



# Red Hat Ceph Storage 3

## Red Hat Enterprise Linux 설치 가이드

Red Hat Enterprise Linux에 Red Hat Ceph Storage 설치



# Red Hat Ceph Storage 3 Red Hat Enterprise Linux 설치 가이드

---

Red Hat Enterprise Linux에 Red Hat Ceph Storage 설치

## 법적 공지

Copyright © 2024 Red Hat, Inc.

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 AMD64 및 Intel 64 아키텍처에서 실행되는 Red Hat Enterprise Linux 7에 Red Hat Ceph Storage를 설치하는 방법에 대해 설명합니다.

## 차례

<b>1장. RED HAT CEPH STORAGE는 무엇입니까?</b> .....	<b>4</b>
<b>2장. RED HAT CEPH STORAGE 설치 요구사항</b> .....	<b>6</b>
2.1. 사전 요구 사항	6
2.2. RED HAT CEPH STORAGE 설치를 위한 요구 사항 확인	6
2.3. RED HAT CEPH STORAGE의 운영 체제 요구 사항	7
2.4. CDN 및 서브스크립션에 RED HAT CEPH STORAGE 노드 등록	7
2.5. RED HAT CEPH STORAGE 리포지토리 활성화	9
2.6. OSD 노드에서 RAID 컨트롤러 사용 고려 사항(선택 사항)	10
2.7. 오브젝트 게이트웨이에서 NVME 사용 고려 사항(선택 사항)	10
2.8. RED HAT CEPH STORAGE 네트워크 구성 확인	10
2.9. RED HAT CEPH STORAGE 방화벽 설정	11
2.10. SUDO 액세스를 사용하여 ANSIBLE 사용자 생성	15
2.11. ANSIBLE에 대해 암호 없는 SSH 활성화	17
<b>3장. RED HAT CEPH STORAGE 배포</b> .....	<b>19</b>
3.1. 사전 요구 사항	19
3.2. RED HAT CEPH STORAGE 클러스터 설치	19
3.3. 모든 NVME 스토리지에 대한 OSD ANSIBLE 설정 구성	35
3.4. 메타데이터 서버 설치	37
3.5. CEPH 클라이언트 역할 설치	38
3.6. CEPH OBJECT GATEWAY 설치	40
3.7. NFS-GANESHA 게이트웨이 설치	47
3.8. 제한 옵션 이해	49
3.9. 추가 리소스	49
<b>4장. RED HAT CEPH STORAGE 클러스터 업그레이드</b> .....	<b>50</b>
사전 요구 사항	52
4.1. 스토리지 클러스터 업그레이드	54
4.2. RED HAT CEPH STORAGE 대시보드 업그레이드	59
<b>5장. 다음을 어떻게 해야 합니까?</b> .....	<b>61</b>
<b>부록 A. 문제 해결</b> .....	<b>62</b>
A.1. ANSIBLE은 설치를 중지하기 때문에 LESS DEVICES THAN IT EXPECTED	62
<b>부록 B. 수동으로 RED HAT CEPH STORAGE 설치</b> .....	<b>64</b>
B.1. 사전 요구 사항	64
B.2. 수동으로 CEPH MANAGER 설치	76
<b>부록 C. CEPH 명령줄 인터페이스 설치</b> .....	<b>87</b>
사전 요구 사항	87
절차	87
<b>부록 D. 수동으로 CEPH 블록 장치 설치</b> .....	<b>89</b>
사전 요구 사항	89
절차	89
<b>부록 E. 수동으로 CEPH OBJECT GATEWAY 설치</b> .....	<b>93</b>
사전 요구 사항	93
절차	93
추가 세부 정보	98
<b>부록 F. CEPH 기본 설정 덮어쓰기</b> .....	<b>99</b>

부록 G. RED HAT CEPH STORAGE 2에서 3으로 수동 업그레이드 .....	100
모니터 노드 업그레이드 .....	101
G.1. 수동으로 CEPH MANAGER 설치 .....	104
부록 H. 버전 2와 3의 ANSIBLE 변수 변경 .....	114
부록 I. ANSIBLE로 기존 CEPH 클러스터 가져오기 .....	115
부록 J. ANSIBLE을 사용하여 CEPH 클러스터 제거 .....	117



# 1장. RED HAT CEPH STORAGE는 무엇입니까?

Red Hat Ceph Storage는 가장 안정적인 버전의 Ceph 스토리지 시스템을 Ceph 관리 플랫폼, 배포 유틸리티 및 지원 서비스와 결합하는 확장 가능한 오픈 소프트웨어 정의 스토리지 플랫폼입니다.

Red Hat Ceph Storage는 클라우드 인프라 및 웹 규모 개체 스토리지를 위해 설계되었습니다. Red Hat Ceph Storage 클러스터는 다음과 같은 유형의 노드로 구성됩니다.

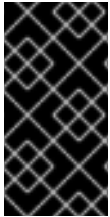
## Red Hat Ceph Storage Ansible 관리 노드

이러한 유형의 노드는 이전 버전의 Red Hat Ceph Storage에서 수행한 기존 Ceph 관리 노드 역할을 합니다. 이 유형의 노드는 다음과 같은 기능을 제공합니다.

- 중앙 집중식 스토리지 클러스터 관리
- Ceph 구성 파일 및 키
- 보안상의 이유로 인터넷에 액세스할 수 없는 노드에 Ceph를 설치하는 경우 선택적으로 로컬 리포지토리

## 노드 모니터링

각 모니터 노드는 클러스터 맵의 마스터 사본을 유지 관리하는 모니터 데몬(**ceph-mon**)을 실행합니다. 클러스터 맵에는 클러스터 토폴로지가 포함됩니다. Ceph 클러스터에 연결된 클라이언트는 모니터에서 현재 클러스터 맵 사본을 검색하고 이를 통해 클라이언트가 데이터를 읽고 클러스터에 쓸 수 있습니다.



### 중요

그러나 프로덕션 클러스터에서고가용성을 보장하기 위해 Ceph를 하나의 모니터로 실행할 수 있지만, Red Hat은 최소 3개의 모니터 노드가 있는 배포만 지원합니다. Red Hat은 750 OSD를 초과하는 스토리지 클러스터에 대해 총 5개의 Ceph Monitor를 배포하는 것이 좋습니다.

## OSD 노드

각 OSD(오브젝트 스토리지 장치) 노드는 Ceph OSD 데몬(**ceph-osd**)을 실행하여 노드에 연결된 논리 디스크와 상호 작용합니다. Ceph는 이러한 OSD 노드에 데이터를 저장합니다.

Ceph는 기본값이 3개지만 프로덕션 클러스터는 매우 적은 OSD 노드로 실행할 수 있지만, 프로덕션 클러스터는 모드 크기(예: 스토리지 클러스터의 50개 OSD)에서 성능이 향상될 수 있습니다. Ceph 클러스터에는 여러 OSD 노드가 있으므로 CRUSH 맵을 생성하여 분리된 오류 도메인을 허용하는 것이 좋습니다.

## MDS 노드

각 메타데이터 서버(MDS) 노드는 MDS 데몬(**ceph-mds**)을 실행하여 Ceph 파일 시스템(CephFS)에 저장된 파일과 관련된 메타데이터를 관리합니다. MDS 데몬은 공유 클러스터에 대한 액세스도 조정합니다.

## Object Gateway 노드

Ceph Object Gateway 노드는 Ceph RADOS Gateway 데몬(**ceph-radosgw**)을 실행하고 **librados** 상단에 구축된 오브젝트 스토리지 인터페이스로, Ceph Storage 클러스터에 RESTful 게이트웨이를 제공합니다. Ceph Object Gateway는 다음 두 가지 인터페이스를 지원합니다.

### S3

Amazon S3 RESTful API의 대규모 하위 집합과 호환되는 인터페이스가 포함된 오브젝트 스토리지 기능을 제공합니다.



## Swift

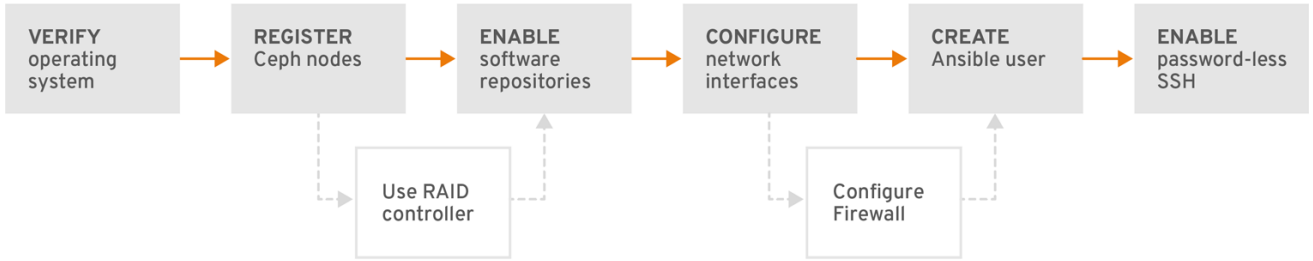
OpenStack Swift API의 대규모 하위 집합과 호환되는 인터페이스가 포함된 오브젝트 스토리지 기능을 제공합니다.

Ceph 아키텍처에 대한 자세한 내용은 Red Hat Ceph Storage 3 [아키텍처 가이드를 참조하십시오](#).

최소 권장 하드웨어의 경우 [Red Hat Ceph Storage Hardware Selection Guide](#) 3을 참조하십시오.

## 2장. RED HAT CEPH STORAGE 설치 요구사항

그림 2.1. 사전 요구 사항 워크플로



CEPH\_459707\_0818

RHCS(Red Hat Ceph Storage)를 설치하기 전에 다음 요구 사항을 검토하고 그에 따라 각 모니터, OSD, 메타데이터 서버 및 클라이언트 노드를 준비합니다.

### 2.1. 사전 요구 사항

- 하드웨어가 최소 요구 사항을 충족하는지 확인합니다. 자세한 내용은 Red Hat Ceph Storage 3의 [하드웨어 가이드](#) 를 참조하십시오.

### 2.2. RED HAT CEPH STORAGE 설치를 위한 요구 사항 확인

Task	필수 항목	섹션	권장 사항
운영 체제 버전 확인	있음	2.3절. "Red Hat Ceph Storage의 운영 체제 요구 사항"	
Ceph 노드 등록	있음	2.4절. "CDN 및 서비스 크립션에 Red Hat Ceph Storage 노드 등록"	
Ceph 소프트웨어 리포지토리 활성화	있음	2.5절. "Red Hat Ceph Storage 리포지토리 활성화"	
OSD 노드에서 RAID 컨트롤러 사용	없음	2.6절. "OSD 노드에서 RAID 컨트롤러 사용 고려 사항(선택 사항)"	RAID 컨트롤러에서 쓰기-백 캐시를 활성화하면 OSD 노드의 소규모 I/O 쓰기 처리량이 증가할 수 있습니다.
네트워크 구성	있음	2.8절. "Red Hat Ceph Storage 네트워크 구성 확인"	최소한 공용 네트워크가 필요합니다. 그러나 클러스터 통신용 사설 네트워크가 권장됩니다.
방화벽 구성	없음	2.9절. "Red Hat Ceph Storage 방화벽 설정"	방화벽은 네트워크에 대한 신뢰 수준을 높일 수 있습니다.

Task	필수 항목	섹션	권장 사항
Ansible 사용자 생성	있음	2.10절. " <b>sudo</b> 액세스를 사용하여 Ansible 사용자 생성"	모든 Ceph 노드에 Ansible 사용자를 생성해야 합니다.
암호 없는 SSH 활성화	있음	2.11절. "Ansible에 대해 암호 없는 SSH 활성화"	Ansible에 필요합니다.

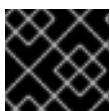


### 참고

기본적으로 **ceph-ansible** 은 NTP를 요구 사항으로 설치합니다. NTP가 사용자 지정되면 수동으로 *Red Hat Ceph Storage 설치에서 네트워크 시간 프로토콜* 구성을 참조하십시오.

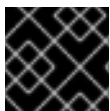
## 2.3. RED HAT CEPH STORAGE의 운영 체제 요구 사항

Red Hat Ceph Storage 3에는 Red Hat Enterprise Linux 7 update 5 이상이 필요합니다. 클러스터의 모든 노드에서 동일한 버전 및 아키텍처를 사용합니다.



### 중요

Red Hat Ceph Storage 3은 Red Hat Enterprise Linux 8에서 지원되지 않습니다.



### 중요

Red Hat은 이기종 운영 체제 또는 버전이 있는 클러스터를 지원하지 않습니다.

### 추가 리소스

- Red Hat Enterprise Linux 7용 [설치 가이드](#).
- Red Hat Enterprise Linux 7용 [시스템 관리자 가이드](#).

[요구 사항 체크리스트로 돌아가기](#)

## 2.4. 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.



## 참고

설치 중에 인터넷에 액세스할 수 없는 RHCS 노드의 경우 Red Hat Satellite 서버를 사용하여 소프트웨어 콘텐츠를 제공합니다. 또는 로컬 Red Hat Enterprise Linux 7 Server ISO 이미지를 마운트하고 RHCS 노드를 ISO 이미지로 가리킵니다. 자세한 내용은 [Red Hat 지원](#)에 문의하십시오.

Red Hat Satellite 서버에 Ceph 노드를 등록하는 방법에 대한 자세한 내용은 [How to register Ceph with Satellite 6](#) and [how to register Ceph with Satellite 5](#) articles를 참조하십시오.

## 사전 요구 사항

- 유효한 Red Hat 서브스크립션
- RHCS 노드는 인터넷에 연결할 수 있어야 합니다.

## 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.
- [2.5절. "Red Hat Ceph Storage 리포지토리 활성화"](#)

[요구 사항 체크리스트로 돌아가기](#)

## 2.5. RED HAT CEPH STORAGE 리포지토리 활성화

Red Hat Ceph Storage를 설치하려면 먼저 설치 방법을 선택해야 합니다. Red Hat Ceph Storage는 다음 두 가지 설치 방법을 지원합니다.

- **CDN(Content Delivery Network)**  
인터넷에 직접 연결할 수 있는 Ceph 노드가 있는 Ceph Storage 클러스터의 경우 Red Hat Subscription Manager를 사용하여 필요한 Ceph 리포지토리를 활성화합니다.
- **로컬 리포지토리**  
보안 조치에서 인터넷 액세스가 불가능한 Ceph Storage 클러스터의 경우 로컬 리포지토리를 설치할 수 있는 ISO 이미지로 제공되는 단일 소프트웨어 빌드에서 Red Hat Ceph Storage 3.3을 설치합니다.

