 데몬을 다시 시작하려면 다음을 실행합니다.

```
# systemctl restart ceph-osd@osd01
```

**ceph-monitor01** 호스트에서 실행되는 **ceph-mon demon**을 시작하려면 다음을 실행합니다.

```
# systemctl start ceph-mon@ceph-monitor01
```

**ceph-rgw 01** 호스트에서 실행되는 **ceph-rgw** 데몬을 중지하려면 다음을 수행합니다.

```
# systemctl stop ceph-radosgw@ceph-rgw01
```

2. 작업이 성공적으로 완료되었는지 확인합니다.

```
systemctl status ceph-daemon@_ID
```

예를 들면 다음과 같습니다.

```
# systemctl status ceph-mon@ceph-monitor01
```

추가 리소스

- **Red Hat Ceph Storage 3 관리 가이드**의 **systemd** 서비스로 실행 중인 **Ceph** 섹션.

### 3.2. 컨테이너에서 실행되는 CEPH 데몬의 로그 파일 보기

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

## 절차

1.

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

```
journalctl -u ceph-daemon@ID
```

다음과 같습니다.

- **daemon** 은 **Ceph** 데몬이며 **osd,mon** 또는 **rgw**
- **ID** 중 하나
  - **ceph-mon,ceph-mds** 또는 **ceph-rgw** 데몬이 실행중인 짧은 호스트 이름입니다.
  - **osd\_scenario** 매개변수를 **lvm**으로 설정한 경우 **ceph-osd** 데몬의 **ID**
  - **ceph-osd** 데몬이 **collocated** 또는 **non-collocated**로 설정된 **osd\_scenario** 매개변수를 사용하여 배포된 경우 사용하는 장치 이름입니다.

예를 들어, **osd01 ID**를 사용하여 **ceph-osd** 데몬의 전체 로그를 보려면 다음을 수행합니다.

```
# journalctl -u ceph-osd@osd01
```

2.

최근 저널 항목만 표시하려면 **-f** 옵션을 사용합니다.

```
journalctl -fu ceph-daemon@ID
```

예를 들어 **ceph-monitor01** 호스트에서 실행되는 **ceph-mon** 데몬의 최근 저널 항목만 보려면 다음을 수행합니다.

```
# journalctl -fu ceph-mon@ceph-monitor01
```



## 참고

**sosreport** 유틸리티를 사용하여 **journald** 로그를 볼 수도 있습니다. **SOS 보고서에 대한 자세한 내용은 [What is a sosreport and how to create one in Red Hat Enterprise Linux 4.6 이상? on the Red Hat Customer Portal](#)을 참조하십시오.**

## 추가 리소스

- [journalctl\(1\) 도움말 페이지](#)

### 3.3. 명령줄 인터페이스를 사용하여 CEPH OSD 추가

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

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



## 중요

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

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



## 참고

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

## 사전 요구 사항

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

## 절차

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

**Red Hat Enterprise Linux**

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

2. **/etc/ceph/** 디렉토리를 만듭니다.
3. 새 **OSD** 노드에서 **Ceph Monitor** 노드 중 하나에서 **Ceph** 관리 인증 키링 및 구성 파일을 복사합니다.
4. 새 **Ceph OSD** 노드에 **ceph-osd** 패키지를 설치합니다.

**Red Hat Enterprise Linux**

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

5.

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



참고

**--filestore** 옵션이 필요합니다.

a.

배치된 저널이 있는 **OSD**의 경우:

구문

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

예제:

```
[root@osd ~]# docker exec ceph-osd-osd1 ceph-disk --setuser ceph --setgroup ceph
prepare --filestore /dev/sda
```

b.

전용 저널이 있는 **OSD**의 경우:

구문

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

또는

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

예

```
[root@osd ~]# docker exec ceph-osd-osd1 ceph-disk --setuser ceph --setgroup ceph
prepare --filestore /dev/sda /dev/sdb
```

```
[root@osd ~]# docker exec ceph-osd-osd1 ceph-volume lvm prepare --filestore --data
/dev/vg00/lvol1 --journal /dev/sdb
```

6.

**noup** 옵션을 설정합니다.

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

7.

새 **OSD**를 활성화합니다.

구문

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

또는

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

## 예제

```
[root@osd ~]# docker exec ceph-osd-osd1 ceph-disk activate /dev/sda
```

```
[root@osd ~]# docker exec ceph-osd-osd1 ceph-volume lvm activate --filestore 0 6cc43680-4f6e-4feb-92ff-9c7ba204120e
```

8.

**CRUSH 맵에 OSD를 추가합니다.**

## 구문

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

## 예제

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



## 참고

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



## 참고

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



### 중요

루트 버킷만 지정하면 **OSD가 root에 직접 연결되지만 CRUSH 규칙에는 호스트 버킷 내부에 OSD가 있어야 합니다.**

9. **noup** 옵션을 설정 해제합니다.

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

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

### 구문

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

### 예제

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

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

### Red Hat Enterprise Linux

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

**\$CLUSTER\_NAME** 을 사용자 지정 클러스터 이름으로 교체합니다.

12.

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

구문

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

예제

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

### 3.4. 명령줄 인터페이스를 사용하여 CEPH OSD 제거

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

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 스토리지 클러스터가 거의 가득 차지 않도록 사용 가능한 **OSD**가 충분합니다.

- **OSD 노드에 root 액세스 권한이 있어야 합니다.**

## 절차

1. **OSD 서비스를 비활성화하고 중지합니다.**

### 구문

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

### 예제

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

**OSD가 중지되면 종료됩니다.**

2. **스토리지 클러스터에서 OSD를 제거합니다.**

### 구문

```
ceph osd out $DEVICE_NAME
```

### 예제

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

### 중요

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

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

3.

**CRUSH 맵에서 OSD를 제거하여 더 이상 데이터를 받지 않도록 합니다.**

### 구문

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

### 예제

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

### 참고

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

4. **OSD 인증 키를 제거합니다.**

구문

```
ceph auth del osd.$DEVICE_NAME
```

예제

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

5. **OSD를 제거합니다.**

구문

```
ceph osd rm $DEVICE_NAME
```

예제

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

6. 스토리지 클러스터의 구성 파일(기본적으로 `/etc/ceph.conf`)을 편집하고 **OSD 항목이 있는 경우 제거합니다.**

**예제**

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

7.

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

8.

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

**구문**

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

**예제**

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

**3.5. OSD ID를 유지하면서 OSD 드라이브 교체**

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



## 참고

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

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

## 사전 요구 사항

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

## 절차

1. **OSD**를 삭제합니다.

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

### 예제

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

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

```
docker exec $CONTAINER_ID ceph-volume lvm zap $DEVICE
```

### 예제

```
$ docker exec ceph-osd-osd1 ceph-volume lvm zap /dev/sdb
```

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

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

예제

```
$ docker exec ceph-osd-osd1 ceph-volume lvm create --osd-id 1 --data /dev/sdb
```

### 3.6. ANSIBLE로 배포된 클러스터 제거

Ceph 클러스터를 더 이상 사용하지 않으려면 **purge-docker-cluster.yml** 플레이북을 사용하여 클러스터를 삭제합니다. 클러스터 제거는 설치 프로세스가 실패하고 처음부터 다시 시작하려는 경우에도 유용합니다.



