



# Red Hat Ceph Storage 4

## 설치 가이드

Red Hat Enterprise Linux에 Red Hat Ceph Storage 설치



# Red Hat Ceph Storage 4 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 7 및 Red Hat Enterprise Linux 8에 Red Hat Ceph Storage를 설치하는 방법을 설명합니다. Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 CTO Chris Wright의 메시지 에서 참조하십시오.

## 차례

<b>1장. RED HAT CEPH STORAGE란 무엇입니까?</b> .....	<b>4</b>
<b>2장. RED HAT CEPH STORAGE 고려 사항 및 권장 사항</b> .....	<b>6</b>
2.1. 사전 요구 사항	6
2.2. RED HAT CEPH STORAGE의 기본 고려 사항	6
2.3. RED HAT CEPH STORAGE 워크로드 고려 사항	7
2.4. RED HAT CEPH STORAGE의 네트워크 고려 사항	10
2.5. CEPH를 실행할 때 LINUX 커널 튜닝 고려 사항	11
2.6. OSD 노드에서 RAID 컨트롤러 사용 고려 사항	12
2.7. OBJECT GATEWAY에서 NVME 사용 고려 사항	12
2.8. RED HAT CEPH STORAGE의 최소 하드웨어 고려 사항	12
2.9. 추가 리소스	15
<b>3장. RED HAT CEPH STORAGE 설치 요구사항</b> .....	<b>16</b>
3.1. 사전 요구 사항	16
3.2. RED HAT CEPH STORAGE 설치를 위한 요구 사항 체크리스트	16
3.3. RED HAT CEPH STORAGE의 운영 체제 요구 사항	17
3.4. RED HAT CEPH STORAGE 노드를 CDN에 등록하고 서브스크립션 연결	18
3.5. RED HAT CEPH STORAGE 리포지토리 활성화	22
3.6. RED HAT CEPH STORAGE의 네트워크 구성 확인	24
3.7. RED HAT CEPH STORAGE의 방화벽 설정	25
3.8. SUDO 액세스를 사용하여 ANSIBLE 사용자 생성	29
3.9. ANSIBLE에 대해 암호 없는 SSH 활성화	30
<b>4장. COCKPIT 웹 인터페이스를 사용하여 RED HAT CEPH STORAGE 설치</b> .....	<b>33</b>
4.1. 사전 요구 사항	33
4.2. 설치 요구사항	33
4.3. COCKPIT CEPH 설치 및 구성	33
4.4. COCKPIT CEPH INSTALLER SSH 키를 클러스터의 모든 노드에 복사	37
4.5. COCKPIT에 로그인합니다.	38
4.6. COCKPIT CEPH 설치 프로그램의 환경 페이지 완료	40
4.7. COCKPIT CEPH 설치 프로그램의 HOSTS 페이지 완료	44
4.8. COCKPIT CEPH 설치 프로그램의 검증 페이지 완료	48
4.9. COCKPIT CEPH 설치 프로그램의 네트워크 페이지 완료	51
4.10. 설치 구성 검토	53
4.11. CEPH 클러스터 배포	55
<b>5장. ANSIBLE을 사용하여 RED HAT CEPH STORAGE 설치</b> .....	<b>61</b>
5.1. 사전 요구 사항	61
5.2. RED HAT CEPH STORAGE 클러스터 설치	61
5.3. 모든 NVME 스토리지에 대한 OSD ANSIBLE 설정 구성	77
5.4. 메타데이터 서버 설치	78
5.5. CEPH 클라이언트 역할 설치	80
5.6. CEPH OBJECT GATEWAY 설치	83
5.7. 다중 사이트 CEPH 개체 게이트웨이 구성	86
5.8. 동일한 호스트에서 다른 하드웨어를 사용하여 OSD 배포	108
5.9. NFS-GANESHA 게이트웨이 설치	111
5.10. 제한 옵션 이해	114
5.11. 배치 그룹 자동 스케일러	115
5.12. 추가 리소스	118
<b>6장. 컨테이너화된 CEPH 데몬 배치</b> .....	<b>119</b>

6.1. 코로케이션 작동 방식 및 이점	119
6.2. 배치된 데몬을 위한 전용 리소스 설정	123
6.3. 추가 리소스	125
<b>7장. RED HAT CEPH STORAGE 클러스터 업그레이드</b>	<b>126</b>
7.1. 지원되는 RED HAT CEPH STORAGE 업그레이드 시나리오	129
7.2. 업그레이드 준비	131
7.3. ANSIBLE을 사용하여 스토리지 클러스터 업그레이드	136
7.4. 명령줄 인터페이스를 사용하여 스토리지 클러스터 업그레이드	150
7.5. CEPH 파일 시스템 메타데이터 서버 노드를 수동으로 업그레이드	155
7.6. 추가 리소스	158
<b>8장. RED HAT CEPH STORAGE 클러스터 및 운영 체제를 수동으로 업그레이드</b>	<b>159</b>
8.1. 사전 요구 사항	160
8.2. CEPH MONITOR 노드 및 해당 운영 체제 수동 업그레이드	160
8.3. CEPH OSD 노드 및 운영 체제 수동 업그레이드	166
8.4. CEPH OBJECT GATEWAY 노드 및 해당 운영 체제를 수동으로 업그레이드	175
8.5. CEPH 대시보드 노드 및 운영 체제를 수동으로 업그레이드	178
8.6. CEPH ANSIBLE 노드 수동 업그레이드 및 설정 재구성	180
8.7. CEPH FILE SYSTEM 메타데이터 서버 노드 및 운영 체제를 수동으로 업그레이드	182
8.8. OSD 노드에서 운영 체제 업그레이드 실패에서 복구	186
8.9. 추가 리소스	188
<b>9장. 다음에 어떻게 해야 하나요?</b>	<b>189</b>
<b>부록 A. 문제 해결</b>	<b>190</b>
A.1. ANSIBLE은 예상보다 적은 장치를 감지하므로 설치를 중지합니다.	190
<b>부록 B. 명령줄 인터페이스를 사용하여 CEPH 소프트웨어 설치</b>	<b>192</b>
B.1. CEPH 명령줄 인터페이스 설치	192
B.2. RED HAT CEPH STORAGE 수동 설치	193
B.3. 수동으로 CEPH MANAGER 설치	210
B.4. 수동으로 CEPH 블록 장치 설치	212
B.5. 수동으로 CEPH OBJECT GATEWAY 설치	216
<b>부록 C. ANSIBLE 인벤토리 위치 구성</b>	<b>221</b>
<b>부록 D. CEPH 기본 설정 덮어쓰기</b>	<b>223</b>
<b>부록 E. 기존 CEPH 클러스터를 ANSIBLE로 가져오기</b>	<b>224</b>
<b>부록 F. ANSIBLE에서 배포한 스토리지 클러스터 제거</b>	<b>226</b>
<b>부록 G. ANSIBLE을 사용하여 CEPH 대시보드 삭제</b>	<b>229</b>
<b>부록 H. ANSIBLE-VAULT를 사용하여 ANSIBLE 암호 변수 암호화</b>	<b>231</b>
<b>부록 I. 일반 ANSIBLE 설정</b>	<b>236</b>
<b>부록 J. OSD ANSIBLE 설정</b>	<b>241</b>



# 1장. RED HAT CEPH STORAGE란 무엇입니까?

Red Hat Ceph Storage는 엔터프라이즈급 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

Ansible 관리 노드는 이전 버전의 Red Hat Ceph Storage에서 사용된 기존 Ceph 관리 노드를 대체합니다. Ansible 관리 노드는 다음 기능을 제공합니다.

- 중앙 집중식 스토리지 클러스터 관리.
- Ceph 구성 파일 및 키.
- 보안상의 이유로 인터넷에 액세스할 수 없는 노드에 Ceph를 설치하기 위한 로컬 리포지토리가 필요한 경우도 있습니다.

## Ceph Monitor

각 Ceph Monitor 노드는 스토리지 클러스터 맵의 마스터 복사본을 유지 관리하는 **ceph-mon** 데몬을 실행합니다. 스토리지 클러스터 맵에는 스토리지 클러스터 토폴로지가 포함됩니다. Ceph 스토리지 클러스터에 연결된 클라이언트는 Ceph Monitor에서 스토리지 클러스터 맵의 현재 복사본을 검색하여 클라이언트가 스토리지 클러스터에 데이터를 읽고 쓸 수 있습니다.



### 중요

스토리지 클러스터는 하나의 Ceph 모니터에서만 실행할 수 있지만 프로덕션 환경의 스토리지 클러스터에서 고가용성을 보장하기 위해 Red Hat은 적어도 3개의 Ceph Monitor 노드를 사용하는 배포만 지원합니다. Red Hat은 750 Ceph OSD 이상의 스토리지 클러스터에 대해 총 5개의 Ceph Monitor를 배포할 것을 권장합니다.

## Ceph OSD

각 Ceph OSD(Ceph Object Storage Device) 노드는 **ceph-osd** 데몬을 실행하고 노드에 연결된 논리 디스크와 상호 작용합니다. 스토리지 클러스터는 데이터를 이러한 Ceph OSD 노드에 저장합니다.

Ceph는 매우 적은 수의 OSD 노드에서 실행할 수 있습니다. 기본값은 3개이지만, 프로덕션 스토리지 클러스터는 크기가 조정될 때 더 나은 성능을 구현할 수 있습니다. 예를 들어 스토리지 클러스터에 있는 50개의 Ceph OSD가 있습니다. Ceph 스토리지 클러스터에는 여러 개의 OSD 노드가 있으므로 CRUSH 매핑을 적절하게 구성하여 장애 도메인을 격리할 수 있습니다.

## Ceph MDS

각 Ceph 메타데이터 서버(MDS) 노드는 Ceph 파일 시스템(CephFS)에 저장된 파일과 관련된 메타데이터를 관리하는 **ceph-mds** 데몬을 실행합니다. Ceph MDS 데몬은 공유 스토리지 클러스터에 대한 액세스도 조정합니다.

## Ceph Object Gateway

Ceph Object Gateway 노드는 **ceph-radosgw** 데몬을 실행하며, Ceph 스토리지 클러스터에 대한 RESTful 액세스 지점을 애플리케이션에 제공하기 위해 **librados** 상단에 빌드된 오브젝트 스토리지 인터페이스입니다. Ceph Object Gateway는 다음 두 개의 인터페이스를 지원합니다.

- S3



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

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

#### 추가 리소스

- Ceph 아키텍처에 대한 자세한 내용은 [Red Hat Ceph Storage Architecture Guide](#) 를 참조하십시오.
- 최소 하드웨어 권장 사항은 [Red Hat Ceph Storage 하드웨어 선택 가이드](#) 를 참조하십시오.

## 2장. RED HAT CEPH STORAGE 고려 사항 및 권장 사항

스토리지 관리자는 Red Hat Ceph Storage 클러스터를 실행하기 전에 고려해야 할 사항을 기본적으로 이해해야 합니다. 하드웨어 및 네트워크 요구 사항, Red Hat Ceph Storage 클러스터와 호환되는 워크로드 유형의 이해 및 Red Hat의 권장 사항을 검토하십시오. Red Hat Ceph Storage는 특정 비즈니스 요구 사항 또는 일련의 요구 사항에 따라 다양한 워크로드에 사용할 수 있습니다. Red Hat Ceph Storage를 설치하기 전에 필요한 계획을 수행하는 것은 Ceph 스토리지 클러스터를 효율적으로 실행하는 데 있어 비즈니스 요구 사항을 달성하는 데 매우 중요합니다.



### 참고

특정 사용 사례에 맞게 Red Hat Ceph Storage 클러스터를 계획하는 데 도움이 필요하십니까? 자세한 내용은 Red Hat 담당자에게 문의하십시오.

### 2.1. 사전 요구 사항

- 스토리지 솔루션을 이해, 검토 및 계획할 시간을 확보합니다.

### 2.2. RED HAT CEPH STORAGE의 기본 고려 사항

Red Hat Ceph Storage를 사용하는 첫 번째 고려 사항은 데이터를 위한 스토리지 전략을 개발하는 것입니다. 스토리지 전략은 특정 사용 사례에 서비스를 제공하는 데이터를 저장하는 방법입니다. OpenStack과 같은 클라우드 플랫폼에 대한 볼륨과 이미지를 저장해야 하는 경우 저널용 SSD(Solid State Drives)로 더 빠른 SAS( Serial Attached SCSI) 드라이브에 데이터를 저장하도록 선택할 수 있습니다. 반대로 S3 또는 Swift 호환 게이트웨이에 대한 오브젝트 데이터를 저장해야 하는 경우 기존 SATA(Serial Advanced Technology Attachment) 드라이브와 같이 더 경제적인 것을 사용할 수 있습니다. Red Hat Ceph Storage는 동일한 스토리지 클러스터에 있는 두 시나리오를 모두 수용할 수 있지만, 클라우드 플랫폼에 빠른 스토리지 전략을 제공하는 수단과 개체 저장소에 더 많은 기존 스토리지를 제공하는 수단이 필요합니다.

Ceph 배포의 가장 중요한 단계 중 하나는 스토리지 클러스터의 사용 사례와 워크로드에 적합한 가격대 성능 프로필을 식별하는 것입니다. 사용 사례에 적합한 하드웨어를 선택하는 것이 중요합니다. 예를 들어, 콜드 스토리지 애플리케이션에 최적화된 IOPS 하드웨어를 선택하면 하드웨어 비용이 불필요하게 증가합니다. 반면, IOPS 집약적인 워크로드에서 용량에 최적화된 하드웨어를 선택하는 경우 성능 저하에 대해 불만을 제기하는 사용자가 불만을 줄 수 있습니다.

Red Hat Ceph Storage는 여러 스토리지 전략을 지원할 수 있습니다. 사운드 스토리지 전략을 개발하는 데 도움이 되는 사용 사례, 비용 대비 성능 장단점 및 데이터 지속성이 가장 중요한 고려 사항입니다.

#### 사용 사례

Ceph는 대용량 스토리지 용량을 제공하며 다음과 같은 다양한 사용 사례를 지원합니다.

- Ceph 블록 장치 클라이언트는 COW(Copy-On-Write 복제)와 같은 고성능 기능을 갖춘 볼륨 및 이미지에 대한 무제한 스토리지를 제공하는 클라우드 플랫폼용 최고의 스토리지 백엔드입니다.
- Ceph Object Gateway 클라이언트는 오디오, 비트맵, 비디오 및 기타 데이터와 같은 오브젝트에 RESTful S3 호환 및 Swift 호환 개체 스토리지를 제공하는 클라우드 플랫폼용 최고의 스토리지 백엔드입니다.
- 기존 파일 스토리지를 위한 Ceph 파일 시스템입니다.

#### 비용 vs. 성능 이점

속도, 크기, 내구성 등이 더 우수합니다. 그러나 각 최상급 품질에는 각각 비용이 들기 때문에 비용 효율적인 측면에서 그에 상응하는 비용 대 편익 절충이 있습니다. 성능 관점에서 다음 사용 사례를 고려하십시오.

SSD는 상대적으로 적은 양의 데이터와 저널링에 매우 빠른 스토리지를 제공할 수 있습니다. 데이터베이스 또는 개체 인덱스를 저장하면 매우 빠른 SSD 풀의 이점을 얻을 수 있지만 다른 데이터에 너무 많은 비용이 듭니다. SSD 저널링을 사용하는 SAS 드라이브는 볼륨과 이미지의 경제적인 가격으로 빠른 성능을 제공합니다. SSD 저널링이 없는 SATA 드라이브는 전체 성능이 낮은 저렴한 스토리지를 제공합니다. OSD의 CRUSH 계층 구조를 생성할 때 사용 사례 및 허용 가능한 비용 대신 성능 트레이드를 고려해야 합니다.

## 데이터 내결함성

대규모 클러스터에서는 하드웨어 장애가 예상되며 예외는 아닙니다. 그러나 데이터 손실 및 서비스 중단은 허용되지 않습니다. 이러한 이유로 데이터 지속성이 매우 중요합니다. Ceph는 여러 오브젝트 복제본 사본 또는 삭제 코딩 및 여러 코딩 청크를 사용하여 데이터 지속성을 처리합니다. 여러 복사본 또는 여러 코딩 청크를 사용하면 추가 비용과 혜택의 장단점이 있습니다. 즉, 복사본 또는 코딩 청크를 더 적게 저장하는 것이 더 저렴하지만 성능 저하된 상태로 서비스 쓰기 요청을 사용할 수 없게 될 수 있습니다. 일반적으로 두 개의 추가 사본이 있는 하나의 오브젝트 또는 두 개의 코딩 청크를 사용하면 스토리지 클러스터를 복구하는 동안 성능이 저하된 상태로 스토리지 클러스터가 기록되도록 할 수 있습니다.

복제는 하드웨어 장애 발생 시 장애 도메인에서 데이터의 중복 복사본을 하나 이상 저장합니다. 그러나 데이터의 중복 복사본은 규모에 따라 비용이 많이 들 수 있습니다. 예를 들어, 3개 복제를 사용하여 1페타바이트의 데이터를 저장하려면 최소 3페타바이트 이상의 스토리지 용량이 있는 클러스터가 필요합니다.

코딩 삭제는 데이터를 데이터 청크 및 코딩 청크로 저장합니다. 데이터 청크가 손실되는 경우 삭제 코딩은 나머지 데이터 청크 및 코딩 청크를 사용하여 손실된 데이터 청크를 복구할 수 있습니다. 삭제 코딩은 복제보다 상당히 경제적입니다. 예를 들어 데이터 청크 8개와 코딩 청크 3개가 있는 삭제 코딩을 사용하면 데이터 복사본 3개와 동일한 중복성이 제공됩니다. 그러나 이러한 인코딩 체계는 복제와 3x에 비해 저장된 초기 데이터의 약 1.5x를 사용합니다.

CRUSH 알고리즘은 Ceph가 스토리지 클러스터 내의 다른 위치에 추가 사본 또는 코딩 청크를 저장할 수 있도록 하여 이 프로세스를 지원합니다. 이렇게 하면 단일 스토리지 장치 또는 노드에 장애가 발생해도 데이터 손실을 방지하는 데 필요한 모든 복사본 또는 코딩 청크가 손실되지 않습니다. 비용 대비 이점의 장단점이 있는 스토리지 전략을 계획하고 데이터 지속성을 염두에 두고 Ceph 클라이언트에 스토리지 풀로 제공할 수 있습니다.



### 중요

데이터 스토리지 풀만 삭제 코딩을 사용할 수 있습니다. 서비스 데이터 및 버킷 인덱스를 저장하는 풀은 복제를 사용합니다.



### 중요

Ceph의 개체 복사본 또는 코딩 청크를 사용하면 RAID 솔루션이 더 이상 사용되지 않습니다. Ceph는 이미 데이터 지속성을 처리하므로 RAID를 사용하지 마십시오. 성능이 저하된 RAID는 성능에 부정적인 영향을 미치며 RAID를 사용하여 데이터를 복구하는 것은 깊은 복사본 또는 코딩 청크를 사용하는 것보다 훨씬 느립니다.

## 추가 리소스

- 자세한 내용은 Red Hat Ceph Storage *설치 가이드*의 [Red Hat Ceph Storage의 최소 하드웨어 고려 사항](#) 섹션을 참조하십시오.

## 2.3. RED HAT CEPH STORAGE 워크로드 고려 사항

Ceph 스토리지 클러스터의 주요 이점 중 하나는 성능 도메인을 사용하여 동일한 스토리지 클러스터 내에서 다양한 유형의 워크로드를 지원할 수 있다는 것입니다. 각 성능 도메인과 다른 하드웨어 구성을 연결할 수 있습니다. 스토리지 관리자는 적절한 성능 도메인에 스토리지 풀을 배포할 수 있으므로 애플리케이션

에 특정 성능 및 비용 프로필에 맞는 스토리지를 제공할 수 있습니다. 이러한 성능 도메인에 맞게 크기가 조정되고 최적화된 서버를 선택하는 것은 Red Hat Ceph Storage 클러스터를 설계하는 데 있어 필수적인 부분입니다.

데이터를 읽고 쓰는 Ceph 클라이언트 인터페이스의 경우 Ceph 스토리지 클러스터는 클라이언트가 데이터를 저장하는 간단한 풀로 나타납니다. 그러나 스토리지 클러스터는 클라이언트 인터페이스에 완전히 투명한 방식으로 많은 복잡한 작업을 수행합니다. Ceph OSD라고 하는 Ceph 클라이언트 및 Ceph 개체 스토리지 데몬은 모두 오브젝트 스토리지 및 검색에 대해 CRUSH(Controlled Replication Under Scalable Hashing) 알고리즘을 사용합니다. Ceph OSD는 컨테이너 또는 RPM 기반 배포를 사용하여 스토리지 클러스터 내의 베어 메탈 서버 또는 가상 시스템에서 실행할 수 있습니다.

CRUSH 맵은 클러스터 리소스의 토폴로지를 설명하고 이 맵은 클라이언트 노드와 클러스터 내의 Ceph 모니터 노드에 모두 존재합니다. Ceph 클라이언트 및 Ceph OSD는 모두 CRUSH 맵과 CRUSH 알고리즘을 사용합니다. Ceph 클라이언트는 OSD와 직접 통신하여 중앙 집중식 개체 조회 및 잠재적인 성능 병목 현상을 제거합니다. CRUSH 맵과 해당 피어의 통신을 인식하면 OSD에서 복제, 백업 및 복구 기능을 처리할 수 있으므로 동적 오류 복구를 수행할 수 있습니다.

Ceph는 CRUSH 맵을 사용하여 장애 도메인을 구현합니다. 또한 Ceph는 CRUSH 맵을 사용하여 성능 도메인을 구현합니다. 이 도메인은 단순히 기본 하드웨어의 성능 프로파일을 고려합니다. CRUSH 맵은 Ceph가 데이터를 저장하는 방법을 설명하고, 이는 단순한 계층 구조, 특히 재할용 그래프 및 규칙 세트로 구현됩니다. CRUSH 맵은 여러 계층 구조를 지원하여 한 가지 유형의 하드웨어 성능 프로필을 분리할 수 있습니다. Ceph는 장치 "클래스"로 성능 도메인을 구현합니다.

예를 들어 동일한 Red Hat Ceph Storage 클러스터에 이러한 성능 도메인을 공존할 수 있습니다.

- 일반적으로 HDD(하드 디스크 드라이브)는 비용 및 용량 중심 워크로드에 적합합니다.
- 처리량에 민감한 워크로드는 일반적으로 SSD(반도체 드라이브)에서 Ceph 쓰기 저널과 함께 HDD를 사용합니다.
- MySQL 및 MariaDB와 같은 IOPS 집약적인 워크로드에서는 SSD를 사용하는 경우가 많습니다.

## 워크로드

Red Hat Ceph Storage는 세 가지 주요 워크로드에 최적화되어 있습니다:

- **IOPS 최적화:** IOPS(초당 입력, 출력당) 최적화 배포는 OpenStack에서 가상 시스템으로 RuntimeClass 또는 MariaDB 인스턴스를 실행하는 것과 같은 클라우드 컴퓨팅 작업에 적합합니다. IOPS 최적화된 배포에는 15k RPM SAS 드라이브와 같은 고성능 스토리지 및 자주 쓰기 작업을 처리하기 위해 별도의 SSD 저널이 필요합니다. 일부 IOPS 시나리오에서는 모든 Flash 스토리지를 사용하여 IOPS 및 총 처리량을 개선합니다. IOPS가 최적화된 스토리지 클러스터에는 다음과 같은 속성이 있습니다.

- IOPS당 최소 비용.
- GB당 최대 IOPS.
- 99번째 백분위 대기 시간 일관성.

IOPS에 최적화된 스토리지 클러스터의 용도는 다음과 같습니다.

- 일반적인 블록 스토리지.
- 하드 드라이브(HDD)의 3x 복제 또는 솔리드 스테이트 드라이브(SSD)의 2x 복제.
- OpenStack 클라우드의 MySQL.
- **최적화된 처리량:** 처리량 최적화 배포는 그래픽, 오디오 및 비디오 콘텐츠와 같은 상당한 양의 데

이터를 제공하는 데 적합합니다. 처리량에 최적화된 배포에는 높은 대역폭 네트워크 하드웨어, 컨트롤러 및 하드 디스크 드라이브, 순차적 읽기 및 쓰기 기능이 필요합니다. 빠른 데이터 액세스가 요구 사항인 경우 처리량 최적화 스토리지 전략을 사용합니다. 또한 빠른 쓰기 성능이 필요한 경우 저널에 Solid State Disks (SSD)를 사용하면 쓰기 성능이 크게 향상됩니다. 처리량 최적화 스토리지 클러스터에는 다음과 같은 속성이 있습니다.

- MBps당 가장 낮은 비용(처리량)
- TB당 최대 MBps
- BTU당 최대 MBps
- Watt당 가장 높은 MBps
- 97%의 대기 시간 일관성

처리량 최적화 스토리지 클러스터에 대한 사용은 다음과 같습니다.

- 블록 또는 오브젝트 스토리지
- 3x 복제
- 비디오, 오디오 및 이미지를 위한 활성 성능 스토리지
- 4K 영상 등의 스트리밍 미디어
- **용량 최적화:** 용량 최적화 배포는 많은 양의 데이터를 최대한 저렴하게 저장하는 데 적합합니다. 용량 최적화 배포는 일반적으로 더 큰 가격대에 대한 성능을 거래합니다. 예를 들어 용량 최적화 배포에서는 저널링에 SSD를 사용하지 않고 속도가 느리고 비용이 적게 드는 SATA 드라이브를 사용하는 경우가 많습니다. 비용 및 용량 최적화 스토리지 클러스터에는 다음과 같은 속성이 있습니다.

- TB당 최소 비용
- TB당 최소 BTU 수
- TB당 필요한 최소 Watt

비용 및 용량 최적화 스토리지 클러스터에 대한 사용은 다음과 같습니다.

- 일반적으로 오브젝트 스토리지
- 사용 가능한 용량을 극대화하기 위한 지우기 코딩
- 오브젝트 아카이브
- 비디오, 오디오 및 이미지 오브젝트 리포지토리

## 중요

스토리지 클러스터의 가격 및 성능에 큰 영향을 미칠 수 있으므로 어떤 하드웨어를 구입할지 고려하기 전에 Red Hat Ceph Storage 클러스터에서 실행되는 워크로드를 신중하게 고려합니다. 예를 들어 워크로드가 용량에 최적화되고 처리량이 최적화된 워크로드에 하드웨어가 더 적합한 경우 하드웨어는 필요한 것보다 비용이 많이 듭니다. 반대로 워크로드가 처리량에 최적화되고 하드웨어가 용량이 최적화된 워크로드에 더 적합한 경우 스토리지 클러스터가 성능이 저하될 수 있습니다.



## 2.4. RED HAT CEPH STORAGE의 네트워크 고려 사항

클라우드 스토리지 솔루션의 중요한 측면은 네트워크 대기 시간 및 기타 요인으로 인해 스토리지 클러스터가 IOPS에서 실행될 수 있다는 것입니다. 또한 스토리지 클러스터에 스토리지 용량이 부족하기 전에 대역폭 제약으로 인해 처리량이 부족할 수 있습니다. 즉, 가격 대비 성능 요구 사항을 충족하기 위해 네트워크 하드웨어 구성에서 선택한 워크로드를 지원해야 합니다.

스토리지 관리자는 스토리지 클러스터를 최대한 빨리 복구하는 것을 선호합니다. 저장소 클러스터 네트워크에 대한 대역폭 요구 사항을 신중하게 고려하고 네트워크 링크 초과 서브스크립션에 유의하고 클러스터 내 트래픽을 클라이언트 간 트래픽에서 분리합니다. 또한 SSD(Solid State Disk), 플래시, NVMe 및 기타 고성능 저장 장치의 사용을 고려할 때 네트워크 성능이 점점 더 중요해진다는 점을 고려하십시오.

Ceph는 공용 네트워크 및 스토리지 클러스터 네트워크를 지원합니다. 공용 네트워크는 Ceph 모니터와의 통신 및 클라이언트 트래픽을 처리합니다. 스토리지 클러스터 네트워크는 Ceph OSD 하트비트, 복제, 백필링 및 복구 트래픽을 처리합니다. **최소한** 하나의 10GB 이더넷 링크를 스토리지 하드웨어에 사용해야 하며 연결 및 처리량을 위해 10GB 이더넷 링크를 추가할 수 있습니다.



### 중요

Red Hat은 스토리지 클러스터 네트워크에 대역폭을 할당하는 것이 좋습니다. 따라서 복제된 풀에서 여러 풀에 대한 기준으로 **osd\_pool\_default\_size**를 사용하는 공용 네트워크의 배수입니다. 또한 별도의 네트워크 카드에서 공용 및 스토리지 클러스터 네트워크를 실행하는 것이 좋습니다.