### 사전 요구 사항

- 유효한 고객 서브스크립션.
- CDN 설치의 경우 RHCS 노드가 인터넷에 연결할 수 있어야 합니다.
- CDN 설치의 경우 [CDN에 클러스터 노드를 등록합니다](#).
- EPEL 소프트웨어 리포지토리를 비활성화합니다.

```
[root@monitor ~]# yum install yum-utils vim -y
[root@monitor ~]# yum-config-manager --disable epel
```

### 절차

#### CDN 설치의 경우:

**Ansible 관리 노드에서** 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
```

#### ISO 설치의 경우:

1. Red Hat 고객 포털에 로그인합니다.
2. **소프트웨어 및 다운로드 센터**를 방문하려면 **다운로드**를 클릭합니다.
3. Red Hat Ceph Storage 영역에서 **Download Software**를 클릭하여 최신 버전의 소프트웨어를 다운로드합니다.

## 추가 리소스

- Red Hat Enterprise Linux용 시스템 관리자 가이드의 [서브스크립션 등록 및 관리](#) 장.

[요구 사항 체크리스트로 돌아가기](#)

## 2.6. OSD 노드에서 RAID 컨트롤러 사용 고려 사항(선택 사항)

2GB의 캐시가 설치된 OSD 노드에 RAID 컨트롤러가 있는 경우 나중 쓰기 캐시를 활성화하면 적은 I/O 쓰기 처리량이 증가할 수 있습니다. 그러나 캐시는 비휘발성이어야 합니다.

최신 RAID 컨트롤러는 일반적으로 정전 이벤트 중에 비volatile NAND 메모리에 휘발성 메모리를 드레인할 수 있는 충분한 전력을 제공하는 슈퍼 캐패시터가 있습니다. 전원을 복원한 후 특정 컨트롤러와 펌웨어가 작동하는 방식을 이해하는 것이 중요합니다.

일부 RAID 컨트롤러는 수동 개입이 필요합니다. 하드 드라이브는 일반적으로 운영 체제에 디스크 캐시를 사용하도록 설정하거나 비활성화해야 하는지를 알립니다. 그러나 특정 RAID 컨트롤러와 일부 펌웨어는 이러한 정보를 제공하지 않습니다. 파일 시스템 손상을 방지하려면 디스크 수준 캐시가 비활성화되어 있는지 확인합니다.

나중 쓰기 캐시가 활성화된 각 Ceph OSD 데이터 드라이브에 대해 쓰기-back으로 단일 RAID 0 볼륨을 생성합니다.

Serial Attached SCSI (SAS) 또는 SATA 연결 Solid-state Drive (SSD) 디스크도 RAID 컨트롤러에 있는 경우 컨트롤러 및 펌웨어가 *캐시쓰루* 모드를 지원하는지 여부를 조사합니다. *캐시쓰루* 모드를 활성화하면 캐싱 논리를 방지할 수 있으며 일반적으로 빠른 미디어의 대기 시간이 훨씬 짧습니다.

[요구 사항 체크리스트로 돌아가기](#)

## 2.7. 오브젝트 게이트웨이에서 NVME 사용 고려 사항(선택 사항)

Red Hat Ceph Storage의 Object Gateway 기능을 사용하고 OSD 노드에 NVMe 기반 SSD 또는 SATA SSD가 있는 경우 [Ceph Object Gateway](#)의 절차에 따라 [LVM에서 NVMe를 최적으로 사용하는](#) 것이 좋습니다. 이 절차에서는 SSD에 저널 및 버킷 인덱스를 함께 배치하는 특수 설계된 Ansible 플레이북을 사용하는 방법을 설명합니다. 이렇게 하면 한 장치에 모든 저널이 있는 것보다 성능이 향상됩니다. LVM에서 NVMe를 최적으로 사용하는 방법에 대한 정보는 이 설치 가이드와 함께 참조해야 합니다.

[요구 사항 체크리스트로 돌아가기](#)

## 2.8. RED HAT CEPH STORAGE 네트워크 구성 확인

모든 RHCS(Red Hat Ceph Storage) 노드에는 공용 네트워크가 필요합니다. Ceph 클라이언트가 Ceph 모니터 및 Ceph OSD 노드에 연결할 수 있는 공용 네트워크에 네트워크 인터페이스 카드가 구성되어 있어야 합니다.

Ceph가 공용 네트워크와 별도로 네트워크에서 심부착, 피어링, 복제 및 복구를 수행할 수 있도록 클러스터 네트워크의 네트워크 인터페이스 카드가 있을 수 있습니다.

네트워크 인터페이스 설정을 구성하고 변경 사항을 지속하도록 합니다.



### 중요

Red Hat은 공용 및 사설 네트워크 모두에 단일 네트워크 인터페이스 카드를 사용하지 않는 것이 좋습니다.

## 사전 요구 사항

- 네트워크에 연결된 네트워크 인터페이스 카드입니다.

### 절차

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

1. **/etc/sysconfig/network-scripts/ifcfg-\*** 파일에 공용을 대상으로 하는 네트워크 인터페이스 카드가 있는지 확인합니다.
  - a. **BOOTPROTO** 매개변수는 고정 IP 주소에 대해 **none** 으로 설정됩니다.
  - b. **ONBOOT** 매개변수는 **yes** 로 설정해야 합니다.  
**no** 로 설정되면 재부팅 시 Ceph 스토리지 클러스터가 피어링되지 않을 수 있습니다.
  - c. IPv6 주소 지정을 사용하려면 **IPV6\_FAILURE\_FATAL** 매개변수를 제외하고 **IPV6INIT** 와 같은 IPv6 매개변수를 **yes** 로 설정해야 합니다.  
또한 Ceph 구성 파일 **/etc/ceph/ceph.conf** 를 편집하여 Ceph에 IPv6를 사용하도록 지시합니다. 그렇지 않으면 Ceph에서 IPv4를 사용합니다.

### 추가 리소스

- Red Hat Enterprise Linux 7에 대한 네트워크 인터페이스 스크립트 구성에 대한 자세한 내용은 Red Hat Enterprise Linux 7의 [네트워킹 가이드](#)의 *ifcfg 파일을 사용하여 네트워크 인터페이스 구성* 장을 참조하십시오.
- 네트워크 구성에 대한 자세한 내용은 Red Hat Ceph Storage 3 [구성 가이드](#)의 *네트워크 구성* 장을 참조하십시오.

[요구 사항 체크리스트로 돌아가기](#)

## 2.9. 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
```

2. 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
```

3. 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.

4. 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.

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

- 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
```

- 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
```

### 추가 리소스

- 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.

[요구 사항 체크리스트로 돌아가기](#)

## 2.10. 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.

- 이 사용자에게 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.

- 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
```

- 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.

[요구 사항 체크리스트로 돌아가기](#)

## 2.11. 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
```

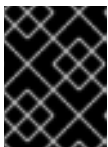
#### replace

- **\$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 가이드의 OpenSSH](#) 장

[요구 사항 체크리스트로 돌아가기](#)

## 3장. RED HAT CEPH STORAGE 배포

이 장에서는 Ansible 애플리케이션을 사용하여 Red Hat Ceph Storage 클러스터 및 메타데이터 서버 또는 Ceph Object 게이트웨이와 같은 기타 구성 요소를 배포하는 방법을 설명합니다.

- Red Hat Ceph Storage 클러스터를 설치하려면 3.2절. “Red Hat Ceph Storage 클러스터 설치” 를 참조하십시오.
- 메타데이터 서버를 설치하려면 3.4절. “메타데이터 서버 설치” 를 참조하십시오.
- **ceph-client** 역할을 설치하려면 3.5절. “Ceph 클라이언트 역할 설치” 를 참조하십시오.
- Ceph Object Gateway를 설치하려면 3.6절. “Ceph Object Gateway 설치” 를 참조하십시오.
- 다중 사이트 Ceph 개체 게이트웨이를 구성하려면 3.6.1절. “다중 사이트 Ceph Object Gateway 구성” 을 참조하십시오.
- Ansible **--limit** 옵션에 대한 자세한 내용은 3.8절. “제한 옵션 이해” 을 참조하십시오.

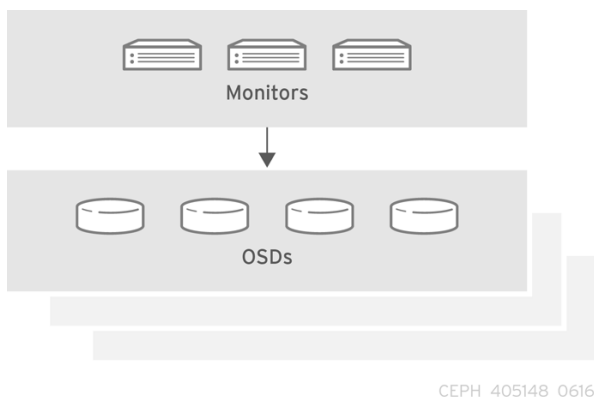
### 3.1. 사전 요구 사항

- 유효한 고객 서브스크립션을 받으십시오.
- 클러스터 노드를 준비합니다. 각 노드에서 다음을 수행합니다.
  - 노드를 CDN(Content Delivery Network)에 등록하고 서브스크립션을 연결합니다.
  - 적절한 소프트웨어 리포지토리를 활성화합니다.
  - Ansible 사용자를 생성합니다.
  - 암호 없는 SSH 액세스를 활성화합니다.
  - 선택 사항: 방화벽 구성.

### 3.2. RED HAT CEPH STORAGE 클러스터 설치

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

프로덕션 Ceph 스토리지 클러스터는 최소 3개의 모니터 호스트 및 여러 OSD 데몬이 포함된 OSD 노드 3개로 시작합니다.



#### 사전 요구 사항

- Ansible 관리 노드에서 root 계정을 사용하여 **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** 디렉터리를 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.yml.sample site.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 3.1. 일반 Ansible 설정

옵션	값	필수 항목	참고
----	---	-------	----



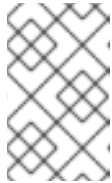
옵션	값	필수 항목	참고
<b>ceph_origin</b>	리포지토리 또는 디트로 또는 로컬	있음	리포지토리 값은 새 리포지토리를 통해 Ceph가 설치됨을 의미합니다. <b>distro</b> 값은 별도의 리포지토리 파일이 추가되지 않으며 Linux 배포에 포함된 모든 Ceph 버전을 가져옵니다. <b>local</b> 값은 Ceph 바이너리가 로컬 시스템에서 복사됨을 의미합니다.
<b>ceph_repository_type</b>	CDN 또는 iso	있음	
<b>ceph_rhcs_version</b>	3	있음	
<b>ceph_rhcs_iso_path</b>	ISO 이미지 경로	예: ISO 이미지를 사용하는 경우	
<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>	ipv6	예: IPv6 주소를 사용하는 경우	
<b>public_network</b>	Ceph 공용 네트워크의 IP 주소 및 넷마스크 또는 IPv6를 사용하는 경우 해당 IPv6 주소	있음	2.8절. "Red Hat Ceph Storage 네트워크 구성 확인"
<b>cluster_network</b>	Ceph 클러스터 네트워크의 IP 주소 및 넷마스크	아니요, 기본값은 <b>public_network</b> 입니다.	
<b>configure_firewall</b>	Ansible에서 적절한 방화벽 규칙을 설정하려고 합니다.	아니요. 값을 <b>true</b> 또는 <b>false</b> 로 설정합니다.	

**all.yml** 파일의 예는 다음과 같습니다.

```

ceph_origin: distro
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 3
monitor_interface: eth0
public_network: 192.168.0.0/24

```



참고

**ceph\_origin** 을 **all.yml** 파일에 배포하도록 설정해야 합니다. 이렇게 하면 설치 프로세스에서 올바른 다운로드 리포지토리를 사용합니다.



참고

**ceph\_rhcs\_version** 옵션을 3 으로 설정하면 최신 버전의 **Red Hat Ceph Storage 3**이 가져옵니다.



**주의**

기본적으로 **Ansible**은 설치를 다시 시작하려고 하지만 마스킹된 **firewalld** 서비스는 **Red Hat Ceph Storage** 배포가 실패할 수 있습니다. 이 문제를 해결하려면 **all.yml** 파일에서 **configure\_firewall** 옵션을 **false** 로 설정합니다. **firewalld** 서비스를 실행하는 경우 **all.yml** 파일에 **configure\_firewall** 옵션을 사용할 필요가 없습니다.

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 **OSD**를 설치합니다. 운영 체제와 **OSD** 간에 동일한 장치를 공유하면 성능에 문제가 발생합니다.

표 3.2. 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 로깅 및 키/값 데이터(BlueStore) 또는 저널 데이터(FileStore)를 저장합니다.  <b>LVM</b> 에서 논리 볼륨 관리자를 사용하여 OSD 데이터를 저장합니다.	있음	<b>osd_scenario</b> 를 사용하는 경우, <b>ceph-ansible</b> 은 장치 및 <b>dedicated_devices</b> 의 변수 수가 일치할 것으로 예상합니다. 예를 들어 <b>장치에서</b> 10 디스크를 지정하는 경우 <b>dedicated_devices</b> 에 10 개의 항목을 지정해야 합니다.
<b>osd_auto_discovery</b>	OSD를 자동으로 검색하려면 <b>true</b>	예: <b>osd_scenario: collocated</b>	장치 설정을 사용할 때는 사용할 수 없습니다.
<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.

옵션	값	필수 항목	참고
<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.
<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 [모든 NVMe 스토리지 for Red Hat Enterprise Linux에 대한 OSD Ansible 설정 구성](#) 또는 [Red Hat Ceph Storage 설치 가이드의 모든 NVMe 스토리지 for Ubuntu에 대한 OSD Ansible 설정](#) 구성을 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_NODE_NAME1
MONITOR_NODE_NAME2
MONITOR_NODE_NAME3
```

b.

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

```
[osds]
OSD_NODE_NAME1[1:10]
```



참고

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

i.

선택적으로 **devices** 및 **dedicated\_devices** 옵션을 사용하여 **OSD** 노드에서 사용할 장치를 지정합니다. 쉘표로 구분된 목록을 사용하여 여러 장치를 나열합니다.

구문

```
[osds]
CEPH_NODE_NAME devices="[DEVICE_1, 'DEVICE_2']" dedicated_devices="
[DEVICE_3, 'DEVICE_4']"
```

예제

```
[osds]
ceph-osd-01 devices="[/dev/sdc, '/dev/sdd']" dedicated_devices="[/dev/sda,
'/dev/sdb']"
ceph-osd-02 devices="[/dev/sdc, '/dev/sdd', '/dev/sde']" dedicated_devices="
[/dev/sdf, '/dev/sdg']"
```



장치를 지정하지 않으면 `osds.yml` 파일에서 `osd_auto_discovery` 옵션을 `true` 로 설정합니다.



참고

**devices** 및 **dedicated\_devices** 매개 변수를 사용하면 **OSD**에서 이름이 다른 장치를 사용하거나 장치 중 하나가 **OSD**에서 실패한 경우 유용합니다.

7.

선택적으로 모든 배포, 베어 메탈 또는 컨테이너에 대해 호스트 특정 매개 변수를 사용하려면 **host\_vars** 디렉터리에 호스트 파일을 생성하여 호스트와 관련된 모든 매개 변수를 포함합니다.

a.

스토리지 클러스터에 추가된 새 **Ceph OSD** 노드의 새 파일을 `/etc/ansible/host_vars/` 디렉터리에 생성합니다.

구문

```
touch /etc/ansible/host_vars/OSD_NODE_NAME
```

예제

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

b.

호스트별 매개 변수를 사용하여 파일을 업데이트합니다. 베어 메탈 배포에서는 **devices:** 및 **dedicated\_devices:** 섹션을 파일에 추가할 수 있습니다.

예제

```

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

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

```

8.

필요한 경우 모든 배포, 베어 메탈 또는 컨테이너의 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
```

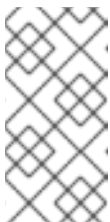
**SSD** 장치를 사용하는 경우 다음과 같이 매개변수를 편집합니다.