주의

**Ceph 클러스터를 종료하면 OSD의 모든 데이터가 손실됩니다.**

#### 사전 요구 사항

- **/var/log/ansible.log** 파일에 쓸 수 있는지 확인합니다.

#### 절차

**Ansible** 관리 노드의 다음 명령을 사용합니다.

1. **root** 사용자로 **/usr/share/ceph-ansible/** 디렉터리로 이동합니다.

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

2.

`/usr/share/infrastructure-playbooks/` 디렉터리의 `purge-docker-cluster.yml` 플레이북을 현재 디렉터리에 복사합니다.

```
[root@admin ceph-ansible]# cp infrastructure-playbooks/purge-docker-cluster.yml .
```

3.

**Ansible** 사용자로 `remove-docker-cluster.yml` 플레이북을 사용하여 **Ceph** 클러스터를 제거합니다.

a.

모든 패키지, 컨테이너, 구성 파일 및 **ceph-ansible** 플레이북에서 생성한 모든 데이터를 제거하려면 다음을 수행합니다.

```
[user@admin ceph-ansible]$ ansible-playbook purge-docker-cluster.yml
```

b.

기본 인벤토리 파일(`/etc/ansible/hosts`)과 다른 인벤토리 파일을 지정하려면 `-i` 매개변수를 사용합니다.

```
ansible-playbook purge-docker-cluster.yml -i inventory-file
```

**inventory-file** 을 인벤토리 파일의 경로로 바꿉니다.

예를 들면 다음과 같습니다.

```
[user@admin ceph-ansible]$ ansible-playbook purge-docker-cluster.yml -i ~/ansible/hosts
```

c.

**Ceph** 컨테이너 이미지 제거를 건너뛰려면 `--skip-tags="remove_img"` 옵션을 사용합니다.

```
[user@admin ceph-ansible]$ ansible-playbook --skip-tags="remove_img" purge-docker-cluster.yml
```

d.

설치 중에 설치된 패키지 제거를 건너뛰려면 `--skip-tags="with_pkg"` 옵션을 사용합니다.

```
[user@admin ceph-ansible]$ ansible-playbook --skip-tags="with_pkg" purge-docker-cluster.yml
```

## 4장. 컨테이너 내에서 RED HAT CEPH STORAGE 업그레이드

**Ansible** 애플리케이션은 컨테이너 내에서 실행되는 **Red Hat Ceph Storage** 업그레이드를 사전에 구성합니다.

### 4.1. 사전 요구 사항

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

### 4.2. 컨테이너에서 실행되는 RED HAT CEPH STORAGE 클러스터 업그레이드

이 섹션에서는 **Red Hat Ceph Storage** 컨테이너 이미지의 최신 마이너 버전 또는 주요 버전으로 업그레이드하는 방법을 설명합니다.

- 스토리지 클러스터를 업그레이드하려면 [4.3절. “스토리지 클러스터 업그레이드”](#) 을 참조하십시오.
- **Red Hat Ceph Storage** 대시보드를 업그레이드하려면 [4.4절. “Red Hat Ceph Storage 대시보드 업그레이드”](#) 를 참조하십시오.

관리 노드의 `/usr/share/ceph-ansible/infrastructure-playbooks/` 디렉터리에 있는 **Ansible rolling\_update.yml** 플레이북을 사용하여 **Red Hat Ceph Storage**의 두 가지 주요 버전 또는 마이너 버전 간에 업그레이드하거나 비동기 업데이트를 적용합니다.

**Ansible**은 다음과 같은 순서로 **Ceph** 노드를 업그레이드합니다.

- 노드 모니터링
- MGR 노드
- OSD 노드
- MDS 노드

- **Ceph Object Gateway 노드**

- 기타 모든 **Ceph** 클라이언트 노드



### 참고

**Red Hat Ceph Storage 3**에서는 `/usr/share/ceph-ansible/group_vars/` 디렉터리에 있는 **Ansible** 구성 파일에 몇 가지 변경 사항이 추가되었습니다. 특정 매개변수의 이름이 변경되거나 제거되었습니다. 따라서 버전 **3**으로 업그레이드한 후 `all.yml.sample` 및 `osds.yml.sample` 파일에서 새 복사본을 생성하기 전에 `all.yml` 및 `osds.yml.yml` 파일의 백업 사본을 만듭니다. 변경 사항에 대한 자세한 내용은 **부록 A. 버전 2와 3의 Ansible 변수 변경** 을 참조하십시오.



### 참고

**Red Hat Ceph Storage 3.1** 이상에서는 **Object Gateway** 및 고속 **NVMe** 기반 **SSD**(및 **SATA SSD**)를 사용할 때 성능을 위해 스토리지를 최적화하는 새로운 **Ansible** 플레이북을 도입합니다. 플레이북은 **SSD**에 저널 및 버킷 인덱스를 함께 배치하여 이를 수행하므로 하나의 장치에 모든 저널이 있는 것보다 성능이 향상될 수 있습니다. 이러한 플레이북은 **Ceph**를 설치할 때 사용하도록 설계되었습니다. 기존 **OSD**는 계속 작동하고 업그레이드하는 동안 추가 단계가 필요하지 않습니다. 이러한 방식으로 스토리지를 최적화하도록 **OSD**를 동시에 재구성하는 동안 **Ceph** 클러스터를 업그레이드할 수 없습니다. 저널 또는 버킷 인덱스에 다른 장치를 사용하려면 **OSD**를 재프로비저닝해야 합니다. 자세한 내용은 프로덕션을 위해 **Ceph Object Gateway** 에서 **LVM**으로 **NVMe**를 최적으로 사용하여 을 참조하십시오.



### 중요

`rolling_update.yml` 플레이북에는 동시에 업데이트할 노드 수를 조정하는 `serial` 변수가 포함되어 있습니다. **Red Hat**은 기본값(1)을 사용하는 것이 좋습니다. 그러면 **Ansible**에서 클러스터 노드를 하나씩 업그레이드합니다.

**중요**

`rolling_update.yml` 플레이북을 사용하여 Red Hat Ceph Storage 3.x 버전으로 업그레이드하는 경우 Ceph File System(CephFS)을 사용하는 사용자는 Metadata Server(MDS) 클러스터를 수동으로 업데이트해야 합니다. 이는 알려진 문제로 인해 발생합니다.

`ceph-ansible rolling_update.yml` 을 사용하여 전체 클러스터를 업그레이드하기 전에 `/etc/ansible/hosts` 에서 MDS 호스트를 주석 처리한 다음 MDS를 수동으로 업그레이드합니다. `/etc/ansible/hosts` 파일에서 다음을 수행합니다.

```
#[mdss]
#host-abc
```

MDS 클러스터 업데이트 방법을 비롯한 알려진 문제에 대한 자세한 내용은 Red Hat Ceph Storage 3.0 릴리스 노트를 참조하십시오.

**중요**

Red Hat Ceph Storage 클러스터를 이전 버전에서 3.2로 업그레이드할 때 Ceph Ansible 구성은 기본적으로 오브젝트 저장소 유형을 BlueStore로 설정합니다. 여전히 FileStore를 OSD 오브젝트 저장소로 사용하려는 경우 Ceph Ansible 구성을 FileStore로 명시적으로 설정합니다. 이렇게 하면 새로 배포되고 교체된 OSD가 FileStore를 사용합니다.