### 중요

Red Hat은 프로덕션 환경에서 Red Hat Ceph Storage 배포에 10GB 이더넷을 사용할 것을 권장합니다. 1GB 이더넷 네트워크는 프로덕션 스토리지 클러스터에 적합하지 않습니다.

드라이브 오류가 발생하는 경우 1GB 이더넷 네트워크에서 1TB의 데이터를 복제하는 데 3시간이 걸리며 3TB의 데이터를 9시간이 소요됩니다. 3TB를 사용하는 것이 일반적인 드라이브 구성입니다. 반대로 10GB 이더넷 네트워크를 사용하면 각각 20분과 1시간이 걸립니다. Ceph OSD가 실패하면 풀 내의 다른 Ceph OSD에 포함된 데이터를 복제하여 스토리지 클러스터를 복구합니다.

랙과 같은 더 큰 도메인에 장애가 발생하면 스토리지 클러스터가 훨씬 더 많은 대역폭을 활용하게 됩니다. 대규모 스토리지 구현에 공통적인 여러 랙으로 구성된 스토리지 클러스터를 구축할 때 최적의 성능을 위해 "팻 트리" 설계의 스위치 간 네트워크 대역폭을 최대한 활용하는 것이 좋습니다. 일반적인 10GB 이더넷 스위치에는 48개의 10GB 포트와 40GB 포트 4개가 있습니다. 최대 처리량을 위해 스파인에서 40GB 포트를 사용합니다. 또는 다른 랙 및 스파인 라우터에 연결하기 위해 사용하지 않는 10GB 포트를 40GB 이상의 포트에 집계하는 것이 좋습니다. 또한 네트워크 인터페이스를 결합하는 데 LACP 모드 4를 사용하는 것이 좋습니다. 또한 점보 프레임, MTU(최대 전송 단위) 9000, 특히 백엔드 또는 클러스터 네트워크에서 사용합니다.

Red Hat Ceph Storage 클러스터를 설치하고 테스트하기 전에 네트워크 처리량을 확인합니다. Ceph에서 대부분의 성능 관련 문제는 일반적으로 네트워킹 문제로 시작합니다. kinked 또는 bent cat-6 케이블과 같은 간단한 네트워크 문제로 인해 대역폭이 저하될 수 있습니다. 전면 네트워크에 최소 10GB 이더넷을 사용합니다. 대규모 클러스터의 경우 백엔드 또는 클러스터 네트워크에 40GB 이더넷을 사용하는 것이 좋습니다.



### 중요

네트워크 최적화를 위해 Red Hat은 대역폭당 CPU를 개선하고 차단되지 않는 네트워크 스위치 백 플레인 에 접보 프레임 을 사용하는 것이 좋습니다. Red Hat Ceph Storage는 공용 네트워크와 클러스터 네트워크 모두에 대해 통신 경로의 모든 네트워킹 장치에 걸쳐 동일한 MTU 값이 필요합니다. 프로덕션에서 Red Hat Ceph Storage 클러스터를 사용하기 전에 MTU 값이 환경의 모든 노드 및 네트워킹 장비에서 같은지 확인합니다.

### 추가 리소스

- 자세한 내용은 *Red Hat Ceph Storage 구성 가이드* 의 [MTU 값 확인 및 구성](#) 을 참조하십시오.

## 2.5. CEPH를 실행할 때 LINUX 커널 튜닝 고려 사항

프로덕션 Red Hat Ceph Storage 클러스터는 일반적으로 운영 체제 조정, 특히 제한 및 메모리 할당의 이 점을 제공합니다. 스토리지 클러스터 내의 모든 노드에 대해 조정이 설정되어 있는지 확인합니다. 추가 지침을 요구하는 Red Hat 지원 케이스를 열 수도 있습니다.

### Ceph OSD의 여유 메모리 예약

Ceph OSD 메모리 할당 요청 중에 메모리 관련 오류가 충분하지 않도록 하려면 예약된 물리적 메모리 양을 설정합니다. Red Hat은 시스템 RAM 용량에 따라 다음 설정을 권장합니다.

- 64GB의 경우 1GB를 예약합니다.

```
vm.min_free_kbytes = 1048576
```

- 128GB의 경우 2GB를 예약합니다.

```
vm.min_free_kbytes = 2097152
```

- 256GB의 경우 3GB를 예약합니다.

```
vm.min_free_kbytes = 3145728
```

### 파일 디스크립터 증가

Ceph Object Gateway는 파일 디스크립터가 부족하면 중지될 수 있습니다. Ceph Object Gateway 노드에서 **/etc/security/limits.conf** 파일을 수정하여 Ceph Object Gateway의 파일 설명자를 늘릴 수 있습니다.

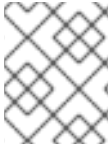
```
ceph soft nofile unlimited
```

### 대규모 스토리지 클러스터의 ulimit 값 조정

대규모 스토리지 클러스터에서 Ceph 관리 명령을 실행하는 경우 예를 들어 1024개의 Ceph OSD가 있는 경우 다음 콘텐츠로 관리 명령을 실행하는 각 노드에 **/etc/security/limits.d/50-ceph.conf** 파일을 생성합니다.

```
USER_NAME soft nproc unlimited
```

**USER\_NAME**을 Ceph 관리 명령을 실행하는 루트가 아닌 사용자 계정의 이름으로 바꿉니다.



## 참고

루트 사용자의 **ulimit** 값은 Red Hat Enterprise Linux에서 기본적으로 **ulimit**으로 이미 설정되어 있습니다.

## 2.6. OSD 노드에서 RAID 컨트롤러 사용 고려 사항

선택적으로 OSD 노드에서 RAID 컨트롤러를 사용하는 것을 고려할 수 있습니다. 다음은 고려해야 할 몇 가지 사항입니다.

- OSD 노드에 1-2GB의 캐시가 설치된 RAID 컨트롤러가 있는 경우 나중 쓰기 캐시를 활성화하면 I/O 쓰기 처리량이 줄어들 수 있습니다. 그러나 캐시는 비휘발성이어야 합니다.
- 대부분의 최신 RAID 컨트롤러에는 전원 손실 이벤트 중에 휘발성 메모리를 비휘발성 NAND 메모리에 트레이닝할 수 있는 충분한 성능을 제공하는 슈퍼 캐패시터가 있습니다. 전원이 복원된 후 특정 컨트롤러와 펌웨어가 작동하는 방식을 이해하는 것이 중요합니다.
- 일부 RAID 컨트롤러에는 수동 조작이 필요합니다. 하드 드라이브는 일반적으로 디스크 캐시를 사용하도록 설정해야 하는지 여부를 운영 체제에 알립니다. 그러나 특정 RAID 컨트롤러와 일부 펌웨어는 이러한 정보를 제공하지 않습니다. 파일 시스템 손상을 방지하기 위해 디스크 수준 캐시가 비활성화되어 있는지 확인합니다.
- 나중 쓰기 캐시가 활성화된 각 Ceph OSD 데이터 드라이브에 다시 쓰기를 설정하여 단일 RAID 0 볼륨을 생성합니다.
- 직렬 연결 SCSI (SAS) 또는 SATA 연결 Solid-state Drive (SSD) 디스크가 RAID 컨트롤러에도 존재하는 경우 컨트롤러 및 펌웨어가 *pass-through* 모드를 지원하는지 확인합니다. *pass-through* 모드를 활성화하면 캐싱 논리를 방지할 수 있으며 일반적으로 빠른 미디어에서 대기 시간이 훨씬 짧아집니다.

## 2.7. OBJECT GATEWAY에서 NVME 사용 고려 사항

선택적으로 Ceph Object Gateway에 NVMe를 사용할 수 있습니다.

Red Hat Ceph Storage의 오브젝트 게이트웨이 기능을 사용하고 OSD 노드에서 NVMe 기반 SSD를 사용하는 경우 프로덕션 [가이드의 Ceph Object Gateway](#) 를 [최적으로 사용하여 NVMe 사용](#) 섹션에서 다음 절차를 고려하십시오. 이 절차에서는 SSD에 저널 및 버킷 인덱스를 함께 배치하는 특수 설계된 Ansible 플레이북을 사용하는 방법을 설명하므로 한 장치에 모든 저널을 사용하는 것보다 성능을 향상시킬 수 있습니다.

## 2.8. RED HAT CEPH STORAGE의 최소 하드웨어 고려 사항

Red Hat Ceph Storage는 비독점 상용 하드웨어에서 실행할 수 있습니다. 소규모 프로덕션 클러스터 및 개발 클러스터는 모드형 하드웨어를 사용하여 성능 최적화 없이 실행될 수 있습니다.

Red Hat Ceph Storage는 베어 메탈 또는 컨테이너화된 배포에 따라 약간 다른 요구 사항이 있습니다.



## 참고

디스크 공간 요구 사항은 `/var/lib/ceph/` 디렉터리에 있는 Ceph 데몬의 기본 경로를 기반으로 합니다.

### 표 2.1. 베어 메탈



프로세스	기준	최소 권장 사항
<b>ceph-osd</b>	프로세서	1x AMD64 또는 Intel 64
	RAM	<b>BlueStore</b> OSD의 경우 Red Hat은 일반적으로 데몬당 5GB의 RAM과 함께 OSD 호스트당 16GB의 RAM으로 기준선을 정하는 것이 좋습니다.
	OS 디스크	호스트당 1x OS 디스크
	블록 스토리지	데몬당 1x 스토리지 드라이브
	<b>block.db</b>	선택 사항이지만 Red Hat은 1x SSD 또는 NVMe 또는 Optane 파티션 또는 데몬당 논리 볼륨을 권장합니다. 오브젝트, 파일 및 혼합된 워크로드의 경우 BlueStore용 <b>block.data</b> 의 크기 조정은 블록 장치, Openstack cinder, Openstack cinder 워크로드용 <b>block.data</b> 의 1%입니다.
	<b>block.wal</b>	데몬당 1x SSD 또는 NVMe 또는 Optane 파티션 또는 논리 볼륨을 선택합니다. 작은 크기(예: 10GB)를 사용하고 <b>block.db</b> 장치보다 빠른 경우에만 사용합니다.
	네트워크	2x 10GB 이더넷 NIC
<b>ceph-mon</b>	프로세서	1x AMD64 또는 Intel 64
	RAM	데몬당 1GB
	디스크 공간	데몬당 15GB
	모니터 디스크	<b>leveldb</b> 모니터 데이터에 필요한 경우, 1x SSD 디스크입니다.
	네트워크	2x 1GB 이더넷 NIC
<b>ceph-mgr</b>	프로세서	1x AMD64 또는 Intel 64
	RAM	데몬당 1GB
	네트워크	2x 1GB 이더넷 NIC
<b>ceph-radosgw</b>	프로세서	1x AMD64 또는 Intel 64
	RAM	데몬당 1GB
	디스크 공간	데몬당 5GB
	네트워크	1x 1GB 이더넷 NIC

프로세스	기준	최소 권장 사항
<b>ceph-mds</b>	프로세서	1x AMD64 또는 Intel 64
	RAM	데몬당 2GB  이 수는 구성 가능한 MDS 캐시 크기에 따라 다릅니다. RAM 요구 사항은 일반적으로 <b>mds_cache_memory_limit</b> 구성 설정에 설정된 양보다 두 배가 됩니다. 이는 전체 시스템 메모리가 아니라 데몬의 메모리이기도 합니다.
	디스크 공간	데몬당 2MB와 로깅에 필요한 공간은 구성된 로그 수준에 따라 다를 수 있습니다.
	네트워크	2x 1GB 이더넷 NIC  OSD와 동일한 네트워크입니다. OSD에 10GB 네트워크가 있는 경우 대기 시간과 관련하여 MDS의 문제가 발생하지 않도록 MDS에서 동일하게 사용해야 합니다.

표 2.2. 컨테이너

프로세스	기준	최소 권장 사항
<b>ceph-osd-container</b>	프로세서	OSD 컨테이너당 1x AMD64 또는 Intel 64 CPU CORE
	RAM	OSD 컨테이너당 최소 5GB의 RAM
	OS 디스크	호스트당 1x OS 디스크
	OSD 스토리지	OSD 컨테이너당 1x 스토리지 드라이브. OS 디스크와 공유할 수 없습니다.
	<b>block.db</b>	선택 사항이지만 Red Hat은 데몬당 1x SSD 또는 NVMe 또는 Optane 파티션 또는 lvm을 권장합니다. 오브젝트, 파일 및 혼합된 워크로드의 경우 BlueStore용 <b>block.data</b> 의 크기 조정은 블록 장치, Openstack cinder, Openstack cinder 워크로드용 <b>block.data</b> 의 1%입니다.
	<b>block.wal</b>	데몬당 1x SSD 또는 NVMe 또는 Optane 파티션 또는 논리 볼륨 옵션으로 사용할 수 있습니다. 작은 크기(예: 10GB)를 사용하고 <b>block.db</b> 장치보다 빠른 경우에만 사용합니다.
	네트워크	2x 10GB 이더넷 NIC, 10GB 권장
<b>ceph-mon-container</b>	프로세서	mon-container당 1x AMD64 또는 Intel 64 CPU CORE
	RAM	<b>mon-container</b> 당 3GB

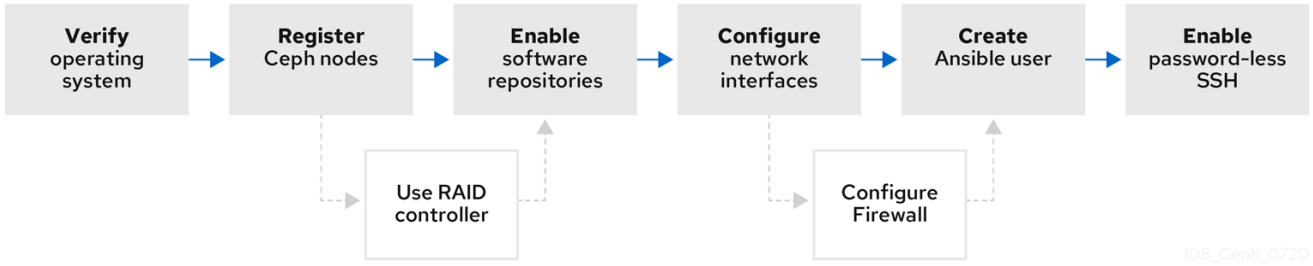
프로세스	기준	최소 권장 사항
	디스크 공간	<b>mon-container</b> 당 10GB, 50GB 권장
	모니터 디스크	선택적으로 <b>Monitor</b> Stensdb 데이터를 위한 1x SSD 디스크
	네트워크	2x 1GB 이더넷 NIC, 10GB 권장
<b>ceph-mgr-container</b>	프로세서	<b>mgr-container</b> 당 1x AMD64 또는 Intel 64 CPU CORE
	RAM	<b>mgr-container</b> 당 3GB
	네트워크	2x 1GB 이더넷 NIC, 10GB 권장
<b>ceph-radosgw-container</b>	프로세서	radosgw-container 당 1x AMD64 또는 Intel 64 CPU CORE
	RAM	데몬당 1GB
	디스크 공간	데몬당 5GB
	네트워크	1x 1GB 이더넷 NIC
<b>ceph-mds-container</b>	프로세서	mds-container 당 1x AMD64 또는 Intel 64 CPU CORE
	RAM	<b>mds-container</b> 당 3GB  이 수는 구성 가능한 MDS 캐시 크기에 따라 다릅니다. RAM 요구 사항은 일반적으로 <b>mds_cache_memory_limit</b> 구성 설정에 설정된 양보다 두 배가 됩니다. 이는 전체 시스템 메모리가 아니라 데몬의 메모리이기도 합니다.
	디스크 공간	2GB의 <b>mds-container</b> 와 가능한 디버그 로깅에 필요한 추가 공간을 고려할 때 20GB가 좋은 출발점입니다.
	네트워크	2x 1GB 이더넷 NIC, 10GB 권장  OSD 컨테이너와 동일한 네트워크입니다. OSD에 10GB 네트워크가 있는 경우 대기 시간과 관련하여 MDS의 문제가 발생하지 않도록 MDS에서 동일하게 사용해야 합니다.

## 2.9. 추가 리소스

- Ceph의 다양한 내부 구성 요소와 해당 구성 요소와 관련된 전략을 자세히 살펴보려면 [Red Hat Ceph Storage Storage Strategies Guide](#) 를 참조하십시오.

### 3장. RED HAT CEPH STORAGE 설치 요구사항

그림 3.1. 사전 요구 사항 워크플로



108\_Ceph\_0720

Red Hat Ceph Storage를 설치하기 전에 다음 요구 사항을 검토하고 그에 따라 각 Monitor, OSD, Metadata Server 및 클라이언트 노드를 준비합니다.



**참고**

Red Hat Ceph Storage 릴리스 및 해당 Red Hat Ceph Storage 패키지 버전에 대해 알아보려면 Red Hat 고객 포털의 [Red Hat Ceph Storage 릴리스 정보 및 해당 Ceph 패키지 버전 문서](#)를 참조하십시오.

#### 3.1. 사전 요구 사항

- [하드웨어](#)가 Red Hat Ceph Storage 4의 최소 요구 사항을 충족하는지 확인합니다.

#### 3.2. RED HAT CEPH STORAGE 설치를 위한 요구 사항 체크리스트

Task	필수 항목	섹션	권장 사항
운영 체제 버전 확인	있음	3.3절. "Red Hat Ceph Storage의 운영 체제 요구 사항"	
Ceph 노드 등록	있음	3.4절. "Red Hat Ceph Storage 노드를 CDN에 등록하고 서브스크립션 연결"	
Ceph 소프트웨어 리포지토리 활성화	있음	3.5절. "Red Hat Ceph Storage 리포지토리 활성화"	
OSD 노드에서 RAID 컨트롤러 사용	없음	2.6절. "OSD 노드에서 RAID 컨트롤러 사용 고려 사항"	RAID 컨트롤러에서 나중 쓰기 캐시를 활성화하면 OSD 노드의 I/O 쓰기 처리량이 증가할 수 있습니다.
네트워크 구성	있음	3.6절. "Red Hat Ceph Storage의 네트워크 구성 확인"	최소한 퍼블릭 네트워크가 필요합니다. 그러나 클러스터 통신용 프라이빗 네트워크를 사용하는 것이 좋습니다.

Task	필수 항목	섹션	권장 사항
방화벽 구성	없음	3.7절. "Red Hat Ceph Storage의 방화벽 설정"	방화벽은 네트워크에 대한 신뢰 수준을 높일 수 있습니다.
Ansible 사용자 생성	있음	3.8절. "sudo 액세스를 사용하여 Ansible 사용자 생성"	모든 Ceph 노드에 Ansible 사용자를 생성해야 합니다.
암호 없는 SSH 활성화	있음	3.9절. "Ansible에 대해 암호 없는 SSH 활성화"	Ansible에 필요합니다.



### 참고

기본적으로 **ceph-ansible**은 NTP/chronyd를 요구 사항으로 설치합니다. NTP/chronyd를 사용자 지정하는 경우 수동으로 [Red Hat Ceph Storage 설치에서 Network Time Protocol for Red Hat Ceph Storage](#) 섹션을 참조하여 NTP/chronyd가 Ceph에서 제대로 작동하도록 구성해야 합니다.

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

Red Hat Enterprise Linux 권한은 Red Hat Ceph Storage 서브스크립션에 포함되어 있습니다.

Red Hat Ceph Storage 4의 초기 릴리스는 Red Hat Enterprise Linux 7.7 또는 Red Hat Enterprise Linux 8.1에서 지원됩니다. Red Hat Ceph Storage 4.3의 현재 버전은 Red Hat Enterprise Linux 7.9, 8.2 EUS, 8.4 EUS, 8.5, 8.6, 8.7, 8.8에서 지원됩니다.

Red Hat Ceph Storage 4는 RPM 기반 배포 또는 컨테이너 기반 배포에서 지원됩니다.



### 중요

Red Hat Enterprise Linux 7에서 실행되는 컨테이너에 Red Hat Ceph Storage 4를 배포하는 경우 Red Hat Enterprise Linux 8 컨테이너 이미지에서 실행되는 Red Hat Ceph Storage 4를 배포합니다.

모든 노드에서 동일한 운영 체제 버전, 아키텍처 및 배포 유형을 사용합니다. 예를 들어, AMD64 및 Intel 64 아키텍처와 함께 노드를 혼합하여 사용하지 마십시오. Red Hat Enterprise Linux 7 및 Red Hat Enterprise Linux 8 운영 체제가 있는 노드가 혼합되거나 RPM 기반 배포 및 컨테이너 기반 배포가 모두 사용되는 노드가 혼합되어 있습니다.



### 중요

Red Hat은 이기종 아키텍처, 운영 체제 버전 또는 배포 유형이 있는 클러스터를 지원하지 않습니다.

## SELinux

기본적으로 SELinux는 **Enforcing** 모드로 설정되고 **ceph-selinux** 패키지가 설치됩니다. SELinux에 대한 자세한 내용은 *Data Security and Hardening Guide*, *Red Hat Enterprise Linux 7 SELinux 사용자 및 관리자 가이드*, *SELinux 가이드* 를 사용하는 Red Hat Enterprise Linux 8을 참조하십시오 .

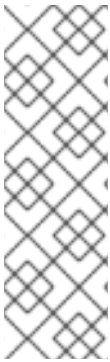
### 추가 리소스

- Red Hat Enterprise Linux 8용 문서는 [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/)에서 확인할 수 있습니다.
- Red Hat Enterprise Linux 7에 대한 문서는 [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/)에서 확인할 수 있습니다.

[요구 사항 체크리스트로 돌아가기](#)

## 3.4. RED HAT CEPH STORAGE 노드를 CDN에 등록하고 서브스크립션 연결

각 Red Hat Ceph Storage 노드를 CDN(Content Delivery Network)에 등록하고 노드가 소프트웨어 리포지토리에 액세스할 수 있도록 적절한 서브스크립션을 연결합니다. 각 Red Hat Ceph Storage 노드는 전체 Red Hat Enterprise Linux 8 기본 콘텐츠와 추가 리포지토리 콘텐츠에 액세스할 수 있어야 합니다. 달리 명시하지 않는 한 스토리지 클러스터의 모든 베어 메탈 및 컨테이너 노드에서 다음 단계를 수행합니다.



### 참고

설치 중에 인터넷에 액세스할 수 없는 베어 메탈 Red Hat Ceph Storage 노드의 경우 Red Hat Satellite 서버를 사용하여 소프트웨어 콘텐츠를 제공합니다. 또는 로컬 Red Hat Enterprise Linux 8 Server ISO 이미지를 마운트하고 Red Hat Ceph Storage 노드 ISO 이미지를 가리키도록 합니다. 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](#) 및 [How to Register Ceph with the Red Hat Customer Portal](#) 문서를 참조하십시오.

### 사전 요구 사항

- 유효한 Red Hat 서브스크립션.
- Red Hat Ceph Storage 노드는 인터넷에 연결할 수 있어야 함.
- Red Hat Ceph Storage 노드에 대한 루트 수준 액세스.

### 절차

1. 컨테이너 배포의 경우 Red Hat Ceph Storage 노드가 배포 중에 인터넷에 액세스할 수 **없는** 경우입니다. 인터넷 액세스가 가능한 노드에서 먼저 다음 단계를 수행해야 합니다.
  - a. 로컬 컨테이너 레지스트리를 시작합니다.

#### Red Hat Enterprise Linux 7

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

#### Red Hat Enterprise Linux 8

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

- b. **registry.redhat.io**가 컨테이너 레지스트리 검색 경로에 있는지 확인합니다.  
/etc/containers/registries.conf 파일을 편집하려면 엽니다.

```
[registries.search]
registries = ['registry.access.redhat.com', 'registry.fedoraproject.org',
'registry.centos.org', 'docker.io']
```

**registry.redhat.io**가 파일에 포함되어 있지 않은 경우 추가합니다.

```
[registries.search]
registries = ['registry.redhat.io', 'registry.access.redhat.com', 'registry.fedoraproject.org',
'registry.centos.org', 'docker.io']
```

- c. Red Hat Customer Portal에서 Red Hat Ceph Storage 4 이미지, Prometheus 이미지 및 대시보드 이미지를 가져옵니다.

### Red Hat Enterprise Linux 7

```
# docker pull registry.redhat.io/rhceph/rhceph-4-rhel8:latest
# docker pull registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
# docker pull registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:latest
# docker pull registry.redhat.io/openshift4/ose-prometheus:v4.6
# docker pull registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
```

### Red Hat Enterprise Linux 8

```
# podman pull registry.redhat.io/rhceph/rhceph-4-rhel8:latest
# podman pull registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
# podman pull registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:latest
# podman pull registry.redhat.io/openshift4/ose-prometheus:v4.6
# podman pull registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
```



### 참고

Red Hat Enterprise Linux 7 및 8은 모두 Red Hat Enterprise Linux 8을 기반으로 동일한 컨테이너 이미지를 사용합니다.

- d. 이미지에 태그를 지정하십시오.  
Prometheus 이미지 태그 버전은 Red Hat Ceph Storage 4.2의 v4.6입니다.

### Red Hat Enterprise Linux 7

```
# docker tag registry.redhat.io/rhceph/rhceph-4-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-rhel8:latest
# docker tag registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-node-exporter:v4.6
# docker tag registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-dashboard-rhel8:latest
# docker tag registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-alertmanager:v4.6
# docker tag registry.redhat.io/openshift4/ose-prometheus:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus:v4.6
```

### 대체 버전

- 로컬 호스트 FQDN을 사용한 `LOCAL_NODE_FQDN`.

## Red Hat Enterprise Linux 8

```
# podman tag registry.redhat.io/rhceph/rhceph-4-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-rhel8:latest
# podman tag registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-node-exporter:v4.6
# podman tag registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-dashboard-rhel8:latest
# podman tag registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-alertmanager:v4.6
# podman tag registry.redhat.io/openshift4/ose-prometheus:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus:v4.6
```

### 대체 버전

- 로컬 호스트 FQDN을 사용한 `LOCAL_NODE_FQDN`.

- e. `/etc/containers/registries.conf` 파일을 편집하고 파일의 포트에 노드의 FQDN을 추가하고 저장합니다.

```
[registries.insecure]
registries = ['LOCAL_NODE_FQDN:5000']
```



### 참고

이 단계는 로컬 Docker 레지스트리에 액세스하는 모든 스토리지 클러스터 노드에서 수행해야 합니다.

- f. 시작한 로컬 Docker 레지스트리로 이미지를 푸시합니다.

## Red Hat Enterprise Linux 7

```
# docker push --remove-signatures LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-rhel8
# docker push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus-node-exporter:v4.6
# docker push --remove-signatures LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-
dashboard-rhel8
# docker push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus-alertmanager:v4.6
# docker push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus:v4.6
```

### 대체 버전

- 로컬 호스트 FQDN을 사용한 `LOCAL_NODE_FQDN`.

## Red Hat Enterprise Linux 8

■



```
# podman push --remove-signatures LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-rhel8
# podman push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus-node-exporter:v4.6
# podman push --remove-signatures LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-
dashboard-rhel8
# podman push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus-alertmanager:v4.6
# podman push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus:v4.6
```

#### 대체 버전

- 로컬 호스트 FQDN을 사용한 `LOCAL_NODE_FQDN`.

- g. Red Hat Enterprise Linux 7의 경우 **docker** 서비스를 다시 시작하십시오.

```
# systemctl restart docker
```



#### 참고

*Red Hat Ceph Storage 노드가 배포 중에 인터넷에 액세스할 수 없는 경우 [all.yml](#) 파일의 예는 Red Hat Ceph Storage 클러스터 설치를 참조하십시오.*

2. 모든 배포의 경우 **베어 메탈** 또는 **컨테이너**에 다음을 수행합니다.

- a. 노드를 등록하고 메시지가 표시되면 적절한 Red Hat Customer Portal 자격 증명을 입력합니다.

```
# subscription-manager register
```

- b. CDN에서 최신 서브스크립션 데이터를 가져옵니다.

```
# subscription-manager refresh
```

- c. Red Hat Ceph Storage에 사용 가능한 모든 서브스크립션을 나열합니다.

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

Red Hat Ceph Storage에 사용 가능한 서브스크립션 목록에서 Pool ID를 복사합니다.

- d. 서브스크립션을 연결합니다.

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

#### 대체 버전

- 이전 단계에서 식별한 풀 ID인 `POOL_ID`.

- e. 기본 소프트웨어 리포지토리를 비활성화하고 해당 Red Hat Enterprise Linux 버전에서 서버 및 추가 리포지토리를 활성화합니다.