```
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 정의해야 하는 **class** 매개 변수가 포함되어 있으므로 **ssd** 및 **hdd OSD**가 모두 배포되지 않으면 기본 **CRUSH** 규칙이 실패합니다.



참고

또한 사용자 지정 **CRUSH** 계층 구조를 위의 단계에 설명된 대로 **host\_vars** 디렉터리의 **OSD** 파일에 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
```

9.

모든 배포, 베어 메탈 또는 컨테이너의 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>
```

10.

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

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

11. `/etc/ansible/ansible.cfg` 파일에 다음 행을 추가합니다.

```
retry_files_save_path = ~/
```

12. **root** 로서 `/var/log/ansible/` 디렉토리를 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
```

13. **Ansible** 사용자로 `/usr/share/ceph-ansible/` 디렉토리로 변경합니다.

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

14. **ceph-ansible** 플레이북을 실행합니다.

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



#### 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 30개의 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.

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

```
[root@monitor ~]# ceph health
HEALTH_OK
```

16.

**rados** 를 사용하여 클러스터가 작동하는지 확인합니다.

a.

모니터 노드에서 **8**개의 배치 그룹이 있는 테스트 풀을 생성합니다.

구문

```
[root@monitor ~]# ceph osd pool create <pool-name> <pg-number>
```

예제

```
[root@monitor ~]# ceph osd pool create test 8
```

b.

**hello-world.txt** 라는 파일을 생성합니다.

구문

```
[root@monitor ~]# vim <file-name>
```

예제

```
[root@monitor ~]# vim hello-world.txt
```

c.

**hello-world.txt** 오브젝트 이름 **hello-world** 를 사용하여 **test** 풀에 업로드합니다.

구문

```
[root@monitor ~]# rados --pool <pool-name> put <object-name> <object-file>
```

예제

```
[root@monitor ~]# rados --pool test put hello-world hello-world.txt
```

d.

**test** 풀에서 **fetch.txt** 로 **hello-world** 를 다운로드합니다.

구문

```
[root@monitor ~]# rados --pool <pool-name> get <object-name> <object-file>
```

예제

```
[root@monitor ~]# rados --pool test get hello-world fetch.txt
```

e.

**fetch.txt**의 내용을 확인합니다.

```
[root@monitor ~]# cat fetch.txt
```

출력은 다음과 같아야 합니다.

```
"Hello World!"
```



참고

클러스터 상태를 확인하는 것 외에도 **ceph-medic** 유틸리티를 사용하여 **Ceph Storage** 클러스터를 전반적으로 진단할 수 있습니다. **Red Hat Ceph Storage 3 관리 가이드**의 [Ceph Storage 클러스터 진단 장을 사용하여 ceph-medic 사용 장을 참조하십시오.](#)

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

스토리지에 **NVMe(Non-volatile Memory express)** 장치만 사용할 때 성능을 최적화하려면 각 **NVMe** 장치에서 **4**개의 **OSD**를 구성합니다. 일반적으로 하나의 **OSD**만 장치별로 구성되며 **NVMe** 장치의 처리량이 낮습니다.



참고

**SSD**와 **HDD**를 혼합하는 경우 **OSD**가 아닌 저널 또는 **block.db**에 **SSD**가 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](#)

### 3.4. 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.

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 참조하십시오.

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

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

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

추가 리소스

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

**3.5. CEPH 클라이언트 역할 설치**

**ceph-ansible** 유틸리티는 **Ceph** 구성 파일 및 관리 인증 키를 노드에 복사하는 **ceph-client** 역할을 제공합니다. 또한 이 역할을 사용하여 사용자 지정 풀 및 클라이언트를 생성할 수 있습니다.

사전 요구 사항

- 실행 중인 **Ceph** 스토리지 클러스터(특히 활성 + 정리 상태)입니다.
- **2장. Red Hat Ceph Storage 설치 요구사항**에 나열된 작업을 수행합니다.

절차

**Ansible** 관리 노드에서 다음 작업을 수행합니다.

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

```
[clients]
<client-hostname>
```

<client-hostname> 을 **ceph-client** 역할을 설치하려는 노드의 호스트 이름으로 바꿉니다.

2.

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

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

3.

**clients.yml.yml** 라는 **clients.yml.sample** 파일의 새 사본을 만듭니다.

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

4.

**group\_vars/clients.yml** 파일을 열고 다음 행의 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

**ceph-authtool --gen-print-key** 명령은 새 클라이언트 키를 생성할 수 있습니다.

5.

선택적으로 **ceph-client** 에 풀 및 클라이언트를 생성하도록 지시합니다.

a.

**clients.yml** 을 업데이트합니다.

- **user\_config** 설정의 주석을 제거하고 **true** 로 설정합니다.
- 풀 및 키 섹션의 주석을 제거하고 필요에 따라 업데이트합니다. **KnativeServing** 기능을 사용하여 사용자 정의 풀 및 클라이언트 이름을 모두 정의할 수 있습니다.

b. **osd\_pool\_default\_pg\_num** 설정을 **all.yml** 파일의 **ceph\_conf\_overrides** 섹션에 추가합니다.

```
ceph_conf_overrides:
  global:
    osd_pool_default_pg_num: <number>
```

<number> 를 기본 배치 그룹 수로 바꿉니다.

6. **Ansible Playbook**을 실행합니다.

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

추가 리소스

- [3.8절. “제한 옵션 이해”](#)

### 3.6. CEPH OBJECT GATEWAY 설치

**RADOS** 게이트웨이라고도 하는 **Ceph Object Gateway**는 **librados API** 상단에 구축된 오브젝트 스토리지 인터페이스로, **Ceph** 스토리지 클러스터에 **RESTful** 게이트웨이를 제공합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터, 바람직하게는 활성 + 클린 상태입니다.
- **Ceph Object Gateway** 노드에서 [2장. Red Hat Ceph Storage 설치 요구사항](#)에 나열된 작업을 수행합니다.

절차

**Ansible** 관리 노드에서 다음 작업을 수행합니다.

1.

게이트웨이 호스트를 **[rgws]** 섹션 아래의 **/etc/ansible/hosts** 파일에 추가하여 **Ansible**에 대한 역할을 확인합니다. 호스트에 순차적 이름 지정이 있는 경우 범위를 사용합니다. 예를 들면 다음과 같습니다.

```
[rgws]
<rgw_host_name_1>
<rgw_host_name_2>
<rgw_host_name[3..10]>
```

2.

**Ansible** 구성 디렉터리로 이동합니다.

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

3.

샘플 파일에서 **rgws.yml** 파일을 생성합니다.

```
[root@ansible ~]# cp group_vars/rgws.yml.sample group_vars/rgws.yml
```

4.

**group\_vars/rgws.yml** 파일을 열고 편집합니다. 관리자 키를 **Ceph Object Gateway** 노드에 복사하려면 **copy\_admin\_key** 옵션의 주석을 제거합니다.

```
copy_admin_key: true
```

5.

**rgws.yml** 파일은 기본 포트 **7480** 과 다른 기본 포트를 지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
ceph_rgw_civetweb_port: 80
```

6.

**all.yml** 파일은 **radosgw\_interface** 를 지정해야 합니다. 예를 들면 다음과 같습니다.

```
radosgw_interface: eth0
```

인터페이스를 지정하면 **Civetweb**이 동일한 호스트에서 여러 인스턴스를 실행할 때 다른 **Civetweb** 인스턴스와 동일한 IP 주소에 바인딩되지 않습니다.

7.

일반적으로 기본 설정을 변경하려면 **rgw.yml** 파일의 설정 주석 처리를 해제하고 적절하게 변

경합합니다. `rgw.yml` 파일에 없는 설정을 추가로 변경하려면 `all.yml` 파일에서 `ceph_conf_overrides:` 를 사용합니다. 예를 들어 DNS 서버의 호스트를 사용하여 `rgw_dns_name:` 을 설정하고 와일드카드가 S3 하위 도메인을 사용하도록 클러스터의 DNS 서버를 구성해야 합니다.

```
ceph_conf_overrides:
  client.rgw.rgw1:
    rgw_dns_name: <host_name>
    rgw_override_bucket_index_max_shards: 16
    rgw_bucket_default_quota_max_objects: 1638400
```

고급 구성 정보는 [Red Hat Ceph Storage 3 \*Ceph Object Gateway for Production\* 가이드](#) 를 참조하십시오. 고급 주제는 다음과 같습니다.

- [Ansible 그룹 구성](#)
- [스토리지 전략](#) 개발. 풀을 생성하고 구성하는 방법에 대한 자세한 내용은 [루트풀 생성, 시스템 풀 생성, 데이터 배치 전략](#) 생성 섹션을 참조하십시오.

버킷 샤딩에 대한 구성 정보는 [Bucket Sharding](#) 을 참조하십시오.

8.

`group_vars/all.yml` 파일의 `radosgw_interface` 매개변수의 주석을 제거합니다.

```
radosgw_interface: <interface>
```

*교체:*

- **Ceph Object Gateway** 노드가 수신하는 인터페이스를 사용하여 `<interface>`

자세한 내용은 `all.yml` 파일을 참조하십시오.

9.

**Ansible Playbook**을 실행합니다.

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



## 참고

**Ansible**은 각 **Ceph Object Gateway**가 실행 중인지 확인합니다.

단일 사이트 구성의 경우 **Ansible** 구성에 **Ceph Object Gateway**를 추가합니다.

다중 사이트 배포의 경우 각 영역에 대한 **Ansible** 구성이 있어야 합니다. 즉, **Ansible**은 **Ceph** 스토리지 클러스터 및 해당 영역의 게이트웨이 인스턴스를 만듭니다.

다중 사이트 클러스터에 대한 설치가 완료되면 *Red Hat Enterprise Linux*의 *오브젝트 게이트웨이 가이드*에 있는 **다중 사이트** 장으로 이동하여 다중 사이트용 클러스터를 구성하는 방법에 대해 자세히 알아보십시오.

## 추가 리소스

- [3.8절. “제한 옵션 이해”](#)
- [Red Hat Enterprise Linux용 오브젝트 게이트웨이 가이드](#)

## 3.6.1. 다중 사이트 Ceph Object Gateway 구성

**Ansible**은 다중 사이트 환경에서 **Ceph Object Gateway**의 마스터 및 보조 영역과 함께 영역 **zonegroup**을 구성합니다.

## 사전 요구 사항

- **Red Hat Ceph Storage** 클러스터를 실행하는 두 개.
- **Ceph Object Gateway** 노드에서 **Red Hat Ceph Storage** [설치 가이드](#)에 있는 **Red Hat Ceph Storage** [설치 요구 사항](#)에 나열된 작업을 수행합니다.
- 스토리지 클러스터당 하나의 **Ceph Object Gateway**를 설치하고 구성합니다.

## 절차

1.

기본 스토리지 클러스터에 대해 **Ansible** 노드에서 다음 단계를 수행합니다.

a.

시스템 키를 생성하고 다중 사이트-**keys.txt** 파일에서 출력을 캡처합니다.

```
[root@ansible ~]# echo system_access_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold
-w 20 | head -n 1) > multi-site-keys.txt
[root@ansible ~]# echo system_secret_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold
-w 40 | head -n 1) >> multi-site-keys.txt
```

b.

**Ansible** 구성 디렉터리 **/usr/share/ceph-ansible** 로 이동합니다.

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

c.

**group\_vars/all.yml** 파일을 열고 편집합니다.

**\$ZONE\_NAME**, **\$ZONE\_GROUP\_NAME**, **\$REALM\_NAME**, **\$ACCESS\_KEY**, 및 **\$SECRET\_KEY** 값을 업데이트하는 것과 함께 다음 옵션을 추가하여 다중 사이트 지원을 활성화합니다.

**Ceph Object Gateway**가 두 개 이상 마스터 영역에 있는 경우 **rgw\_multisite\_endpoints** 옵션을 설정해야 합니다. **rgw\_multisite\_endpoints** 옵션의 값은 공백 없이 쉼표로 구분된 목록입니다.

예제

```
rgw_multisite: true
rgw_zone: $ZONE_NAME
rgw_zonemaster: true
rgw_zonessecondary: false
rgw_multisite_endpoint_addr: "{{ ansible_fqdn }}"
rgw_multisite_endpoints:
http://foo.example.com:8080,http://bar.example.com:8080,http://baz.example.com:8080
rgw_zonegroup: $ZONE_GROUP_NAME
rgw_zone_user: zone.user
rgw_realm: $REALM_NAME
system_access_key: $ACCESS_KEY
system_secret_key: $SECRET_KEY
```





참고

**ansible\_fqdn** 도메인 이름은 보조 스토리지 클러스터에서 확인할 수 있어야 합니다.



참고

새 오브젝트 게이트웨이를 추가할 때 **Ansible** 플레이북을 실행하기 전에 새 오브젝트 게이트웨이의 엔드포인트 **URL**을 사용하여 **rgw\_multisite\_endpoints** 목록 끝에 추가합니다.

d.

**Ansible Playbook**을 실행합니다.

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

e.

**Ceph Object Gateway** 데몬을 다시 시작합니다.

```
[root@rgw ~]# systemctl restart ceph-radosgw@rgw.`hostname -s`
```

2.

보조 스토리지 클러스터에 대해 **Ansible** 노드에서 다음 단계를 수행합니다.

a.

**Ansible** 구성 디렉터리 **/usr/share/ceph-ansible** 로 이동합니다.

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

b.

**group\_vars/all.yml** 파일을 열고 편집합니다.

**\$ZONE\_NAME, \$ZONE\_GROUP\_NAME, \$REALM\_NAME, \$ACCESS\_KEY**, 및 **\$SECRET\_KEY** 값을 업데이트하는 것과 함께 다음 옵션을 추가하여 다중 사이트 지원을 활성화합니다. **rgw\_zone\_user, system\_access\_key, system\_secret\_key** 는 마스터 영역 구성에 사용된 것과 동일한 값이어야 합니다. **rgw\_pullhost** 옵션은 마스터 영역의 **Ceph Object Gateway**여야 합니다.

**Ceph Object Gateway**가 보조 영역에 있는 경우 **rgw\_multisite\_endpoints** 옵션을 설정해야 합니다. **rgw\_multisite\_endpoints** 옵션의 값은 공백 없이 쉼표로 구분된 목록입니다.

예제

-

```

rgw_multisite: true
rgw_zone: $ZONE_NAME
rgw_zonemaster: false
rgw_zonesecondary: true
rgw_multisite_endpoint_addr: "{{ ansible_fqdn }}"
rgw_multisite_endpoints:
http://foo.example.com:8080,http://bar.example.com:8080,http://baz.example.com:8080
rgw_zonegroup: $ZONE_GROUP_NAME
rgw_zone_user: zone.user
rgw_realm: $REALM_NAME
system_access_key: $ACCESS_KEY
system_secret_key: $SECRET_KEY
rgw_pull_proto: http
rgw_pull_port: 8080
rgw_pullhost: $MASTER_RGW_NODE_NAME

```



참고

**ansible\_fqdn** 도메인 이름은 기본 스토리지 클러스터에서 확인할 수 있어야 합니다.



참고

새 오브젝트 게이트웨이를 추가할 때 **Ansible** 플레이북을 실행하기 전에 새 오브젝트 게이트웨이의 엔드포인트 **URL**을 사용하여 **rgw\_multisite\_endpoints** 목록 끝에 추가합니다.

c. **Ansible Playbook**을 실행합니다.