**중요**

`rolling_update.yml` 플레이북을 사용하여 Red Hat Ceph Storage 3.x 버전으로 업그레이드할 때 다중 사이트 Ceph Object Gateway 구성을 사용하는 경우 다중 사이트 구성을 지정하기 위해 `all.yml` 파일을 수동으로 업데이트할 필요가 없습니다.

**사전 요구 사항**

- 스토리지 클러스터의 모든 노드에 root 사용자로 로그인합니다.
- 스토리지 클러스터의 모든 노드에서 `rhel-7-server-extras-rpms` 리포지토리를 활성화합니다.

```
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

- **Ansible** 관리 노드 및 **RBD** 미러링 노드에서 **Red Hat Ceph Storage 2.x**에서 **3.x**로 업그레이드하는 경우 **Red Hat Ceph Storage 3** 툴 리포지토리를 활성화합니다.

```
# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

- **Ansible** 관리 노드에서 **Ansible** 리포지토리를 활성화합니다.

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-ansible-2.6-rpms
```

- **Ansible** 관리 노드에서 최신 버전의 **ansible** 및 **ceph-ansible** 패키지가 설치되어 있는지 확인합니다.

```
[root@admin ~]# yum update ansible ceph-ansible
```

### 4.3. 스토리지 클러스터 업그레이드

#### 절차

**Ansible** 관리 노드의 다음 명령을 사용합니다.

1. **root** 사용자로 **/usr/share/ceph-ansible/** 디렉터리로 이동합니다.

```
[root@admin ~]# cd /usr/share/ceph-ansible/
```

2. **Red Hat Ceph Storage** 버전 **3.x**에서 최신 버전으로 업그레이드할 때 이 단계를 건너뛰니다. **group\_vars/all.yml** 및 **group\_vars/osds.yml** 파일을 백업합니다.

```
[root@admin ceph-ansible]# cp group_vars/all.yml group_vars/all_old.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml group_vars/osds_old.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml group_vars/clients_old.yml
```

3. **Red Hat Ceph Storage** 버전 **3.x**에서 최신 버전으로 업그레이드할 때 이 단계를 건너뛰니다. **Red Hat Ceph Storage 2.x**에서 **3.x**로 업그레이드할 때 각각 **group\_vars/all.yml.sample**, **group\_vars/osds.yml.sample** 및 **group\_vars/clients.yml.sample** 파일의 새 사본을 생성하고 **group\_vars/all.yml**, **group\_vars/osds.yml**, **group\_vars/clients.yml** 파일의 새 사본을 각각 생성합니다. 를 열고 그에 따라 편집합니다. 자세한 내용은 **부록 A. 버전 2와 3의 Ansible 변수 변경 및 1.2절. “컨테이너에 Red Hat Ceph Storage 클러스터 설치”** 를 참조하십시오.

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4.

**Red Hat Ceph Storage 버전 3.x에서 최신 버전으로 업그레이드할 때 이 단계를 건너뛰니다. Red Hat Ceph Storage 2.x에서 3.x로 업그레이드하는 경우 group\_vars/clients.yml 파일을 열고 다음 줄의 주석을 제거합니다.**

```
keys:
- { name: client.test, caps: { mon: "allow r", osd: "allow class-read object_prefix
  rbd_children, allow rwx pool=test" }, mode: "{{ ceph_keyring_permissions }}" }
```

a.

**client.test** 를 실제 클라이언트 이름으로 바꾸고 클라이언트 정의 행에 클라이언트 키를 추가합니다. 예를 들면 다음과 같습니다.

```
key: "ADD-KEYRING-HERE=="
```

이제 전체 예제는 다음과 유사합니다.

```
- { name: client.test, key: "AQAIN8tUMICVFBAALRHNRV0Z4MXupRw4v9JQ6Q==", caps:
  { mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" },
  mode: "{{ ceph_keyring_permissions }}" }
```



참고

클라이언트 키를 가져오려면 **ceph auth get-or-create** 명령을 실행하여 **named** 클라이언트의 키를 확인합니다.

5.

**2.x에서 3.x로 업그레이드할 때 group\_vars/all.yml 파일에서 ceph\_docker\_image 매개 변수가 Ceph 3 컨테이너 버전을 가리키도록 변경합니다.**

```
ceph_docker_image: rhceph/rhceph-3-rhel7
```

6.

**fetch\_directory** 매개변수를 **group\_vars/all.yml** 파일에 추가합니다.

```
fetch_directory: <full_directory_path>
```

교체:

- **Ansible** 사용자의 홈 디렉터리와 같은 쓰기 가능한 위치가 있는 `<full_directory_path>` 초기 스토리지 클러스터 설치에 사용된 기존 경로를 제공합니다.

기존 경로가 손실되거나 누락된 경우 다음을 먼저 수행합니다.

- a. 기존 `group_vars/all.yml` 파일에 다음 옵션을 추가합니다.

```
fsid: <add_the_fsid>
generate_fsid: false
```

- b. `take-over-existing-cluster.yml` Ansible 플레이북을 실행합니다.

```
[user@admin ceph-ansible]$ cp infrastructure-playbooks/take-over-existing-cluster.yml .
[user@admin ceph-ansible]$ ansible-playbook take-over-existing-cluster.yml
```

7. 업그레이드하려는 클러스터에 **Ceph Object Gateway** 노드가 포함된 경우 `radosgw_interface` 매개변수를 `group_vars/all.yml` 파일에 추가합니다.

```
radosgw_interface: <interface>
```

교체:

- **Ceph Object Gateway** 노드가 수신 대기하는 인터페이스가 `<interface>` 입니다.

8. **Red Hat Ceph Storage 3.2**부터 기본 OSD 오브젝트 저장소는 **BlueStore**입니다. 기존의 OSD 오브젝트 저장소를 유지하려면 `osd_objectstore` 옵션을 `group_vars/all.yml` 파일의 `filestore` 로 명시적으로 설정해야 합니다.

```
osd_objectstore: filestore
```



참고

`osd_objectstore` 옵션이 `filestore` 로 설정된 경우 OSD를 교체하면 **BlueStore** 대신 **FileStore**를 사용합니다.

- 9.

`/etc/ansible/hosts` 에 있는 **Ansible** 인벤토리 파일에서 `[mgrs]` 섹션에 **Ceph Manager(ceph-mgr)** 노드를 추가합니다. **Ceph Manager** 데몬을 모니터 노드와 분리합니다. 버전 **3.x**에서 최신 버전으로 업그레이드할 때 이 단계를 건너뛸 수 있습니다.

```
[mgrs]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

10.

`infrastructure-playbooks` 디렉터리의 `rolling_update.yml` 을 현재 디렉터리로 복사합니다.

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

11.

`/var/log/ansible/` 디렉터를 생성하고 **ansible** 사용자에게 적절한 권한을 할당합니다.

```
[root@admin ceph-ansible]# mkdir /var/log/ansible
[root@admin ceph-ansible]# chown ansible:ansible /var/log/ansible
[root@admin ceph-ansible]# chmod 755 /var/log/ansible
```

a.

`/usr/share/ceph-ansible/ansible.cfg` 파일을 편집하여 다음과 같이 `log_path` 값을 업데이트합니다.

```
log_path = /var/log/ansible/ansible.log
```

12.

**Ansible** 사용자로 **Playbook**을 실행합니다.

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

**Ansible** 인벤토리 파일의 특정 노드 그룹에만 플레이북을 사용하려면 `--limit` 옵션을 사용합니다. 자세한 내용은 [1.8절. “제한 옵션 이해”](#)의 내용을 참조하십시오.

13.