## Red Hat Enterprise Linux 7

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

## Red Hat Enterprise Linux 8

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
# subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
```

3. 최신 패키지를 수신하도록 시스템을 업데이트합니다.

a. Red Hat Enterprise Linux 7의 경우:

```
# yum update
```

b. Red Hat Enterprise Linux 8의 경우:

```
# dnf update
```

### 추가 리소스

- Red Hat 서브스크립션 [관리를 위한 Red Hat Subscription Manager 사용 및](#) 구성을 참조하십시오.
- [Red Hat Ceph Storage 리포지토리](#) 활성화를 참조하십시오.

[요구 사항 체크리스트로 돌아가기](#)

## 3.5. 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 Storage 클러스터의 경우 Red Hat Subscription Manager를 사용하여 필수 Ceph 리포지토리를 활성화합니다.
- 로컬 리포지토리  
보안 조치가 인터넷에 액세스하지 못하도록 하는 Ceph Storage 클러스터의 경우 ISO 이미지로 제공되는 단일 소프트웨어 빌드에서 Red Hat Ceph Storage 4를 설치하여 로컬 리포지토리를 설치할 수 있습니다.

### 사전 요구 사항

- 유효한 고객 서브스크립션.
- CDN 설치의 경우:
  - Red Hat Ceph Storage 노드는 인터넷에 연결할 수 있어야 합니다.
  - [CDN에 클러스터 노드를 등록합니다.](#)
- 활성화된 경우 EPEL(Extra Packages for Enterprise Linux) 소프트웨어 리포지토리를 비활성화합니다.

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

## 절차

- CDN 설치의 경우:  
**Ansible** 관리 노드 에서 Red Hat Ceph Storage 4 Tools 리포지토리 및 Ansible 리포지토리를 활성화합니다.

### Red Hat Enterprise Linux 7

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms --enable=rhel-7-server-ansible-2.9-rpms
```

### Red Hat Enterprise Linux 8

```
[root@admin ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
```

- 기본적으로 Red Hat Ceph Storage 리포지토리는 해당 노드에서 **ceph-ansible** 에 의해 활성화됩니다. 리포지토리를 수동으로 활성화하려면 다음을 수행합니다.



## 참고

컨테이너화된 배포에서 이러한 리포지토리가 필요하지 않으므로 활성화하지 마십시오.

**Ceph Monitor** 노드에서 Red Hat Ceph Storage 4 Monitor 리포지토리를 활성화합니다.

### Red Hat Enterprise Linux 7

```
[root@monitor ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-mon-rpms
```

### Red Hat Enterprise Linux 8

```
[root@monitor ~]# subscription-manager repos --enable=rhceph-4-mon-for-rhel-8-x86_64-rpms
```

**Ceph OSD** 노드에서 Red Hat Ceph Storage 4 OSD 리포지토리를 활성화합니다.

### Red Hat Enterprise Linux 7

```
[root@osd ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-osd-rpms
```

### Red Hat Enterprise Linux 8

```
[root@osd ~]# subscription-manager repos --enable=rhceph-4-osd-for-rhel-8-x86_64-rpms
```

다음 노드 유형에서 Red Hat Ceph Storage 4 Tools 리포지토리를 활성화합니다. **RBD** 미러링, **Ceph 클라이언트**, **Ceph 개체 게이트웨이**, **메타데이터 서버**, **NFS**, **iSCSI 게이트웨이** 및 **대시보드 서버**.

## Red Hat Enterprise Linux 7

```
[root@client ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
```

## Red Hat Enterprise Linux 8

```
[root@client ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

- ISO 설치의 경우:
  1. Red Hat 고객 포털에 로그인합니다.
  2. 다운로드를 클릭하여 소프트웨어 및 다운로드 센터를 방문하십시오.
  3. Red Hat Ceph Storage 영역에서 소프트웨어 다운로드를 클릭하여 최신 버전의 소프트웨어를 다운로드합니다.

### 추가 리소스

- Red Hat 서브스크립션 관리 1에 대한 [Red Hat Subscription Manager 사용 및 구성 가이드](#)

[요구 사항 체크리스트로 돌아가기](#)

## 3.6. RED HAT CEPH STORAGE의 네트워크 구성 확인

모든 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 분리된 네트워크에서 제어, 피어링, 복제 및 복구할 수 있도록 클러스터 네트워크용 네트워크 인터페이스 카드가 있을 수 있습니다.

네트워크 인터페이스 설정을 구성하고 변경 사항을 영구적으로 수행해야 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

스토리지 클러스터의 모든 Red Hat Ceph Storage 노드에서 **root** 사용자로 다음 단계를 수행합니다.

1. `/etc/sysconfig/network-scripts/ifcfg-*` 파일에 공용 네트워크 인터페이스 카드에 해당하는 다음 설정이 있는지 확인합니다.
  - a. **BOOTPROTO** 매개변수는 고정 IP 주소에 대해 **none** 으로 설정됩니다.
  - b. **ONBOOT** 매개변수를 **yes** 로 설정해야 합니다.  
이 값이 **no** 로 설정되면 재부팅 시 Ceph 스토리지 클러스터가 피어링되지 않을 수 있습니다.

- c. IPv6 주소를 사용하려는 경우 **IPV6INIT** 와 같은 IPv6 매개변수를 **IPV6\_FAILURE\_FATAL** 매개변수를 제외하고 **yes** 로 설정해야 합니다.  
또한 Ceph 구성 파일 **/etc/ceph/ceph.conf** 를 편집하여 Ceph에 IPv6를 사용하도록 지시하고, Ceph는 IPv4를 사용합니다.

### 추가 리소스

- Red Hat Enterprise Linux 8에 대한 네트워크 인터페이스 스크립트 구성에 대한 자세한 내용은 Red Hat Enterprise Linux 8 [네트워크 구성 가이드의 ifcfg 파일로 IP 네트워킹](#) 구성 장을 참조하십시오.
- 네트워크 구성에 대한 자세한 내용은 Red Hat [Ceph Storage 4용 구성 가이드의 Ceph 네트워크](#) 구성 섹션을 참조하십시오.

[요구 사항 체크리스트로 돌아가기](#)

## 3.7. RED HAT CEPH STORAGE의 방화벽 설정

Red Hat Ceph Storage는 **firewalld** 서비스를 사용합니다. **firewalld** 서비스에는 각 데몬의 포트 목록이 포함되어 있습니다.

Ceph 모니터 데몬은 Ceph 스토리지 클러스터 내에서의 통신에 포트 **3300** 및 **6789** 를 사용합니다.

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

- 클라이언트와 통신하고 공용 네트워크를 통해 모니터링하기 위한 하나
- 클러스터 네트워크를 통해 다른 OSD로 데이터를 전송하는 방법(사용 가능한 경우), 공용 네트워크를 통해 데이터 전송
- 클러스터 네트워크를 통해 하트비트 패킷의 교환, 사용 가능한 경우; 그렇지 않으면 공용 네트워크를 통해

Ceph Manager(**ceph-mgr**) 데몬은 범위 **6800-7300** 의 포트를 사용합니다. 동일한 노드에서 Ceph Monitors를 사용하여 **ceph-mgr** 데몬을 조정하는 것이 좋습니다.

Ceph Metadata Server 노드(**ceph-mds**)는 포트 범위 **6800-7300** 을 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.

**firewalld** 가 활성화된 경우 다음 단계는 선택 사항입니다. 기본적으로 **ceph-ansible** 에는 **group\_vars/all.yml** 의 아래 설정이 포함되어 있으며 해당 포트가 자동으로 열립니다.

```
configure_firewall: True
```

### 사전 요구 사항

- 네트워크 하드웨어가 연결되어 있습니다.
- 스토리지 클러스터의 모든 노드에 **root** 또는 **sudo** 액세스 권한을 부여합니다.

### 절차

1. 스토리지 클러스터의 모든 노드에서 **firewalld** 서비스를 시작합니다. 부팅 시 실행되도록 활성화하고 실행 중인지 확인합니다.

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

2. 모든 모니터 노드에서 공용 네트워크에서 포트 **3300** 및 **6789** 를 엽니다.

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

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' --permanent
```

#### 대체 버전

- 모니터 노드의 네트워크 주소가 있는 *IP\_ADDRESS*.
- CIDR 표기법의 넷마스크가 포함된 *NETMASK\_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' --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
[root@osd ~]# firewall-cmd --permanent --add-service=ceph
[root@osd ~]# firewall-cmd --add-service=ceph
```

별도의 클러스터 네트워크가 있는 경우 적절한 영역을 사용하여 명령을 반복합니다.

4. 모든 Ceph Manager(**ceph-mgr**) 노드에서 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 적절한 영역을 사용하여 명령을 반복합니다.

5. 모든 Ceph Metadata Server(**ceph-mds**) 노드에서 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 적절한 영역을 사용하여 명령을 반복합니다.

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

a. 기본 Ansible 구성 포트 **8080** 을 시작하려면 다음을 수행합니다.

```
[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
```

대체 버전

- 모니터 노드의 네트워크 주소가 있는 *IP\_ADDRESS*.
- CIDR 표기법의 넷마스크가 포함된 *NETMASK\_PREFIX*

예제

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

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

b. 선택적으로 Ansible을 사용하여 Ceph Object Gateway를 설치하고 Ansible이 **8080** 에서 포트 **80** 으로 사용할 Ceph Object Gateway를 구성하는 기본 포트를 변경한 경우, 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 주소에 따라 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
```

#### 대체 버전

- 모니터 노드의 네트워크 주소가 있는 *IP\_ADDRESS*.
- CIDR 표기법의 넷마스크가 포함된 *NETMASK\_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 주소에 따라 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
```

#### 대체 버전

- 모니터 노드의 네트워크 주소가 있는 *IP\_ADDRESS*.
- CIDR 표기법의 넷마스크가 포함된 *NETMASK\_PREFIX*

#### 예제

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

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



- 공용 및 클러스터 네트워크에 대한 자세한 내용은 [Red Hat Ceph Storage의 네트워크 구성 확인](#)을 참조하십시오.
- **firewalld**에 대한 자세한 내용은 Red Hat Enterprise Linux 8용 네트워크 [보안 가이드의 방화벽 사용 및 구성](#) 장을 참조하십시오.

[요구 사항 체크리스트로 돌아가기](#)

### 3.8. SUDO 액세스를 사용하여 ANSIBLE 사용자 생성

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

#### 사전 요구 사항

- 스토리지 클러스터의 모든 노드에 **root** 또는 **sudo** 액세스 권한을 부여합니다.

#### 절차

1. **root** 사용자로 노드에 로그인합니다.

```
ssh root@HOST_NAME
```

#### 대체 버전

- *HOST\_NAME* 및 Ceph 노드의 호스트 이름입니다.

#### 예제

```
# ssh root@mon01
```

메시지가 표시되면 **root** 암호를 입력합니다.

2. 새 Ansible 사용자를 생성합니다.

```
adduser USER_NAME
```

#### 대체 버전

- *USER\_NAME* - Ansible 사용자의 새 사용자 이름입니다.

#### 예제

```
# adduser admin
```



#### 중요

**ceph**를 사용자 이름으로 사용하지 마십시오. **ceph** 사용자 이름은 Ceph 데몬용으로 예약되어 있습니다. 클러스터 전체의 균일한 사용자 이름은 쉽게 사용할 수 있지만 일반적으로 무차별 공격에 사용되므로 명확한 사용자 이름을 사용하지 않도록 합니다.

- 이 사용자의 새 암호를 설정합니다.

```
# passwd USER_NAME
```

#### 대체 버전

- `USER_NAME` - Ansible 사용자의 새 사용자 이름입니다.

#### 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
```

#### 대체 버전

- `USER_NAME` - Ansible 사용자의 새 사용자 이름입니다.

#### 예제

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

- 새 파일에 올바른 파일 권한을 할당합니다.

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

#### 대체 버전

- `USER_NAME` - Ansible 사용자의 새 사용자 이름입니다.

#### 예제

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

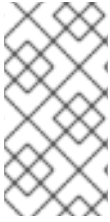
#### 추가 리소스

- 기본 시스템 설정 구성 가이드 Red Hat Enterprise Linux 8의 [사용자 계정 관리](#) 섹션

[요구 사항 체크리스트로 돌아가기](#)

## 3.9. ANSIBLE에 대해 암호 없는 SSH 활성화

Ansible 관리 노드에서 SSH 키 쌍을 생성하고, 암호를 입력하라는 메시지가 표시되지 않고 Ansible이 노드에 액세스할 수 있도록 스토리지 클러스터의 각 노드에 공개 키를 배포합니다.



## 참고

Cockpit 웹 기반 인터페이스를 사용하여 Red Hat Ceph Storage를 설치하는 경우에는 이 절차가 필요하지 않습니다. Cockpit Ceph 설치 프로그램이 자체 SSH 키를 생성하기 때문입니다. [Cockpit SSH 키를 클러스터의 모든 노드에 복사하는 방법은 Cockpit 웹 인터페이스를 사용하여 Red Hat Ceph Storage 설치](#) 장에 있습니다.

## 사전 요구 사항

- Ansible 관리 노드에 액세스
- [sudo 액세스 권한을 사용하여 Ansible 사용자 만들기](#).

## 절차

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

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

2. 공개 키를 스토리지 클러스터의 모든 노드에 복사합니다.

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

### 대체 버전

- *USER\_NAME* - Ansible 사용자의 새 사용자 이름입니다.
- *HOST\_NAME* 및 Ceph 노드의 호스트 이름입니다.

### 예제

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

3. 사용자의 SSH **config** 파일을 생성합니다.

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

4. **config** 파일을 편집하려면 `을` 엽니다. 스토리지 클러스터의 각 노드에 대한 **Hostname** 및 **User** 옵션의 값을 설정합니다.

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

### 대체 버전

- `HOST_NAME` 및 Ceph 노드의 호스트 이름입니다.
- `USER_NAME` - Ansible 사용자의 새 사용자 이름입니다.

### 예제

```
Host node1
  Hostname monitor
  User admin
Host node2
  Hostname osd
  User admin
Host node3
  Hostname gateway
  User admin
```



### 중요

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

5. `~/.ssh/config` 파일에 대한 올바른 파일 권한을 설정합니다.

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

### 추가 리소스

- **ssh\_config(5)** 매뉴얼 페이지.
- Red Hat Enterprise Linux 8용 [네트워크 보안](#)의 **OpenSSH**를 사용하여 두 시스템 간에 보안 통신 사용 장을 참조하십시오.

[요구 사항 체크리스트로 돌아가기](#)

## 4장. COCKPIT 웹 인터페이스를 사용하여 RED HAT CEPH STORAGE 설치

이 장에서는 Cockpit 웹 기반 인터페이스를 사용하여 Red Hat Ceph Storage 클러스터 및 메타데이터 서버, Ceph 클라이언트 또는 Ceph Object Gateway와 같은 기타 구성 요소를 설치하는 방법을 설명합니다.

이 프로세스는 Cockpit Ceph 설치 프로그램 설치, Cockpit에 로그인한 후 설치 프로그램 내의 다른 페이지를 사용하여 클러스터 설치 구성 및 시작으로 구성됩니다.



### 참고

Cockpit Ceph 설치 프로그램은 Ansible 및 **ceph-ansible** RPM에서 제공하는 Ansible 플레이북을 사용하여 실제 설치를 수행합니다. Cockpit 없이 이러한 플레이북을 사용하여 Ceph를 설치할 수 있습니다. 해당 프로세스는 이 장과 관련이 있으며 [직접 Ansible 설치 또는 Ansible 플레이북을 직접 사용하는 것](#)입니다.



### 중요

Cockpit Ceph 설치 프로그램은 현재 IPv6 네트워킹을 지원하지 않습니다. IPv6 네트워킹이 필요한 경우 [Ansible 플레이북을 사용하여 Ceph를 직접 설치](#)합니다.



### 참고

Ceph의 관리 및 모니터링에 사용되는 대시보드 웹 인터페이스는 기본적으로 Cockpit이 백엔드에서 사용하는 **ceph-ansible** RPM의 Ansible 플레이북에서 설치됩니다. 따라서 Ansible 플레이북을 직접 사용하거나 Cockpit을 사용하여 Ceph를 설치할 때 대시보드 웹 인터페이스도 설치됩니다.

### 4.1. 사전 요구 사항

- Ansible Red Hat Ceph Storage를 직접 설치하는 데 필요한 [일반 사전 요구 사항](#)을 완료합니다.
- Firefox 또는 Chrome 최신 버전입니다.
- 다중 네트워크를 사용하여 클러스터 내부 트래픽, 클라이언트 간 트래픽, RADOS Gateway 트래픽 또는 iSCSI 트래픽을 세그먼트하는 경우 관련 네트워크가 이미 호스트에 이미 구성되어 있는지 확인합니다. 자세한 내용은 [Cockpit Ceph 설치 프로그램의 네트워크 페이지 완료에 대한 하드웨어 가이드](#)의 네트워크 고려 사항 및 이 장에 있는 섹션을 참조하십시오.
- Cockpit 웹 기반 인터페이스의 기본 포트(**9090**)에 액세스할 수 있는지 확인합니다.

### 4.2. 설치 요구 사항

- Ansible 관리 노드로 작동할 노드 1개입니다.
- 성능 지표 및 경고 플랫폼을 제공하는 하나의 노드입니다. 이는 Ansible 관리 노드와 함께 배치될 수 있습니다.
- Ceph 클러스터를 구성하는 하나 이상의 노드입니다. 설치 프로그램은 [개발/POC](#) 라는 옵션을 설치할 수 있으며 데이터 복제는 기본적으로 호스트 수준 보호가 아닌 디스크로 설정됩니다.

### 4.3. COCKPIT CEPH 설치 및 구성

Cockpit Ceph 설치 프로그램을 사용하여 Red Hat Ceph Storage 클러스터를 설치하려면 먼저 Ansible 관리 노드에 Cockpit Ceph 설치 프로그램을 설치해야 합니다.

### 사전 요구 사항

- Ansible 관리 노드에 대한 루트 수준 액세스.
- Ansible 애플리케이션과 함께 사용할 **ansible** 사용자 계정입니다.

### 절차

1. Cockpit이 설치되었는지 확인합니다.

```
$ rpm -q cockpit
```

예제:

```
[admin@jb-ceph4-admin ~]$ rpm -q cockpit
cockpit-196.3-1.el8.x86_64
```

위의 예제와 유사한 출력이 표시되면 *Cockpit* 확인 단계로 건너뛵니다. 출력이 **package cockpit** 이 설치되지 않은 경우 *Cockpit* 설치 단계를 계속합니다.

2. 선택 사항: Cockpit을 설치합니다.

- a. Red Hat Enterprise Linux 8의 경우:

```
# dnf install cockpit
```

- b. Red Hat Enterprise Linux 7의 경우:

```
# yum install cockpit
```

3. Cockpit이 실행 중인지 확인합니다.

```
# systemctl status cockpit.socket
```

출력에 **Active: active(가져오기)** 가 표시되면 *Red Hat Ceph Storage*용 *Cockpit* 플러그인 설치 단계로 건너뛵니다. **Active: inactive(dead)** 가 표시되는 경우 *Cockpit* 활성화 단계를 계속 진행합니다.

4. 선택 사항: Cockpit을 활성화합니다.

- a. **systemctl** 명령을 사용하여 Cockpit을 활성화합니다.

```
# systemctl enable --now cockpit.socket
```

다음과 같은 행이 표시됩니다.

```
Created symlink /etc/systemd/system/sockets.target.wants/cockpit.socket →
/usr/lib/systemd/system/cockpit.socket.
```

- b. Cockpit이 실행 중인지 확인합니다.

■

```
# systemctl status cockpit.socket
```

다음과 같은 행이 표시됩니다.

```
Active: active (listening) since Tue 2020-01-07 18:49:07 EST; 7min ago
```

5. Red Hat Ceph Storage용 Cockpit Ceph Installer를 설치합니다.

a. Red Hat Enterprise Linux 8의 경우:

```
# dnf install cockpit-ceph-installer
```

b. Red Hat Enterprise Linux 7의 경우:

```
# yum install cockpit-ceph-installer
```

6. Ansible 사용자로 sudo를 사용하여 컨테이너 카탈로그에 로그인합니다.



### 참고

기본적으로 Cockpit Ceph 설치 프로그램은 **root** 사용자를 사용하여 Ceph를 설치합니다. 사전 요구 사항의 일부로 생성된 Ansible 사용자를 사용하여 Ceph를 설치하려면 Ansible 사용자로 **sudo** 를 사용하여 이 절차의 나머지 명령을 실행합니다.

### Red Hat Enterprise Linux 7

```
$ sudo docker login -u CUSTOMER_PORTAL_USERNAME https://registry.redhat.io
```

#### 예제

```
[admin@jb-ceph4-admin ~]$ sudo docker login -u myusername https://registry.redhat.io
Password:
Login Succeeded!
```

### Red Hat Enterprise Linux 8

```
$ sudo podman login -u CUSTOMER_PORTAL_USERNAME https://registry.redhat.io
```

#### 예제

```
[admin@jb-ceph4-admin ~]$ sudo podman login -u myusername https://registry.redhat.io
Password:
Login Succeeded!
```

7. **registry.redhat.io**가 컨테이너 레지스트리 검색 경로에 있는지 확인합니다.

a. **/etc/containers/registries.conf** 파일을 편집하려면 엽니다.

```
[registries.search]
registries = ['registry.access.redhat.com', 'registry.fedoraproject.org',
'registry.centos.org', 'docker.io']
```

**registry.redhat.io**가 파일에 포함되어 있지 않은 경우 추가합니다.

```
[registries.search]
registries = ['registry.redhat.io', 'registry.access.redhat.com', 'registry.fedoraproject.org',
'registry.centos.org', 'docker.io']
```

8. Ansible 사용자로 sudo를 사용하여 **ansible-runner-service** 를 시작합니다.

```
$ sudo ansible-runner-service.sh -s
```

#### 예제

```
[admin@jb-ceph4-admin ~]$ sudo ansible-runner-service.sh -s
Checking environment is ready
Checking/creating directories
Checking SSL certificate configuration
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
writing RSA key
Signature ok
subject=C = US, ST = North Carolina, L = Raleigh, O = Red Hat, OU = RunnerServer, CN =
jb-ceph4-admin
Getting CA Private Key
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
..+++++
e is 65537 (0x010001)
writing RSA key
Signature ok
subject=C = US, ST = North Carolina, L = Raleigh, O = Red Hat, OU = RunnerClient, CN = jb-
ceph4-admin
Getting CA Private Key
Setting ownership of the certs to your user account(admin)
Setting target user for ansible connections to admin
Applying SELINUX container_file_t context to '/etc/ansible-runner-service'
Applying SELINUX container_file_t context to '/usr/share/ceph-ansible'
Ansible API (runner-service) container set to rhceph/ansible-runner-rhel8:latest
Fetching Ansible API container (runner-service). Please wait...
Trying to pull registry.redhat.io/rhceph/ansible-runner-rhel8:latest...Getting image source
signatures
Copying blob c585fd5093c6 done
Copying blob 217d30c36265 done
Copying blob e61d8721e62e done
Copying config b96067ea93 done
Writing manifest to image destination
Storing signatures
b96067ea93c8d6769eaea86854617c63c61ea10c4ff01ecf71d488d5727cb577
```



```
Starting Ansible API container (runner-service)
Started runner-service container
Waiting for Ansible API container (runner-service) to respond
The Ansible API container (runner-service) is available and responding to requests
```

Login to the cockpit UI at <https://jb-ceph4-admin:9090/cockpit-ceph-installer> to start the install

출력의 마지막 행에는 Cockpit Ceph 설치 프로그램에 대한 URL이 포함됩니다. 위의 예에서 URL은 <https://jb-ceph4-admin:9090/cockpit-ceph-installer> 입니다. 사용자 환경에 인쇄된 URL을 기록해 두십시오.

## 4.4. COCKPIT CEPH INSTALLER SSH 키를 클러스터의 모든 노드에 복사

Cockpit Ceph 설치 프로그램은 SSH를 사용하여 클러스터에 노드를 연결하고 구성합니다. 이를 위해 설치 프로그램이 자동으로 SSH 키 쌍을 생성하므로 암호를 입력하라는 메시지가 표시되지 않고 노드에 액세스할 수 있습니다. SSH 공개 키는 클러스터의 모든 노드로 전송되어야 합니다.

### 사전 요구 사항

- `sudo` 액세스 권한이 있는 Ansible 사용자가 생성되었습니다.
- Cockpit Ceph 설치 프로그램이 설치 및 구성되어 있습니다.

### 절차

1. Ansible 관리 노드에 Ansible 사용자로 로그인합니다.

```
ssh ANSIBLE_USER@HOST_NAME
```

예제:

```
$ ssh admin@jb-ceph4-admin
```

2. SSH 공개 키를 첫 번째 노드에 복사합니다.

```
sudo ssh-copy-id -f -i /usr/share/ansible-runner-service/env/ssh_key.pub
_ANSIBLE_USER @_HOST_NAME_
```

예제:

```
$ sudo ssh-copy-id -f -i /usr/share/ansible-runner-service/env/ssh_key.pub admin@jb-ceph4-
mon
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/usr/share/ansible-runner-
service/env/ssh_key.pub"
admin@192.168.122.182's password:
```

```
Number of key(s) added: 1
```

```
Now try logging into the machine, with: "ssh 'admin@jb-ceph4-mon'"
and check to make sure that only the key(s) you wanted were added.
```

클러스터의 모든 노드에 대해 이 단계를 반복합니다.

## 4.5. COCKPIT에 로그인합니다.

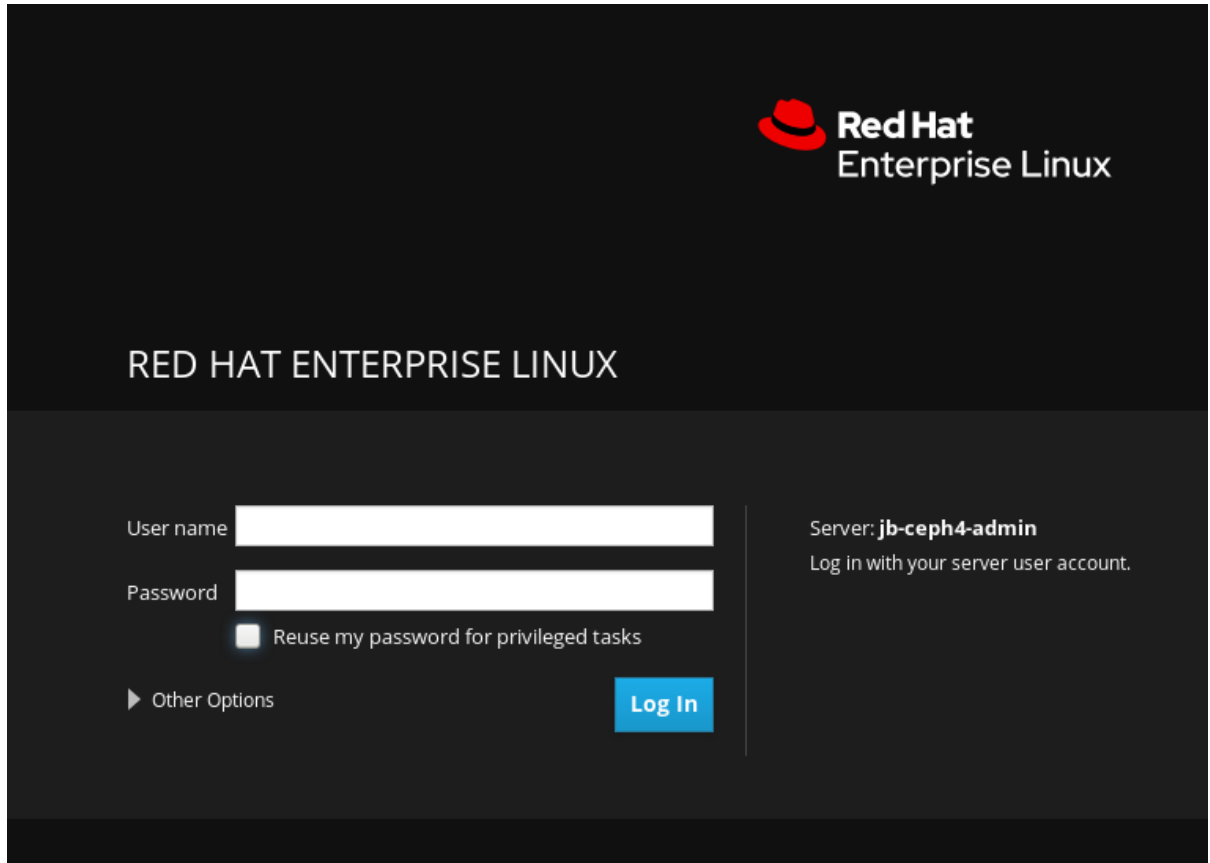
Cockpit에 로그인하면 Cockpit Ceph 설치 프로그램 웹 인터페이스를 볼 수 있습니다.