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

d. **Ceph Object Gateway** 데몬을 다시 시작합니다.

```
[root@rgw ~]# systemctl restart ceph-radosgw@rgw.`hostname -s`
```

3. 마스터 및 보조 스토리지 클러스터에서 **Ansible** 플레이북을 실행하면 **활성-활성 Ceph Object Gateway** 구성이 실행됩니다.

ㄱ.

다중 사이트 **Ceph Object Gateway** 구성을 확인합니다.

a.

각 사이트의 **Ceph Monitor** 및 **Object Gateway** 노드에서 기본 및 보조 사이트를 컬링할 수 있어야 합니다.

b.

두 사이트 모두에서 **radosgw-admin** 동기화 **status** 명령을 실행합니다.

### 3.7. 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

- [3.8절. “제한 옵션 이해”](#)
- [Red Hat Enterprise Linux용 오브젝트 게이트웨이 가이드](#)

### 3.8. 제한 옵션 이해

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

**Ansible**은 인벤토리 파일의 특정 섹션에 대해 `site`, `site-docker`, `rolling_upgrade` **Ansible** 플레이북을 사용할 수 있는 **--limit** 옵션을 지원합니다.

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

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

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

#### 중요

하나의 노드에 **Ceph** 구성 요소를 배치하는 경우 **Ansible**은 **limit** 옵션을 사용하여 하나의 구성 요소 유형만 지정했음에도 불구하고 노드의 모든 구성 요소에 플레이북을 적용합니다. 예를 들어 **OSD** 및 메타데이터 서버(**MDS**)가 포함된 노드에서 **--limit osds** 옵션을 사용하여 **rolling\_update** 플레이북을 실행하면 **Ansible**에서 구성 요소, **OSD** 및 **MDSs**를 모두 업그레이드합니다.

### 3.9. 추가 리소스

- [Ansible 문서](#)

## 4장. RED HAT CEPH STORAGE 클러스터 업그레이드

이 섹션에서는 **Red Hat Ceph Storage**의 새 주요 버전 또는 마이너 버전으로 업그레이드하는 방법을 설명합니다.

- 스토리지 클러스터를 업그레이드하려면 [4.1절. “스토리지 클러스터 업그레이드”](#) 을 참조하십시오.
- **Red Hat Ceph Storage** 대시보드를 업그레이드하려면 [4.2절. “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 3에서는 `/usr/share/ceph-ansible/group_vars/` 디렉터리에 있는 **Ansible** 구성 파일에 몇 가지 변경 사항이 추가되었습니다. 특정 매개변수의 이름이 변경되거나 제거되었습니다. 따라서 버전 3으로 업그레이드한 후 **all.yml.sample** 및 **osds.yml.sample** 파일에서 새 복사본을 생성하기 전에 **all.yml** 및 **osds.yml** 파일의 백업 사본을 만듭니다. 변경 사항에 대한 자세한 내용은 [부록 H. 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 SSD에 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

언제든지 업그레이드에 실패하면 **ceph status** 명령으로 클러스터 상태를 확인하여 업그레이드 실패 이유를 확인합니다. 실패 이유 및 해결 방법이 확실하지 않은 경우 [Red Hat 지원팀](#)에 문의하십시오.



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-upgrade.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 노트 를 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
```

-



Ceph 노드가 Red Hat CDN(Content Delivery Network)에 연결되어 있지 않고 ISO 이미지를 사용하여 Red Hat Ceph Storage를 설치한 경우 로컬 리포지토리를 최신 Red Hat Ceph Storage로 업데이트합니다. 자세한 내용은 2.5절. “Red Hat Ceph Storage 리포지토리 활성화”을 참조하십시오.

- Ansible 관리 노드 및 RBD 미러링 노드에서 Red Hat Ceph Storage 2.x에서 3.x로 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
```

- rolling\_update.yml 플레이북에서 health\_osd\_check\_retries 및 health\_osd\_check\_delay 값을 각각 50 및 30 으로 변경합니다.

```
health_osd_check_retries: 50
health_osd_check_delay: 30
```

이러한 값을 설정하면 각 OSD 노드에 대해 Ansible이 최대 25분 동안 대기하고 업그레이드 프로세스를 계속하기 전에 30초마다 스토리지 클러스터 상태를 대기합니다.



#### 참고

스토리지 클러스터의 사용된 스토리지 용량에 따라 health\_osd\_check\_retries 옵션 값을 위 또는 아래로 조정합니다. 예를 들어 436TB 중 218TB를 사용하는 경우 스토리지 용량의 50%를 사용하는 경우 health\_osd\_check\_retries 옵션을 50 으로 설정합니다.

- 업그레이드하려는 클러스터에 exclusive-lock 기능을 사용하는 Ceph Block Device 이미지가 포함된 경우 모든 Ceph Block Device 사용자에게 클라이언트를 블랙리스트로 지정할 수 있는 권한이 있는지 확인하십시오.

```
ceph auth caps client.<ID> mon 'allow r, allow command "osd blacklist"' osd '<existing-OSD-user-capabilities>'
```

#### 4.1. 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 건너뛴니다. `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 건너뛴니다. **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 H. 버전 2와 3의 Ansible 변수 변경](#) 및 [3.2절. "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.x**에서 최신 버전으로 업그레이드할 때 이 단계를 건너뛴니다. **Red Hat Ceph Storage 2.x**에서 **3.x**로 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. `group_vars/all.yml` 파일에서 `upgrade_ceph_packages` 옵션의 주석을 제거하고 `True` 로 설정합니다.

```
upgrade_ceph_packages: True
```

6. `group_vars/all.yml` 파일에서 `ceph_rhcs_version` 을 3 으로 설정합니다.

```
ceph_rhcs_version: 3
```



참고

`ceph_rhcs_version` 옵션을 3 으로 설정하면 최신 버전의 **Red Hat Ceph Storage 3**이 가져옵니다.

7. `ceph_origin` 매개변수를 `group_vars/all.yml` 파일에서 `distro` 로 설정합니다.

```
ceph_origin: distro
```

8. `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
```

9. 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.

10. **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 **OSD**를 교체하면 **BlueStore** 대신 **FileStore**를 사용합니다.

- 11.

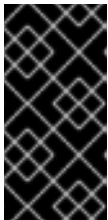
`/etc/ansible/hosts` 에 있는 **Ansible** 인벤토리 파일에서 **[mgrs]** 섹션에 **Ceph Manager(ceph-mgr)** 노드를 추가합니다. **Ceph Manager** 데몬을 모니터 노드와 분리합니다. 버전 **3.x**에서 최신 버전으로 업그레이드할 때 이 단계를 건너뛸 수 있습니다.

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

12.

**infrastructure-playbooks** 디렉터리의 **rolling\_update.yml** 을 현재 디렉터리로 복사합니다.

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



중요

**rolling\_update.yml** 플레이북과 함께 **limit ansible** 옵션을 사용하지 마십시오.

13.

`/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
```

14.

**Ansible** 사용자로 **Playbook**을 실행합니다.

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

15.

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

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

테몬을 다시 시작하십시오.

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

16.

클러스터 상태가 **OK**인지 확인합니다. **..log**를 통해 **root** 사용자로 모니터 노드에 로그인하고 **ceph status** 명령을 실행합니다.

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

1.

**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.

## 4.2. 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 **3.1**에서 **3.2**로 업그레이드되었는지 확인합니다. **4.1**을 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. 다음을 어떻게 해야 하나요?

이는 최신 데이터 센터의 어려운 스토리지 요구 사항을 충족하기 위해 **Red Hat Ceph Storage**가 수행할 수 있는 초기 단계일 뿐입니다. 다음은 다양한 주제에 대한 자세한 정보의 링크입니다.

- 성능 벤치마킹 및 성능 카운터 액세스, **Red Hat Ceph Storage 3** 관리 가이드의 [성능 벤치마크](#) 장을 참조하십시오.
- 스냅샷 생성 및 관리, **Red Hat Ceph Storage 3**용 블록 장치 가이드의 [스냅샷](#) 장을 참조하십시오.
- **Red Hat Ceph Storage** 클러스터 확장은 **Red Hat Ceph Storage 3** 관리 가이드의 [클러스터 크기](#) 관리 장을 참조하십시오.
- **Ceph** 블록 장치 미러링은 **Red Hat Ceph Storage 3**용 블록 장치 가이드의 블록 장치 [미러링](#) 장을 참조하십시오.
- 프로세스 관리, **Red Hat Ceph Storage 3** 관리 가이드의 [프로세스](#) 관리 장을 참조하십시오.
- 조정 가능한 매개변수는 **Red Hat Ceph Storage 3**의 [구성 가이드](#) 를 참조하십시오.
- **Ceph**를 **OpenStack**의 백엔드 스토리지로 사용하여 **Red Hat OpenStack Platform**의 스토리지 가이드의 [백엔드](#) 섹션을 참조하십시오.

## 부록 A. 문제 해결

## A.1. ANSIBLE은 설치를 중지하기 때문에 LESS DEVICES THAN IT EXPECTED

**Ansible** 자동화 애플리케이션은 설치 프로세스를 중지하고 다음 오류를 반환합니다.

```
- name: fix partitions gpt header or labels of the osd disks (autodiscover disks)
  shell: "sgdisk --zap-all --clear --mbrtogpt -- /dev/{{ item.0.item.key }} || sgdisk --zap-all --clear --
mbrtogpt -- /dev/{{ item.0.item.key }}"
  with_together:
    - "{{ osd_partition_status_results.results }}"
    - "{{ ansible_devices }}"
  changed_when: false
  when:
    - ansible_devices is defined
    - item.0.item.value.removable == "0"
    - item.0.item.value.partitions|count == 0
    - item.0.rc != 0
```

이것이 의미하는 것은 다음과 같습니다.

`/usr/share/ceph-ansible/group_vars/osds.yml` 파일에서 `osd_auto_discovery` 매개 변수가 `true` 로 설정된 경우 **Ansible**은 사용 가능한 모든 장치를 자동으로 탐지하고 구성합니다. 이 프로세스 중에 **Ansible**은 모든 **OSD**에서 동일한 장치를 사용할 것으로 예상합니다. 장치는 **Ansible**이 감지하는 것과 동일한 순서로 이름을 가져옵니다. **OSD** 중 하나에서 장치 중 하나가 실패하면 **Ansible**에서 실패한 장치를 감지하지 못하고 전체 설치 프로세스를 중지합니다.

## 예제 상황:

1. 3개의 **OSD** 노드(`host1,host2,host3`)는 `/dev/sdb, /dev/sdc, dev/sdd` 디스크를 사용합니다.
2. `host2` 에서 `/dev/sdc` 디스크가 실패하고 제거됩니다.
3. 다음 재부팅 시 **Ansible**은 제거된 `/dev/sdc` 디스크를 감지하지 못하고 두 디스크만 `host2, /dev/sdb` 및 `/dev/sdc`(이전 `/dev/sdd`)에 사용될 것으로 예상됩니다.
4. **Ansible**은 설치 프로세스를 중지하고 위의 오류 메시지를 반환합니다.

문제를 해결하려면 다음을 수행하십시오.

`/etc/ansible/hosts` 파일에서 오류가 발생한 디스크(위의 예 상황에서 `host 2`)에서 `OSD` 노드에서 사용하는 장치를 지정합니다.

```
[osds]
host1
host2 devices="[ '/dev/sdb', '/dev/sdc' ]"
host3
```

자세한 내용은 [3장. Red Hat Ceph Storage 배포](#) 을 참조하십시오.

## 부록 B. 수동으로 RED HAT CEPH STORAGE 설치



### 중요

Red Hat은 수동으로 배포된 클러스터 업그레이드를 지원하지 않거나 테스트하지 않습니다. 따라서 Red Hat은 Ansible을 사용하여 Red Hat Ceph Storage 3을 사용하여 새 클러스터를 배포하는 것이 좋습니다. 자세한 내용은 3장. Red Hat Ceph Storage 배포 을 참조하십시오.

YUM과 같은 명령줄 유틸리티를 사용하여 수동으로 배포된 클러스터를 설치할 수 있습니다.

모든 Ceph 클러스터에는 하나 이상의 모니터가 필요하며, 클러스터에 저장된 오브젝트의 사본으로 최소 개수의 OSD가 필요합니다. Red Hat은 프로덕션 환경에 세 개의 모니터와 최소 3개의 오브젝트 스토리지 장치(OSD)를 사용하는 것이 좋습니다.

명령줄 인터페이스를 사용하여 Ceph 스토리지 클러스터를 설치하려면 다음 단계를 수행해야 합니다.

- 초기 모니터 노드를 부트스트래핑합니다.
- Ceph Manager 데몬 설치.
- OSD(오브젝트 스토리지 장치) 노드 추가.

### B.1. 사전 요구 사항

#### Red Hat Ceph Storage의 네트워크 시간 프로토콜 구성

모든 Ceph 모니터 및 OSD 노드는 NTP(Network Time Protocol)를 구성해야 합니다. Ceph 노드가 NTP 피어인지 확인합니다. NTP는 클럭 드리프트에서 발생하는 문제를 선점하는 데 도움이 됩니다.



### 참고

Ansible을 사용하여 Red Hat Ceph Storage 클러스터를 배포할 때 Ansible은 자동으로 NTP를 설치, 구성 및 활성화합니다.

### 사전 요구 사항

- 유효한 시간 소스에 대한 네트워크 액세스.

프로시저: RHCS의 네트워크 시간 프로토콜 구성

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

1. **ntp** 패키지를 설치합니다.

```
# yum install ntp
```

2. 재부팅 시 **NTP** 서비스를 시작하고 활성화합니다.

```
# systemctl start ntpd
# systemctl enable ntpd
```

3. **NTP**가 클럭을 올바르게 동기화하는지 확인합니다.

```
$ ntpq -p
```

추가 리소스

- [Red Hat Enterprise Linux 7용 시스템 관리자 가이드의 ntpd를 사용하여 NTP 구성 장.](#)

## Bootstrapping 모니터링

모니터를 부트 스트랩하고 **Ceph** 스토리지 클러스터를 확장하여 다음 데이터가 필요합니다.

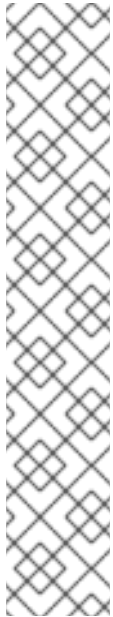
고유 식별자

파일 시스템 식별자(**fsid**)는 클러스터의 고유 식별자입니다. **fsid** 는 **Ceph** 스토리지 클러스터가 기본적으로 **Ceph** 파일 시스템에 사용될 때 사용되었습니다. **Ceph**는 이제 기본 인터페이스, 블록 장치 및 오브젝트 스토리지 게이트웨이 인터페이스를 지원하므로 **fsid** 는 약간 **misnomer**입니다.

클러스터 이름

**Ceph** 클러스터에는 공백이 없는 간단한 문자열인 클러스터 이름이 있습니다. 기본 클러스터 이름은 **ceph** 이지만 다른 클러스터 이름을 지정할 수 있습니다. 기본 클러스터 이름을 재정의하는 것은 여러 클러스터에서 작업할 때 특히 유용합니다.