**RBD** 미러링 데몬 노드에 **root** 사용자로 로그인한 동안 **rbd-mirror** 를 수동으로 업그레이드합니다.

```
# yum upgrade rbd-mirror
```

데몬을 다시 시작하십시오.

```
# systemctl restart ceph-rbd-mirror@<client-id>
```

14.

클러스터 상태가 **OK**인지 확인합니다.

a.

**root** 사용자로 모니터 노드에 로그인하고 실행 중인 모든 컨테이너를 나열합니다.

```
[root@monitor ~]# docker ps
```

b.

클러스터 상태가 **OK**인지 확인합니다.

```
[root@monitor ~]# docker exec ceph-mon-<mon-id> ceph -s
```

교체:

•

첫 번째 단계에서 찾은 **Monitor** 컨테이너의 이름이 있는 **<mon-id>** 입니다.

예를 들면 다음과 같습니다.

```
[root@monitor ~]# docker exec ceph-mon-monitor ceph -s
```

15.

**OpenStack** 환경에서 작업하는 경우 풀에 **RBD** 프로필을 사용하도록 모든 **KnativeServing** 사용자를 업데이트합니다. 다음 명령을 **root** 사용자로 실행해야 합니다.

•

**Glance** 사용자

```
ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd pool=<glance-pool-name>'
```

예제

```
[root@monitor ~]# ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd pool=images'
```

- 

### Cinder 사용자

```
ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd pool=<cinder-volume-pool-name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=<glance-pool-name>'
```

예제

```
[root@monitor ~]# ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```

- 

### OpenStack 일반 사용자

```
ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-only pool=<cinder-volume-pool-name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=<glance-pool-name>'
```

예제

```
[root@monitor ~]# ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-only pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```



#### 중요

라이브 클라이언트 마이그레이션을 수행하기 전에 이러한 **CAPS** 업데이트를 수행합니다. 이를 통해 클라이언트는 메모리에서 실행되는 새 라이브러리를 사용하므로 이전 **CAPS** 설정이 캐시에서 삭제되고 새 **RBD** 프로필 설정을 적용할 수 있습니다.

## 4.4. RED HAT CEPH STORAGE 대시보드 업그레이드

다음 절차에서는 **Red Hat Ceph Storage** 대시보드를 버전 **3.1**에서 **3.2**로 업그레이드하는 단계를 간략하게 설명합니다.

업그레이드하기 전에 **Red Hat Ceph Storage**가 3.1에서 3.2로 업그레이드되었는지 확인합니다. 4.1을 참조하십시오. 자세한 내용은 **Storage Cluster** 를 업그레이드합니다.



#### 주의

업그레이드 절차에서는 이전 스토리지 대시보드 데이터를 제거합니다.

#### 절차

1. **root** 사용자로 **Ansible** 관리 노드에서 **cephmetrics-ansible** 패키지를 업데이트합니다.

```
[root@admin ~]# yum update cephmetrics-ansible
```

2. **/usr/share/cephmetrics-ansible** 디렉터리로 변경합니다.

```
[root@admin ~]# cd /usr/share/cephmetrics-ansible
```

3. 업데이트된 **Red Hat Ceph Storage** 대시보드를 설치합니다.

```
[root@admin cephmetrics-ansible]# ansible-playbook -v playbook.yml
```

## 5장. RED HAT CEPH STORAGE 대시보드를 사용하여 컨테이너에서 실행 중인 CEPH 클러스터 모니터링

**Red Hat Ceph Storage** 대시보드는 **Ceph Storage** 클러스터의 상태를 시각화하는 모니터링 대시보드를 제공합니다. 또한 **Red Hat Ceph Storage** 대시보드 아키텍처는 스토리지 클러스터에 기능을 추가하는 추가 모듈을 위한 프레임워크를 제공합니다.

- 대시보드에 대한 자세한 내용은 [5.1절. “Red Hat Ceph Storage 대시보드”](#) 을 참조하십시오.
- 대시보드를 설치하려면 [5.2절. “Red Hat Ceph Storage 대시보드 설치”](#) 를 참조하십시오.
- 대시보드에 액세스하려면 [5.3절. “Red Hat Ceph Storage 대시보드 액세스”](#) 을 참조하십시오.
- 대시보드 설치 후 기본 암호를 변경하려면 [5.4절. “기본 Red Hat Ceph Storage 대시보드 암호 변경”](#) 을 참조하십시오.
- **Prometheus** 플러그인에 대한 자세한 내용은 [5.5절. “Red Hat Ceph Storage용 Prometheus 플러그인”](#) 을 참조하십시오.
- **Red Hat Ceph Storage Dashboard** 경고 및 구성 방법에 대한 자세한 내용은 [5.6절. “Red Hat Ceph Storage 대시보드 경고”](#) 을 참조하십시오.

### 사전 요구 사항

- 컨테이너에서 실행되는 **Red Hat Ceph Storage** 클러스터

### 5.1. RED HAT CEPH STORAGE 대시보드

**Red Hat Ceph Storage** 대시보드는 **Ceph** 클러스터에 스토리지 클러스터 상태를 시각화할 수 있는 모니터링 대시보드를 제공합니다. 대시보드는 웹 브라우저에서 액세스할 수 있으며 클러스터, 모니터, OSD, 풀 또는 네트워크의 상태에 대한 다양한 지표와 그래프를 제공합니다.

이전 **Red Hat Ceph Storage** 릴리스에서는 **collectd** 플러그인을 통해 모니터링 데이터가 제공되었으며, 이를 통해 데이터를 **Graphite** 모니터링 유틸리티 인스턴스로 전송했습니다. **Red Hat Ceph Storage**

3.3부터는 **ceph-mgr Prometheus** 플러그인을 사용하여 **ceph-mgr** 데몬에서 모니터링 데이터를 직접 가져옵니다.

모니터링 데이터 소스로 **Prometheus**가 도입되면 **Red Hat Ceph Storage Dashboard** 솔루션의 배포 및 운영 관리를 단순화하고 전체 하드웨어 요구 사항을 줄일 수 있습니다. **Red Hat Ceph Storage** 대시보드 솔루션은 **Ceph** 모니터링 데이터를 직접 가져와 컨테이너에 배포된 **Ceph** 클러스터를 지원할 수 있습니다.



#### 참고

아키텍처가 변경되면 **Red Hat Ceph Storage 2.x** 및 **3.0**에서 **Red Hat Ceph Storage 3.3**으로 데이터를 모니터링할 수 있는 마이그레이션 경로가 없습니다.

**Red Hat Ceph Storage** 대시보드는 다음 유틸리티를 사용합니다.

- 배포를 위한 **Ansible** 자동화 애플리케이션입니다.
- 포함된 **Prometheus ceph-mgr** 플러그인입니다.
- 스토리지 클러스터의 각 노드에서 실행되는 **Prometheus node-exporter** 데몬.
- 사용자 인터페이스 및 경고를 제공하는 **Grafana** 플랫폼.