### 사전 요구 사항

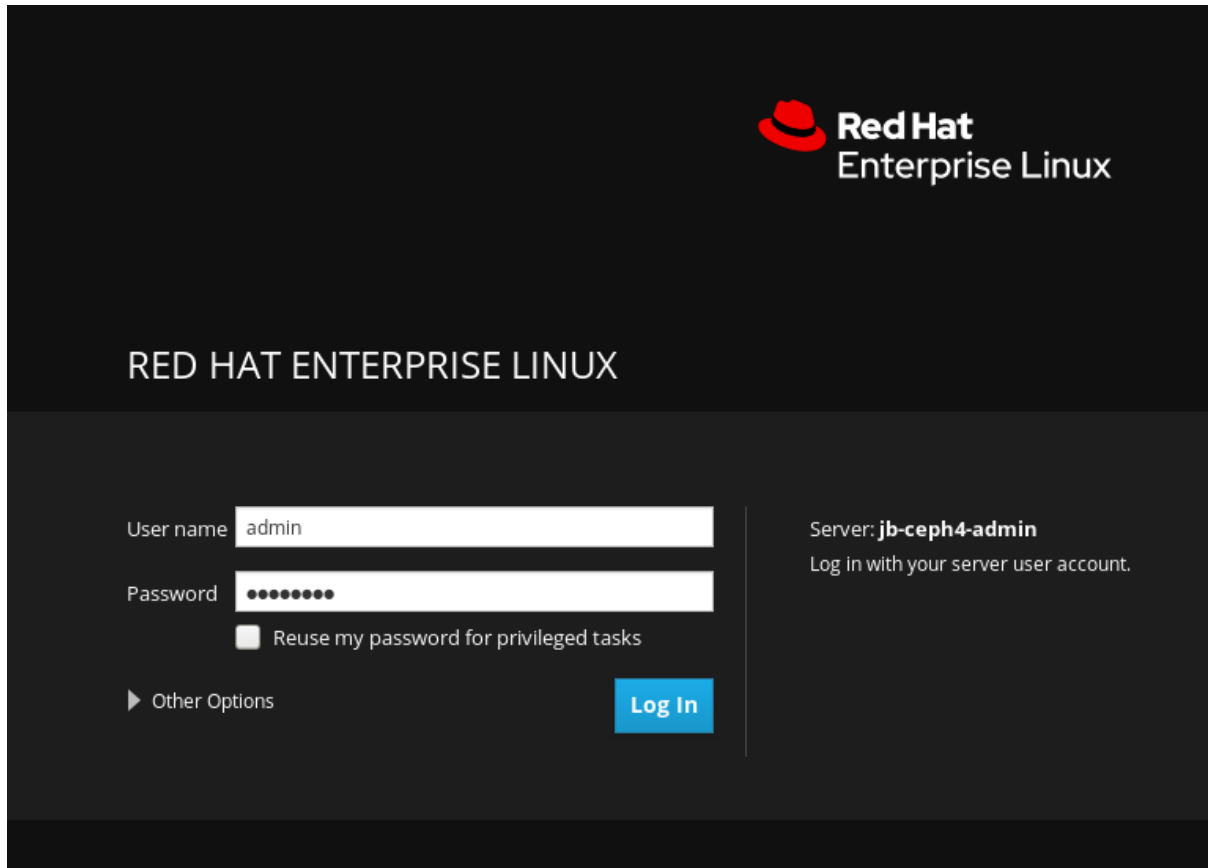
- Cockpit Ceph 설치 프로그램이 [설치 및 구성되어 있습니다](#).
- Cockpit Ceph 설치 프로그램 구성의 일부로 URL이 인쇄되어 있습니다.

### 절차

1. 웹 브라우저에서 URL을 엽니다.

The image shows a login interface for Red Hat Enterprise Linux. At the top right, there is the Red Hat logo and the text "Red Hat Enterprise Linux". Below this, the text "RED HAT ENTERPRISE LINUX" is centered. The main area contains a login form with two input fields: "User name" and "Password". Below the "Password" field is a checkbox labeled "Reuse my password for privileged tasks". To the left of the "Log In" button is a link "Other Options". To the right of the form, the text "Server: jb-ceph4-admin" and "Log in with your server user account." is displayed. The "Log In" button is a blue rectangle with white text.

2. Ansible 사용자 이름과 암호를 입력합니다.



Red Hat Enterprise Linux

RED HAT ENTERPRISE LINUX

User name

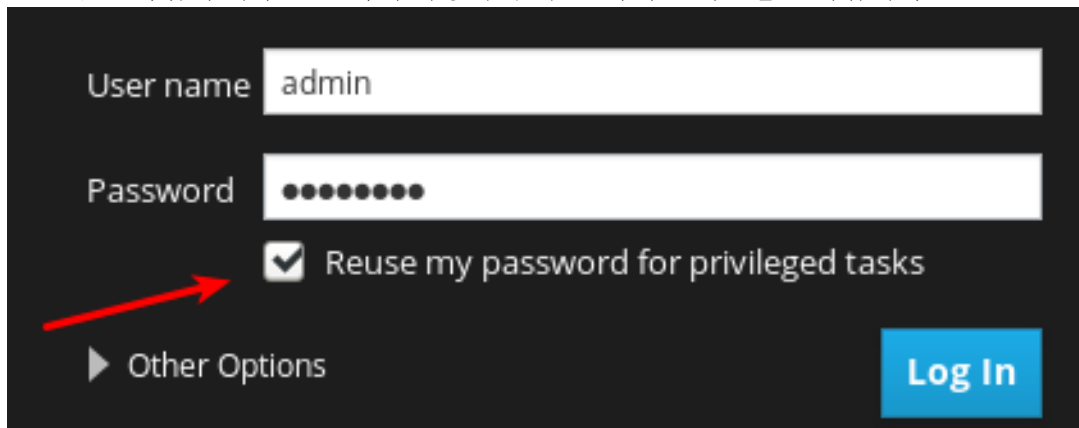
Password

Reuse my password for privileged tasks

▶ Other Options

Server: **jb-ceph4-admin**  
Log in with your server user account.

3. 권한 있는 작업에 내 암호를 다시 사용하기 위한 라디오 버튼을 클릭합니다.



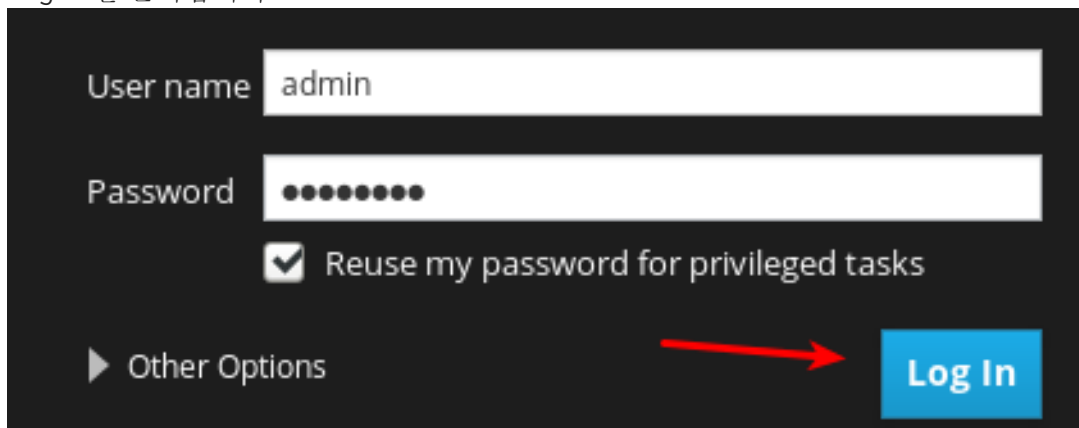
User name

Password

Reuse my password for privileged tasks

▶ Other Options

4. Log In 을 클릭합니다.



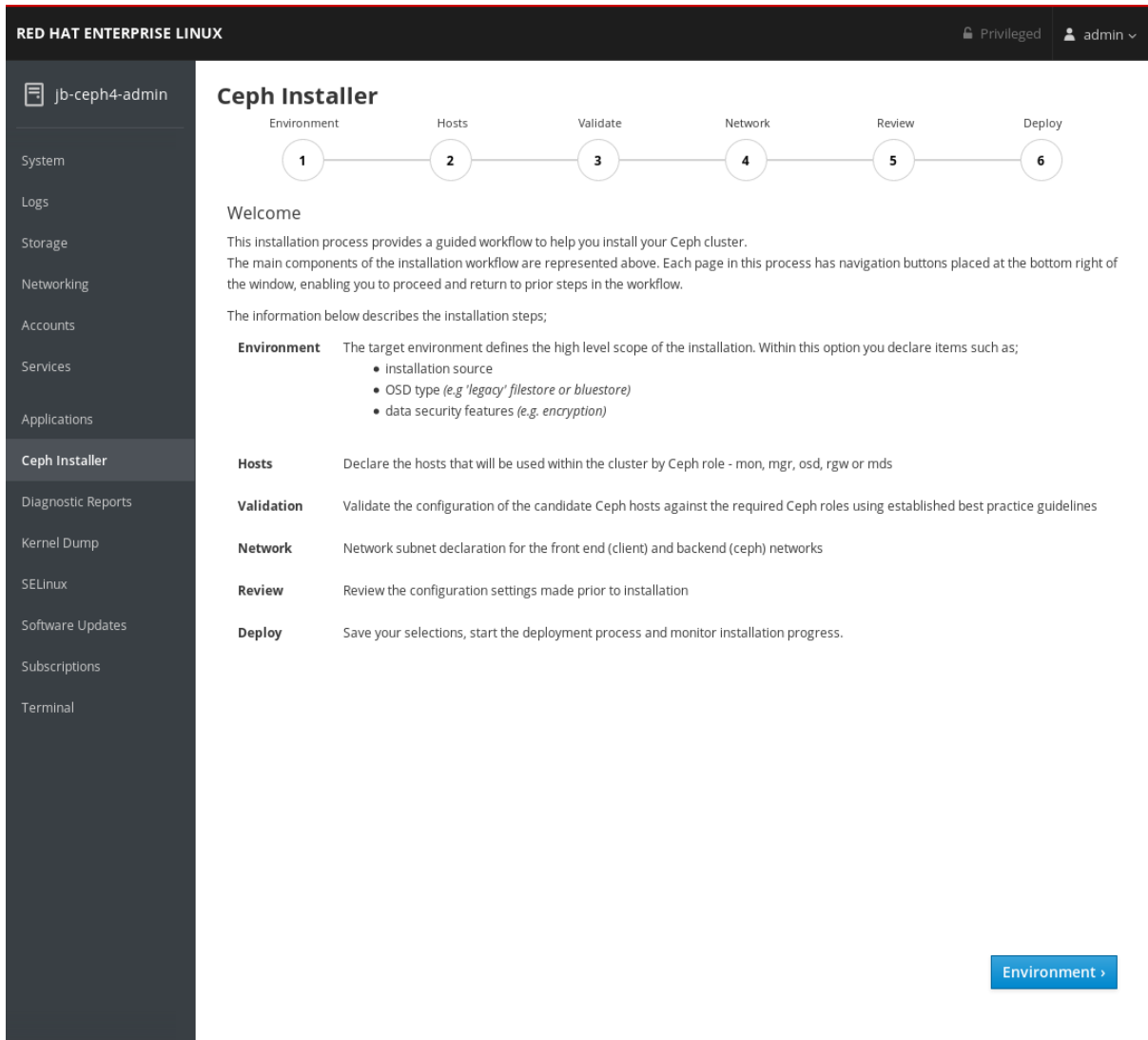
User name

Password

Reuse my password for privileged tasks

▶ Other Options

5. 시작 페이지를 검토하여 설치 관리자의 작동 방식 및 설치 프로세스의 전반적인 흐름을 확인합니다.



시작 페이지의 정보를 검토한 후 웹 페이지의 오른쪽 아래에 있는 환경 버튼을 클릭합니다.

## 4.6. COCKPIT CEPH 설치 프로그램의 환경 페이지 완료

환경 페이지를 사용하면 사용할 설치 소스 및 스토리지에 하드 디스크 드라이브(HDD) 및 Solid State Drives(SSD)를 사용하는 방법과 같이 클러스터의 전반적인 측면을 구성할 수 있습니다.

### 사전 요구 사항

- Cockpit Ceph 설치 프로그램이 설치 및 구성되어 있습니다.
- Cockpit Ceph 설치 프로그램 구성의 일부로 URL이 인쇄됩니다.
- 레지스트리 서비스 계정을 생성했습니다.

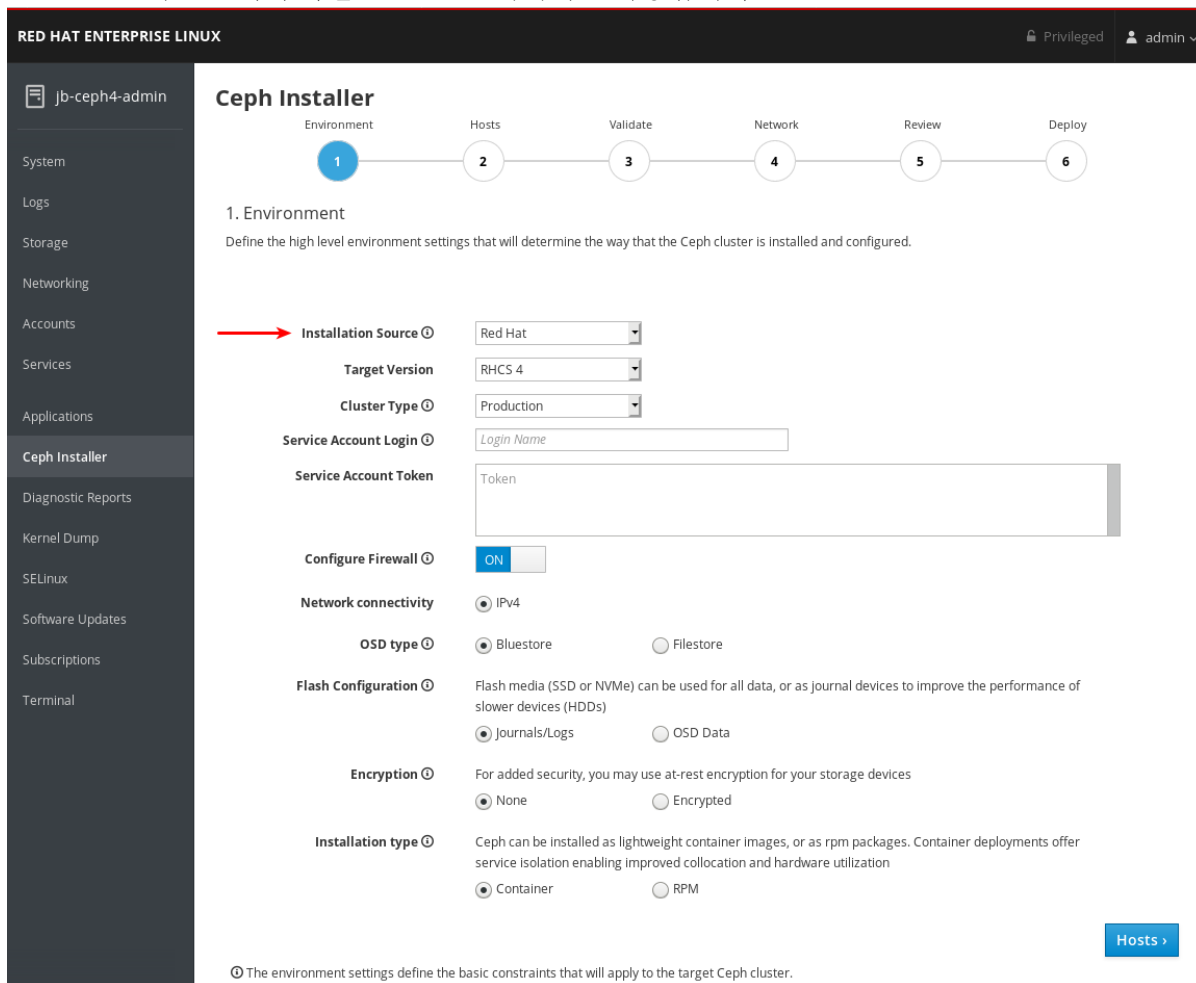


### 참고

따를 대화 상자에 일부 설정의 오른쪽에는 툴팁이 있습니다. 이를 보려면 점선을 가진 *i*와 같은 아이콘 위에 마우스 커서를 올려 놓습니다.

### 절차

1. 설치 소스를 선택합니다. Red Hat Subscription Manager 또는 ISO에서 리포지토리를 사용하여 Red Hat 고객 포털에서 다운로드한 CD 이미지를 사용합니다.



Red Hat 을 선택하면 다른 옵션 없이 *Target Version* 이 *RHCS 4* 로 설정됩니다. *ISO* 를 선택하면 대상 버전이 *ISO* 이미지 파일로 설정됩니다.



### 중요

ISO를 선택하는 경우 이미지 파일이 `/usr/share/ansible-runner-service/iso` 디렉터리에 있어야 하며 SELinux 컨텍스트를 `container_file_t` 로 설정해야 합니다.



### 중요

설치 소스에 대한 커뮤니티 및 배포 옵션은 지원되지 않습니다.

2. 클러스터 유형을 선택합니다. CPU 번호 및 메모리 크기와 같은 특정 리소스 요구 사항이 충족되지 않는 경우 프로덕션 선택을 통해 설치가 진행되지 않습니다. 리소스 요구 사항이 충족되지 않는 경우에도 클러스터 설치를 허용하려면 *Development/POC* 를 선택합니다.



5. 현재 Cockpit Ceph 설치 프로그램은 IPv4 만 지원합니다. IPv6 지원이 필요한 경우 Cockpit Ceph Installer 를 그대로 사용하고 [Ansible 스크립트를 사용하여 Ceph 설치를 진행합니다.](#)

**Configure Firewall** ⓘ  ON

**Network connectivity**  IPv4 

**OSD type** ⓘ  Bluestore  Filestore

6. OSD 유형을 BlueStore 또는 FileStore 로 설정합니다.

**Network connectivity**  IPv4

 **OSD type** ⓘ  Bluestore  Filestore



### 중요

bluestore가 기본 OSD 유형입니다. 이전에는 Ceph에서 FileStore를 오브젝트 저장 소로 사용했습니다. BlueStore는 더 많은 기능과 향상된 성능을 제공하기 때문에 이 형식은 새로운 Red Hat Ceph Storage 4.0 설치에서 더 이상 사용되지 않습니다. FileStore를 계속 사용할 수는 있지만 이를 사용하려면 지원 예외가 필요합니다. BlueStore에 대한 자세한 내용은 [아키텍처 가이드의 Ceph BlueStore](#) 를 참조하십시오.

7. Flash Configuration 을 journal /Logs 또는 OSD 데이터로 설정합니다. NVMe 또는 기존 SATA/SAS 인터페이스를 사용하는지 여부에 관계없이 Solid State Drives(SSD)가 있는 경우 실제 데이터가 하드 디스크 드라이브(HDD)로 전환되거나 저널, 로그, 데이터에 대해 SSD를 사용할 수 있고 모든 Ceph OSD 기능에 대해 SSD를 사용할 수 있지만 NVMe/SAS 인터페이스를 사용할 수 있습니다.

**Flash Configuration** ⓘ Flash media (SSD or NVMe) can be used for all data, or as journal devices to improve the performance of slower devices (HDDs)

Journals/Logs  OSD Data

8. Encryption 을 None 또는 Encrypted 로 설정합니다. LUKS1 형식을 사용하여 스토리지 장치의 미 사용 암호화를 나타냅니다.

**Encryption** ⓘ For added security, you may use at-rest encryption for your storage devices

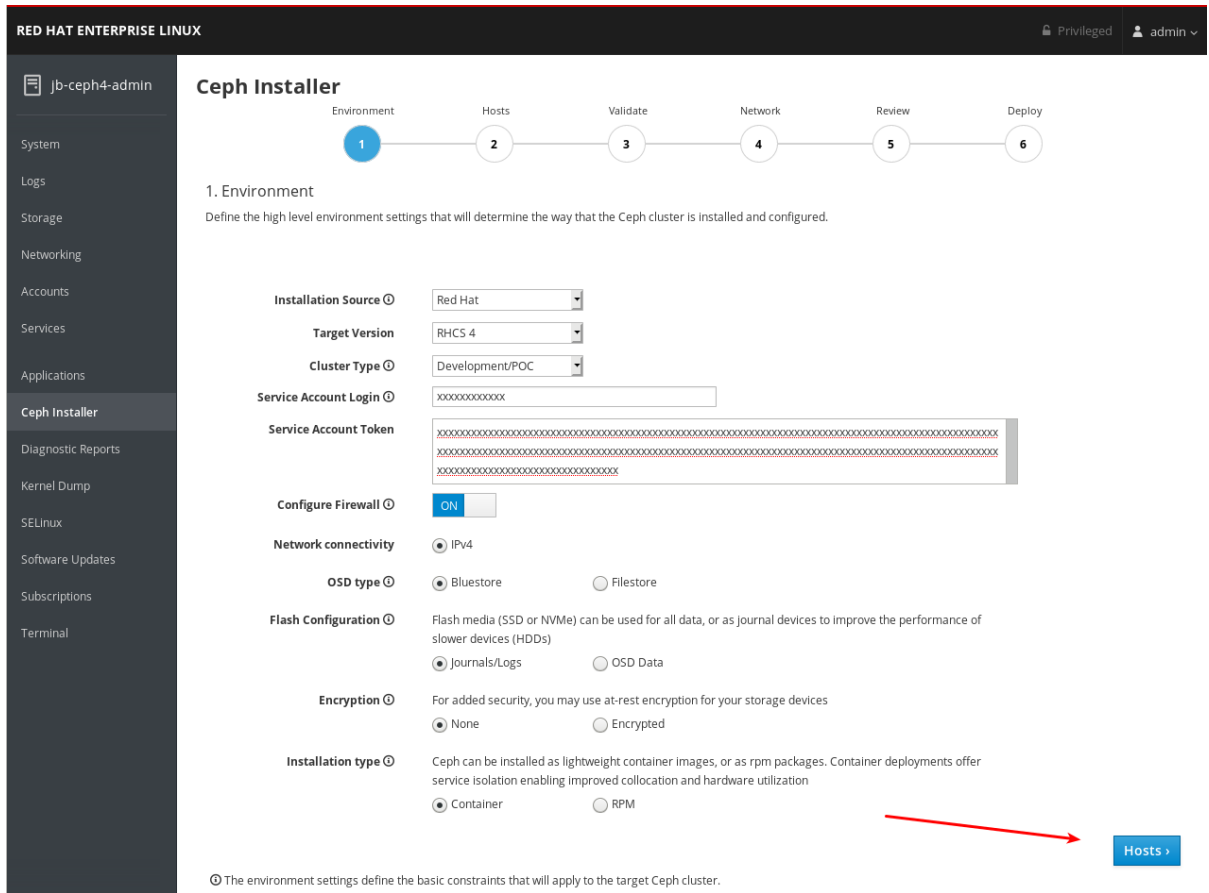
None  Encrypted

9. 설치 유형을 Container 또는 RPM 으로 설정합니다. 일반적으로 Red Hat Package Manager(RPM)는 Red Hat Enterprise Linux에 소프트웨어를 설치하는 데 사용되었습니다. 이제 RPM 또는 컨테이너를 사용하여 Ceph를 설치할 수 있습니다. 컨테이너를 사용하여 Ceph를 설치하면 서비스를 분리하고 공동 배치할 수 있으므로 개선된 하드웨어 사용률을 제공할 수 있습니다.

**Installation type** ⓘ Ceph can be installed as lightweight container images, or as rpm packages. Container deployments offer service isolation enabling improved collocation and hardware utilization

Container  RPM

10. 모든 환경 설정을 검토하고 웹 페이지의 오른쪽 하단에 있는 Hosts (호스트) 버튼을 클릭합니다.



## 4.7. COCKPIT CEPH 설치 프로그램의 HOSTS 페이지 완료

Hosts (호스트) 페이지를 사용하면 Cockpit Ceph 설치 프로그램에 Ceph를 설치할 호스트와 각 호스트가 사용할 역할을 알릴 수 있습니다. 호스트를 추가하면 설치 프로그램에서 SSH 및 DNS 연결을 확인합니다.

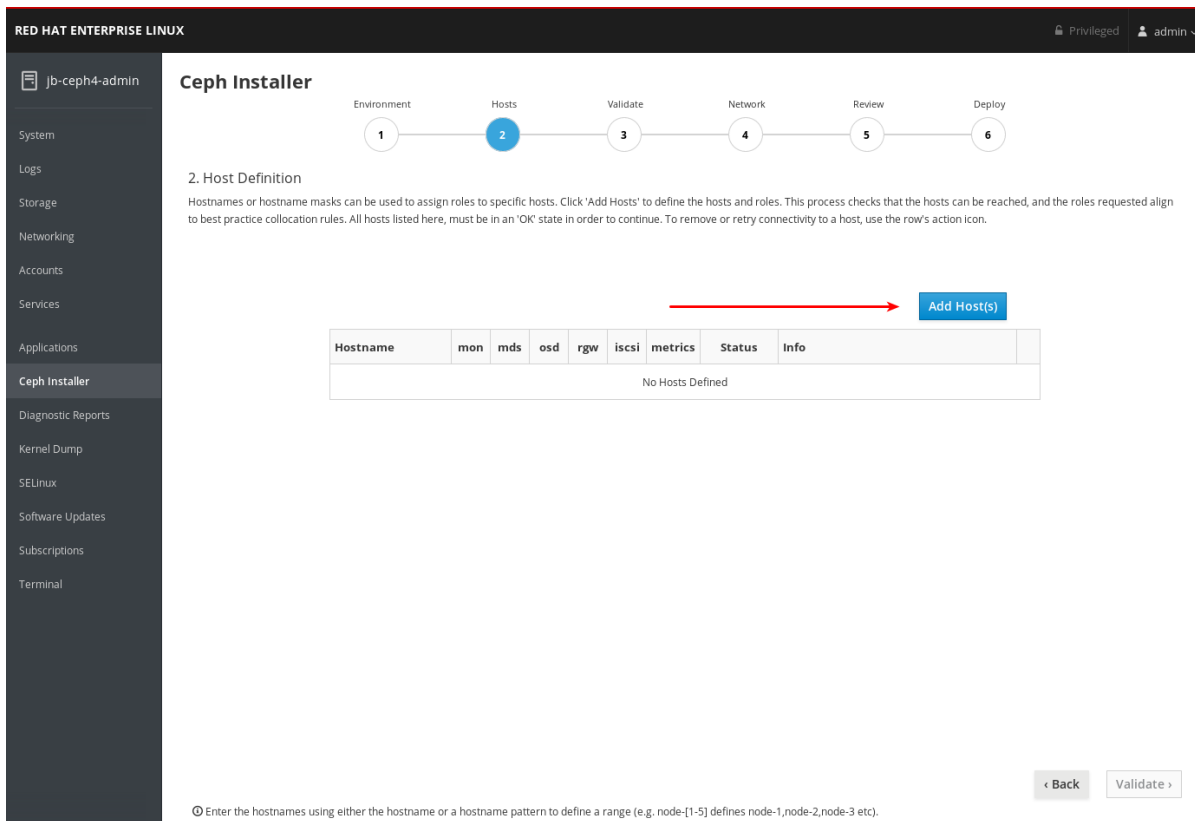
### 사전 요구 사항

- Cockpit Ceph 설치 프로그램의 환경 페이지가 완료되었습니다.
- Cockpit Ceph Installer SSH 키가 클러스터의 모든 노드에 복사되었습니다.