다중 사이트 아키텍처에서 여러 클러스터를 실행하는 경우 클러스터 이름(예: **us-west**)은 현재 명령줄 세션에 대한 클러스터를 식별합니다.



참고

명령줄 인터페이스에서 클러스터 이름을 확인하려면 클러스터 이름(예: **ceph.conf,us-west.conf,us-east.conf** 등)을 사용하여 **Ceph** 구성 파일을 지정합니다.

예제:

```
# ceph --cluster us-west.conf ...
```

모니터링 이름

클러스터 내의 각 모니터 인스턴스마다 고유한 이름이 있습니다. 일반적으로 **Ceph Monitor** 이름은 노드 이름입니다. **Red Hat**은 노드당 하나의 **Ceph Monitor**를 권장하고 **Ceph OSD** 데몬을 함께 배치하지 않는 것이 좋습니다. 짧은 노드 이름을 검색하려면 **hostname -s** 명령을 사용합니다.

모니터링 맵

초기 모니터를 부트 스트랩하려면 모니터 맵을 생성해야 합니다. 모니터 맵에는 다음이 필요합니다.

- 파일 시스템 식별자(**fsid**)
- 클러스터 이름 또는 **ceph**의 기본 클러스터 이름이 사용됩니다.
- 하나 이상의 호스트 이름과 **IP** 주소.

키 링 모니터링

모니터는 시크릿 키를 사용하여 서로 통신합니다. **Monitor** 시크릿 키를 사용하여 인증 키를 생성하고 초기 모니터를 부트스트랩할 때 제공해야 합니다.

관리자 키 링

**ceph** 명령줄 인터페이스 유틸리티를 사용하려면 **client.admin** 사용자를 생성하고 인증 키를 생성합니다. 또한 **Monitor** 인증 키에 **client.admin** 사용자를 추가해야 합니다.

예상 요구 사항은 **Ceph** 구성 파일 생성을 보장하지 않습니다. 하지만 모범 사례로 **Ceph** 구성 파일을 만들고 **fsid**, **mon initial members** 및 **mon host** 설정으로 채우는 것이 좋습니다.

런타임 시 모든 **Monitor** 설정을 가져오고 설정할 수 있습니다. 그러나 **Ceph** 구성 파일에는 기본값을 재정의하는 설정만 포함될 수 있습니다. **Ceph** 구성 파일에 설정을 추가하면 이러한 설정이 기본 설정을 재정의합니다. **Ceph** 구성 파일에서 이러한 설정을 유지 관리하면 클러스터를 보다 쉽게 유지 관리할 수 있습니다.

초기 모니터를 부트 스트랩하려면 다음 단계를 수행합니다.

1. **Red Hat Ceph Storage 3 Monitor** 리포지토리를 활성화합니다.

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

2. 초기 모니터 노드에서 **root** 로 **ceph-mon** 패키지를 설치합니다.

```
# yum install ceph-mon
```

3. **root** 로서 **/etc/ceph/** 디렉터리에 **Ceph** 구성 파일을 만듭니다. 기본적으로 **Ceph**는 **ceph.conf** 를 사용합니다. 여기서 **ceph** 는 클러스터 이름을 반영합니다.

구문

```
# touch /etc/ceph/<cluster_name>.conf
```

예제

```
# touch /etc/ceph/ceph.conf
```

4.

**root** 로 클러스터의 고유 식별자를 생성하고 **Ceph** 구성 파일의 **[global]** 섹션에 고유 식별자를 추가합니다.

구문

```
# echo "[global]" > /etc/ceph/<cluster_name>.conf
# echo "fsid = `uuidgen`" >> /etc/ceph/<cluster_name>.conf
```

예제

```
# echo "[global]" > /etc/ceph/ceph.conf
# echo "fsid = `uuidgen`" >> /etc/ceph/ceph.conf
```

5.

현재 **Ceph** 구성 파일을 확인합니다.

```
$ cat /etc/ceph/ceph.conf
[global]
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
```

6.

**root** 로서 초기 **Monitor**를 **Ceph** 구성 파일에 추가합니다.

구문

```
# echo "mon initial members = <monitor_host_name>[,<monitor_host_name>]" >>
/etc/ceph/<cluster_name>.conf
```



## 예제

```
# echo "mon initial members = node1" >> /etc/ceph/ceph.conf
```

7.

**root** 로서 초기 **Monitor**의 IP 주소를 **Ceph** 구성 파일에 추가합니다.

## 구문

```
# echo "mon host = <ip-address>[,<ip-address>]" >> /etc/ceph/<cluster_name>.conf
```

## 예제

```
# echo "mon host = 192.168.0.120" >> /etc/ceph/ceph.conf
```



## 참고

**IPv6** 주소를 사용하려면 **ms bind ipv6** 옵션을 **true** 로 설정합니다. 자세한 내용은 **Red Hat Ceph Storage 3** 구성 가이드의 **Bind** 섹션을 참조하십시오.

8.

**root** 로 클러스터에 대한 인증 키를 생성하고 **Monitor** 시크릿 키를 생성합니다.

## 구문

```
# ceph-authtool --create-keyring /tmp/<cluster_name>.mon.keyring --gen-key -n mon. --cap mon '<capabilities>'
```

## 예제

```
# ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --cap mon 'allow *'  
creating /tmp/ceph.mon.keyring
```

9.

**root** 로 관리자 인증 키를 생성하고 **<cluster\_name>.client.admin.keyring** 사용자를 생성하고 사용자를 인증 키에 추가합니다.

## 구문

```
# ceph-authtool --create-keyring /etc/ceph/<cluster_name>.client.admin.keyring --gen-key -n  
client.admin --set-uid=0 --cap mon '<capabilities>' --cap osd '<capabilities>' --cap mds  
'<capabilities>'
```

## 예제

```
# ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n  
client.admin --set-uid=0 --cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow'  
creating /etc/ceph/ceph.client.admin.keyring
```

10.

**루트** 로서 **<cluster\_name>.client.admin.keyring** 키를 **<cluster\_name>.mon.keyring**에 추가합니다.

## 구문

```
# ceph-authtool /tmp/<cluster_name>.mon.keyring --import-keyring
/etc/ceph/<cluster_name>.client.admin.keyring
```

#### 예제

```
# ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring
importing contents of /etc/ceph/ceph.client.admin.keyring into /tmp/ceph.mon.keyring
```

11.

모니터 맵을 생성합니다. 노드 이름, IP 주소 및 **fsid** 를 초기 모니터로 사용하여 지정하고 **/tmp/monmap** 으로 저장합니다.

#### 구문

```
$ monmaptool --create --add <monitor_host_name> <ip-address> --fsid <uuid>
/tmp/monmap
```

#### 예제

```
$ monmaptool --create --add node1 192.168.0.120 --fsid a7f64266-0894-4f1e-a635-
d0aeaca0e993 /tmp/monmap
monmaptool: monmap file /tmp/monmap
monmaptool: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993
monmaptool: writing epoch 0 to /tmp/monmap (1 monitors)
```

12.

초기 **Monitor** 노드에서 **root** 로 기본 데이터 디렉토리를 생성합니다.

#### 구문

```
# mkdir /var/lib/ceph/mon/<cluster_name>-<monitor_host_name>
```

#### 예제

```
# mkdir /var/lib/ceph/mon/ceph-node1
```

13.

**root** 로 초기 **Monitor** 데몬을 **Monitor** 맵 및 인증 키로 채웁니다.

#### 구문

```
# ceph-mon [--cluster <cluster_name>] --mkfs -i <monitor_host_name> --monmap  
/tmp/monmap --keyring /tmp/<cluster_name>.mon.keyring
```

#### 예제

```
# ceph-mon --mkfs -i node1 --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring  
ceph-mon: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993  
ceph-mon: created monfs at /var/lib/ceph/mon/ceph-node1 for mon.node1
```

14.

현재 **Ceph** 구성 파일을 확인합니다.

```
# cat /etc/ceph/ceph.conf  
[global]
```

```
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
mon_initial_members = node1
mon_host = 192.168.0.120
```

다양한 Ceph 구성 설정에 대한 자세한 내용은 **Red Hat Ceph Storage 3 구성 가이드**를 참조하십시오. Ceph 구성 파일의 다음 예제에는 가장 일반적인 구성 설정이 나열되어 있습니다.

#### 예제

```
[global]
fsid = <cluster-id>
mon initial members = <monitor_host_name>[, <monitor_host_name>]
mon host = <ip-address>[, <ip-address>]
public network = <network>[, <network>]
cluster network = <network>[, <network>]
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
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 ppg num = <n>
osd crush chooseleaf type = <n>
```

15. **root** 로서 완료된 파일을 생성합니다.

#### 구문

```
# touch /var/lib/ceph/mon/<cluster_name>-<monitor_host_name>/done
```

#### 예제

```
# touch /var/lib/ceph/mon/ceph-node1/done
```

16.

**root** 로서 새로 생성된 디렉터리 및 파일에 대한 소유자 및 그룹 권한을 업데이트합니다.

구문

```
# chown -R <owner>:<group> <path_to_directory>
```

예제

```
# chown -R ceph:ceph /var/lib/ceph/mon
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown ceph:ceph /etc/ceph/ceph.client.admin.keyring
# chown ceph:ceph /etc/ceph/ceph.conf
# chown ceph:ceph /etc/ceph/rbdmap
```



참고

**Ceph Monitor** 노드가 **OpenStack** 컨트롤러 노드와 함께 배치되는 경우 **Glance** 및 **Cinder** 인증 키 파일을 각각 **glance** 및 **cinder** 에서 소유해야 합니다. 예를 들면 다음과 같습니다.

```
# ls -l /etc/ceph/
...
-rw-----. 1 glance glance    64 <date> ceph.client.glance.keyring
-rw-----. 1 cinder cinder    64 <date> ceph.client.cinder.keyring
...
```

17.

사용자 지정 이름이 있는 스토리지 클러스터의 경우 **root** 로 다음 행을 추가합니다.

구문

```
# echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
```

예제

```
# echo "CLUSTER=test123" >> /etc/sysconfig/ceph
```

18.

**root** 로 초기 **Monitor** 노드에서 **ceph-mon** 프로세스를 시작하고 활성화합니다.

구문

```
# systemctl enable ceph-mon.target  
# systemctl enable ceph-mon@<monitor_host_name>  
# systemctl start ceph-mon@<monitor_host_name>
```

예제

```
# systemctl enable ceph-mon.target  
# systemctl enable ceph-mon@node1  
# systemctl start ceph-mon@node1
```

19.

**root** 로서 모니터 데몬이 실행 중인지 확인합니다.

구문

```
# systemctl status ceph-mon@<monitor_host_name>
```

## 예제

```
# systemctl status ceph-mon@node1
● ceph-mon@node1.service - Ceph cluster monitor daemon
   Loaded: loaded (/usr/lib/systemd/system/ceph-mon@.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2018-06-27 11:31:30 PDT; 5min ago
   Main PID: 1017 (ceph-mon)
   CGroup: /system.slice/system-ceph\x2dmon.slice/ceph-mon@node1.service
           └─1017 /usr/bin/ceph-mon -f --cluster ceph --id node1 --setuser ceph --setgroup ceph

Jun 27 11:31:30 node1 systemd[1]: Started Ceph cluster monitor daemon.
Jun 27 11:31:30 node1 systemd[1]: Starting Ceph cluster monitor daemon...
```

스토리지 클러스터에 **Red Hat Ceph Storage** 모니터를 더 추가하려면 **Red Hat Ceph Storage 3** 관리 가이드의 [모니터 추가](#) 섹션을 참조하십시오.

## B.2. 수동으로 CEPH MANAGER 설치

일반적으로 **Ansible** 자동화 유틸리티는 **Red Hat Ceph Storage** 클러스터를 배포할 때 **Ceph Manager** 데몬(**ceph-mgr**)을 설치합니다. 그러나 **Ansible**을 사용하여 **Red Hat Ceph Storage**를 관리하지 않는 경우 **Ceph Manager**를 수동으로 설치할 수 있습니다. **Red Hat**은 동일한 노드에서 **Ceph Manager** 및 **Ceph Monitor** 데몬을 배치하는 것이 좋습니다.

### 사전 요구 사항

- 작동하는 **Red Hat Ceph Storage** 클러스터
- **root** 또는 **sudo** 액세스



- **rhel-7-server-rhceph-3-mon-els-rpms** 리포지토리 활성화
- 방화벽이 사용되는 경우 공용 네트워크에서 포트 **6800-7300**

## 절차

**ceph-mgr** 가 배포되는 노드에서 및 **root** 사용자로 또는 **sudo** 유틸리티를 사용하여 다음 명령을 사용합니다.

1. **ceph-mgr** 패키지를 설치합니다.

```
[root@node1 ~]# yum install ceph-mgr
```

2. **/var/lib/ceph/mgr/ceph-호스트 이름/** 디렉터리를 생성합니다.

```
mkdir /var/lib/ceph/mgr/ceph-hostname
```

**hostname** 을 **ceph-mgr** 데몬이 배포될 노드의 호스트 이름으로 교체합니다. 예를 들면 다음과 같습니다.

```
[root@node1 ~]# mkdir /var/lib/ceph/mgr/ceph-node1
```

3. 새로 생성된 디렉터리에서 **ceph-mgr** 데몬에 대한 인증 키를 생성합니다.

```
[root@node1 ~]# ceph auth get-or-create mgr.`hostname -s` mon 'allow profile mgr' osd 'allow *' mds 'allow *' -o /var/lib/ceph/mgr/ceph-node1/keyring
```

4. **/var/lib/ceph/mgr/** 디렉터리의 소유자 및 그룹을 **ceph:ceph** 로 변경합니다.

```
[root@node1 ~]# chown -R ceph:ceph /var/lib/ceph/mgr
```

5. **ceph-mgr** 대상을 활성화합니다.

```
[root@node1 ~]# systemctl enable ceph-mgr.target
```

6. **ceph-mgr** 인스턴스를 활성화하고 시작합니다.

```
systemctl enable ceph-mgr@hostname
systemctl start ceph-mgr@hostname
```

**hostname** 을 **ceph-mgr** 이 배포할 노드의 호스트 이름으로 교체합니다. 예를 들면 다음과 같습니다.

```
[root@node1 ~]# systemctl enable ceph-mgr@node1
[root@node1 ~]# systemctl start ceph-mgr@node1
```

7. **ceph-mgr** 데몬이 시작되었는지 확인합니다.

```
ceph -s
```

출력에는 **services:** 섹션 아래의 다음 행과 유사한 행이 포함됩니다.

```
mgr: node1(active)
```

8. 현재 활성화 데몬이 실패하는 경우 더 많은 **ceph-mgr** 데몬을 설치하여 대기 데몬으로 작동합니다.

#### 추가 리소스

- 

[Red Hat Ceph Storage 설치 요구사항](#)

#### OSD 부팅

초기 모니터가 실행되면 **OSD**(오브젝트 스토리지 장치)를 추가할 수 있습니다. 오브젝트 사본 수를 처리하기에 충분한 **OSD**가 있을 때까지 클러스터는 활성화 + 클린 상태에 도달할 수 없습니다.