**Red Hat Ceph Storage** 대시보드는 다음과 같은 기능을 지원합니다.

#### 일반 기능

- **Red Hat Ceph Storage 3.1** 이상 지원
- **SELinux** 지원
- **FileStore** 및 **BlueStore OSD** 백엔드 지원

- 암호화된 및 암호화되지 않은 **OSD** 지원
- 모니터, **OSD**, **Ceph Object Gateway** 및 **iSCSI** 역할 지원
- 메타데이터 서버(**MDS**)에 대한 초기 지원
- **east down** 및 **dashboard** 링크
- 15초 단위
- 하드 디스크 드라이브(**HDD**), 솔리드 스테이트 드라이브(**SSD**), **NVMe(Non-volatile Memory Express)** 인터페이스, **Intel® Cache Acceleration** 소프트웨어(**Intel® CAS**) 지원

#### 노드 지표

- **CPU** 및 **RAM** 사용량
- 네트워크 로드

#### 구성 가능한 경고

- **OOB(Out-of-Band)** 경고 및 트리거
- 알림 채널은 설치 중에 자동으로 정의됩니다.
- 기본적으로 생성된 **Ceph** 상태 요약 대시보드

자세한 내용은 [Red Hat Ceph Storage Dashboard Alerts](#) 섹션을 참조하십시오.

#### 클러스터 요약

- **OSD** 구성 요약

- **OSD 파일 저장소 및 BlueStore 요약**
- **역할별 클러스터 버전 분석**
- **디스크 크기 요약**
- **용량 및 디스크 수별 호스트 크기**
- **PG(배치 그룹) 상태 분석**
- **플 수**
- **장치 클래스 요약, HDD vs. SSD**

#### 클러스터 세부 정보

- **클러스터 플래그 상태(noout,nodown)**
- **OSD 또는 Ceph Object Gateway 호스트 up 및 down 상태**
- **플 용량 사용량당**
- **원시 용량 사용**
- **활성 스크립 및 복구 프로세스에 대한 지표**
- **증가 추적 및 예측(raw capacity)**

- **OSD 호스트 및 디스크를 포함하여 다운되거나 거의 전체 OSD에 대한 정보**
- **OSD당 PG 배포**
- **PG 수별 OSD 수를 초과하거나 사용되는 OSD를 강조 표시합니다.**

### **OSD 성능**

- **IOPS(초당 I/O 작업 수) 및 풀별 처리량에 대한 정보**
- **OSD 성능 지표**
- **OSD당 디스크 통계**
- **클러스터 전체 디스크 처리량**
- **읽기/쓰기 비율(클라이언트 IOPS)**
- **디스크 사용률 heat map**
- **Ceph 역할별 네트워크 로드**

### **Ceph Object Gateway 세부 정보**

- **집계된 로드 보기**
- **호스트 대기 시간 및 처리량당**
- **HTTP 작업별 워크로드 분석**

### **Ceph iSCSI 게이트웨이 세부 정보**

- 집계된 뷰
- 설정
- 성능
- 게이트웨이 리소스 사용률당
- 클라이언트 로드 및 구성당
- Ceph 블록 장치 이미지 성능당

## 5.2. RED HAT CEPH STORAGE 대시보드 설치

Red Hat Ceph Storage 대시보드는 실행 중인 Ceph Storage 클러스터에서 다양한 지표를 모니터링할 수 있는 시각적 대시보드를 제공합니다.



참고

Red Hat Ceph Storage 대시보드 업그레이드에 대한 자세한 내용은 [Red Hat Enterprise Linux 설치 가이드의 Red Hat Ceph Storage 대시보드 업그레이드](#)를 참조하십시오.

사전 요구 사항

- Ansible 자동화 애플리케이션과 함께 배포된 컨테이너에서 실행되는 Ceph Storage 클러스터입니다.
- 스토리지 클러스터 노드는 Red Hat Enterprise Linux 7을 사용합니다.

자세한 내용은 1.1.1절. “CDN 및 서브스크립션에 Red Hat Ceph Storage 노드 등록”의 내용을 참조하십시오.

- 클러스터 노드에서 데이터를 수신하고 **Red Hat Ceph Storage** 대시보드를 제공하는 별도의 노드인 **Red Hat Ceph Storage** 대시보드 노드.
- **Red Hat Ceph Storage** 대시보드 노드를 준비합니다.
  - **Red Hat CDN(Red Hat Content Delivery Network)**에 시스템을 등록하고 서브스크립션을 연결한 다음 **Red Hat Enterprise Linux** 리포지토리를 활성화합니다. 자세한 내용은 **1.1.1절. “CDN 및 서브스크립션에 Red Hat Ceph Storage 노드 등록”**의 내용을 참조하십시오.
  - 모든 노드에서 툴 리포지토리를 활성화합니다.
 

자세한 내용은 **Red Hat Enterprise Linux용 Red Hat Ceph Storage 3 설치 가이드의 Red Hat Ceph Storage 리포지토리 활성화** 섹션을 참조하십시오.
  - 방화벽을 사용하는 경우 다음 **TCP** 포트가 열려 있는지 확인합니다.

**표 5.1. TCP 포트 요구 사항**

포트	사용	어디에 있습니까?
<b>3000</b>	Grafana	Red Hat Ceph Storage 대시보드 노드.
<b>9090</b>	기본 Prometheus 그래프	Red Hat Ceph Storage 대시보드 노드.
<b>9100</b>	Prometheus의 <b>node-exporter</b> 데몬	모든 스토리지 클러스터 노드.
<b>9283</b>	Ceph 데이터 수집	모든 <b>ceph-mgr</b> 노드.
<b>9287</b>	Ceph iSCSI 게이트웨이 데이터	모든 Ceph iSCSI 게이트웨이 노드.

자세한 내용은 **Red Hat Enterprise Linux 7의 보안 가이드의 방화벽 사용** 장을 참조하십시오.

**절차**

**Ansible** 관리 노드에서 **root** 사용자로 다음 명령을 실행합니다.

1. **cephmetrics-ansible** 패키지를 설치합니다.

```
[root@admin ~]# yum install cephmetrics-ansible
```

2. **Ceph Ansible** 인벤토리를 기반으로 사용하여 기본적으로 `/etc/ansible/hosts` 에 있는 **Ansible** 인벤토리 파일의 `[ceph-grafana]` 섹션 아래에 **Red Hat Ceph Storage Dashboard** 노드를 추가합니다.

```
[ceph-grafana]
$HOST_NAME
```

교체:

- **\$HOST\_NAME (Red Hat Ceph Storage 대시보드 노드 이름)**

예를 들면 다음과 같습니다.

```
[ceph-grafana]
node0
```

3. `/usr/share/cephmetrics-ansible/` 디렉터리로 변경합니다.

```
[root@admin ~]# cd /usr/share/cephmetrics-ansible
```

4. **Ansible** 플레이북을 실행합니다.

```
[root@admin cephmetrics-ansible]# ansible-playbook -v playbook.yml
```



중요

클러스터 구성을 업데이트할 때마다 (예: **MON** 또는 **OSD** 노드를 추가 또는 제거) **cephmetrics Ansible** 플레이북을 다시 실행해야 합니다.