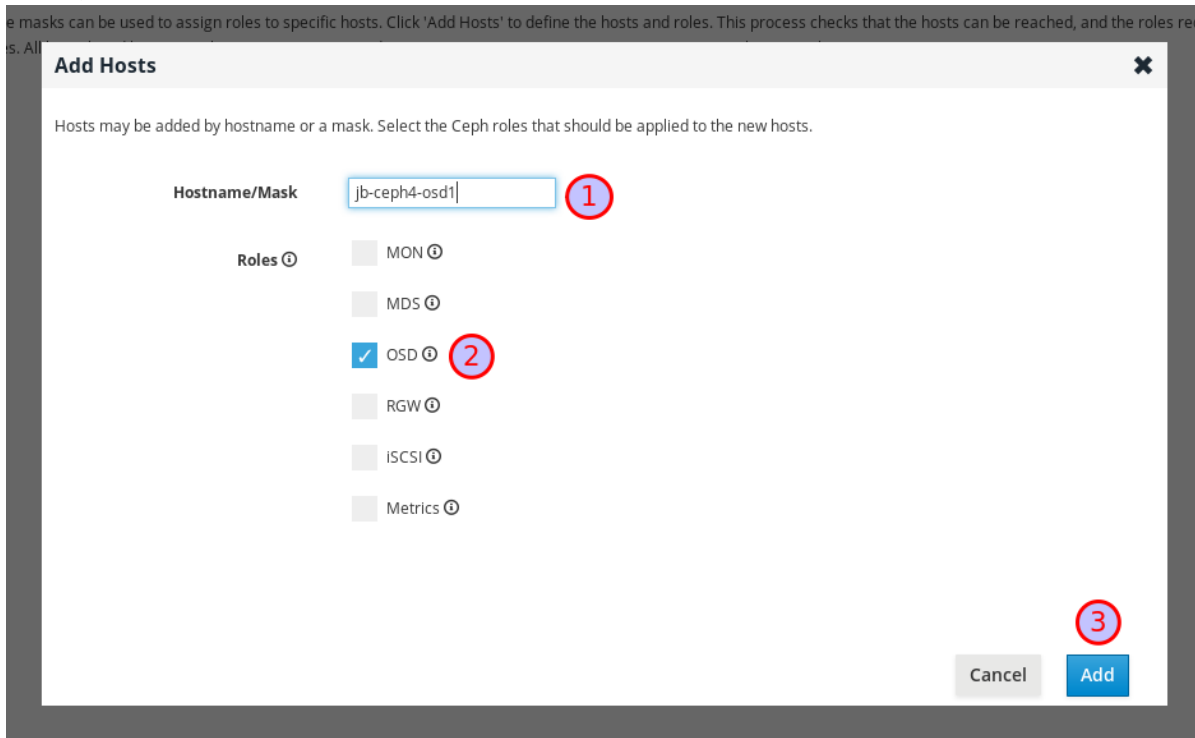
### 절차

1. Add Host(s) (호스트 추가) 버튼을 클릭합니다.





2. Ceph OSD 노드의 호스트 이름을 입력하고 OSD의 확인란을 선택한 다음 Add (추가) 버튼을 클릭합니다.

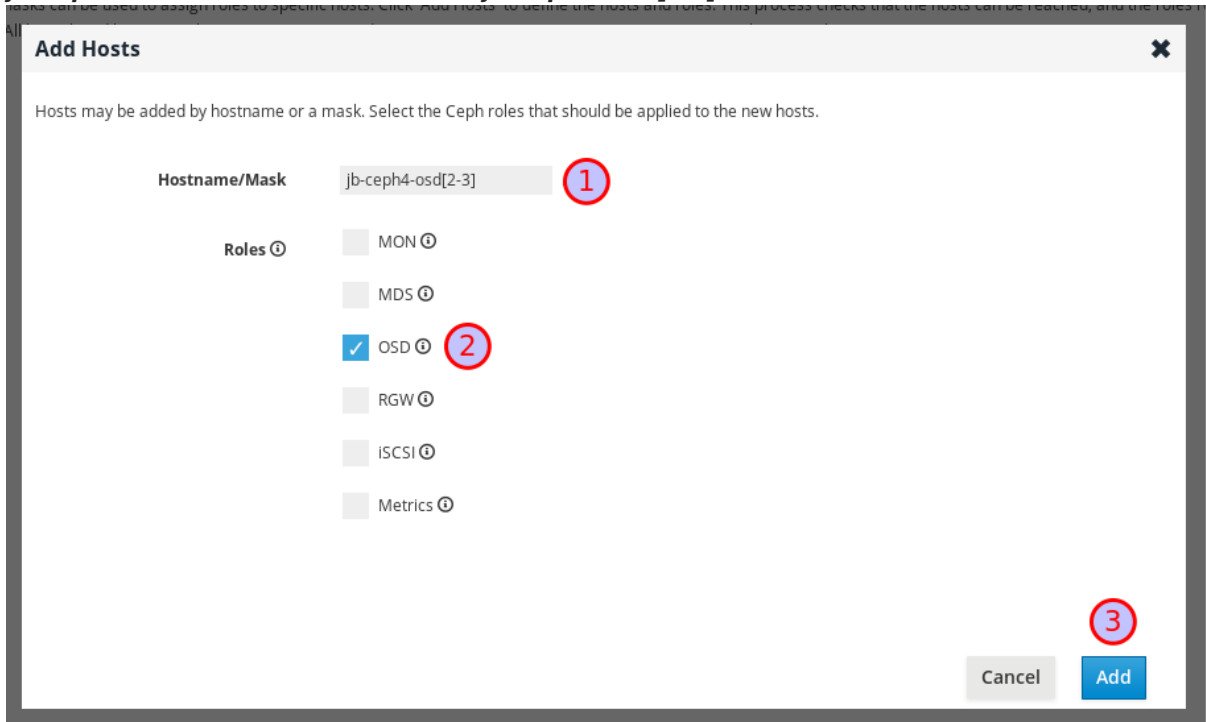


첫 번째 Ceph OSD 노드가 추가되었습니다.



프로덕션 클러스터의 경우 3개 이상의 Ceph OSD 노드를 추가할 때까지 이 단계를 반복합니다.

3. 선택 사항: 호스트 이름 패턴을 사용하여 노드 범위를 정의합니다. 예를 들어 **jb-ceph4-osd2** 및 **jb-ceph4-osd3** 을 동시에 추가하려면 **jb-ceph4-osd[2-3]** 을 입력합니다.



**jb-ceph4-osd2** 및 **jb-ceph4-ods3** 이 추가되었습니다.

[Add Host\(s\)](#)

Hostname	mon	mds	osd	rgw	iscsi	metrics	Status	Info
jb-ceph4-osd3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-osd2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-osd1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory

4. 클러스터의 다른 노드에 대해 위의 단계를 반복합니다.

- a. 프로덕션 클러스터의 경우 3개 이상의 Ceph Monitor 노드를 추가합니다. 대화 상자에서는 **MON** 으로 역할이 나열됩니다.
- b. **Metrics** 역할이 있는 노드를 추가합니다. **Metrics** 역할은 Grafana 및 Prometheus를 설치하여 Ceph 클러스터의 성능에 대한 실시간 정보를 제공합니다. 이러한 지표는 Ceph 대시보드에 제공되어 클러스터를 모니터링하고 관리할 수 있습니다. 대시보드, Grafana 및 Prometheus를 설치해야 합니다. Ansible 관리 노드에서 지표 기능을 배치할 수 있습니다. 이 경우 노드의 시스템 리소스가 독립형 매트릭 노드에 필요한 것보다 크지 확인하십시오.
- c. 선택 사항: **MDS** 역할이 있는 노드를 추가합니다. **MDS** 역할은 Ceph 메타데이터 서버(MDS)를 설치합니다. 메타데이터 서버 데몬은 Ceph 파일 시스템을 배포하는 데 필요합니다.
- d. 선택 사항: **RGW** 역할이 있는 노드를 추가합니다. **RGW** 역할은 Ceph 개체 게이트웨이를 설치하고 RADOS 게이트웨이라고도 하며, RADOS 게이트웨이는 Ceph 스토리지 클러스터에 RESTful 게이트웨이를 제공하기 위해 librados API 상단에 구축된 오브젝트 스토리지 인터페이스입니다. Amazon S3 및 OpenStack Swift API를 지원합니다.

- e. 선택 사항: **iSCSI** 역할로 노드를 추가합니다. **iSCSI** 역할은 iSCSI 게이트웨이를 설치하므로 iSCSI를 통해 Ceph 블록 장치를 공유할 수 있습니다. Ceph에서 iSCSI를 사용하려면 다중 경로 I/O의 두 개 이상의 노드에 iSCSI 게이트웨이를 설치해야 합니다.
5. 선택 사항: 노드를 추가할 때 여러 역할을 선택하여 동일한 노드에서 두 개 이상의 서비스를 연결합니다.

데몬 배치에 대한 자세한 내용은 [설치 가이드의 컨테이너화된 Ceph 데몬 배치를 참조하십시오](#).

6. 선택 사항: 표에서 역할을 확인하거나 선택 해제하여 노드에 할당된 역할을 수정합니다.

[Add Host\(s\)](#)

Hostname	mon	mds	osd	rgw	iscsi	metrics	Status	Info
jb-ceph4-admin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-mon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-osd1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-osd2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-osd3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-rgw	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory

7. 선택 사항: 노드를 삭제하려면 삭제할 노드 행의 맨 오른쪽에 있는 kebab 아이콘을 클릭한 다음 삭제를 클릭합니다.

[Add Host\(s\)](#)

Hostname	mon	mds	osd	rgw	iscsi	metrics	Status	Info
jb-ceph4-admin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-mon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory

Delete

- 8. 클러스터의 모든 노드를 추가하고 필요한 모든 역할을 설정한 후 페이지의 오른쪽 아래에 있는 Validate 버튼을 클릭합니다.

**2. Host Definition**

Hostnames or hostname masks can be used to assign roles to specific hosts. Click 'Add Hosts' to define the hosts and roles. This process checks that the hosts can be reached, and the roles requested align to best practice collocation rules. All hosts listed here, must be in an 'OK' state in order to continue. To remove or retry connectivity to a host, use the row's action icon.

Hostname	mon	mds	osd	rgw	iscsi	metrics	Status	Info
jb-ceph4-admin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-mon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-osd1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-osd2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-osd3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jb-ceph4-rgw	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory

ⓘ Enter the hostnames using either the hostname or a hostname pattern to define a range (e.g. node-[1-5] defines node-1,node-2,node-3 etc).



**참고**

프로덕션 클러스터의 경우 3개 또는 5개의 모니터를 사용하지 않는 한 Cockpit Ceph 설치 프로그램이 진행되지 않습니다. 이 예제에서 Cluster Type 은 Development/POC 로 설정 되어 하나의 모니터만으로 설치가 진행될 수 있습니다.

**4.8. COCKPIT CEPH 설치 프로그램의 검증 페이지 완료**

Validate (유효성 검사) 페이지를 사용하면 Hosts (호스트) 페이지에서 제공한 노드를 검색하여 해당 노드를 사용할 역할의 하드웨어 요구 사항을 충족하는지 확인할 수 있습니다.

**사전 요구 사항**

- Cockpit Ceph 설치 프로그램의 Hosts 페이지가 완료되었습니다.

**절차**

1. 프rob 호스트 버튼을 클릭합니다.

**Ceph Installer**

Environment   Hosts   **Validate**   Network   Review   Deploy

1   2   3   4   5   6

3. Validate Host Selection

The hosts have been checked for DNS and passwordless SSH.  
The next step is to probe the hosts that Ceph will use to validate that their hardware configuration is compatible with their intended Ceph role. Once the probe is complete you must select the hosts to use for deployment using the checkboxes (only hosts in an "OK" state can be selected)

<input type="checkbox"/>	Hostname	mon	mds	osd	rgw	iscsi	CPU	RAM	NIC	HDD	SSD	Raw Capacity (HDD/SSD)	Status
<input type="checkbox"/>	j-b-ceph4-mon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>							
<input type="checkbox"/>	j-b-ceph4-osd1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>							
<input type="checkbox"/>	j-b-ceph4-osd2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>							
<input type="checkbox"/>	j-b-ceph4-osd3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>							
<input type="checkbox"/>	j-b-ceph4-rgw	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>							

◀ Back   **Probe Hosts**   Network ▶

© The probe process compares hardware configurations against the intended Ceph roles

계속하려면 OK Status 가 있는 호스트를 3개 이상 선택해야 합니다.

2. 선택 사항: 호스트에 대한 경고 또는 오류가 생성된 경우 호스트에 대한 확인 표시 왼쪽에 있는 화살표를 클릭하여 문제를 확인합니다.

✔ 5/5 probes complete

<input type="checkbox"/>	Hostname	mon	mds	osd	rgw	iscsi	CPU	RAM	NIC	HDD	SSD	Raw Capacity (HDD/SSD)	Status
<input type="checkbox"/>	j-b-ceph4-mon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	0	0	0 / 0	NOTOK <b>3 errors</b> <b>1 warning</b>
<input type="checkbox"/>	j-b-ceph4-osd1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	NOTOK <b>3 errors</b> <b>2 warnings</b>
<input type="checkbox"/>	j-b-ceph4-osd2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	NOTOK <b>2 errors</b> <b>2 warnings</b>
<input type="checkbox"/>	j-b-ceph4-osd3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	NOTOK <b>2 errors</b> <b>2 warnings</b>
<input type="checkbox"/>	j-b-ceph4-rgw	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	1	1	0	0	0 / 0	NOTOK <b>3 errors</b> <b>2 warnings</b>

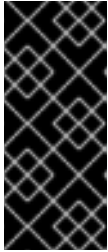
  

<input type="checkbox"/>	Hostname	mon	mds	osd	rgw	iscsi	CPU	RAM	NIC	HDD	SSD	Raw Capacity (HDD/SSD)	Status
<input checked="" type="checkbox"/>	j-b-ceph4-mon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	0	0	0 / 0	NOTOK <b>3 errors</b> <b>1 warning</b>
<p><b>error</b> #CPU's too low (min 6 needed)</p> <p><b>error</b> Freespace on /var/lib is too low (&lt;30GB)</p> <p><b>error</b> RAM too low (min 12G needed)</p> <p><b>warning</b> hosts should have a minimum of 2 networks</p>													
<input type="checkbox"/>	j-b-ceph4-osd1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	NOTOK <b>3 errors</b> <b>2 warnings</b>
<input type="checkbox"/>	j-b-ceph4-osd2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	NOTOK <b>2 errors</b> <b>2 warnings</b>
<input type="checkbox"/>	j-b-ceph4-osd3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	NOTOK <b>2 errors</b> <b>2 warnings</b>
<input type="checkbox"/>	j-b-ceph4-rgw	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	1	1	0	0	0 / 0	NOTOK <b>3 errors</b> <b>2 warnings</b>



**중요**

Cluster Type 을 Production 으로 설정하면 생성된 모든 오류로 인해 상태가 NOTOK 로 되어 설치 시 해당 상태를 선택할 수 없습니다. 오류 해결 방법에 대한 자세한 내용은 다음 단계를 참조하십시오.



**중요**

Cluster Type 을 Development/POC 로 설정하면 생성된 모든 오류가 경고로 나열 되므로 상태가 항상 OK 입니다. 이를 통해 호스트가 요구 사항 또는 제안 사항을 충족하는지 여부와 관계없이 호스트를 선택하고 Ceph를 설치할 수 있습니다. 원하는 경우 경고를 계속 해결할 수 있습니다. 경고를 해결하는 방법에 대한 자세한 내용은 다음 단계를 참조하십시오.

3. 선택 사항: 오류 및 경고를 해결하려면 다음 방법 중 하나 이상을 사용합니다.

- a. 오류 또는 경고를 해결하는 가장 쉬운 방법은 특정 역할을 완전히 비활성화하거나 한 호스트에서 역할을 비활성화하고 필요한 리소스가 있는 다른 호스트에서 활성화하는 것입니다. 개발/POC 클러스터를 설치하는 경우 또는 프로덕션 클러스터를 설치하는 경우, 3개 이상의 호스트에 할당된 역할에 필요한 모든 리소스가 있으므로 나머지 경고를 찾을 때까지 역할을 활성화 또는 비활성화하기 전에 역할을 활성화하거나 비활성화할 수 있습니다.
- b. 필요한 역할에 대한 요구 사항을 충족하는 새 호스트를 사용할 수도 있습니다. 먼저 Hosts (호스트) 페이지로 돌아가 문제가 있는 호스트를 삭제합니다.

[Add Host\(s\)](#)

Hostname	mon	mds	osd	rgw	iscsi	metrics	Status	Info
jib-ceph4-mon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jib-ceph4-osd1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jib-ceph4-osd2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jib-ceph4-osd3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory
jib-ceph4-rgw	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	OK	Connectivity verified, added to the inventory

그런 다음 새 호스트를 추가합니다.

- c. 호스트에서 하드웨어를 업그레이드하거나 다른 방식으로 수정하려면 요구 사항 또는 제안을 충족하고 먼저 호스트를 원하는 대로 변경한 다음 프로브 호스트를 다시 클릭합니다. 운영 체제를 다시 설치해야 하는 경우 SSH 키를 다시 복사 해야 합니다.

4. 호스트 옆에 있는 상자를 선택하여 Red Hat Ceph Storage 를 설치할 호스트를 선택합니다.

✔ 5/5 probes complete

<input type="checkbox"/>	Hostname	mon	mds	osd	rgw	iscsi	CPU	RAM	NIC	HDD	SSD	Raw Capacity (HDD/SSD)	Status
<input checked="" type="checkbox"/>	jib-ceph4-mon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	0	0	0 / 0	OK <span style="background-color: yellow;">3 warnings</span>
<input checked="" type="checkbox"/>	jib-ceph4-osd1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	OK <span style="background-color: yellow;">3 warnings</span>
<input checked="" type="checkbox"/>	jib-ceph4-osd2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	OK <span style="background-color: yellow;">3 warnings</span>
<input checked="" type="checkbox"/>	jib-ceph4-osd3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	OK <span style="background-color: yellow;">3 warnings</span>
<input checked="" type="checkbox"/>	jib-ceph4-rgw	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	1	1	0	0	0 / 0	OK <span style="background-color: yellow;">3 warnings</span>



**중요**

프로덕션 클러스터를 설치하는 경우 설치용으로 선택하기 전에 오류를 해결해야 합니다.

5. 페이지의 오른쪽 아래에 있는 네트워크 버튼을 클릭하여 클러스터의 네트워킹을 검토 및 구성합니다.

**Ceph Installer**

Environment (1) → Hosts (2) → **Validate (3)** → Network (4) → Review (5) → Deploy (6)

3. Validate Host Selection

The hosts have been checked for DNS and passwordless SSH. The next step is to probe the hosts that Ceph will use to validate that their hardware configuration is compatible with their intended Ceph role. Once the probe is complete you must select the hosts to use for deployment using the checkboxes (only hosts in an 'OK' state can be selected)

5/5 probes complete

<input type="checkbox"/>	Hostname	mon	mds	osd	rgw	iscsi	CPU	RAM	NIC	HDD	SSD	Raw Capacity (HDD/SSD)	Status
<input checked="" type="checkbox"/>	jb-ceph4-mon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	0	0	0 / 0	OK <span style="background-color: yellow;">3 warnings</span>
<input checked="" type="checkbox"/>	jb-ceph4-osd1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	OK <span style="background-color: yellow;">3 warnings</span>
<input checked="" type="checkbox"/>	jb-ceph4-osd2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	OK <span style="background-color: yellow;">3 warnings</span>
<input checked="" type="checkbox"/>	jb-ceph4-osd3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	1	1	1	0	25G / 0	OK <span style="background-color: yellow;">3 warnings</span>
<input checked="" type="checkbox"/>	jb-ceph4-rgw	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	1	1	0	0	0 / 0	OK <span style="background-color: yellow;">3 warnings</span>

© The probe process compares hardware configurations against the intended Ceph roles

◀ Back Probe Hosts **Network ▶**

## 4.9. COCKPIT CEPH 설치 프로그램의 네트워크 페이지 완료

네트워크 페이지를 사용하면 특정 클러스터 통신 유형을 특정 네트워크에 분리할 수 있습니다. 이를 위해서는 클러스터의 호스트 전체에 구성된 여러 다른 네트워크가 필요합니다.



### 중요

네트워크 페이지는 Validate 페이지에서 수행한 프로브에서 수집된 정보를 사용하여 호스트에서 액세스할 수 있는 네트워크를 표시합니다. 현재 네트워크 페이지를 이미 진행한 경우 호스트에 새 네트워크를 추가하고, Validate (유효성 검사) 페이지로 돌아가 호스트를 재프로필하고, 네트워크 페이지로 다시 이동하여 새 네트워크를 사용할 수 없습니다. 선택이 표시되지 않습니다. 네트워크 페이지로 이동한 후 호스트에 추가된 네트워크를 사용하려면 웹 페이지를 완전히 새로 고침한 후 처음부터 설치를 다시 시작해야 합니다.



### 중요

프로덕션 클러스터의 경우 별도의 NIC에서 클라이언트 간 트래픽과 클러스터 내 트래픽을 분리해야 합니다. 클러스터 트래픽 유형을 분리하는 것 외에도 Ceph 클러스터를 설정할 때 고려해야 할 다른 네트워킹 고려 사항이 있습니다. 자세한 내용은 [하드웨어 가이드의 네트워크 고려 사항을 참조하십시오](#).

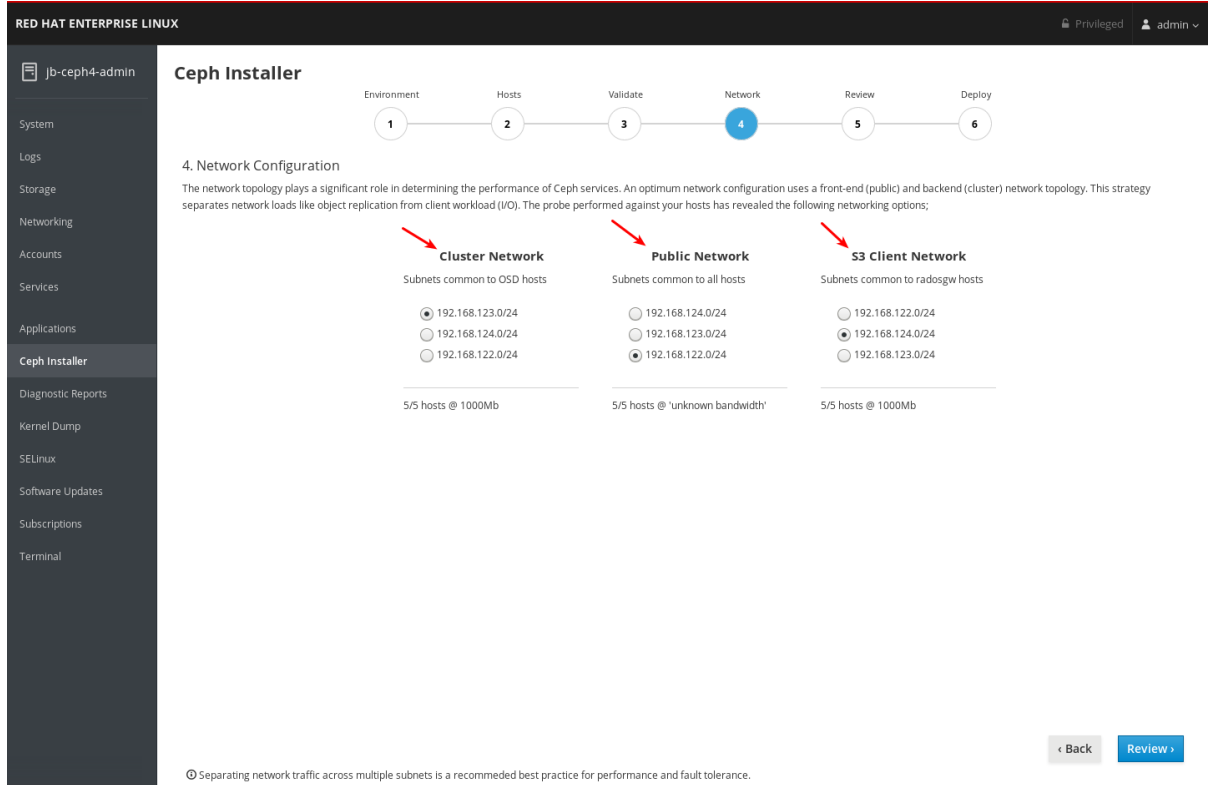
### 사전 요구 사항

- Cockpit Ceph 설치 프로그램의 검증 페이지가 완료되었습니다.

### 절차

1. 네트워크 페이지에서 구성할 수 있는 네트워크 유형을 기록해 두십시오. 각 유형에는 자체 열이 있습니다. 클러스터 네트워크 및 공용 네트워크의 열이 항상 표시됩니다. RADOS Gateway 역할

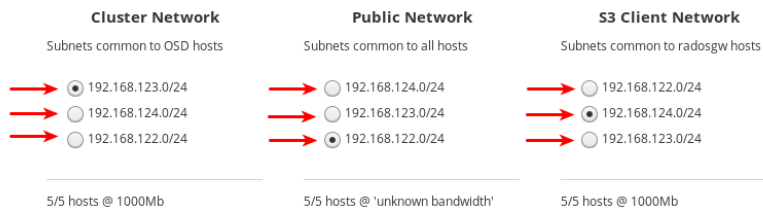
을 사용하여 호스트를 설치하는 경우 S3 네트워크 열이 표시됩니다. iSCSI 역할로 호스트를 설치하는 경우 iSCSI 네트워크 열이 표시됩니다. 아래 예에서는 Cluster Network, Public Network, S3 Network 에 대한 열이 표시됩니다.



2. 각 네트워크 유형에 대해 선택할 수 있는 네트워크를 기록해 두십시오. 특정 네트워크 유형을 구성하는 모든 호스트에서 사용 가능한 네트워크만 표시됩니다. 아래 예제에는 클러스터의 모든 호스트에서 사용할 수 있는 세 개의 네트워크가 있습니다. 네트워크 유형을 구성하는 모든 호스트에서 세 개의 네트워크를 모두 사용할 수 있으므로 각 네트워크 유형은 동일한 세 개의 네트워크를 나열합니다.

4. Network Configuration

The network topology plays a significant role in determining the performance of Ceph services. An optimum network configuration uses a front-end (public) and backend (cluster) network topology. This strategy separates network loads like object replication from client workload (I/O). The probe performed against your hosts has revealed the following networking options;

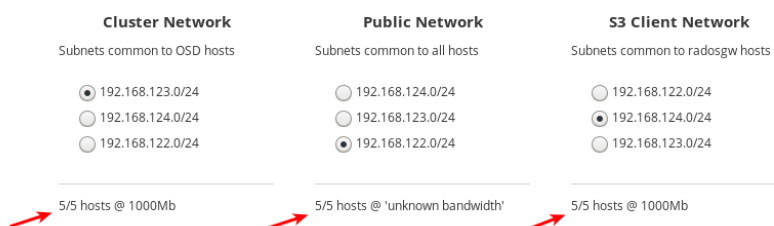


사용 가능한 세 개의 네트워크는 **192.168.122.0/24, 192.168.123.0/24, 192.168.124.0/24** 입니다.

3. 각 네트워크가 작동하는 속도를 기록해 두십시오. 이는 특정 네트워크에 사용되는 NIC의 속도입니다. 아래 예에서 **192.168.123.0/24** 및 **192.168.124.0/24** 는 1,000mbps 입니다. Cockpit Ceph 설치 프로그램에서 **192.168.122.0/24** 네트워크의 속도를 확인할 수 없습니다.

4. Network Configuration

The network topology plays a significant role in determining the performance of Ceph services. An optimum network configuration uses a front-end (public) and backend (cluster) network topology. This strategy separates network loads like object replication from client workload (I/O). The probe performed against your hosts has revealed the following networking options;

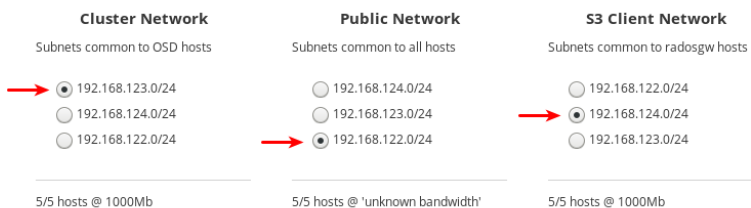




4. 각 네트워크 유형에 사용할 네트워크를 선택합니다. 프로덕션 클러스터의 경우 클러스터 네트워크 및 공용 네트워크에 대해 별도의 네트워크를 선택해야 합니다. 개발/POC 클러스터의 경우 두 유형 모두에 대해 동일한 네트워크를 선택하거나 모든 호스트에 하나의 네트워크만 구성되어 있는 경우 해당 네트워크만 표시되고 다른 네트워크를 선택할 수 없습니다.