오브젝트의 기본 사본 수는 **3**입니다. 최소 **3**개의 **OSD** 노드가 필요합니다. 그러나 오브젝트의 두 사본만 원하는 경우 두 개의 **OSD** 노드만 추가한 다음, **Ceph** 구성 파일에서 **osd** 풀 기본 크기 및 **osd** 풀의 기본 **min size** 설정을 업데이트합니다.

자세한 내용은 **Red Hat Ceph Storage 3 구성 가이드의 OSD 구성 참조** 섹션을 참조하십시오.

초기 모니터를 부트 스트랩하면 클러스터에 기본 **CRUSH** 맵이 있습니다. 그러나 **CRUSH** 맵에는 **Ceph** 노드에 매핑된 **Ceph OSD** 데몬이 없습니다.

클러스터에 **OSD**를 추가하고 기본 **CRUSH** 맵을 업데이트하려면 각 **OSD** 노드에서 다음을 실행합니다.

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

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

2. **root** 로 **Ceph OSD** 노드에 **ceph-osd** 패키지를 설치합니다.

```
# yum install ceph-osd
```

3. 초기 **Monitor** 노드에서 **OSD** 노드로 **Ceph** 구성 파일 및 관리 인증 키 파일을 복사합니다.

구문

```
# scp <user_name>@<monitor_host_name>:<path_on_remote_system>  
<path_to_local_file>
```

예제

```
# scp root@node1:/etc/ceph/ceph.conf /etc/ceph  
# scp root@node1:/etc/ceph/ceph.client.admin.keyring /etc/ceph
```

4. **OSD**에 대한 **UUID(Universally Unique Identifier)**를 생성합니다.

```
$ uuidgen  
b367c360-b364-4b1d-8fc6-09408a9cda7a
```

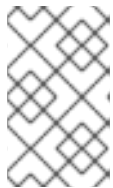
5. **root** 로서 **OSD** 인스턴스를 생성합니다.

구문

```
# ceph osd create <uuid> [<osd_id>]
```

예제

```
# ceph osd create b367c360-b364-4b1d-8fc6-09408a9cda7a  
0
```



참고

이 명령은 후속 단계에 필요한 **OSD** 번호 식별자를 출력합니다.

6. 루트 로서 새 **OSD**에 대한 기본 디렉토리를 만듭니다.

구문

```
# mkdir /var/lib/ceph/osd/<cluster_name>-<osd_id>
```

예제

```
# mkdir /var/lib/ceph/osd/ceph-0
```

7.

루트 로서 **OSD**로 사용할 드라이브를 준비한 후 방금 생성한 디렉터리에 마운트합니다. **Ceph** 데이터 및 저널용 파티션을 만듭니다. 저널 및 데이터 파티션은 동일한 디스크에 있을 수 있습니다. 이 예에서는 **15GB** 디스크를 사용합니다.

구문

```
# parted <path_to_disk> mklabel gpt
# parted <path_to_disk> mkpart primary 1 10000
# mkfs -t <fstype> <path_to_partition>
# mount -o noatime <path_to_partition> /var/lib/ceph/osd/<cluster_name>-<osd_id>
# echo "<path_to_partition> /var/lib/ceph/osd/<cluster_name>-<osd_id> xfs
defaults,noatime 1 2" >> /etc/fstab
```

예제

```
# parted /dev/sdb mklabel gpt
# parted /dev/sdb mkpart primary 1 10000
# parted /dev/sdb mkpart primary 10001 15000
# mkfs -t xfs /dev/sdb1
# mount -o noatime /dev/sdb1 /var/lib/ceph/osd/ceph-0
# echo "/dev/sdb1 /var/lib/ceph/osd/ceph-0 xfs defaults,noatime 1 2" >> /etc/fstab
```

8.

루트 로서 **OSD** 데이터 디렉터리를 초기화합니다.

구문

```
# ceph-osd -i <osd_id> --mkfs --mkkey --osd-uuid <uuid>
```

예제

```
# ceph-osd -i 0 --mkfs --mkkey --osd-uuid b367c360-b364-4b1d-8fc6-09408a9cda7a
... auth: error reading file: /var/lib/ceph/osd/ceph-0/keyring: can't open /var/lib/ceph/osd/ceph-0/keyring: (2) No such file or directory
... created new key in keyring /var/lib/ceph/osd/ceph-0/keyring
```



참고

**--mkkey** 옵션을 사용하여 **ceph-osd** 를 실행하기 전에 디렉터리가 비어 있어야 합니다. 사용자 지정 클러스터 이름이 있는 경우 **ceph-osd** 유틸리티에 **--cluster** 옵션이 필요합니다.

9.

**root** 로 **OSD** 인증 키를 등록합니다. 클러스터 이름이 **ceph** 와 다른 경우 클러스터 이름을 대신 삽입합니다.

구문

```
# ceph auth add osd.<osd_id> osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/<cluster_name>-<osd_id>/keyring
```

예제

```
# ceph auth add osd.0 osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-0/keyring
added key for osd.0
```

10.

**root** 로서 **OSD** 노드를 **CRUSH** 맵에 추가합니다.

그림

---

구문

```
# ceph [--cluster <cluster_name>] osd crush add-bucket <host_name> host
```

예제

```
# ceph osd crush add-bucket node2 host
```

11. **root** 로 **OSD** 노드를 기본 **CRUSH** 트리 아래에 배치합니다.

구문

```
# ceph [--cluster <cluster_name>] osd crush move <host_name> root=default
```

예제

```
# ceph osd crush move node2 root=default
```

12. **root** 로서 **OSD** 디스크를 **CRUSH** 맵에 추가합니다.

구문

```
# ceph [--cluster <cluster_name>] osd crush add osd.<osd_id> <weight> [<bucket_type>=
<bucket-name> ...]
```

### 예제

```
# ceph osd crush add osd.0 1.0 host=node2
add item id 0 name 'osd.0' weight 1 at location {host=node2} to crush map
```



### 참고

**CRUSH** 맵을 컴파일하고 장치 목록에 **OSD**를 추가할 수도 있습니다. **OSD** 노드를 버킷으로 추가한 다음 장치를 **OSD** 노드의 항목으로 추가하고 **OSD**에 가중치를 할당하고 **CRUSH** 맵을 다시 컴파일하고 **CRUSH** 맵을 설정합니다. 자세한 내용은 **Red Hat Ceph Storage 3용 스토리지 전략 가이드**의 **CRUSH 맵 편집** 섹션을 참조하십시오.

13.

**root**로서 새로 생성된 디렉터리 및 파일에 대한 소유자 및 그룹 권한을 업데이트합니다.

### 구문

```
# chown -R <owner>:<group> <path_to_directory>
```

### 예제

```
# chown -R ceph:ceph /var/lib/ceph/osd
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```



14.

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

구문

```
# echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
```

예제

```
# echo "CLUSTER=test123" >> /etc/sysconfig/ceph
```

15.

**OSD** 노드는 **Ceph** 스토리지 클러스터 구성에 있습니다. 그러나 **OSD** 데몬은 다운 되어 에서입니다. 새 **OSD**는 데이터를 받기 전에 실행해야 합니다. **root** 로 **OSD** 프로세스를 활성화하고 시작합니다.

구문

```
# systemctl enable ceph-osd.target  
# systemctl enable ceph-osd@<osd_id>  
# systemctl start ceph-osd@<osd_id>
```

예제

```
# systemctl enable ceph-osd.target
# systemctl enable ceph-osd@0
# systemctl start ceph-osd@0
```

**OSD** 데몬을 시작하면 **up** 및 **입니**다.

이제 모니터와 일부 **OSD**가 실행 중입니다. 다음 명령을 실행하여 배치 그룹 피어를 확인할 수 있습니다.

```
$ ceph -w
```

**OSD** 트리를 보려면 다음 명령을 실행합니다.

```
$ ceph osd tree
```

예제

ID	WEIGHT	TYPE	NAME	UP/DOWN	REWEIGHT	PRIMARY-AFFINITY
-1	2	root	default			
-2	2	host	node2			
0	1	osd	osd.0	up	1	1
-3	1	host	node3			
1	1	osd	osd.1	up	1	1

스토리지 클러스터에 새 **OSD**를 추가하여 스토리지 용량을 확장하려면 **Red Hat Ceph Storage 3 관리 가이드**의 **OSD 추가** 섹션을 참조하십시오.

## 부록 C. CEPH 명령줄 인터페이스 설치

**Ceph CLI**(명령줄 인터페이스)를 사용하면 관리자가 **Ceph** 관리 명령을 실행할 수 있습니다. **CLI**는 **ceph-common** 패키지에서 제공하며 다음 유틸리티를 포함합니다.

- **Ceph**
- **ceph-authtool**
- **ceph-dencoder**
- **rados**

## 사전 요구 사항

- 실행 중인 **Ceph** 스토리지 클러스터(특히 활성 + 정리 상태)입니다.

## 절차

1. 클라이언트 노드에서 **Red Hat Ceph Storage 3** 툴 리포지토리를 활성화합니다.

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

2. 클라이언트 노드에서 **ceph-common** 패키지를 설치합니다.

```
# yum install ceph-common
```

3. 초기 모니터 노드에서 **Ceph** 구성 파일(이 경우 **ceph.conf**) 및 관리 인증 키를 클라이언트 노드에 복사합니다.

## 구문

```
# scp /etc/ceph/<cluster_name>.conf <user_name>@<client_host_name>:/etc/ceph/  
# scp /etc/ceph/<cluster_name>.client.admin.keyring  
<user_name>@<client_host_name>:/etc/ceph/
```

#### 예제

```
# scp /etc/ceph/ceph.conf root@node1:/etc/ceph/  
# scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

**<client\_host\_name>** 을 클라이언트 노드의 호스트 이름으로 바꿉니다.

## 부록 D. 수동으로 CEPH 블록 장치 설치

다음 절차에서는 썬 프로비저닝된 **Ceph** 블록 장치를 설치하고 마운트하는 방법을 보여줍니다.



## 중요

**Ceph** 블록 장치는 **Ceph Monitor** 및 **OSD** 노드에서 별도의 노드에 배포해야 합니다. 동일한 노드에서 커널 클라이언트 및 커널 서버 데몬을 실행하면 커널 교착 상태가 발생할 수 있습니다.

## 사전 요구 사항

- [부록 C. Ceph 명령줄 인터페이스 설치](#) 섹션에 나열된 작업을 수행해야 합니다.
- **Ceph** 블록 장치를 **QEMU**를 사용하는 가상 머신(**VM**)의 백엔드로 사용하는 경우 기본 파일 설명자를 늘립니다. 자세한 내용은 [Ceph - VM을 RBD 디스크 지식베이스에 대량의 데이터를 전송할 때](#) 참조하십시오.

## 절차

1. **OSD** 노드의 파일에 대한 전체 권한을 사용하여 **client.rbd** 라는 **Ceph** 블록 장치 사용자를 만들고 결과를 인증 키 파일로 출력합니다.

```
ceph auth get-or-create client.rbd mon 'profile rbd' osd 'profile rbd pool=<pool_name>' \
-o /etc/ceph/rbd.keyring
```

**<pool\_name>** 을 **client.rbd** 가 액세스할 수 있는 풀 이름으로 바꿉니다(예: **rbd**).

```
# ceph auth get-or-create \
client.rbd mon 'allow r' osd 'allow rwx pool=rbd' \
-o /etc/ceph/rbd.keyring
```

**사용자** 생성에 대한 자세한 내용은 [Red Hat Ceph Storage 3 관리 가이드](#)의 사용자 관리 섹션을 참조하십시오.

2. 블록 장치 이미지를 생성합니다.

```

rbd create <image_name> --size <image_size> --pool <pool_name> \
--name client.rbd --keyring /etc/ceph/rbd.keyring

```

예를 들어 **<image\_name>**, **<image\_size>**, **<pool\_name>** 을 지정합니다.

```

$ rbd create image1 --size 4096 --pool rbd \
--name client.rbd --keyring /etc/ceph/rbd.keyring

```



### 주의

기본 **Ceph** 구성에는 다음과 같은 **Ceph** 블록 장치 기능이 포함됩니다.

- 계층 지정
- **exclusive-lock**
- **object-map**
- **deep-flatten**
- **fast-diff**

커널 **RBD(octetsd)** 클라이언트를 사용하는 경우 **Red Hat Enterprise Linux 7.3**에 포함된 현재 커널 버전이 객체 맵, 딥-플러튼 및 **fast-diff** 를 지원하지 않기 때문에 블록 장치 이미지를 매핑할 수 없습니다.

이 문제를 해결하려면 지원되지 않는 기능을 비활성화합니다. 다음을 수행하려면 다음 옵션 중 하나를 사용합니다.

- 지원되지 않는 기능을 동적으로 비활성화합니다.

```

rbd feature disable <image_name> <feature_name>

```

예를 들면 다음과 같습니다.

```
# rbd feature disable image1 object-map deep-flatten fast-diff
```

- **rbd create** 명령에 **--image-feature layering** 옵션을 사용하여 새로 생성된 블록 장치 이미지에서 계층화만 활성화합니다.

- **Ceph** 구성 파일에서 기능을 기본값으로 비활성화합니다.

```
rbd_default_features = 1
```

이는 **Red Hat Ceph Storage 3 릴리스 노트**에서 [알려진 문제](#) 장을 참조하십시오.

이러한 모든 기능은 사용자 공간 **RBD** 클라이언트를 사용하여 블록 장치 이미지에 액세스하는 사용자에게 적용됩니다.

3.

새로 생성된 이미지를 블록 장치에 매핑합니다.

```
rbd map <image_name> --pool <pool_name>\
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

예를 들면 다음과 같습니다.

```
# rbd map image1 --pool rbd --name client.rbd \
--keyring /etc/ceph/rbd.keyring
```



## 중요

커널 블록 장치는 현재 **CRUSH** 맵에서 기존 **straw** 버킷 알고리즘만 지원합니다. **CRUSH** 튜닝 가능 항목을 최적으로 설정한 경우 기존 또는 이전 주요 릴리스로 설정해야 합니다. 그렇지 않으면 이미지를 매핑할 수 없습니다.

또는 **straw2** 를 **CRUSH** 맵의 **straw** 로 바꿉니다. 자세한 내용은 **Red Hat Ceph Storage 3용 스토리지 전략 가이드**의 **CRUSH 맵 편집** 장을 참조하십시오.

4. 파일 시스템을 생성하여 블록 장치를 사용합니다.

```
mkfs.ext4 -m5 /dev/rbd/<pool_name>/<image_name>
```

풀 이름과 이미지 이름을 지정합니다. 예를 들면 다음과 같습니다.

```
# mkfs.ext4 -m5 /dev/rbd/rbd/image1
```

이 작업은 몇 분 정도 걸릴 수 있습니다.

5. 새로 생성된 파일 시스템을 마운트합니다.

```
mkdir <mount_directory>
mount /dev/rbd/<pool_name>/<image_name> <mount_directory>
```

예를 들면 다음과 같습니다.