참고

**cephmetrics Ansible** 플레이북은 다음 작업을 수행합니다.

- **prometheus** 플러그인을 활성화하고 **TCP 포트 9283**을 열도록 **ceph-mgr** 인스턴스를 업데이트합니다.
- **Prometheus node-exporter** 데몬을 스토리지 클러스터의 각 노드에 배포합니다.
  - **TCP 포트 9100**을 엽니다.
  - **node-exporter** 데몬을 시작합니다.
- **Red Hat Ceph Storage** 대시보드 노드의 **Docker/systemd**에 **Grafana** 및 **Prometheus** 컨테이너를 배포합니다.
  - **Prometheus**는 **ceph-mgr** 노드 및 각 **ceph** 호스트에서 실행 중인 **node-exporter**에서 데이터를 수집하도록 구성되어 있습니다.
  - **TCP 포트 3000**을 엽니다.
  - 대시보드, 주제 및 사용자 계정은 모두 **Grafana**에 생성됩니다.
  - 관리자의 **Grafana URL**을 출력합니다.

**5.3. RED HAT CEPH STORAGE** 대시보드 액세스

**Red Hat Ceph Storage** 대시보드에 액세스하면 **Red Hat Ceph Storage** 클러스터를 관리하는 웹 기반 관리 툴에 액세스할 수 있습니다.

사전 요구 사항

- **Red Hat Ceph Storage** 대시보드를 설치합니다.
- 노드가 제대로 동기화되지 않은 경우 **Ceph Storage Dashboard** 노드, 클러스터 노드 및 브라우저 간에 시간 지연이 발생할 수 있으므로 **NTP**가 올바르게 클럭을 동기화하는지 확인합니다. **Red Hat Enterprise Linux** 또는 **Ubuntu** 용 **Red Hat Ceph Storage 3** 설치 가이드의 **Red Hat Ceph Storage**용 네트워크 시간 프로토콜 구성 섹션을 참조하십시오.

## 절차

1. 웹 브라우저에 다음 **URL**을 입력합니다.

```
http://$HOST_NAME:3000
```

교체:

- **\$HOST\_NAME** (Red Hat Ceph Storage 대시보드 노드 이름)

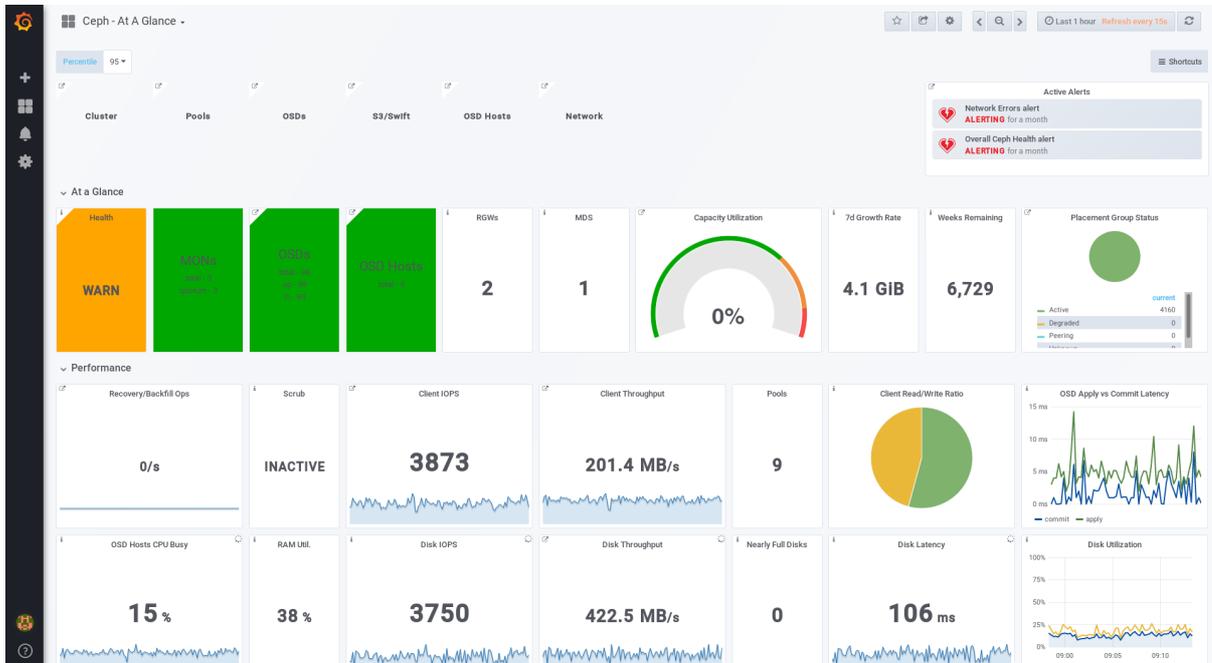
예를 들면 다음과 같습니다.

```
http://cephmetrics:3000
```

2. **admin** 사용자의 암호를 입력합니다. 설치 중에 암호를 설정하지 않은 경우 기본 암호인 **admin**을 사용합니다.

로그인하면 **Ceph At a Glance** 대시보드에 자동으로 배치됩니다. **Ceph Glance** 대시보드는 용량, 성능 및 노드 수준 성능 정보에 대한 대략적인 개요를 제공합니다.

## 예제



추가 리소스

- [Red Hat Ceph Storage 관리 가이드의 기본 Red Hat Ceph Storage Dashboard Password](#) 섹션을 참조하십시오.

5.4. 기본 RED HAT CEPH STORAGE 대시보드 암호 변경

Red Hat Ceph Storage 대시보드에 액세스하는 기본 사용자 이름 및 암호는 `admin` 및 `admin` 으로 설정됩니다. 보안상의 이유로 설치 후 암호를 변경해야 할 수 있습니다.



참고

암호가 기본값으로 재설정되지 않도록 하려면 `/usr/share/cephmetrics-ansible/group_vars/all.yml` 파일에서 사용자 지정 암호를 업데이트합니다.

사전 요구 사항

- [Red Hat Ceph Storage 대시보드를 설치합니다.](#)
- [Red Hat Ceph Storage 대시보드에 로그인합니다.](#)

절차

1. 왼쪽 상단에 있는 **Grafana** 아이콘을 클릭합니다.
2. 암호를 수정할 사용자 이름을 마우스로 가리키십시오. 이 경우 **admin**.
3. 프로필을 클릭합니다.
4. 암호 변경을 클릭합니다.
5. 새 암호를 두 번 입력하고 암호 변경을 클릭합니다.

#### 추가 리소스

- 암호를 잊어 버린 경우 **Grafana** 웹 페이지의 [관리자 암호 재설정](#) 절차를 따르십시오.

### 5.5. RED HAT CEPH STORAGE용 PROMETHEUS 플러그인

스토리지 관리자는 성능 데이터를 수집하고 **Red Hat Ceph Storage** 대시보드에 대한 **Prometheus** 플러그인 모듈을 사용하여 해당 데이터를 내보낸 다음 이 데이터에 대한 쿼리를 수행할 수 있습니다. **Prometheus** 모듈을 사용하면 **ceph-mgr** 이 **Ceph** 관련 상태 및 성능 데이터를 **Prometheus** 서버에 노출할 수 있습니다.

#### 5.5.1. 사전 요구 사항

- **Red Hat Ceph Storage 3.1** 이상을 실행합니다.
- **Red Hat Ceph Storage** 대시보드 설치.