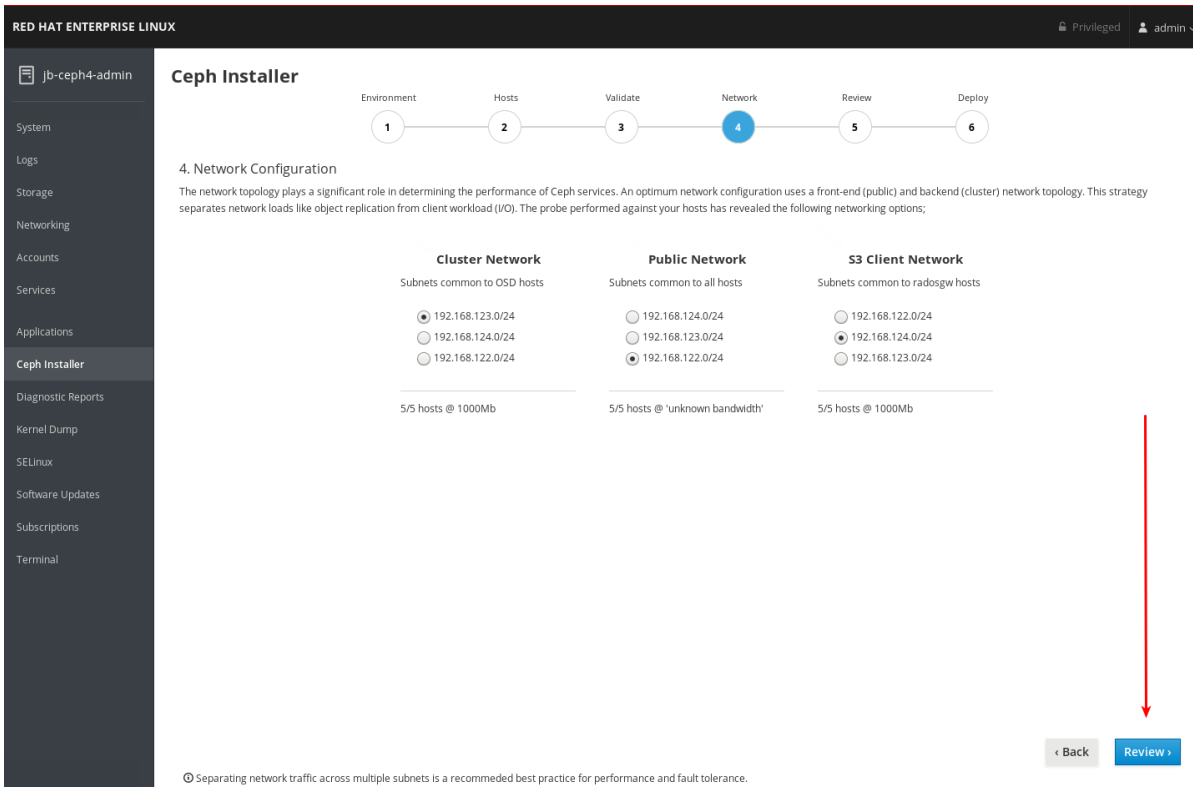
4. Network Configuration

The network topology plays a significant role in determining the performance of Ceph services. An optimum network configuration uses a front-end (public) and backend (cluster) network topology. This strategy separates network loads like object replication from client workload (I/O). The probe performed against your hosts has revealed the following networking options;



**192.168.122.0/24** 네트워크는 공용 네트워크에 사용되며 **192.168.123.0/24** 네트워크는 클러스터 네트워크 용으로 사용되며 **192.168.124.0/24** 네트워크는 S3 네트워크에 사용됩니다.

5. 페이지 오른쪽 하단의 검토 버튼을 클릭하여 설치 전에 전체 클러스터 구성을 검토합니다.



### 4.10. 설치 구성 검토

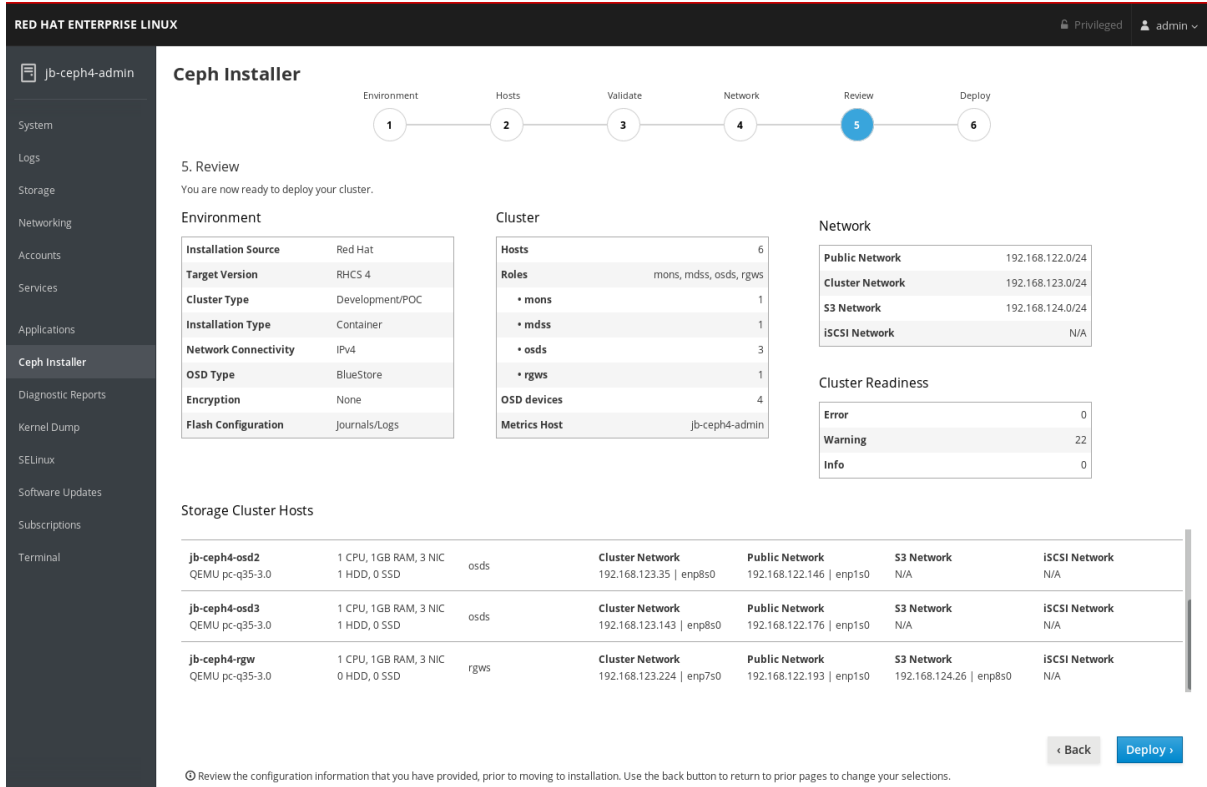
검토 페이지를 통해 이전 페이지에 설정한 Ceph 클러스터 설치 구성의 모든 세부 정보와 이전 페이지에 포함되지 않은 호스트에 대한 세부 정보를 볼 수 있습니다.

#### 사전 요구 사항

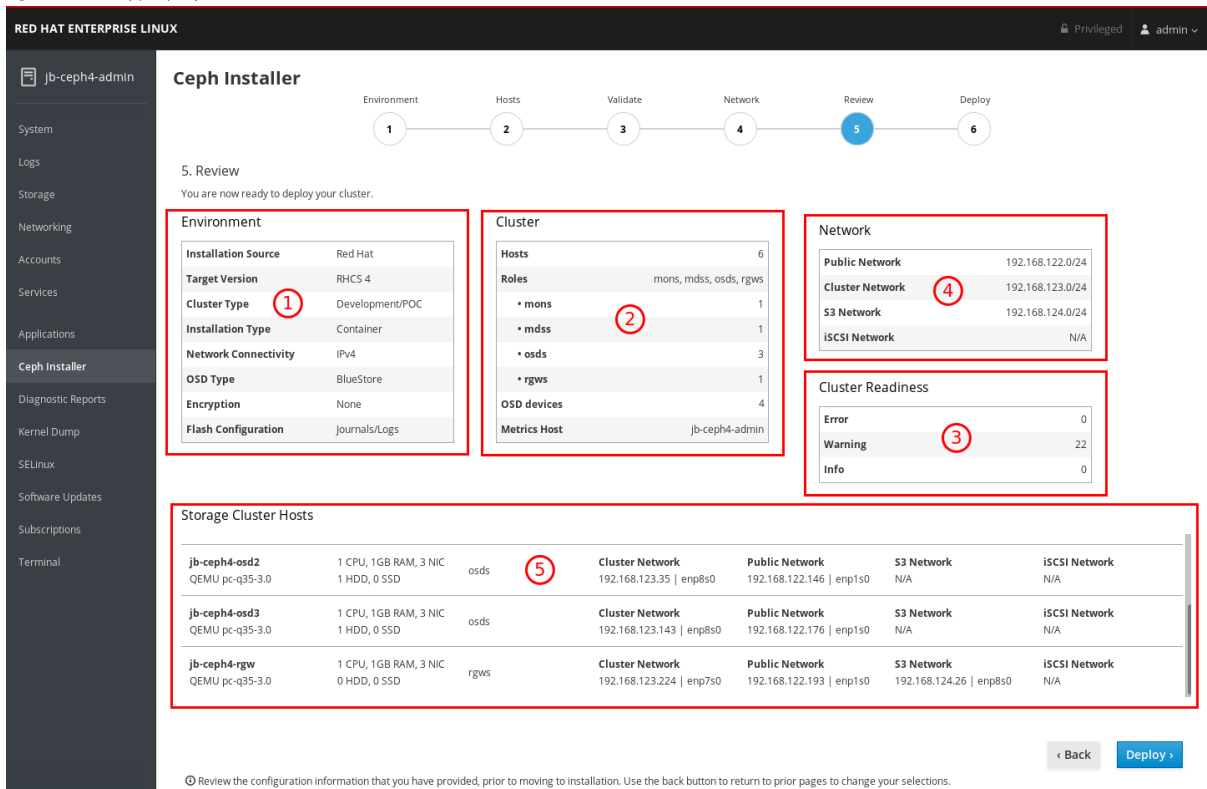
- Cockpit Ceph 설치 프로그램의 네트워크 페이지가 완료되었습니다.

#### 절차

1. 검토 페이지를 봅니다.



2. 검토 페이지에 표시된 대로 이전 페이지의 정보가 예상과 같은지 확인합니다. 환경 페이지의 정보 요약은 1에 있는 호스트 페이지 2에서 호스트 페이지, 3에서 네트워크 페이지, 4에서 네트워크 페이지 및 이전 페이지에 포함되지 않은 몇 가지 추가 세부 정보를 포함하여 호스트에 대한 세부 정보는 5입니다.



3. 페이지의 오른쪽 하단 모서리에 있는 Deploy (배포) 버튼을 클릭하여 Deploy (배포) 페이지로 이동하여 실제 설치 프로세스를 종료하고 시작할 수 있습니다.

RED HAT ENTERPRISE LINUX Privileged admin

jb-ceph4-admin

### Ceph Installer

Environment Hosts Validate Network Review Deploy

1 2 3 4 5 6

5. Review  
You are now ready to deploy your cluster.

**Environment**

Installation Source	Red Hat
Target Version	RHCS 4
Cluster Type	Development/POC
Installation Type	Container
Network Connectivity	IPv4
OSD Type	BlueStore
Encryption	None
Flash Configuration	Journals+Logs

**Cluster**

Hosts	6
Roles	mons, mdss, osds, rgws
• mons	1
• mdss	1
• osds	3
• rgws	1
OSD devices	4
Metrics Host	jb-ceph4-admin

**Network**

Public Network	192.168.122.0/24
Cluster Network	192.168.123.0/24
S3 Network	192.168.124.0/24
iSCSI Network	N/A

**Cluster Readiness**

Error	0
Warning	22
Info	0

**Storage Cluster Hosts**

Host	Hardware	Roles	Cluster Network	Public Network	S3 Network	iSCSI Network
jb-ceph4-osd2	1 CPU, 1GB RAM, 3 NIC QEMU pc-q35-3.0 1 HDD, 0 SSD	osds	192.168.123.35   enp8s0	192.168.122.146   enp1s0	N/A	N/A
jb-ceph4-osd3	1 CPU, 1GB RAM, 3 NIC QEMU pc-q35-3.0 1 HDD, 0 SSD	osds	192.168.123.143   enp8s0	192.168.122.176   enp1s0	N/A	N/A
jb-ceph4-rgw	1 CPU, 1GB RAM, 3 NIC QEMU pc-q35-3.0 0 HDD, 0 SSD	rgws	192.168.123.224   enp7s0	192.168.122.193   enp1s0	192.168.124.26   enp8s0	N/A

© Review the configuration information that you have provided, prior to moving to installation. Use the back button to return to prior pages to change your selections.

Deploy

## 4.11. CEPH 클러스터 배포

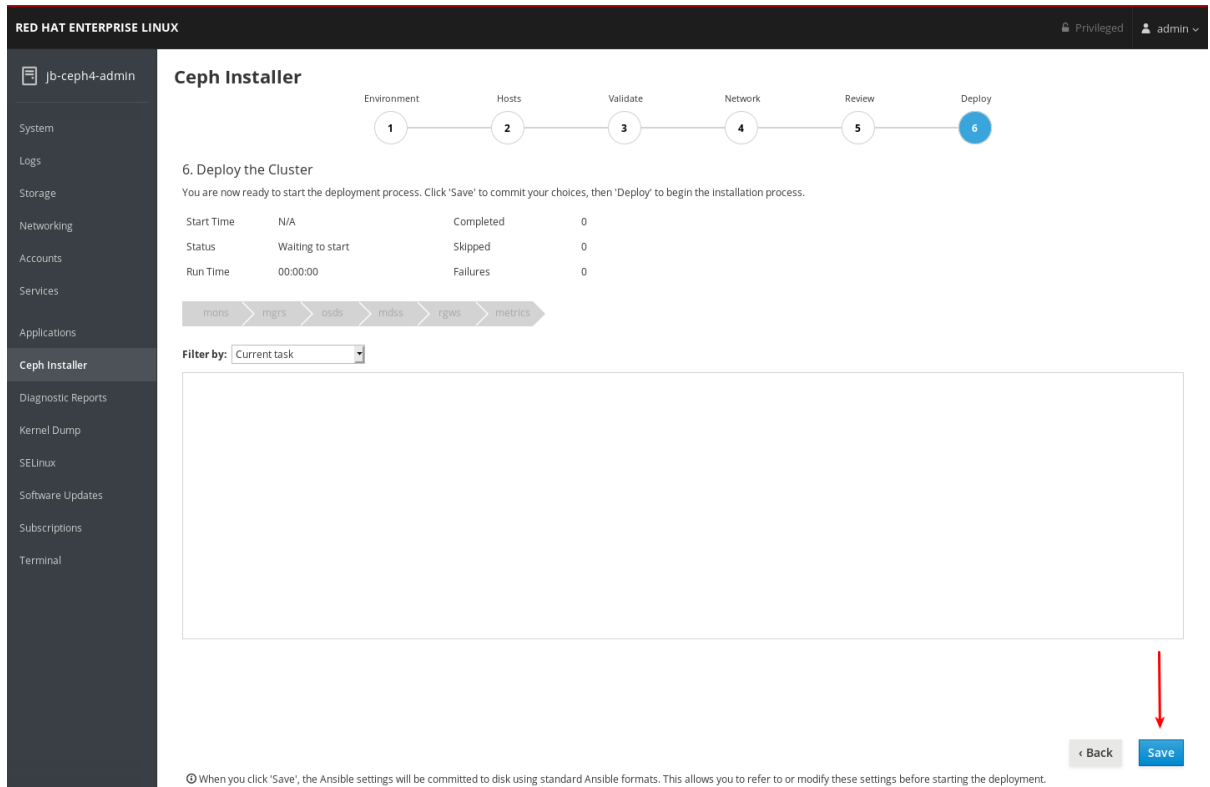
배포 페이지를 사용하면 설치 설정을 네이티브 Ansible 형식으로 저장하고, 필요한 경우 이를 검토하거나 수정하고, 설치를 시작하고, 진행 상황을 모니터링하며, 설치가 완료된 후 클러스터 상태를 볼 수 있습니다.

### 사전 요구 사항

- 검토 페이지의 설치 구성 설정이 확인되었습니다.

### 절차

- 페이지의 오른쪽 아래에 있는 Save 버튼을 클릭하여 Ansible에서 실제 설치를 수행하는 데 사용할 Ansible 플레이북에 설치 설정을 저장합니다.



RED HAT ENTERPRISE LINUX Privileged admin

jb-ceph4-admin

### Ceph Installer

Environment Hosts Validate Network Review Deploy

1 2 3 4 5 6

#### 6. Deploy the Cluster

You are now ready to start the deployment process. Click 'Save' to commit your choices, then 'Deploy' to begin the installation process.

Start Time	N/A	Completed	0
Status	Waiting to start	Skipped	0
Run Time	00:00:00	Failures	0

mions > mgrs > osds > mdss > rgws > metrics

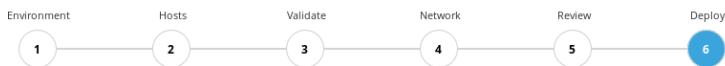
Filter by: Current task

Back Save

© When you click 'Save', the Ansible settings will be committed to disk using standard Ansible formats. This allows you to refer to or modify these settings before starting the deployment.

- 선택 사항: Ansible 관리 노드에 있는 Ansible 플레이북의 설정을 확인하거나 추가로 사용자 지정합니다. 플레이북은 `/usr/share/ceph-ansible` 에 있습니다. Ansible 플레이북과 설치를 사용자 지정하는 방법에 대한 자세한 내용은 [Red Hat Ceph Storage 클러스터 설치](#)를 참조하십시오.
- Grafana 및 대시보드의 기본 사용자 이름 및 암호를 보호합니다. Red Hat Ceph Storage 4.1부터 `/usr/share/ceph-ansible/group_vars/all.yml` 에서 `dashboard_admin_password` 및 `grafana_admin_password` 의 주석 처리를 해제하거나 설정해야 합니다. 각각에 대해 보안 암호를 설정합니다. `dashboard_admin_user` 및 `grafana_admin_user` 의 사용자 지정 사용자 이름도 설정합니다.
- 페이지의 오른쪽 하단 모서리에 있는 Deploy (배포) 버튼을 클릭하여 설치를 시작합니다.

### Ceph Installer



#### 6. Deploy the Cluster

You are now ready to start the deployment process. Click 'Save' to commit your choices, then 'Deploy' to begin the installation process.

Start Time	N/A	Completed	0
Status	Waiting to start	Skipped	0
Run Time	00:00:00	Failures	0



Filter by:

[Back](#) [Deploy](#)

Variables have been stored within the host\_vars and group\_vars directories of /usr/share/ceph-ansible.

#### 5. 실행 중인 동안 설치 진행 상황을 관찰합니다.

1의 정보는 설치가 실행 중인지, 시작 시간, 경과된 시간입니다. 2의 정보는 시도된 Ansible 작업의 요약을 보여줍니다. 3의 정보는 설치 또는 설치 중인 역할을 보여줍니다. 녹색은 해당 역할에 할당된 모든 호스트가 해당 역할에 설치된 역할을 나타냅니다. Blue는 해당 역할이 할당된 호스트가 여전히 설치되어 있는 역할을 나타냅니다. 4에서 현재 작업에 대한 세부 정보를 보거나 실패한 작업을 볼 수 있습니다. 필터 기준 메뉴를 사용하여 현재 작업과 실패한 작업 간에 전환합니다.

### Ceph Installer



#### 6. Deploy the Cluster

You are now ready to start the deployment process. Click 'Save' to commit your choices, then 'Deploy' to begin the installation process.

Start Time	13:21:23	Completed	576
Status	Running	Skipped	1128
Run Time	00:06:27	Failures	0



Filter by:

**Task Name:** [ceph-facts] set\_fact rbd\_client\_directory\_mode 0770

**Started:** 13:28:02

**Role:** ceph-facts

**Pattern:** osds

**Task Path:** /usr/share/ceph-ansible/roles/ceph-facts/tasks/facts.yml:202

**Action:** set\_fact

[Back](#) [Running](#)

역할 이름은 Ansible 인벤토리 파일에서 가져옵니다. 동등성: **mons** 는 Monitors 이고, **mgrs** 는 Managers 이고, Manager 역할은 Monitor 역할과 함께 설치되고, **osds** 는 Object Storage Devices 이고, **mdss** 는 Metadata Servers 이고, **rgws** 는 RADOS Gateways 이고, **지표** 는 Grafana 및 Prometheus 서비스 대시보드입니다. 예제 스크린샷에는 표시되지 않습니다. **iscsigws** 는 iSCSI 게이트웨이입니다.

- 설치가 완료되면 페이지의 오른쪽 아래에 있는 완료 버튼을 클릭합니다. 그러면 명령 **ceph** 상태 및 대시보드 액세스 정보의 출력이 표시되는 창이 열립니다.

### Ceph Installer



#### 6. Deploy the Cluster

You are now ready to start the deployment process. Click 'Save' to commit your choices, then 'Deploy' to begin the installation process.

Start Time	13:21:23	Completed	1139
Status	<b>Successful</b>	Skipped	1795
Run Time	00:12:04	Failures	0



Filter by:

**Task Name:** show ceph status for cluster ceph

**Started:** 13:34:06

**Role:**

**Pattern:** mons

**Task Path:** /usr/share/ceph-ansible/site-container.yml:446

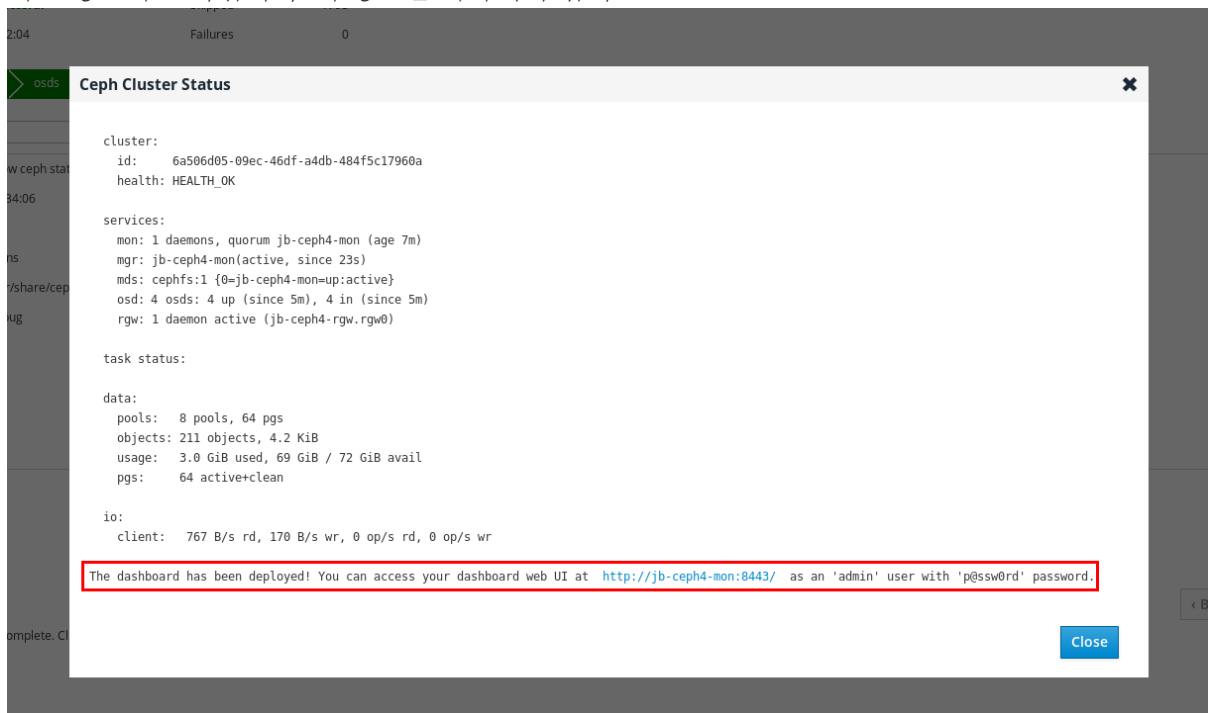
**Action:** debug

ⓘ Ceph deployment is complete. Click 'Complete' to show current state and login URL

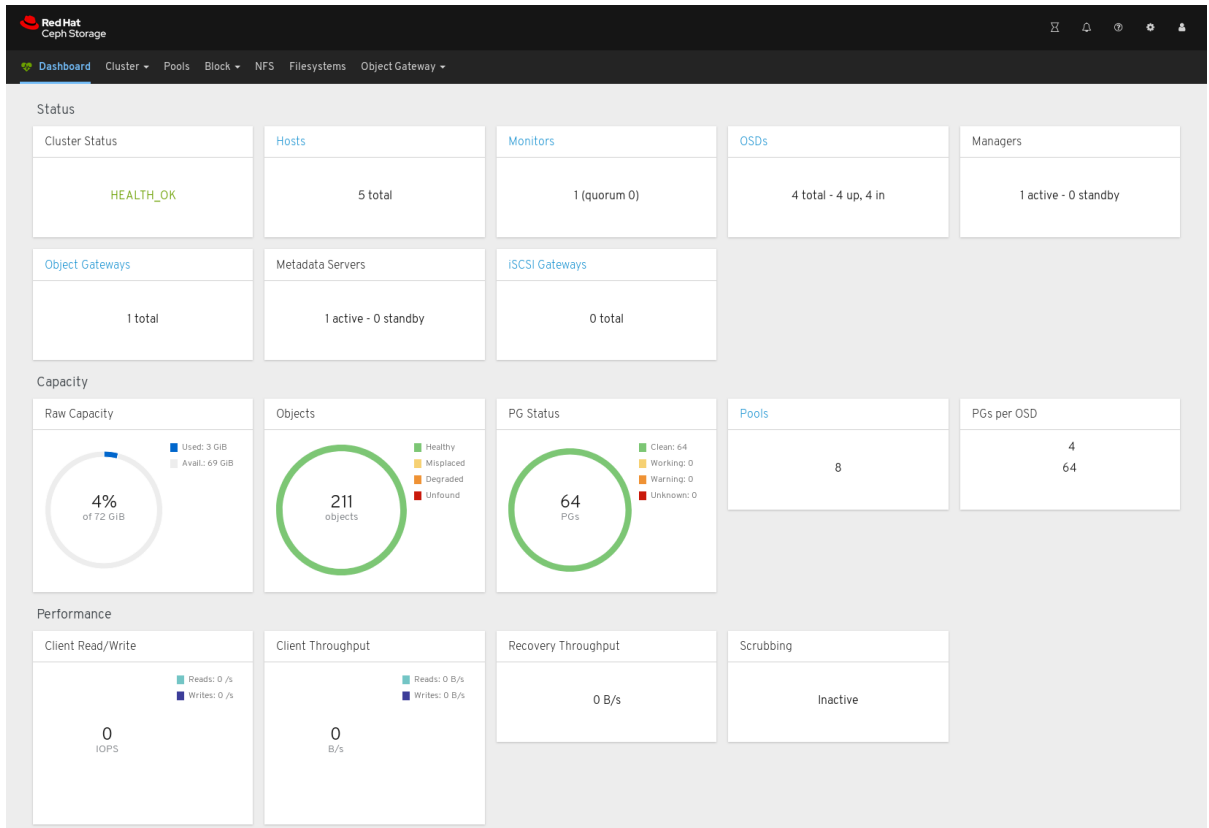
- 아래 예에서 클러스터 상태 정보를 클러스터의 클러스터 상태 정보와 비교합니다. 이 예에서는 모든 OSD up 및 in 및 모든 서비스가 활성 상태인 정상 클러스터를 보여줍니다. PGS는 **active+clean** 상태입니다. 클러스터의 일부 측면이 동일하지 않은 경우 문제 해결 방법에 대한 자세한 내용은 [Troubleshooting Guide](#) 를 참조하십시오.



8. Ceph Cluster Status(Ceph 클러스터 상태) 창의 맨 아래에 URL, 사용자 이름, 암호 등 대시보드 액세스 정보가 표시됩니다. 이 정보를 기록해 두십시오.



9. 대시보드 가이드와 함께 이전 단계의 정보를 사용하여 대시보드에 액세스 합니다.



대시보드는 웹 인터페이스를 제공하므로 Red Hat Ceph Storage 클러스터를 관리하고 모니터링할 수 있습니다. 자세한 내용은 [대시보드 가이드](#) 를 참조하십시오.

- 선택 사항: **cockpit-ceph-installer.log** 파일을 봅니다. 이 파일은 선택한 로그와 프로브 프로세스가 생성된 경고를 기록합니다. 설치 프로그램 스크립트 **ansible-runner-service.sh** 를 실행한 사용자의 홈 디렉터리에 있습니다.