```
# mkdir /mnt/ceph-block-device
# mount /dev/rbd/rbd/image1 /mnt/ceph-block-device
```

자세한 내용은 **Red Hat Ceph Storage 3**의 **블록 장치 가이드** 를 참조하십시오.



## 부록 E. 수동으로 CEPH OBJECT GATEWAY 설치

**RADOS** 게이트웨이라고도 하는 **Ceph** 개체 게이트웨이는 **librados** API 상단에 빌드된 오브젝트 스토리지 인터페이스로, 애플리케이션을 **Ceph** 스토리지 클러스터에 **RESTful** 게이트웨이를 제공합니다.

### 사전 요구 사항

- 실행 중인 **Ceph** 스토리지 클러스터(특히 활성 + 정리 상태)입니다.
- [2장. Red Hat Ceph Storage 설치 요구사항](#)에 나열된 작업을 수행합니다.

### 절차

1. **Red Hat Ceph Storage 3** 툴 리포지토리를 활성화합니다.

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

2. **Object Gateway** 노드에서 **ceph-radosgw** 패키지를 설치합니다.

```
# yum install ceph-radosgw
```

3. 초기 모니터 노드에서 다음 단계를 수행합니다.

- a. 다음과 같이 **Ceph** 구성 파일을 업데이트합니다.

```
[client.rgw.<obj_gw_hostname>]
host = <obj_gw_hostname>
rgw frontends = "civetweb port=80"
rgw dns name = <obj_gw_hostname>.example.com
```

여기서 **<obj\_gw\_hostname>** 은 게이트웨이 노드의 짧은 호스트 이름입니다. 짧은 호스트 이름을 보려면 **hostname -s** 명령을 사용합니다.

- b. 업데이트된 구성 파일을 **Ceph** 스토리지 클러스터의 새 **Object Gateway** 노드 및 기타 모든 노드에 복사합니다.

구문

```
# scp /etc/ceph/<cluster_name>.conf <user_name>@<target_host_name>:/etc/ceph
```

예제

```
# scp /etc/ceph/ceph.conf root@node1:/etc/ceph/
```

c.

**<cluster\_name>.client.admin.keyring** 파일을 새 **Object Gateway** 노드에 복사합니다.

구문

```
# scp /etc/ceph/<cluster_name>.client.admin.keyring  
<user_name>@<target_host_name>:/etc/ceph/
```

예제

```
# scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

4.

**Object Gateway** 노드에서 데이터 디렉토리를 생성합니다.

구문

```
# mkdir -p /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`
```

예제

```
# mkdir -p /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`
```

5.

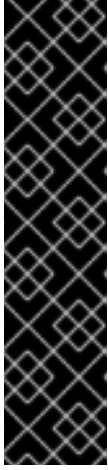
**Object Gateway** 노드에서 사용자 및 인증 키를 추가하여 오브젝트 게이트웨이를 부트스트랩합니다.

구문

```
# ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`/keyring
```

예제

```
# ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/keyring
```



### 중요

게이트웨이 키에 기능을 제공할 때 읽기 기능을 제공해야 합니다. 그러나 **Monitor** 쓰기 기능을 제공하는 것은 선택 사항입니다. 사용자가 제공하는 경우 **Ceph Object Gateway**는 자동으로 풀을 생성할 수 있습니다.

이러한 경우 풀에 적절한 수의 배치 그룹을 지정해야 합니다. 그렇지 않으면 게이트웨이가 기본 번호를 사용하므로 필요에 적합하지 않을 수 있습니다. 자세한 내용은 풀당 **Ceph PG(배치 그룹)** 를 참조하십시오.

6. **Object Gateway** 노드에서 완료된 파일을 생성합니다.

### 구문

```
# touch /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname`-s`/done
```

### 예제

```
# touch /var/lib/ceph/radosgw/ceph-rgw.`hostname`-s`/done
```

7. **Object Gateway** 노드에서 소유자 및 그룹 권한을 변경합니다.

```
# chown -R ceph:ceph /var/lib/ceph/radosgw
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```

8. 사용자 지정 이름이 있는 스토리지 클러스터의 경우 **root** 로 다음 행을 추가합니다.

### 구문

```
# echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
```

예제

```
# echo "CLUSTER=test123" >> /etc/sysconfig/ceph
```

9.

**Object Gateway** 노드에서 **TCP 포트 80**을 엽니다.

```
# firewall-cmd --zone=public --add-port=80/tcp  
# firewall-cmd --zone=public --add-port=80/tcp --permanent
```

10.

**Object Gateway** 노드에서 **ceph-radosgw** 프로세스를 시작하고 활성화합니다.

구문

```
# systemctl enable ceph-radosgw.target  
# systemctl enable ceph-radosgw@rgw.<rgw_hostname>  
# systemctl start ceph-radosgw@rgw.<rgw_hostname>
```

예제

```
# systemctl enable ceph-radosgw.target  
# systemctl enable ceph-radosgw@rgw.node1  
# systemctl start ceph-radosgw@rgw.node1
```

설치가 완료되면 **Ceph Object Gateway**가 모니터에 쓰기 기능이 설정된 경우 자동으로 풀을 생성합니다. 수동으로 풀 생성에 대한 자세한 내용은 스토리지 전략 가이드의 풀 장을 참조하십시오.

추가 세부 정보

- **Red Hat Ceph Storage 3 Red Hat Enterprise Linux용 오브젝트 게이트웨이 가이드**

## 부록 F. CEPH 기본 설정 덮어쓰기

Ansible 구성 파일에 달리 지정하지 않는 한 Ceph는 기본 설정을 사용합니다.

Ansible은 Ceph 구성 파일을 관리하므로 `/usr/share/ceph-ansible/group_vars/all.yml` 파일을 편집하여 Ceph 구성을 변경합니다. `ceph_conf_overrides` 설정을 사용하여 기본 Ceph 구성을 재정의합니다.

Ansible은 Ceph 구성 파일과 동일한 섹션, `[global]`, `[mon]`, `[osd]`, `[mds]`, `[rgw]` 등을 지원합니다. 특정 Ceph Object Gateway 인스턴스와 같은 특정 인스턴스를 덮어쓸 수도 있습니다. 예를 들면 다음과 같습니다.

```
#####
# CONFIG OVERRIDE #
#####

ceph_conf_overrides:
  client.rgw.rgw1:
    log_file: /var/log/ceph/ceph-rgw-rgw1.log
```

## 참고

Ceph 구성 파일의 특정 섹션을 참조할 때 Ansible에는 브레이스가 포함되지 않습니다. 섹션 및 설정 이름은 콜론으로 종료됩니다.

## 중요

CONFIG OVERRIDE 섹션에서 `cluster_network` 매개 변수를 사용하여 클러스터 네트워크를 설정하지 마십시오. 이로 인해 Ceph 구성 파일에 두 개의 충돌하는 클러스터 네트워크가 설정될 수 있기 때문입니다.

클러스터 네트워크를 설정하려면 CEPH CONFIGURATION 섹션에서 `cluster_network` 매개 변수를 사용합니다. 자세한 내용은 3.2절. “Red Hat Ceph Storage 클러스터 설치”의 내용을 참조하십시오.

## 부록 G. RED HAT CEPH STORAGE 2에서 3으로 수동 업그레이드

클러스터가 실행되는 동안 **Ceph Storage** 클러스터를 롤링 방식으로 2에서 3으로 업그레이드할 수 있습니다. 클러스터의 각 노드를 순차적으로 업그레이드하고 이전 노드가 완료된 후 다음 노드로만 이동합니다.

다음과 같은 순서로 **Ceph** 구성 요소를 업그레이드하는 것이 좋습니다.

- 노드 모니터링
- OSD 노드
- Ceph Object Gateway 노드
- 기타 모든 **Ceph** 클라이언트 노드

**Red Hat Ceph Storage 3**에는 새로운 데몬 **Ceph Manager(ceph-mgr)**가 도입되었습니다. **Monitor** 노드를 업그레이드한 후 **ceph-mgr** 을 설치합니다.

**Red Hat Ceph Storage 2**를 3으로 업그레이드하는 방법은 다음 두 가지가 있습니다.

- **Red Hat CDN(Content Delivery Network)** 사용
- **Red Hat** 제공 ISO 이미지 파일 사용

스토리지 클러스터를 업그레이드한 후 기존 튜닝 가능 항목을 사용하여 **CRUSH** 맵에 대한 상태 경고를 받을 수 있습니다. 자세한 내용은 **Red Hat Ceph Storage 3**용 스토리지 전략 가이드의 **CRUSH Tunables** 섹션을 참조하십시오.

예제



```
$ ceph -s
cluster 848135d7-cdb9-4084-8df2-fb5e41ae60bd
health HEALTH_WARN
  crush map has legacy tunables (require bobtail, min is firefly)
monmap e1: 1 mons at {ceph1=192.168.0.121:6789/0}
  election epoch 2, quorum 0 ceph1
osdmap e83: 2 osds: 2 up, 2 in
pgmap v1864: 64 pgs, 1 pools, 38192 kB data, 17 objects
  10376 MB used, 10083 MB / 20460 MB avail
  64 active+clean
```



### 중요

Red Hat은 모든 Ceph 클라이언트가 Ceph 스토리지 클러스터와 동일한 버전을 실행하도록 권장합니다.

### 사전 요구 사항

- 업그레이드하려는 클러스터에 **exclusive-lock** 기능을 사용하는 **Ceph Block Device** 이미지가 포함된 경우 모든 **Ceph Block Device** 사용자에게 클라이언트를 블랙리스트로 지정할 수 있는 권한이 있는지 확인하십시오.

```
ceph auth caps client.<ID> mon 'allow r, allow command "osd blacklist"' osd '<existing-OSD-user-capabilities>'
```

### 모니터 노드 업그레이드

이 섹션에서는 **Ceph Monitor** 노드를 최신 버전으로 업그레이드하는 단계를 설명합니다. 홀수의 모니터가 있어야 합니다. 모니터 하나를 업그레이드하는 동안 스토리지 클러스터에는 쿼럼이 있습니다.

### 절차

스토리지 클러스터의 각 모니터 노드에서 다음 단계를 수행합니다. 한 번에 하나의 **Monitor** 노드만 업그레이드합니다.

- 소프트웨어 리포지토리를 사용하여 **Red Hat Ceph Storage 2**를 설치한 경우 리포지토리를 비활성화합니다.

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2-mon-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

2. **Red Hat Ceph Storage 3 Monitor** 리포지토리를 활성화합니다.

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

3. **root** 로 **monitor** 프로세스를 중지합니다.

구문

```
# service ceph stop <daemon_type>.<monitor_host_name>
```

예제

```
# service ceph stop mon.node1
```

4. **root** 로 **ceph-mon** 패키지를 업데이트합니다.

```
# yum update ceph-mon
```

5. **root** 로서 소유자 및 그룹 권한을 업데이트합니다.

구문

```
# chown -R <owner>:<group> <path_to_directory>
```

예제

```
# chown -R ceph:ceph /var/lib/ceph/mon
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown ceph:ceph /etc/ceph/ceph.client.admin.keyring
# chown ceph:ceph /etc/ceph/ceph.conf
# chown ceph:ceph /etc/ceph/rbdmap
```



## 참고

**Ceph Monitor** 노드가 **OpenStack** 컨트롤러 노드와 함께 배치되는 경우 **Glance** 및 **Cinder** 인증 키 파일을 각각 **glance** 및 **cinder** 에서 소유해야 합니다. 예를 들면 다음과 같습니다.

```
# ls -l /etc/ceph/
...
-rw-----. 1 glance glance    64 <date> ceph.client.glance.keyring
-rw-----. 1 cinder cinder    64 <date> ceph.client.cinder.keyring
...
```

6. **SELinux**가 강제 모드 또는 허용 모드인 경우 다음 재부팅 시 **SELinux** 컨텍스트의 레이블을 다시 지정합니다.

```
# touch /.autorelabel
```



## 주의

**SELinux**가 모든 파일 시스템을 트래버스하고 레이블이 잘못 지정된 파일을 수정해야 하므로 레이블을 다시 지정하는 데 시간이 오래 걸릴 수 있습니다. 디렉토리의 레이블을 다시 레이블로 지정하지 않도록 하려면 재부팅하기 전에 `/etc/selinux/fixfiles_exclude_dirs` 파일에 디렉토리를 추가합니다.

7. **root** 로 **ceph-mon** 프로세스를 활성화합니다.

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@<monitor_host_name>
```

8. **root** 로서 **Monitor** 노드를 재부팅합니다.

```
# shutdown -r now
```

9. **Monitor** 노드가 가동되면 다음 **Monitor** 노드로 이동하기 전에 **Ceph** 스토리지 클러스터의 상태를 확인합니다.

```
# ceph -s
```

### G.1. 수동으로 CEPH MANAGER 설치

일반적으로 **Ansible** 자동화 유틸리티는 **Red Hat Ceph Storage** 클러스터를 배포할 때 **Ceph Manager** 데몬(**ceph-mgr**)을 설치합니다. 그러나 **Ansible**을 사용하여 **Red Hat Ceph Storage**를 관리하지 않는 경우 **Ceph Manager**를 수동으로 설치할 수 있습니다. **Red Hat**은 동일한 노드에서 **Ceph Manager** 및 **Ceph Monitor** 데몬을 배치하는 것이 좋습니다.

#### 사전 요구 사항

- 작동하는 **Red Hat Ceph Storage** 클러스터
- **root** 또는 **sudo** 액세스
- **rhel-7-server-rhceph-3-mon-els-rpms** 리포지토리 활성화
- 방화벽이 사용되는 경우 공용 네트워크에서 포트 **6800-7300**

#### 절차

**ceph-mgr** 가 배포되는 노드에서 및 **root** 사용자로 또는 **sudo** 유틸리티를 사용하여 다음 명령을 사용합니다.

1. **ceph-mgr** 패키지를 설치합니다.

```
[root@node1 ~]# yum install ceph-mgr
```

2.

`/var/lib/ceph/mgr/ceph-호스트 이름/` 디렉터리를 생성합니다.

```
mkdir /var/lib/ceph/mgr/ceph-hostname
```

**hostname** 을 **ceph-mgr** 데몬이 배포될 노드의 호스트 이름으로 교체합니다. 예를 들면 다음과 같습니다.

```
[root@node1 ~]# mkdir /var/lib/ceph/mgr/ceph-node1
```

3.

새로 생성된 디렉터리에서 **ceph-mgr** 데몬에 대한 인증 키를 생성합니다.

```
[root@node1 ~]# ceph auth get-or-create mgr.`hostname` -s` mon 'allow profile mgr' osd 'allow *' mds 'allow *' -o /var/lib/ceph/mgr/ceph-node1/keyring
```

4.

`/var/lib/ceph/mgr/` 디렉터리의 소유자 및 그룹을 **ceph:ceph** 로 변경합니다.

```
[root@node1 ~]# chown -R ceph:ceph /var/lib/ceph/mgr
```

5.

**ceph-mgr** 대상을 활성화합니다.

```
[root@node1 ~]# systemctl enable ceph-mgr.target
```

6.

**ceph-mgr** 인스턴스를 활성화하고 시작합니다.

```
systemctl enable ceph-mgr@hostname
systemctl start ceph-mgr@hostname
```

**hostname** 을 **ceph-mgr** 이 배포할 노드의 호스트 이름으로 교체합니다. 예를 들면 다음과 같습니다.

```
[root@node1 ~]# systemctl enable ceph-mgr@node1
[root@node1 ~]# systemctl start ceph-mgr@node1
```

7.

**ceph-mgr** 데몬이 시작되었는지 확인합니다.

■

```
ceph -s
```

출력에는 **services:** 섹션 아래의 다음 행과 유사한 행이 포함됩니다.

```
mgr: node1(active)
```

8.