#### 5.5.2. Prometheus 플러그인

**Prometheus** 플러그인은 **ceph-mgr** 의 컬렉션 지점에서 **Ceph** 성능 카운터를 전달할 내보내기를 제공합니다. **Red Hat Ceph Storage** 대시보드는 **Ceph Monitor** 및 **OSD**와 같은 모든 **MgrClient** 프로세스에서 **MMgrReport** 메시지를 수신합니다. 마지막 수의 샘플 수의 원형 버퍼는 성능 카운터 스키마 데이터 및 실제 카운터 데이터를 포함합니다. 이 플러그인은 **HTTP** 끝점을 생성하고 폴링할 때 모든 카운터의 최신 샘플을 검색합니다. **HTTP** 경로 및 쿼리 매개 변수는 무시됩니다. 모든 보고 엔터티에 대한 모든 **extant** 카운터는 텍스트 예외 형식으로 반환됩니다.

### 추가 리소스

- 텍스트 계획 형식에 대한 자세한 내용은 [Prometheus 설명서](#) 를 참조하십시오.

### 5.5.3. Prometheus 환경 관리

**Prometheus**를 사용하여 **Ceph** 스토리지 클러스터를 모니터링하려면 **Prometheus** 내보내기를 구성하고 활성화하여 **Ceph** 스토리지 클러스터에 대한 메타데이터 정보를 수집할 수 있습니다.

#### 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage 3.1** 클러스터
- **Red Hat Ceph Storage** 대시보드 설치

#### 절차

1. 루트 사용자로 `/etc/prometheus/prometheus.yml` 파일을 열고 편집합니다.
  - a. 글로벌 섹션에서 `scrape_interval` 및 `evaluation_interval` 옵션을 15초로 설정합니다.

#### 예제

```
global:  
  scrape_interval: 15s  
  evaluation_interval: 15s
```

- b. `scrape_configs` 섹션에서 `honor_labels: true` 옵션을 추가하고 타겟 및 각 `ceph-mgr` 노드의 인스턴스 옵션을 편집합니다.

#### 예제

```

scrape_configs:
- job_name: 'node'
  honor_labels: true
  static_configs:
- targets: ['node1.example.com:9100']
  labels:
    instance: "node1.example.com"
- targets: ['node2.example.com:9100']
  labels:
    instance: "node2.example.com"

```



#### 참고

**honor\_labels** 옵션을 사용하면 Ceph에서 Ceph 스토리지 클러스터의 모든 노드와 관련된 레이블이 적절히 지정된 데이터를 출력할 수 있습니다. 이를 통해 Ceph에서 Prometheus를 덮어쓰지 않고 적절한 인스턴스 레이블을 내보낼 수 있습니다.

c.

새 노드를 추가하려면 다음 형식으로 대상, 인스턴스 옵션을 추가하면 됩니다.

#### 예제

```

- targets: ['new-node.example.com:9100']
  labels:
    instance: "new-node"

```



#### 참고

**instance** 레이블은 노드의 짧은 호스트 이름인 Ceph의 OSD 메타데이터 인스턴스 필드에 표시되는 항목과 일치해야 합니다. 이는 Ceph 통계와 노드의 통계의 상관 관계를 유지하는 데 도움이 됩니다.

2.

다음 형식의 `/etc/prometheus/ceph_targets.yml` 파일에 Ceph 대상을 추가합니다.

#### 예제

```
[
  {
    "targets": [ "cephnode1.example.com:9283" ],
    "labels": {}
  }
]
```

3.

**Prometheus** 모듈을 활성화합니다.

```
# ceph mgr module enable prometheus
```

#### 5.5.4. Prometheus 데이터 및 쿼리 작업

통계 이름은 정확히 **Ceph** 이름이며, 불법 문자는 밑줄로 변환되고 모든 이름에 접두사가 붙습니다. 모든 **Ceph** 데몬 통계에는 **ceph\_daemon** 레이블이 있어 제공되는 데몬의 유형 및 ID(예: **osd.123**)를 식별합니다. 일부 통계는 다양한 유형의 데몬에서 발생할 수 있으므로 쿼리 시 **Ceph** 모니터 및 브라운드 통계에서 혼합되지 않도록 **osd** 로 시작하는 **Ceph** 데몬을 필터링해야 합니다. 글로벌 **Ceph** 스토리지 클러스터 통계에는 보고되는 항목에 적합한 레이블이 있습니다. 예를 들어 풀과 관련된 지표에는 **pool\_id** 레이블이 있습니다. 코어 **Ceph**의 히스토그램을 나타내는 긴 실행 평균은 합계 및 개수 성능 지표 쌍으로 표시됩니다.

다음 예제 쿼리를 **Prometheus** 표현식 브라우저에서 사용할 수 있습니다.

#### OSD의 물리 디스크 사용률 표시

```
(irate(node_disk_io_time_ms[1m]) /10) and on(device,instance)
ceph_disk_occupation{ceph_daemon="osd.1"}
```

#### 운영 체제에 표시된 OSD의 물리적 IOPS 표시

```
irate(node_disk_reads_completed[1m]) + irate(node_disk_writes_completed[1m]) and on (device,
instance) ceph_disk_occupation{ceph_daemon="osd.1"}
```

### 풀 및 OSD 메타데이터 시리즈

특정 메타데이터 필드에 표시 및 쿼리를 사용할 수 있도록 특수 데이터 시리즈가 출력됩니다. 풀에는 `ceph_pool_metadata` 필드가 있습니다. 예를 들면 다음과 같습니다.

```
ceph_pool_metadata{pool_id="2",name="cephfs_metadata_a"} 1.0
```

OSD에는 `ceph_osd_metadata` 필드가 있습니다. 예를 들면 다음과 같습니다.

```
ceph_osd_metadata{cluster_addr="172.21.9.34:6802/19096",device_class="ssd",ceph_daemon="osd.0",public_addr="172.21.9.34:6801/19096",weight="1.0"} 1.0
```

`node_exporter`를 사용하여 드라이브 통계 조정

Ceph의 Prometheus 출력은 Prometheus 노드 내보내기의 일반 노드 모니터링과 함께 사용하도록 설계되었습니다. Ceph OSD 통계와 일반 노드 모니터링 드라이브 통계의 상관 관계, 특수 데이터 시리즈가 출력됩니다. 예를 들면 다음과 같습니다.

```
ceph_disk_occupation{ceph_daemon="osd.0",device="sdd",exported_instance="node1"}
```

OSD ID로 디스크 통계를 가져오려면 Prometheus 쿼리에서 및 연산자(\*) 연산자를 사용합니다. 모든 메타데이터 지표의 값은 1 이므로 별표 연산자를 사용하여 중복적으로 작동합니다. 별표 연산자를 사용하면 결과 메트릭에 쿼리의 한 쪽의 추가 레이블이 있도록 `group_left` 및 `group_right` 그룹화 수정자를 사용할 수 있습니다. 예를 들면 다음과 같습니다.

```
rate(node_disk_bytes_written[30s]) and on (device,instance)
ceph_disk_occupation{ceph_daemon="osd.0"}
```

`label_replace` 사용

`label_replace` 함수는 레이블을 에 추가하거나 쿼리 내의 지표를 변경할 수 있습니다. OSD 및 해당 디스크의 쓰기 속도를 상관시키기 위해 다음 쿼리를 사용할 수 있습니다.

```
label_replace(rate(node_disk_bytes_written[30s]), "exported_instance", "$1", "instance", "(.*):.") and
on (device,exported_instance) ceph_disk_occupation{ceph_daemon="osd.0"}
```

추가 리소스

- [쿼리 구성에 대한 자세한 내용은 Prometheus 쿼리 기본 사항을 참조하십시오.](#)
- [자세한 내용은 Prometheus의 label\\_replace 설명서 를 참조하십시오.](#)

### 5.5.5. Prometheus 표현식 브라우저 사용

내장된 Prometheus 표현식 브라우저를 사용하여 수집된 데이터에 대한 쿼리를 실행합니다.