## 5장. ANSIBLE을 사용하여 RED HAT CEPH STORAGE 설치

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

- Red Hat Ceph Storage 클러스터를 설치하려면 5.2절. "Red Hat Ceph Storage 클러스터 설치" 에서 참조하십시오.
- 메타데이터 서버를 설치하려면 5.4절. "메타데이터 서버 설치" 에서 참조하십시오.
- **ceph-client** 역할을 설치하려면 5.5절. "Ceph 클라이언트 역할 설치" 에서 참조하십시오.
- Ceph Object Gateway를 설치하려면 5.6절. "Ceph Object Gateway 설치" 에서 참조하십시오.
- 다중 사이트 Ceph 오브젝트 게이트웨이를 구성하려면 5.7절. "다중 사이트 Ceph 개체 게이트웨이 구성" 을 참조하십시오.
- Ansible **--limit** 옵션에 대한 자세한 내용은 5.10절. "제한 옵션 이해" 을 참조하십시오.

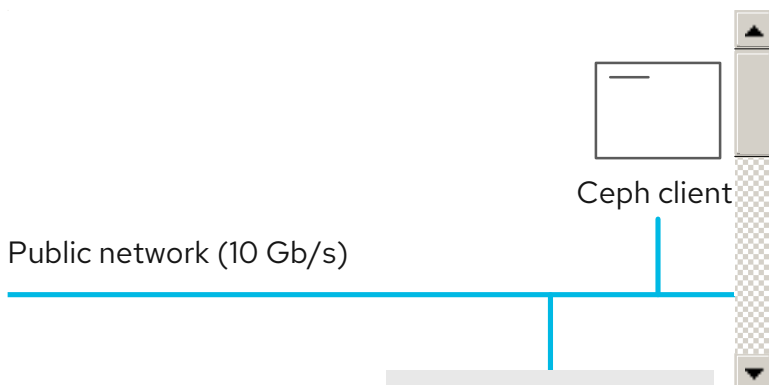
### 5.1. 사전 요구 사항

- 유효한 고객 서브스크립션을 받으십시오.
- 각 노드에서 다음을 수행하여 클러스터 노드를 준비합니다.
  - 노드를 CDN(Content Delivery Network)에 등록하고 서브스크립션을 연결합니다.
  - 적절한 소프트웨어 리포지토리를 활성화합니다.
  - Ansible 사용자를 생성합니다.
  - 암호 없는 SSH 액세스를 활성화합니다.
  - 선택적으로 방화벽을 구성합니다.

### 5.2. RED HAT CEPH STORAGE 클러스터 설치

**ceph-ansible** 플레이북과 함께 Ansible 애플리케이션을 사용하여 베어 메탈 또는 컨테이너에 Red Hat Ceph Storage를 설치합니다. 프로덕션 환경에서 Ceph 스토리지 클러스터를 사용하려면 최소 3개의 모니터 노드와 여러 OSD 데몬이 포함된 OSD 노드 3개가 있어야 합니다. 프로덕션 환경에서 실행되는 일반적인 Ceph 스토리지 클러스터는 일반적으로 10개 이상의 노드로 구성됩니다.

다음 절차에서는 달리 지침이 없는 경우 Ansible 관리 노드에서 명령을 실행합니다. 이 절차는 지정하지 않는 한 베어 메탈 및 컨테이너 배포에 모두 적용됩니다.



**중요**

Ceph는 하나의 모니터로 실행할 수 있지만 프로덕션 클러스터에서 고가용성을 보장하기 위해 Red Hat은 모니터 노드가 세 개 이상인 배포만 지원합니다.

**중요**

Red Hat Enterprise Linux 7.7의 컨테이너에 Red Hat Ceph Storage 4를 배포하면 Red Hat Enterprise Linux 8 컨테이너 이미지에 Red Hat Ceph Storage 4가 배포됩니다.

**사전 요구 사항**

- 유효한 고객 서브스크립션.
- Ansible 관리 노드에 대한 루트 수준 액세스.
- Ansible 애플리케이션과 함께 사용할 **ansible** 사용자 계정입니다.
- Red Hat Ceph Storage Tools 및 Ansible 리포지토리 활성화
- ISO 설치의 경우 Ansible 노드에 최신 ISO 이미지를 다운로드합니다. Red Hat Ceph Storage 설치 가이드 [의 Red Hat Ceph Storage 리포지토리 활성화](#) 장을 참조하십시오.

**절차**

1. Ansible 관리 노드에 **root** 사용자 계정으로 로그인합니다.
2. 모든 배포가 **베어 메탈** 또는 **컨테이너**의 경우 **ceph-ansible** 패키지를 설치합니다.

**Red Hat Enterprise Linux 7**

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

**Red Hat Enterprise Linux 8**

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

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

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

4. 새 **yml** 파일을 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
```

- a. **베어 메탈** 배포:

```
[root@admin ceph-ansible]# cp site.yml.sample site.yml
```

- b. **컨테이너** 배포:

```
[root@admin ceph-ansible]# cp site-container.yml.sample site-container.yml
```

## 5. 새 파일을 편집합니다.

- a. **group\_vars/all.yml** 파일을 편집하려면 을 엽니다.

**중요**

사용자 정의 스토리지 클러스터 이름 사용은 지원되지 않습니다. 클러스터 매개변수를 **ceph** 이외의 값으로 설정하지 마십시오. 사용자 지정 스토리지 클러스터 이름 사용은 **librados**, Ceph Object Gateway 및 RADOS 블록 장치 미러링과 같은 Ceph 클라이언트에서만 지원됩니다.

**주의**

기본적으로 Ansible은 설치된 **firewalld** 서비스를 다시 시작하려고 하지만 마스킹되어 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** 파일에서 **configure\_firewall** 옵션을 사용할 필요가 없습니다.

**참고**

**ceph\_rhcs\_version** 옵션을 4 로 설정하면 최신 버전의 Red Hat Ceph Storage 4가 가져옵니다.

**참고**

Red Hat은 **group\_vars/all.yml** 파일에서 **dashboard\_enabled** 옵션을 **True** 로 설정하고 **False** 로 변경하지 않는 것이 좋습니다. 대시보드를 비활성화하려면 [Ceph 대시보드 비활성화](#)를 참조하십시오.

**참고**

대시보드 관련 구성 요소는 컨테이너화되어 있습니다. 따라서 **베어 메탈** 또는 **컨테이너 배포**의 경우 **ceph\_docker\_registry\_username** 및 **ceph\_docker\_registry\_password** 매개변수를 포함해야 **ceph-ansible**이 대시보드에 필요한 컨테이너 이미지를 가져올 수 있습니다.

**참고**

Red Hat Registry Service 계정이 없는 경우 레지스트리 서비스 계정 웹 페이지를 사용하여 생성합니다. 토큰을 만들고 관리하는 방법에 대한 자세한 내용은 [Red Hat Container Registry Authentication](#) Knowledgebase 문서를 참조하십시오.



## 참고

**ceph\_docker\_registry\_username** 및 **ceph\_docker\_registry\_password** 매개변수에 서비스 계정을 사용하는 것 외에도 고객 포털 자격 증명을 사용하지 만 보안을 보장하기 위해 **ceph\_docker\_registry\_password** 매개변수를 암호화할 수 있습니다. 자세한 내용은 [ansible-vault](#)를 사용하여 Ansible 암호 변수 암호화를 참조하십시오.

- i. CDN 설치를 위한 **all.yml** 파일의 배어 메탈 예:

```
fetch_directory: ~/ceph-ansible-keys
ceph_origin: repository
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 4
monitor_interface: eth0 1
public_network: 192.168.0.0/24
ceph_docker_registry: registry.redhat.io
ceph_docker_registry_auth: true
ceph_docker_registry_username: SERVICE_ACCOUNT_USER_NAME
ceph_docker_registry_password: TOKEN
dashboard_admin_user:
dashboard_admin_password:
node_exporter_container_image: registry.redhat.io/openshift4/ose-prometheus-
node-exporter:v4.6
grafana_admin_user:
grafana_admin_password:
grafana_container_image: registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8
prometheus_container_image: registry.redhat.io/openshift4/ose-prometheus:v4.6
alertmanager_container_image: registry.redhat.io/openshift4/ose-prometheus-
alertmanager:v4.6
```

- 1** 공용 네트워크의 인터페이스입니다.



## 중요

Red Hat Ceph Storage 4.1부터 **/usr/share/ceph-ansible/group\_vars/all.yml**에서 **dashboard\_admin\_password** 및 **grafana\_admin\_password**의 주석 처리를 해제하거나 설정해야 합니다. 각각에 대해 보안 암호를 설정합니다. **dashboard\_admin\_user** 및 **grafana\_admin\_user**의 사용자 지정 사용자 이름도 설정합니다.



## 참고

Red Hat Ceph Storage 4.2의 경우 설치를 위해 로컬 레지스트리를 사용한 경우 Prometheus 이미지 태그에 4.6을 사용합니다.

- ii. ISO 설치를 위한 **all.yml** 파일의 배어 메탈 예:

```
fetch_directory: ~/ceph-ansible-keys
ceph_origin: repository
ceph_repository: rhcs
ceph_repository_type: iso
```

```

ceph_rhcs_iso_path: /home/rhceph-4-rhel-8-x86_64.iso
ceph_rhcs_version: 4
monitor_interface: eth0 1
public_network: 192.168.0.0/24
ceph_docker_registry: registry.redhat.io
ceph_docker_registry_auth: true
ceph_docker_registry_username: SERVICE_ACCOUNT_USER_NAME
ceph_docker_registry_password: TOKEN
dashboard_admin_user:
dashboard_admin_password:
node_exporter_container_image: registry.redhat.io/openshift4/ose-prometheus-
node-exporter:v4.6
grafana_admin_user:
grafana_admin_password:
grafana_container_image: registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8
prometheus_container_image: registry.redhat.io/openshift4/ose-prometheus:v4.6
alertmanager_container_image: registry.redhat.io/openshift4/ose-prometheus-
alertmanager:v4.6

```

**1** 공용 네트워크의 인터페이스입니다.

iii. **all.yml** 파일의 컨테이너 예:

```

fetch_directory: ~/ceph-ansible-keys
monitor_interface: eth0 1
public_network: 192.168.0.0/24
ceph_docker_image: rhceph/rhceph-4-rhel8
ceph_docker_image_tag: latest
containerized_deployment: true
ceph_docker_registry: registry.redhat.io
ceph_docker_registry_auth: true
ceph_docker_registry_username: SERVICE_ACCOUNT_USER_NAME
ceph_docker_registry_password: TOKEN
ceph_origin: repository
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 4
dashboard_admin_user:
dashboard_admin_password:
node_exporter_container_image: registry.redhat.io/openshift4/ose-prometheus-
node-exporter:v4.6
grafana_admin_user:
grafana_admin_password:
grafana_container_image: registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8
prometheus_container_image: registry.redhat.io/openshift4/ose-prometheus:v4.6
alertmanager_container_image: registry.redhat.io/openshift4/ose-prometheus-
alertmanager:v4.6

```

**1** 공용 네트워크의 인터페이스입니다.

**중요**

[Red Hat Ecosystem Catalog](#) 에서 최신 컨테이너 이미지 태그를 조회하여 모든 최신 패치가 적용된 최신 컨테이너 이미지를 설치합니다.

- iv. **all.yml** 파일의 컨테이너 예: Red Hat Ceph Storage 노드가 배포 중에 인터넷에 액세스할 수 없는 경우:

```
fetch_directory: ~/ceph-ansible-keys
monitor_interface: eth0 1
public_network: 192.168.0.0/24
ceph_docker_image: rhceph/rhceph-4-rhel8
ceph_docker_image_tag: latest
containerized_deployment: true
ceph_docker_registry: LOCAL_NODE_FQDN:5000
ceph_docker_registry_auth: false
ceph_origin: repository
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 4
dashboard_admin_user:
dashboard_admin_password:
node_exporter_container_image: LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus-node-exporter:v4.6
grafana_admin_user:
grafana_admin_password:
grafana_container_image: LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-dashboard-
rhel8
prometheus_container_image: LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus:4.6
alertmanager_container_image: LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus-alertmanager:4.6
```

- 1** 공용 네트워크의 인터페이스입니다.

**대체 버전**

- 로컬 호스트 FQDN을 사용한 LOCAL\_NODE\_FQDN.

- v. Red Hat Ceph Storage 4.2에서 **dashboard\_protocol** 은 **https** 로 설정되고 Ansible은 대시보드 및 grafana 키 및 인증서를 생성합니다. 사용자 정의 인증서의 경우 **all.yml** 파일에서 **dashboard\_cert**, **dashboard\_key**, **grafana\_cert**, **grafana\_cert** 에 대한 Ansible 설치 프로그램 호스트의 경로를 업데이트합니다. 베어 메탈 또는 컨테이너 배포의 경우 **grafana\_key** 입니다.

**구문**

```
dashboard_protocol: https
dashboard_port: 8443
dashboard_cert: 'DASHBOARD_CERTIFICATE_PATH'
dashboard_key: 'DASHBOARD_KEY_PATH'
dashboard_tls_external: false
dashboard_grafana_api_no_ssl_verify: "{{ True if dashboard_protocol == 'https' and
```

```
not grafana_cert and not grafana_key else False }}"
grafana_cert: 'GRAFANA_CERTIFICATE_PATH'
grafana_key: 'GRAFANA_KEY_PATH'
```

- b. http 또는 https 프록시와 함께 연결할 수 있는 컨테이너 레지스트리를 사용하여 Red Hat Ceph Storage를 설치하려면 **group\_vars/all.yml** 파일에서 **ceph\_docker\_http\_proxy** 또는 **ceph\_docker\_https\_proxy** 변수를 설정합니다.

#### 예제

```
ceph_docker_http_proxy: http://192.168.42.100:8080
ceph_docker_https_proxy: https://192.168.42.100:8080
```

프록시 구성에 대해 일부 호스트를 제외해야 하는 경우 **group\_vars/all.yml** 파일의 **ceph\_docker\_no\_proxy** 변수를 사용합니다.

#### 예제

```
ceph_docker_no_proxy: "localhost,127.0.0.1"
```

- c. Red Hat Ceph Storage 프록시 설치를 위해 **all.yml** 파일을 편집하는 것 외에도 **/etc/environment** 파일을 편집합니다.

#### 예제

```
HTTP_PROXY: http://192.168.42.100:8080
HTTPS_PROXY: https://192.168.42.100:8080
NO_PROXY: "localhost,127.0.0.1"
```

그러면 podman을 트리거하여 prometheus, grafana-server, alertmanager, node-exporter와 같은 컨테이너화된 서비스를 시작하고 필요한 이미지를 다운로드합니다.

- d. 모든 배포의 경우 베어 메탈 또는 컨테이너에 대해 **group\_vars/osds.yml** 파일을 편집합니다.



#### 중요

운영 체제가 설치된 장치에 OSD를 설치하지 마십시오. 운영 체제와 OSD 간에 동일한 장치를 공유하면 성능 문제가 발생합니다.

Ceph-ansible은 **ceph-volume** 툴을 사용하여 Ceph 사용을 위해 스토리지 장치를 준비합니다. 다양한 방법으로 스토리지 장치를 사용하여 특정 워크로드에 대한 성능을 최적화하도록 **osds.yml**을 구성할 수 있습니다.



#### 중요

아래 모든 예제에서는 Ceph 형식에서 장치의 데이터를 저장하는 데 사용하는 BlueStore 오브젝트 저장소를 사용합니다. 이전에는 Ceph에서 FileStore를 오브젝트 저장소로 사용했습니다. BlueStore는 더 많은 기능과 향상된 성능을 제공하기 때문에 이 형식은 새로운 Red Hat Ceph Storage 4.0 설치에서 더 이상 사용되지 않습니다. FileStore를 계속 사용할 수는 있지만 이를 사용하려면 Red Hat 지원 예외가 필요합니다. BlueStore에 대한 자세한 내용은 [Red Hat Ceph Storage 아키텍처 가이드의 Ceph BlueStore](#)를 참조하십시오.

i. 자동 검색

```
osd_auto_discovery: true
```

위 예제에서는 시스템의 빈 스토리지 장치를 모두 사용하여 OSD를 생성하므로 명시적으로 지정할 필요가 없습니다. **ceph-volume** 툴에서는 빈 장치를 확인하므로 비어 있지 않은 장치는 사용되지 않습니다.



참고

나중에 **purge-docker-cluster.yml** 또는 **purge-cluster.yml** 을 사용하여 클러스터를 제거하도록 결정하는 경우 **osd\_auto\_discovery** 를 주석 처리하고 **osds.yml** 파일에 OSD 장치를 선언해야 합니다. 자세한 내용은 [Ansible에서 배포한 스토리지 클러스터 삭제](#)를 참조하십시오.

ii. 간단한 설정

첫 번째 시나리오

```
devices:
  - /dev/sda
  - /dev/sdb
```

또는

두 번째 시나리오

```
devices:
  - /dev/sda
  - /dev/sdb
  - /dev/nvme0n1
  - /dev/sdc
  - /dev/sdd
  - /dev/nvme1n1
```

또는

세 번째 시나리오

```
lvm_volumes:
  - data: /dev/sdb
  - data: /dev/sdc
```

또는

네 번째 시나리오

```
lvm_volumes:
  - data: /dev/sdb
  - data: /dev/nvme0n1
```

**devices** 옵션을 단독으로 사용하는 경우 **ceph-volume lvm** 배치 모드에서 OSD 구성을 자동으로 최적화합니다.



첫 번째 시나리오에서는 **장치**가 기존 하드 드라이브 또는 SSD인 경우 장치당 하나의 OSD가 생성됩니다.

두 번째 시나리오에서는 기존 하드 드라이브와 SSD가 혼합된 경우 데이터는 기존 하드 드라이브(**sda,sdb**)에 배치되고 BlueStore 데이터베이스는 SSD(**nvme0n1**)에서 최대한 크게 생성됩니다. 마찬가지로 데이터는 기존의 하드 드라이브(**sdс,sdd**)에 배치되고, BlueStore 데이터베이스는 언급된 장치 순서에 관계없이 SSD **nvme1n1**에 생성됩니다.



### 참고

기본적으로 **ceph-ansible**은 **bluestore\_block\_db\_size** 및 **bluestore\_block\_wal\_size**의 기본값을 재정의하지 않습니다. **group\_vars/all.yml** 파일에서 **ceph\_conf\_overrides**를 사용하여 **bluestore\_block\_db\_size**를 설정할 수 있습니다. **bluestore\_block\_db\_size** 값은 2GB보다 커야 합니다.

세 번째 시나리오에서는 데이터가 기존 하드 드라이브(**sdb,sdc**)에 배치되고 BlueStore 데이터베이스는 동일한 장치에 배치됩니다.

네 번째 시나리오에서는 데이터가 기존 하드 드라이브(**sdb**) 및 SSD(**nvme1n1**)에 배치되고 BlueStore 데이터베이스가 동일한 장치에 배치됩니다. 이는 BlueStore 데이터베이스가 SSD에 배치된 **devices** 지시문을 사용하는 것과 다릅니다.



### 중요

**ceph-volume lvm** 배치 모드 명령은 기존 하드 드라이브와 BlueStore 데이터베이스에 데이터를 SSD에 배치하여 최적화된 OSD 구성을 생성합니다. 사용할 논리 볼륨 및 볼륨 그룹을 지정하려면 아래 고급 구성 시나리오를 따라 직접 생성할 수 있습니다.

## iii. 고급 설정

### 첫 번째 시나리오

```
devices:
  - /dev/sda
  - /dev/sdb
dedicated_devices:
  - /dev/sdx
  - /dev/sdy
```

또는

### 두 번째 시나리오

```
devices:
  - /dev/sda
  - /dev/sdb
dedicated_devices:
  - /dev/sdx
  - /dev/sdy
bluestore_wal_devices:
  - /dev/nvme0n1
  - /dev/nvme0n2
```

첫 번째 시나리오에는 두 개의 OSD가 있습니다. **sda** 및 **sdb** 장치에는 각각 자체 데이터 세그먼트와 write-ahead 로그가 있습니다. 추가 사전 **dedicated\_devices** 는 각각 **sdx** 및 **sdz** 에서 **block.db** 라고도 하는 데이터베이스를 격리하는 데 사용됩니다.

두 번째 시나리오에서 다른 추가 사전 **bluestore\_wal\_devices** 는 NVMe 장치 **nvme0n1** 및 **nvme0n2** 에서 write-ahead 로그를 격리하는 데 사용됩니다. 장치, **dedicated\_devices** 및 **bluestore\_wal\_devices** 를 함께 사용하면 OSD의 모든 구성 요소를 별도의 장치에 분리할 수 있습니다. 이와 같이 OSD를 배치하면 전반적인 성능이 향상될 수 있습니다.

#### iv. 미리 생성된 논리 볼륨

##### 첫 번째 시나리오

```
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
```

또는

##### 두 번째 시나리오

```
lvm_volumes:
- data: /dev/sdb
  db: db-lv1
  db_vg: db-vg1
  wal: wal-lv1
  wal_vg: wal-vg1
```

기본적으로 Ceph는 논리 볼륨 관리자를 사용하여 OSD 장치에 논리 볼륨을 생성합니다. 위의 간단한 구성 및 고급 구성 예제에서 Ceph는 장치에 논리 볼륨을 자동으로 생성합니다. **lvm\_volumes** 사전을 지정하여 Ceph에서 이전에 생성한 논리 볼륨을 사용할 수 있습니다.

첫 번째 시나리오에서는 데이터가 전용 논리 볼륨, 데이터베이스 및 WAL에 배치됩니다. 데이터, 데이터 및 WAL 또는 데이터 및 데이터베이스만 지정할 수도 있습니다. **data:** 행은 데이터를 저장할 논리 볼륨 이름을 지정해야 하며 **data\_vg:** 는 데이터 논리 볼륨이 포함된 볼륨 그룹의 이름을 지정해야 합니다. 마찬가지로 **db:** 는 데이터베이스가 저장된 논리 볼륨을 지정하는 데 사용되며, **db\_vg:** 는 논리 볼륨이 있는 볼륨 그룹을 지정하는 데 사용됩니다. **wal:** 행은 WAL이 저장된 논리 볼륨을 지정하고 **wal\_vg:** 행은 해당 볼륨이 포함된 볼륨 그룹을 지정합니다.

두 번째 시나리오에서는 **data:** option에 대해 실제 장치 이름이 설정되며, 이렇게 하면 **data\_vg:** 옵션을 지정할 필요가 없습니다. BlueStore 데이터베이스 및 WAL 장치의 논리 볼륨 이름과 볼륨 그룹 세부 정보를 지정해야 합니다.



### 중요

**lvm\_volumes:** 를 사용하면 블록 그룹과 논리 볼륨을 미리 만들어야 합니다. 블록 그룹과 논리 볼륨은 **ceph-anible**에서 생성되지 않습니다.



### 참고

모든 NVMe SSD를 사용하는 경우 **osds\_per\_device**를 설정합니다. 2 자세한 내용은 Red Hat Ceph Storage 설치 가이드의 모든 NVMe 스토리지에 대한 OSD Ansible 설정 구성을 참조하십시오.



### 참고

Ceph OSD 노드를 재부팅한 후에는 블록 장치 할당이 변경될 수 있습니다. 예를 들어 **sdc** 는 **sdd** 가 될 수 있습니다. 기존 블록 장치 이름 대신 **/dev/disk/by-path/** 장치 경로와 같은 영구적인 이름 지정 장치를 사용할 수 있습니다.

6. 베어 메탈 또는 컨테이너에 있는 모든 배포의 경우 Ansible 인벤토리 파일을 생성한 다음 편집을 위해 엽니다.

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

그에 따라 **hosts** 파일을 편집합니다.



### 참고

Ansible 인벤토리 위치 편집에 대한 자세한 내용은 [Ansible 인벤토리 위치 구성](#)을 참조하십시오.

- a. **[grafana-server]** 아래에 노드를 추가합니다. 이 역할은 Grafana 및 Prometheus를 설치하여 Ceph 클러스터의 성능에 대한 실시간 정보를 제공합니다. 이러한 지표는 Ceph 대시보드에 제공되어 클러스터를 모니터링하고 관리할 수 있습니다. 대시보드, Grafana 및 Prometheus를 설치해야 합니다. Ansible 관리 노드에서 지표 기능을 배치할 수 있습니다. 이 경우 노드의 시스템 리소스가 독립형 매트릭 노드에 필요한 것보다 크지 확인하십시오.

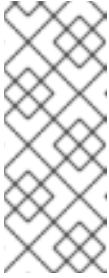
```
[grafana-server]
GRAFANA-SERVER_NODE_NAME
```

- b. **[mons]** 섹션 아래에 모니터 노드를 추가합니다.

```
[mons]
MONITOR_NODE_NAME_1
MONITOR_NODE_NAME_2
MONITOR_NODE_NAME_3
```

- c. **[osds]** 섹션에 OSD 노드를 추가합니다.

```
[osds]
OSD_NODE_NAME_1
OSD_NODE_NAME_2
OSD_NODE_NAME_3
```



**참고**

노드 이름이 숫자 순차적이면 범위 지정자([1:10])를 노드 이름 끝에 추가할 수 있습니다. 예를 들어 다음과 같습니다.

```
[osds]
example-node[1:10]
```



**참고**

새 설치의 OSD의 경우 기본 오브젝트 저장소 형식은 BlueStore입니다.

- d. 선택적으로, 컨테이너 배포에서 **[mon]** 및 **[osd]** 섹션에서 동일한 노드를 추가하여 Ceph OSD 데몬을 하나의 노드의 Ceph OSD 데몬과 함께 연결합니다. 아래 추가 리소스 섹션에서 자세한 내용은 Ceph 데몬 일치 링크를 참조하십시오.
- e. **[mgrs]** 섹션에 Ceph Manager(**ceph-mgr**) 노드를 추가합니다. Ceph Monitor 데몬을 사용하여 Ceph Manager 데몬을 배치합니다.

```
[mgrs]
MONITOR_NODE_NAME_1
MONITOR_NODE_NAME_2
MONITOR_NODE_NAME_3
```

7. 선택적으로 모든 배포, **베어 메탈** 또는 **컨테이너**에 대해 호스트 특정 매개 변수를 사용하려면 호스트 파일과 함께 **host\_vars** 디렉토리를 생성하여 호스트와 관련된 모든 매개 변수를 포함합니다.

- a. **host\_vars** 디렉토리를 생성합니다.

```
[ansible@admin ~]$ mkdir /usr/share/ceph-ansible/host_vars
```

- b. **host\_vars** 디렉터리로 변경합니다.

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

- c. 호스트 파일을 생성합니다. 파일 이름에 host-name-short-name 형식을 사용합니다. 예를 들면 다음과 같습니다.

```
[ansible@admin host_vars]$ touch tower-osd6
```

- d. 호스트별 매개 변수를 사용하여 파일을 업데이트합니다. 예를 들면 다음과 같습니다.

- i. **베어 메탈** 배포에서는 **devices** 매개 변수를 사용하여 OSD 노드에서 사용할 장치를 지정합니다. OSD에서 장치를 사용하면 OSD에서 이름이 다른 장치를 사용하거나 장치 중 하나가 OSD에서 실패한 경우 유용합니다.

```
devices:
  DEVICE_1
  DEVICE_2
```

**예제**

```
devices:
  /dev/sdb
  /dev/sdc
```



### 참고

장치를 지정하지 않는 경우 **group\_vars/osds.yml** 파일에서 **osd\_auto\_discovery** 매개변수를 **true** 로 설정합니다.

8. 필요한 경우 모든 배포, **베어 메탈** 또는 **컨테이너** 의 경우 Ceph Ansible 을 사용하여 사용자 지정 NetNamespace 계층 구조를 만들 수 있습니다.

- a. Ansible 인벤토리 파일을 설정합니다. **osd\_crush\_location** 매개변수를 사용하여 OSD 호스트의 계층 구조에 OSD 호스트가 될 위치를 지정합니다. OSD의 위치를 지정하려면 두 개 이상의 ArgoCD 버킷 유형을 지정해야 하며 하나의 버킷 유형이 호스트여야 합니다. 기본적으로 **root,datacenter,room,row,pod,pdu,rack,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. **group\_vars/osds.yml** 파일을 편집하고 **crush\_rule\_config** 및 **create\_crush\_tree** 매개 변수를 **True** 로 설정합니다. 기본 NetNamespace 규칙을 사용하지 않는 경우 하나 이상의 NetNamespace 규칙을 만듭니다. 예를 들면 다음과 같습니다.

```
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
```

더 빠른 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** OSD가 모두 배포되지 않으면 기본 **nmap** 규칙이 실패합니다.

- c. **group\_vars/clients.yml** 파일에 **crush\_rules** 를 생성한 상태에서 풀 을 생성합니다.

### 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. 트리 보기:

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

- e. 풀을 검증합니다.