현재 활성 데몬이 실패하는 경우 더 많은 **ceph-mgr** 데몬을 설치하여 대기 데몬으로 작동합니다.

추가 리소스

- 

[Red Hat Ceph Storage 설치 요구사항](#)

## OSD 노드 업그레이드

이 섹션에서는 **Ceph OSD** 노드를 최신 버전으로 업그레이드하는 단계를 설명합니다.

사전 요구 사항

**OSD** 노드를 업그레이드할 때 **OSD**가 다운되거나 다시 시작될 수 있으므로 일부 배치 그룹이 성능이 저하됩니다. **Ceph**가 복구 프로세스를 시작하지 못하도록 하려면 **Monitor** 노드에서 **noout** 및 **norebalance** **OSD** 플래그를 설정합니다.

```
[root@monitor ~]# ceph osd set noout
[root@monitor ~]# ceph osd set norebalance
```

절차

스토리지 클러스터의 각 **OSD** 노드에서 다음 단계를 수행합니다. 한 번에 하나의 **OSD** 노드만 업그레이드합니다. **Red Hat Ceph Storage 2.3**에 대해 **ISO** 기반 설치를 수행한 경우 이 첫 번째 단계를 건너뛰니다.

1.

**root** 로서 **Red Hat Ceph Storage 2** 리포지토리를 비활성화합니다.

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2-osd-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

2.

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

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

3. **root** 로 실행 중인 **OSD** 프로세스를 중지합니다.

구문

```
# service ceph stop <daemon_type>.<osd_id>
```

예제

```
# service ceph stop osd.0
```

4. **root** 로 **ceph-osd** 패키지를 업데이트합니다.

```
# yum update ceph-osd
```

5. **root** 로서 새로 생성된 디렉터리 및 파일에 대한 소유자 및 그룹 권한을 업데이트합니다.

구문

```
# chown -R <owner>:<group> <path_to_directory>
```

예제

```
# chown -R ceph:ceph /var/lib/ceph/osd
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```



### 참고

다음 **find** 명령을 사용하면 다수의 디스크가 많은 **Ceph** 스토리지 클러스터에서 **chown** 명령을 병렬로 사용하여 소유권을 변경하는 프로세스를 빠르게 수행할 수 있습니다.

```
# find /var/lib/ceph/osd -maxdepth 1 -mindepth 1 -print | xargs -P12 -n1 chown -R ceph:ceph
```

6.

**SELinux**가 강제 모드 또는 허용 모드로 설정된 경우 다음 재부팅을 위해 파일에서 **SELinux** 컨텍스트 재지정을 설정합니다.

```
# touch /.autorelabel
```



### 주의

**SELinux**는 모든 파일 시스템을 트래버스하고 레이블이 잘못 지정된 파일을 수정해야 하므로 레이블을 다시 지정하는 데 시간이 오래 걸립니다. 디렉토리의 레이블 재지정을 제외하려면 재부팅하기 전에 **/etc/selinux/fixfiles\_exclude\_dirs** 파일에 디렉토리를 추가합니다.





## 참고

배치 그룹(PG)당 많은 개체가 있는 환경에서는 디렉터리 열거 속도가 감소하여 성능에 부정적인 영향을 미칩니다. 이는 SELinux 컨텍스트를 확인하는 **xattr** 쿼리의 추가로 인해 발생합니다. 마운트 시 컨텍스트를 설정하면 컨텍스트에 대한 **xattr** 쿼리가 제거되고 특히 느린 디스크에서 전체 디스크 성능이 도움이 됩니다.

/etc/ceph/ceph.conf 파일의 [osd] 섹션에 다음 행을 추가합니다.

+

```
osd_mount_options_xfs=rw,noatime,inode64,context="system_u:object_r:ceph_var_lib_t:s0"
```

7.

**root** 로서 커널에서 장치 이벤트를 재생합니다.

```
# udevadm trigger
```

8.

**root** 로 **ceph-osd** 프로세스를 활성화합니다.

```
# systemctl enable ceph-osd.target
# systemctl enable ceph-osd@<osd_id>
```

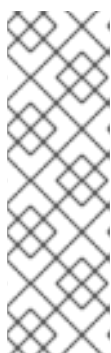
9.

**root** 로서 **OSD** 노드를 재부팅합니다.

```
# shutdown -r now
```

10.

다음 **OSD** 노드로 이동합니다.



## 참고

**noout** 및 **norebalance** 플래그가 설정된 경우 스토리지 클러스터는 **HEALTH\_WARN** 상태입니다.

```
$ ceph health
HEALTH_WARN noout,norebalance flag(s) set
```

**Ceph Storage** 클러스터를 업그레이드한 후 이전에 설정한 **OSD** 플래그를 설정 해제하고 스토리지 클러스터 상태를 확인합니다.

모니터 노드에서 및 모든 **OSD** 노드가 업그레이드된 후 **noout** 및 **norebalance** 플래그를 설정 해제합니다.

```
# ceph osd unset noout
# ceph osd unset norebalance
```

또한 **ceph osd require-osd-release <release>** 명령을 실행합니다. 이 명령을 사용하면 더 이상 **Red Hat Ceph Storage 2.3**이 있는 **OSD**를 스토리지 클러스터에 추가할 수 없습니다. 이 명령을 실행하지 않으면 스토리지 상태가 **HEALTH\_WARN** 이 됩니다.

```
# ceph osd require-osd-release luminous
```

추가 리소스

- 스토리지 클러스터에 새 **OSD**를 추가하여 스토리지 용량을 확장하려면 **Red Hat Ceph Storage 3 관리 가이드**의 **OSD 추가** 섹션을 참조하십시오.

## Ceph Object Gateway 노드 업그레이드

이 섹션에서는 **Ceph Object Gateway** 노드를 이후 버전으로 업그레이드하는 단계를 설명합니다.

사전 요구 사항

- **Red Hat**은 **HAProxy** 와 같은 로드 밸런서 뒤에 **Ceph** 오브젝트 게이트웨이를 두는 것이 좋습니다. 로드 밸런서를 사용하는 경우 요청이 제공되지 않으면 로드 밸런서에서 **Ceph Object Gateway**를 제거합니다.
- **rgw\_region\_root\_pool** 매개변수에 지정된 리전 풀에 사용자 지정 이름을 사용하는 경우 **Ceph** 구성 파일의 **[global]** 섹션에 **rgw\_zonegroup\_root\_pool** 매개변수를 추가합니다. **rgw\_zonegroup\_root\_pool**의 값을 **rgw\_region\_root\_pool** 과 동일하게 설정합니다. 예를 들면 다음과 같습니다.

```
[global]
rgw_zonegroup_root_pool = .us.rgw.root
```

절차

스토리지 클러스터의 각 **Ceph Object Gateway** 노드에서 다음 단계를 수행합니다. 한 번에 하나의 노드만 업그레이드합니다.

1. **Red Hat Ceph Storage**를 설치하기 위해 온라인 리포지토리를 사용한 경우 **2** 리포지토리를 비활성화합니다.

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2.3-tools-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

2. **Red Hat Ceph Storage 3** 툴 리포지토리를 활성화합니다.

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

3. **Ceph Object Gateway** 프로세스(**ceph-radosgw**)를 중지합니다.

```
# service ceph-radosgw stop
```

4. **ceph-radosgw** 패키지를 업데이트합니다.

```
# yum update ceph-radosgw
```

5. 새로 생성된 **/var/lib/ceph/radosgw/** 및 **/var/log/ceph/** 디렉터리에 대한 소유자 및 그룹 권한을 **ceph** 로 변경합니다.

```
# chown -R ceph:ceph /var/lib/ceph/radosgw
# chown -R ceph:ceph /var/log/ceph
```

6. **SELinux**가 강제 또는 허용 모드에서 실행되도록 설정된 경우 다음 부팅 시 **SELinux** 컨텍스트의 레이블을 다시 지정하도록 지시합니다.

```
# touch /.autorelabel
```

### 중요

**SELinux**는 모든 파일 시스템을 통과하고 레이블이 잘못 지정된 파일을 수정해야 하므로 레이블을 다시 지정하는 데 시간이 오래 걸립니다. 디렉터리의 레이블이 다시 지정되지 않은 경우 재부팅하기 전에 **/etc/selinux/fixfiles\_exclude\_dirs** 파일에 추가합니다.

7. **ceph-radosgw** 프로세스를 활성화합니다.

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.<hostname>
```

<hostname> 을 **Ceph Object Gateway** 호스트의 이름으로 바꿉니다(예: **gateway-node** ).

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.gateway-node
```

8. **Ceph Object Gateway** 노드 재부팅

```
# shutdown -r now
```

9. 로드 밸런서를 사용하는 경우 **Ceph Object Gateway** 노드를 다시 로드 밸런서에 추가합니다.

예를 들면 다음과 같습니다.

- [Red Hat Enterprise Linux용 Ceph Object Gateway 가이드](#)

## Ceph 클라이언트 노드 업그레이드

Ceph 클라이언트는 다음과 같습니다.

- **Ceph** 블록 장치
- **OpenStack Nova Compute** 노드
- **QEMU/KVM** 하이퍼바이저
- **Ceph** 클라이언트 측 라이브러리를 사용하는 사용자 정의 애플리케이션

Red Hat은 모든 **Ceph** 클라이언트가 **Ceph** 스토리지 클러스터와 동일한 버전을 실행하도록 권장합니다.

## 사전 요구 사항

- 예기치 않은 오류가 발생하지 않도록 패키지를 업그레이드하는 동안 **Ceph** 클라이언트 노드에 대한 모든 I/O 요청을 중지합니다.

## 절차

1. 소프트웨어 리포지토리를 사용하여 **Red Hat Ceph Storage 2** 클라이언트를 설치한 경우 리포지토리를 비활성화합니다.

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2-tools-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```



### 참고

**Red Hat Ceph Storage 2** 클라이언트에 대해 **ISO** 기반 설치를 수행한 경우 이 첫 번째 단계를 건너뛸 수 있습니다.

2. 클라이언트 노드에서 **Red Hat Ceph Storage Tools 3** 리포지토리를 활성화합니다.

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

3. 클라이언트 노드에서 **ceph-common** 패키지를 업데이트합니다.

```
# yum update ceph-common
```

**ceph-common** 패키지를 업그레이드한 후 **Ceph** 클라이언트 측 라이브러리에 종속된 애플리케이션을 다시 시작합니다.



### 참고

**QEMU/KVM** 인스턴스를 실행하거나 전용 **QEMU/KVM** 클라이언트를 사용하는 **OpenStack Nova** 컴퓨팅 노드를 업그레이드하는 경우 이 경우 인스턴스를 다시 시작할 수 없으므로 **QEMU/KVM** 인스턴스를 중지 및 시작합니다.

부록 H. 버전 2와 3의 ANSIBLE 변수 변경

Red Hat Ceph Storage 3에서는 /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 [Red Hat Ceph Storage 클러스터 업그레이드](#)를 참조하십시오.

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

## 부록 I. ANSIBLE로 기존 CEPH 클러스터 가져오기

**Ansible** 없이 배포된 클러스터를 사용하도록 **Ansible**을 구성할 수 있습니다. 예를 들어 **Red Hat Ceph Storage 1.3** 클러스터를 버전 **2**로 수동으로 업그레이드하는 경우 다음 절차에 따라 **Ansible**을 사용하도록 구성합니다.

1. 버전 **1.3**에서 버전 **2**로 수동으로 업그레이드한 후 관리 노드에 **Ansible**을 설치하고 구성합니다.
2. **Ansible** 관리 노드에 클러스터의 모든 **Ceph** 노드에 대한 암호 없는 **ssh** 액세스 권한이 있는지 확인합니다. 자세한 내용은 [2.11절. “Ansible에 대해 암호 없는 SSH 활성화”](#)를 참조하십시오.
3. **root** 로서 **/etc/ansible/** 디렉터리의 **Ansible group\_vars** 디렉터리에 대한 심볼릭 링크를 만듭니다.

```
# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

4. **root** 로서 **all.yml.sample** 파일에서 **all.yml** 파일을 생성하고 편집을 위해 엽니다.

```
# cd /etc/ansible/group_vars
# cp all.yml.sample all.yml
# vim all.yml
```

5. **group\_vars/all.yml** 에서 **generate\_fsid** 설정을 **false** 로 설정합니다.

6. **ceph fsid** 를 실행하여 현재 클러스터 **fsid**를 가져옵니다.

7. 검색된 **fsid** 를 **group\_vars/all.yml** 에 설정합니다.

8. **Ceph** 호스트를 포함하도록 **/etc/ansible/hosts** 에서 **Ansible** 인벤토리를 수정합니다. **[mons]** 섹션 아래에 있는 모니터와 **[osds]** 섹션 아래에 있는 게이트웨이를 추가하여 **Ansible**에 대한 역할을 식별합니다.

9. **ceph\_conf\_overrides** 가 **all.yml** 파일의 **[global]**, **[osd]**, **[mon]** 및 **[client]** 섹션에 사용되는 원래 **ceph.conf** 옵션으로 업데이트되었는지 확인합니다.

**osd journal,public\_network** 및 **cluster\_network** 와 같은 옵션은 이미 **all.yml** 의 일부이므로 **ceph\_conf\_overrides** 에 추가하지 않아야 합니다. **all.yml** 의 일부가 아니며 원래 **ceph.conf** 에 있는 옵션만 **ceph\_conf\_overrides** 에 추가해야 합니다.

10.

**/usr/share/ceph-ansible/** 디렉터리에서 플레이북을 실행합니다.

```
# cd /usr/share/ceph-ansible/  
# cp infrastructure-playbooks/take-over-existing-cluster.yml .  
$ ansible-playbook take-over-existing-cluster.yml -u <username>
```



## 부록 J. ANSIBLE을 사용하여 CEPH 클러스터 제거

Ansible을 사용하여 Ceph 클러스터를 배포하고 클러스터를 삭제하려면 **infrastructure-playbooks** 디렉터리에 있는 **purge-cluster.yml** Ansible 플레이북을 사용합니다.



중요

Ceph 클러스터를 제거해도 클러스터 OSD에 저장된 데이터가 손실됩니다.

### Ceph 클러스터를 제거하기 전에...

**osds.yml** 파일에서 **osd\_auto\_discovery** 옵션을 확인합니다. 이 옵션을 **true** 로 설정하면 제거가 실패합니다. 실패를 방지하려면 제거를 실행하기 전에 다음 단계를 수행하십시오.

1. **osds.yml** 파일에서 OSD 장치를 선언합니다. 자세한 내용은 [3.2절. “Red Hat Ceph Storage 클러스터 설치”](#)를 참조하십시오.
2. **osds.yml** 파일의 **osd\_auto\_discovery** 옵션을 주석 처리합니다.

### Ceph 클러스터 제거...

1. 루트 로서 **/usr/share/ceph-ansible/** 디렉터리로 이동합니다.

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

2. root 로서 **purge-cluster.yml** Ansible 플레이북을 현재 디렉터리에 복사합니다.

```
# cp infrastructure-playbooks/purge-cluster.yml .
```

3. **purge-cluster.yml** Ansible 플레이북을 실행합니다.

```
$ ansible-playbook purge-cluster.yml
```