#### 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 3.1 클러스터
- Red Hat Ceph Storage 대시보드 설치

#### 절차

1. 웹 브라우저의 URL을 입력합니다.

```
http://$DASHBOARD_SEVER_NAME:9090/graph
```

replace...

- **\$DASHBOARD\_SEVER\_NAME** 은 Red Hat Ceph Storage 대시보드 서버의 이름으로 사용됩니다.
2. 그래프 를 클릭한 다음 쿼리 창에 쿼리를 입력하거나 붙여넣고 **Execute** 버튼을 누릅니다.
    - a. 콘솔 창에서 결과를 봅니다.
  3. 그래프 를 클릭하여 렌더링된 데이터를 봅니다.

#### 추가 리소스

-

자세한 내용은 [Prometheus 웹 사이트의 Prometheus 표현식 브라우저 설명서](#)를 참조하십시오.

### 5.5.6. 추가 리소스

- [Prometheus 웹 페이지](#)

## 5.6. RED HAT CEPH STORAGE 대시보드 경고

이 섹션에는 **Red Hat Ceph Storage** 대시보드의 경고에 대한 정보가 포함되어 있습니다.

- **Red Hat Ceph Storage** 대시보드 경고에 대한 자세한 내용은 [5.6.2절. “경고 정보”](#)을 참조하십시오.
- 알림을 보려면 [5.6.3절. “경고 상태 대시보드에 액세스”](#)을 참조하십시오.
- 알림 대상을 구성하려면 [5.6.4절. “알림 대상 구성”](#)을 참조하십시오.
- 기본 경고를 변경하거나 새 경고를 추가하려면 [5.6.5절. “기본 경고 및 새 항목 추가 변경”](#)를 참조하십시오.

### 5.6.1. 사전 요구 사항

- [Red Hat Ceph Storage](#) 대시보드를 설치합니다.
- [Red Hat Ceph Storage](#) 대시보드에 로그인합니다.

### 5.6.2. 경고 정보

**Red Hat Ceph Storage** 대시보드는 **Grafana** 플랫폼에서 제공하는 경고 메커니즘을 지원합니다. 관심 있는 메트릭이 특정 값에 도달할 때 알림을 보내도록 대시보드를 구성할 수 있습니다. 이러한 지표는 **Alert Status** 대시보드에 있습니다.

기본적으로 경고 상태에는 **Overall Ceph Health**, **OSD Down** 또는 **Pool Capacity** 와 같은 특정 메트

력이 이미 포함되어 있습니다. 이 대시보드에 관심 있는 메트릭을 추가하거나 트리거 값을 변경할 수 있습니다.

다음은 **Red Hat Ceph Storage** 대시보드에 포함된 사전 정의 경고 목록입니다.

- 전체 Ceph 상태
- 디스크 전체(>85%)
- OSD 다운
- OSD 호스트 다운
- PG의 Stuck Inactive
- OSD Host Less - Free Capacity Check
- OSD의 응답 시간이 높습니다.
- 네트워크 오류
- Pool Capacity High
- 아래쪽 모니터링
- 전체 클러스터 용량 감소
- PG 수가 많은 OSD

### 5.6.3. 경고 상태 대시보드에 액세스

특정 **Red Hat Ceph Storage** 대시보드 경고는 기본적으로 경고 상태 대시보드에서 구성됩니다. 이 섹션에서는 두 가지 방법으로 액세스할 수 있습니다.

#### 절차

대시보드에 액세스하려면 다음을 수행합니다.

- **Glance** 대시보드의 오른쪽 상단에 있는 활성 경고 패널을 클릭합니다.

또는..

- **Grafana** 아이콘 옆에 있는 왼쪽 상단에 있는 대시보드 메뉴를 클릭합니다. **Alert Status** 를 선택합니다.

### 5.6.4. 알림 대상 구성

설치 중에 **cephmetrics** 라는 알림 채널이 자동으로 생성됩니다. 사전 구성된 모든 경고는 **cephmetrics** 채널을 참조하지만 경고를 수신하기 전에 원하는 알림 유형을 선택하여 알림 채널 정의를 완료합니다. **Grafana** 플랫폼은 이메일, **Slack** 및 **PagerDuty**를 포함하여 다양한 알림 유형을 지원합니다.

#### 절차

- 알림 채널을 구성하려면 **Grafana** 웹 페이지의 **알림 알림** 섹션의 지침을 따릅니다.

### 5.6.5. 기본 경고 및 새 항목 추가 변경

이 섹션에서는 이미 구성된 경고의 트리거 값을 변경하는 방법과 경고 상태 대시보드에 새 경고를 추가 하는 방법을 설명합니다.

#### 절차

- 경고의 트리거 값을 변경하거나 새 경고를 추가하려면 **Grafana** 웹 페이지의 **경고 엔진 및 규칙 가이드** 를 따르십시오.



### 중요

사용자 정의 경고를 재정의하지 않도록 트리거 값을 변경하거나 새 경고를 추가할 때 **Red Hat Ceph Storage Dashboard** 패키지를 업그레이드할 때 경고 상태 대시보드가 업데이트되지 않습니다.

### 추가 리소스

- [Grafana 웹 페이지](#)

## 부록 A. 버전 2와 3의 ANSIBLE 변수 변경

**Red Hat Ceph Storage 3**에서는 `/usr/share/ceph-ansible/group_vars/` 디렉터리에 있는 구성 파일의 특정 변수가 변경되었거나 삭제되었습니다. 다음 표에는 모든 변경 사항이 나열되어 있습니다. 버전 3으로 업그레이드한 후 `all.yml.sample` 및 `osds.yml.sample` 파일을 다시 복사하여 이러한 변경 사항을 반영합니다. 자세한 내용은 [Containers](#)에서 실행되는 **Red Hat Ceph Storage** 클러스터 업그레이드를 참조하십시오.

이전 옵션	새로운 옵션	파일
<code>mon_containerized_deploy ment</code>	<code>containerized_deployment</code>	<code>all.yml</code>
<code>ceph_mon_docker_interface</code>	<code>monitor_interface</code>	<code>all.yml</code>
<code>ceph_rhcs_cdn_install</code>	<code>ceph_repository_type: cdn</code>	<code>all.yml</code>
<code>ceph_rhcs_iso_install</code>	<code>ceph_repository_type: iso</code>	<code>all.yml</code>
<code>ceph_rhcs</code>	<code>ceph_origin</code> : 리포지토리 및 <code>ceph_repository: rhcs</code> (기본적으로 사용)	<code>all.yml</code>
<code>journal_collocation</code>	<code>osd_scenario: collocated</code>	<code>osds.yml</code>
<code>raw_multi_journal</code>	<code>osd_scenario: non-collocated</code>	<code>osds.yml</code>
<code>raw_journal_devices</code>	<code>dedicated_devices</code>	<code>osds.yml</code>
<code>dmccrypt_journal_collocation</code>	<code>dmccrypt: true</code> + <code>osd_scenario: collocated</code>	<code>osds.yml</code>
<code>dmccrypt_dedicated_journal</code>	<code>dmccrypt: true</code> + <code>osd_scenario: non-collocated</code>	<code>osds.yml</code>