```
[root@mon ~]# for i in $(rados lspools); do echo "pool: $i"; ceph osd pool get $i
crush_rule; done

pool: pool1
crush_rule: HDD
```

9. 모든 배포의 경우 **베어 메탈** 또는 **컨테이너**에 대해 로그인하거나 **ansible** 사용자로 전환합니다.

- a. Ansible 이 **ceph-ansible** 플레이북에서 생성한 임시 값을 저장하는 **ceph-ansible-keys** 디렉토리를 생성합니다.

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

- b. **/usr/share/ceph-ansible/** 디렉토리로 변경합니다.

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

- c. Ansible 이 Ceph 노드에 연결할 수 있는지 확인합니다.

```
[ansible@admin ceph-ansible]$ ansible all -m ping -i hosts
```

10. **ceph-ansible** 플레이북을 실행합니다.

## a. 베어 메탈 배포:

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

## b. 컨테이너 배포:

```
[ansible@admin ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```



## 참고

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

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



## 참고

배포 속도를 높이려면 **ansible-playbook** 에 **--forks** 옵션을 사용합니다. 기본적으로 **ceph-ansible** 세트는 포크를 **20** 으로 설정합니다. 이 설정을 사용하면 최대 20개의 노드가 동시에 설치됩니다. 한 번에 최대 30개의 노드를 설치하려면 **ansible-playbook --forks 30 PLAYBOOK Files - i** 호스트를 실행합니다. 과도하게 사용되지 않도록 관리 노드의 리소스를 모니터링해야 합니다. 이러한 값이 있는 경우 **--forks** 에 전달되는 수를 줄입니다.

## 11. Ceph 배포가 완료될 때까지 기다립니다.

## 출력에

```
INSTALLER STATUS *****
Install Ceph Monitor      : Complete (0:00:30)
Install Ceph Manager     : Complete (0:00:47)
Install Ceph OSD         : Complete (0:00:58)
Install Ceph RGW         : Complete (0:00:34)
Install Ceph Dashboard   : Complete (0:00:58)
Install Ceph Grafana     : Complete (0:00:50)
Install Ceph Node Exporter : Complete (0:01:14)
```

## 12. Ceph 스토리지 클러스터의 상태를 확인합니다.

## a. 베어 메탈 배포:

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

## b. 컨테이너 배포:

## Red Hat Enterprise Linux 7

```
[root@mon ~]# docker exec ceph-mon-ID ceph health
```

## Red Hat Enterprise Linux 8

```
[root@mon ~]# podman exec ceph-mon-ID ceph health
```

#### 대체 버전

- Ceph Monitor 노드의 호스트 이름이 있는 **ID**:

#### 예제

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

13. 모든 배포에서는 **베어 메탈** 또는 **컨테이너**에 대해 **rados** 를 사용하여 스토리지 클러스터가 작동하는지 확인합니다.

- Ceph Monitor 노드에서 8개의 배치 그룹(PG)이 있는 테스트 풀을 생성합니다.

#### 구문

```
[root@mon ~]# ceph osd pool create POOL_NAME PG_NUMBER
```

#### 예제

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

- hello-world.txt** 라는 파일을 생성합니다.

#### 구문

```
[root@mon ~]# vim FILE_NAME
```

#### 예제

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

- hello-world** 라는 오브젝트를 사용하여 test 풀에 **hello-world.txt** 를 업로드합니다.

#### 구문

```
[root@mon ~]# rados --pool POOL_NAME put OBJECT_NAME OBJECT_FILE_NAME
```

#### 예제

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

- 테스트 풀에서 **hello-world** 를 파일 이름 **fetch.txt** 로 다운로드합니다.

#### 구문

```
[root@mon ~]# rados --pool POOL_NAME get OBJECT_NAME OBJECT_FILE_NAME
```



## 예제

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

e. **fetch.txt** 내용을 확인합니다.

```
[root@mon ~]# cat fetch.txt
"Hello World!"
```



## 참고

스토리지 클러스터 상태를 확인하는 것 외에도 **ceph-medic** 유틸리티를 사용하여 Ceph Storage 클러스터를 전반적으로 진단할 수 있습니다. Red Hat Ceph Storage 4 문제 해결 가이드의 [Ceph Storage 클러스터 진단을 위한 ceph-medic 설치 및 사용](#) 장을 참조하십시오.

## 추가 리소스

- 공통 [Ansible 설정](#) 목록입니다.
- 일반 [OSD 설정](#) 목록입니다.
- 자세한 내용은 [컨테이너화된 Ceph 데몬 공동 배치](#)를 참조하십시오.

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

전체 성능을 높이기 위해 스토리지에 NVMe(Non-volatile memory express) 장치만 사용하도록 Ansible을 구성할 수 있습니다. 일반적으로 장치당 하나의 OSD만 구성되어 있으므로 NVMe 장치의 처리량이 향상됩니다.



## 참고

SSD와HD를 혼합하면 OSD의 데이터가 아닌 데이터베이스에 SSD가 사용되거나 **block.db**가 사용됩니다.



## 참고

테스트에서 최적의 성능을 제공하는 각 NVMe 장치에서 OSD를 두 개 구성하는 것이 확인되었습니다. Red Hat은 **osds\_per\_device** 옵션을 **2**로 설정하는 것을 권장하지만 필수는 아닙니다. 다른 값은 환경에 더 나은 성능을 제공할 수 있습니다.

## 사전 요구 사항

- Ansible 관리 노드에 액세스합니다.
- **ceph-ansible** 패키지 설치.

## 절차

1. **osds\_per\_device**를 설정합니다. 2 in **group\_vars/osds.yml**:

```
osds_per_device: 2
```

2. 장치 아래에 NVMe 장치를 나열합니다.

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

3.

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

```
osds_per_device: 2
devices:
- /dev/nvme0n1
- /dev/nvme1n1
- /dev/nvme2n1
- /dev/nvme3n1
```



참고

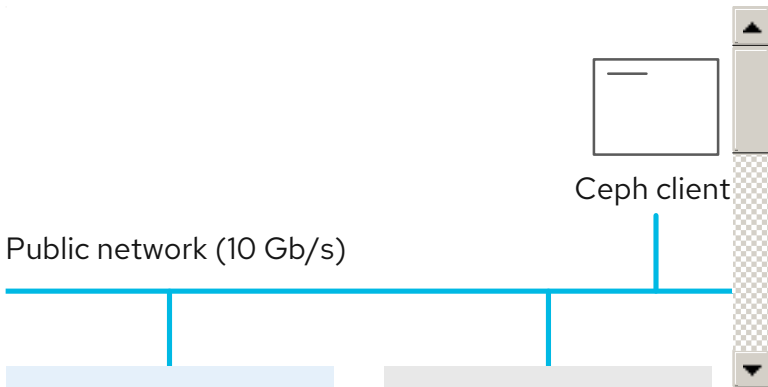
`lvm_volumes` 가 아닌 이 구성으로 장치를 사용해야 합니다. 이는 `lvm_volumes` 가 일반적으로 미리 생성된 논리 볼륨에 사용되며 `osds_per_device` 는 Ceph에서 자동 논리 볼륨 생성을 의미하기 때문입니다.

추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage 설치 가이드](#)에서 [Red Hat Ceph Storage 클러스터 설치](#)를 참조하십시오.

5.4. 메타데이터 서버 설치

**Ansible** 자동화 애플리케이션을 사용하여 **Ceph 메타데이터 서버(MDS)**를 설치합니다. 메타데이터 서버 데몬은 **Ceph** 파일 시스템을 배포하는 데 필요합니다.



## 사전 요구 사항

- 작동 중인 **Red Hat Ceph Storage** 클러스터.
- 암호 없는 **SSH** 액세스를 활성화합니다.

## 절차

**Ansible** 관리 노드에서 다음 단계를 수행합니다.

1. 새 섹션 **[mds]** 을 **/etc/ansible/hosts** 파일에 추가합니다.

```
[mdss]
MDS_NODE_NAME1
MDS_NODE_NAME2
MDS_NODE_NAME3
```

**MDS\_NODE\_NAME** 을 **Ceph** 메타데이터 서버를 설치하려는 노드의 호스트 이름으로 바꿉니다.

또는 **[osds]** 및 **[mds]** 섹션에 동일한 노드를 추가하여 메타데이터 서버를 한 노드의 **OSD** 데이터와 함께 배치할 수 있습니다.

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

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

3. 선택적으로 기본 변수를 변경할 수 있습니다.

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

```
[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.

•

베어 메탈 배포:

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

•

컨테이너 배포:

```
[ansible@admin ceph-ansible]$ ansible-playbook site-container.yml --limit mdss -i hosts
```

5.

이제 메타데이터 서버를 설치한 후 구성할 수 있습니다. 자세한 내용은 **Red Hat Ceph Storage 파일 시스템 가이드의 Ceph 파일 시스템 메타데이터 서버 장**을 참조하십시오.

#### 추가 리소스

•

**Red Hat Ceph Storage 4용 Ceph 파일 시스템 가이드**

•

자세한 내용은 **컨테이너화된 Ceph 데몬 공동 배치**를 참조하십시오.

•

자세한 내용은 **제한 옵션** 이해를 참조하십시오.

### 5.5. 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

•

활성 + 클린 상태의 실행 중인 **Ceph** 스토리지 클러스터.

•

**요구 사항**에 나열된 작업을 수행합니다.

•

암호 없는 **SSH** 액세스를 활성화합니다.

## 절차

**Ansible** 관리 노드에서 다음 작업을 수행합니다.

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

```
[clients]
CLIENT_NODE_NAME
```

**CLIENT\_NODE\_NAME** 을 **ceph-client** 역할을 설치할 노드의 호스트 이름으로 교체합니다.

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

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

3. **clients.yml.sample** 파일 이름을 **clients.yml** 이라는 새 복사본을 생성합니다.

```
[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 줄에 클라이언트 키를 추가합니다. 예를 들면 다음과 같습니다.

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

이제 전체 줄 예는 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.

필요한 경우 **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. **VDDK** 기능을 사용하여 사용자 지정 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** 사용자로 **Ansible** 플레이북을 실행합니다.

a.

베어 메탈 배포:

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

b.

컨테이너 배포:

```
[ansible@admin ceph-ansible]$ ansible-playbook site-container.yml --limit clients -i hosts
```

추가 리소스

- 자세한 내용은 [제한 옵션](#) 이해를 참조하십시오.

## 5.6. CEPH OBJECT GATEWAY 설치

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

### 사전 요구 사항

- **활성 + 클린** 상태에서 실행되는 **Red Hat Ceph Storage** 클러스터.
- **암호 없는 SSH** 액세스를 활성화합니다.
- **Ceph Object Gateway** 노드에서 **3장. Red Hat Ceph Storage 설치 요구사항**에 나열된 작업을 수행합니다.



### 주의

다중 사이트 구성에서 **Ceph Object Gateway**를 사용하려는 경우 **1 - 6** 단계만 완료합니다. 다중 사이트를 구성하기 전에 **Ansible** 플레이북을 실행하지 마십시오. 그러면 단일 사이트 구성에서 **Object Gateway**가 시작됩니다. 단일 사이트 구성에서 이미 시작된 후에는 **Ansible**에서 다중 사이트 설정으로 게이트웨이를 재구성할 수 없습니다. **1~6** 단계를 완료한 후 **다중 사이트 Ceph Object Gateways 구성** 섹션으로 이동하여 다중 사이트를 설정합니다.

### 절차

**Ansible** 관리 노드에서 다음 작업을 수행합니다.

1. **[rgws]** 섹션의 **/etc/ansible/hosts** 파일에 게이트웨이 호스트를 추가하여 **Ansible**에 대한 역할을 확인합니다. 호스트에 순차적인 이름 지정이 있는 경우 다음과 같이 범위를 사용합니다.

```
[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 제거하십시오.

```
copy_admin_key: true
```

5.

**all.yml** 파일에서 **radosgw\_interface** 를 지정해야 합니다.

```
radosgw_interface: <interface>
```

교체:

- **Ceph Object Gateway** 노드가 수신 대기하는 인터페이스를 사용하여 **<interface>**

예를 들어 다음과 같습니다.

```
radosgw_interface: eth0
```

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

자세한 내용은 **all.yml** 파일을 참조하십시오.

6.

일반적으로 기본 설정을 변경하려면 **rgws.yml** 파일의 설정 주석 처리를 제거하고 적절하게 변경합니다. **rgws.yml** 파일에 없는 설정을 추가로 변경하려면 **all.yml** 파일에서



`ceph_conf_overrides`: 를 사용합니다.

```
ceph_conf_overrides:
  client.rgw.rgw1:
    rgw_override_bucket_index_max_shards: 16
    rgw_bucket_default_quota_max_objects: 1638400
```

고급 구성 세부 정보는 **Red Hat Ceph Storage 4 Ceph Object Gateway for Production** 가이드를 참조하십시오. 고급 주제는 다음과 같습니다.

- **Ansible 그룹 구성**
- **스토리지 전략 개발.** 풀 생성 및 구성 방법에 대한 자세한 내용은 루트 풀 생성, 시스템 풀 생성 및 데이터 배치 전략 생성 섹션을 참조하십시오.

버킷 분할에 대한 구성 세부 정보는 **Bucket Sharding** 을 참조하십시오.

7.

**Ansible Playbook**을 실행합니다.



주의

다중 사이트를 설정하려는 경우 **Ansible** 플레이북을 실행하지 마십시오. 다중 사이트 **Ceph** 개체 게이트웨이 구성 섹션으로 이동하여 다중 사이트를 설정합니다.

a.

베어 메탈 배포:

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

b.

컨테이너 배포:

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



## 참고

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

단일 사이트 구성의 경우 **Ceph** 개체 게이트웨이를 **Ansible** 구성에 추가합니다.

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

다중 사이트 클러스터에 대한 설치가 완료된 후 [다중 사이트](#) 용 클러스터 구성에 대한 자세한 내용은 [Red Hat Ceph Storage 4 개체 게이트웨이 가이드](#)의 [다중 사이트 장](#)을 진행합니다.

## 추가 리소스

- 자세한 내용은 [제한 옵션](#) 이해를 참조하십시오.
- [Red Hat Ceph Storage 4 오브젝트 게이트웨이 가이드](#)

## 5.7. 다중 사이트 CEPH 개체 게이트웨이 구성

시스템 관리자는 재해 복구를 위해 클러스터 간에 데이터를 미러링하도록 다중 사이트 **Ceph** 개체 게이트웨이를 구성할 수 있습니다.

하나 이상의 **RGW** 영역으로 다중 사이트를 구성할 수 있습니다. 영역을 사용하면 내부 **RGW**가 독립적이며 영역 외부의 **RGW**와 분리할 수 있습니다. 이러한 방식으로, 한 영역에서 **RGW**에 기록된 데이터는 다른 영역에서 **RGW**에 의해 액세스될 수 없습니다.



## 주의

**Ceph-ansible**은 단일 사이트 구성에서 이미 사용된 후 다중 사이트 설정으로 게이트웨이를 재구성할 수 없습니다. 이 구성을 수동으로 배포할 수 있습니다. [Red Hat 지원팀](#)에 문의하십시오.



## 참고

Red Hat Ceph Storage 4.1에서는 `group_vars/all.yml` 파일에서 `rgw_multisite_endpoints_list` 값을 설정할 필요가 없습니다.

자세한 내용은 **Red Hat Ceph Storage Object Gateway 구성 및 관리 가이드**의 **다중 사이트** 섹션을 참조하십시오.

## 5.7.1. 사전 요구 사항

- **Red Hat Ceph Storage 클러스터 두 개**
- **Ceph Object Gateway 노드에서 Red Hat Ceph Storage 설치 가이드의 Red Hat Ceph Storage 설치 요구 사항** 섹션에 나열된 작업을 수행합니다.
- **각 Object Gateway 노드에 대해 Red Hat Ceph Storage 설치 가이드의 Ceph Object Gateway 설치** 섹션에서 1단계 6단계를 수행합니다.

## 5.7.2. 하나의 영역을 사용하여 다중 사이트 Ceph Object Gateway 구성

**Ceph-ansible**은 여러 **Ceph Object Gateway** 인스턴스가 있는 여러 스토리지 클러스터에서 한 영역의 데이터를 미러링하도록 **Ceph 오브젝트 게이트웨이**를 구성합니다.



## 주의

**Ceph-ansible**은 단일 사이트 구성에서 이미 사용된 후 다중 사이트 설정으로 게이트웨이를 재구성할 수 없습니다. 이 구성을 수동으로 배포할 수 있습니다. **Red Hat 지원팀**에 문의하십시오.

## 사전 요구 사항

- **Red Hat Ceph Storage 클러스터가 실행 중인 2개입니다.**
-

**Ceph Object Gateway** 노드에서 **Red Hat Ceph Storage 설치 가이드의 Red Hat Ceph Storage 설치 요구 사항** 섹션에 나열된 작업을 수행합니다.

- 각 **Object Gateway** 노드에 대해 **Red Hat Ceph Storage 설치 가이드의 Ceph Object Gateway 설치** 섹션에서 1단계 6단계를 수행합니다.

## 절차

1.

시스템 키를 생성하고 **multi-site-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
```

## 기본 스토리지 클러스터

a.

**Ceph-ansible** 구성 디렉터리로 이동합니다.

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

b.

**group\_vars/all.yml** 파일을 열고 편집합니다. **rgw\_multisite** 행의 주석을 제거하고 **true** 로 설정합니다. **rgw\_multisite\_proto** 매개변수의 주석을 제거합니다.

```
rgw_multisite: true
rgw_multisite_proto: "http"
```

c.

**/usr/share/ceph-ansible** 에 **host\_vars** 디렉토리를 만듭니다.

```
[root@ansible ceph-ansible]# mkdir host_vars
```

d.

기본 스토리지 클러스터의 각 **Object Gateway** 노드에 대해 **host\_vars** 에 파일을 만듭니다. 파일 이름은 **Ansible** 인벤토리 파일에서 사용된 이름과 동일해야 합니다. 예를 들어 **Object Gateway** 노드의 이름이 **rgw-primary** 인 경우 **host\_vars/rgw-primary** 파일을 만듭니다.

## 구문

```
touch host_vars/NODE_NAME
```

## 예제

```
[root@ansible ceph-ansible]# touch host_vars/rgw-primary
```



## 참고

다중 사이트 구성에 사용되는 클러스터에 여러 **Ceph Object Gateway** 노드가 있는 경우 각 노드에 대해 별도의 파일을 생성합니다.

e.

파일을 편집하고 해당 **Object Gateway** 노드의 모든 인스턴스에 대한 구성 세부 정보를 추가합니다. **ZONE\_NAME,ZONE\_GROUP\_NAME,ZONE\_USER\_NAME,ZONE\_DISPLAY\_NAME** 및 **REALM\_NAME** 업데이트와 함께 다음 설정을 구성합니다. **ACCESS\_KEY** 및 **ECDHERET\_KEY** 용으로 **multi-site-keys.txt** 파일에 저장된 임의의 문자열을 사용합니다.

## 구문

```
rgw_instances:
  - instance_name: 'INSTANCE_NAME'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: ZONE_NAME_1
    rgw_zonegroup: ZONE_GROUP_NAME_1
    rgw_realm: REALM_NAME_1
    rgw_zone_user: ZONE_USER_NAME_1
    rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
    system_access_key: ACCESS_KEY_1
    system_secret_key: SECRET_KEY_1
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: RGW_PRIMARY_PORT_NUMBER_1
```

## 예제

```

rgw_instances:
- instance_name: 'rgw0'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonesecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: paris
  rgw_zonegroup: idf
  rgw_realm: france
  rgw_zone_user: jacques.chirac
  rgw_zone_user_display_name: "Jacques Chirac"
  system_access_key: P9Eb6S8XNyo4dtZZUUMy
  system_secret_key: qqHCUtdNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
  radosgw_address: "{{_radosgw_address }}"
  radosgw_frontend_port: 8080

```

f.

**선택 사항:** 여러 인스턴스를 생성하려면 파일을 편집하고 각 **Object Gateway** 노드의 모든 인스턴스에 구성 세부 정보를 추가합니다. **rgw\_instances** 에서 항목을 업데이트하는 것과 함께 다음 설정을 구성합니다. **ACCESS\_KEY\_1** 및 **SECRET\_KEY\_1** 의 경우 **multi-site-keys-realm-1.txt** 파일에 저장된 임의의 문자열을 사용합니다.

구문

```

rgw_instances:
- instance_name: 'INSTANCE_NAME_1'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonesecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: ZONE_NAME_1
  rgw_zonegroup: ZONE_GROUP_NAME_1
  rgw_realm: REALM_NAME_1
  rgw_zone_user: ZONE_USER_NAME_1
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
  system_access_key: ACCESS_KEY_1
  system_secret_key: SECRET_KEY_1
  radosgw_address: "{{_radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
- instance_name: 'INSTANCE_NAME_2'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonesecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: ZONE_NAME_1

```

```

rgw_zonegroup: ZONE_GROUP_NAME_1
rgw_realm: REALM_NAME_1
rgw_zone_user: ZONE_USER_NAME_1
rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
system_access_key: ACCESS_KEY_1
system_secret_key: SECRET_KEY_1
radosgw_address: "{{ _radosgw_address }}"
radosgw_frontend_port: PORT_NUMBER_2

```

## 예제

```

rgw_instances:
- instance_name: 'rgw0'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonesecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: paris
  rgw_zonegroup: idf
  rgw_realm: france
  rgw_zone_user: jacques.chirac
  rgw_zone_user_display_name: "Jacques Chirac"
  system_access_key: P9Eb6S8XNyo4dtZZUUMy
  system_secret_key: qqHCUtdNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: 8080
- instance_name: 'rgw1'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonesecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: paris
  rgw_zonegroup: idf
  rgw_realm: france
  rgw_zone_user: jacques.chirac
  rgw_zone_user_display_name: "Jacques Chirac"
  system_access_key: P9Eb6S8XNyo4dtZZUUMy
  system_secret_key: qqHCUtdNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: 8081

```

- a. **Ceph-ansible** 구성 디렉터리로 이동합니다.

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

- b. **group\_vars/all.yml** 파일을 열고 편집합니다. **rgw\_multisite** 행의 주석을 제거하고 **true** 로 설정합니다. **rgw\_multisite\_proto** 매개변수의 주석을 제거합니다.

```
rgw_multisite: true
rgw_multisite_proto: "http"
```

- c. **/usr/share/ceph-ansible** 에 **host\_vars** 디렉토리를 만듭니다.

```
[root@ansible ceph-ansible]# mkdir host_vars
```

- d. 보조 스토리지 클러스터의 각 **Object Gateway** 노드에 대해 **host\_vars** 에 파일을 만듭니다. 파일 이름은 **Ansible** 인벤토리 파일에서 사용된 이름과 동일해야 합니다. 예를 들어 **Object Gateway** 노드의 이름이 **rgw-secondary** 인 경우 **host\_vars/rgw-secondary** 파일을 만듭니다.

구문

```
touch host_vars/NODE_NAME
```

예제

```
[root@ansible ceph-ansible]# touch host_vars/rgw-secondary
```



#### 참고

다중 사이트 구성에 사용되는 클러스터에 여러 **Ceph Object Gateway** 노드가 있는 경우 각 노드에 대해 파일을 생성합니다.



e.

다음 설정을 구성합니다. 첫 번째 클러스터에서 **ZONE\_USER\_NAME, ZONE\_DISPLAY\_NAME, ACCESS\_KEY, ECDHERET\_KEY, REALM\_NAME, ZONE\_GROUP\_NAME** 에 사용된 값과 동일한 값을 사용합니다. 기본 스토리지 클러스터의 **ZONE\_NAME** 에 다른 값을 사용합니다. **ECDHERET\_RGW\_NODE\_NAME** 을 **master** 영역의 **Ceph Object Gateway** 노드로 설정합니다. 기본 스토리지 클러스터와 비교하여 **rgw\_zonemaster, rgw\_zonesecondary, rgw\_zonegroupmaster** 에 대한 설정이 반전됩니다.

구문

```
rgw_instances:
- instance_name: 'INSTANCE_NAME_1'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: ZONE_NAME_2
  rgw_zonegroup: ZONE_GROUP_NAME_1
  rgw_realm: REALM_NAME_1
  rgw_zone_user: ZONE_USER_NAME_1
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
  system_access_key: ACCESS_KEY_1
  system_secret_key: SECRET_KEY_1
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
  endpoint:
    RGW_PRIMARY_HOSTNAME_ENDPOINT:RGW_PRIMARY_PORT_NUMBER_1
```

예제

```
rgw_instances:
- instance_name: 'rgw0'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: lyon
  rgw_zonegroup: idf
  rgw_realm: france
  rgw_zone_user: jacques.chirac
  rgw_zone_user_display_name: "Jacques Chirac"
  system_access_key: P9Eb6S8XNyo4dtZZUUMy
  system_secret_key: qqHCUtdfNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
```

```
radosgw_address: "{{ _radosgw_address }}"
radosgw_frontend_port: 8080
endpoint: http://rgw-primary:8081
```

f.

**선택 사항:** 여러 인스턴스를 생성하려면 파일을 편집하고 각 **Object Gateway** 노드의 모든 인스턴스에 구성 세부 정보를 추가합니다. `rgw_instances` 에서 항목을 업데이트하는 것과 함께 다음 설정을 구성합니다. `ACCESS_KEY_1` 및 `SECRET_KEY_1` 의 경우 `multi-site-keys-realm-1.txt` 파일에 저장된 임의의 문자열을 사용합니다. `RGW_PRIECDHE_HOSTNAME` 을 기본 스토리지 클러스터의 **Object Gateway** 노드로 설정합니다.

구문

```
rgw_instances:
- instance_name: 'INSTANCE_NAME_1'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonessecondary: true
  rgw_zonemaster: false
  rgw_zone: ZONE_NAME_2
  rgw_zonename: ZONE_GROUP_NAME_1
  rgw_realm: REALM_NAME_1
  rgw_zone_user: ZONE_USER_NAME_1
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
  system_access_key: ACCESS_KEY_1
  system_secret_key: SECRET_KEY_1
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
  endpoint: RGW_PRIMARY_HOSTNAME:RGW_PRIMARY_PORT_NUMBER_1
- instance_name: '_INSTANCE_NAME_2_'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonessecondary: true
  rgw_zonemaster: false
  rgw_zone: ZONE_NAME_2
  rgw_zonename: ZONE_GROUP_NAME_1
  rgw_realm: REALM_NAME_1
  rgw_zone_user: ZONE_USER_NAME_1
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
  system_access_key: ACCESS_KEY_1
  system_secret_key: SECRET_KEY_1
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
  endpoint: RGW_PRIMARY_HOSTNAME:RGW_PRIMARY_PORT_NUMBER_2
```

## 예제

```

rgw_instances:
- instance_name: 'rgw0'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonemaster: false
  rgw_zone: lyon
  rgw_zonegroup: idf
  rgw_realm: france
  rgw_zone_user: jacques.chirac
  rgw_zone_user_display_name: "Jacques Chirac"
  system_access_key: P9Eb6S8XNyo4dtZZUUMy
  system_secret_key: qqHCUtfdNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: 8080
  endpoint: http://rgw-primary:8080
- instance_name: 'rgw1'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonemaster: false
  rgw_zone: lyon
  rgw_zonegroup: idf
  rgw_realm: france
  rgw_zone_user: jacques.chirac
  rgw_zone_user_display_name: "Jacques Chirac"
  system_access_key: P9Eb6S8XNyo4dtZZUUMy
  system_secret_key: qqHCUtfdNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: 8081
  endpoint: http://rgw-primary:8081

```

두 사이트 모두에서 다음 단계를 실행합니다.

1.

기본 스토리지 클러스터에서 **Ansible** 플레이북을 실행합니다.

- 

베어 메탈 배포:

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

- 

컨테이너 배포:

```
[user@ansible ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```

2.

보조 스토리지 클러스터가 기본 스토리지 클러스터의 **API**에 액세스할 수 있는지 확인합니다.

보조 스토리지 클러스터의 **Object Gateway** 노드에서 **curl** 또는 다른 **HTTP** 클라이언트를 사용하여 기본 클러스터의 **API**에 연결합니다. **all.yml** 에서 **rgw\_pull\_proto**, **rgw\_pullhost**, **rgw\_pull\_port** 를 구성하는 데 사용되는 정보를 사용하여 **URL**을 작성합니다. 위의 예에 따라 **URL**은 **http://cluster0-rgw-000:8080** 입니다. **API**에 액세스할 수 없는 경우 **URL**이 올바른지 확인하고 필요한 경우 **all.yml** 을 업데이트합니다. **URL**이 작동하고 네트워크 문제가 해결되면 다음 단계를 계속하여 보조 스토리지 클러스터에서 **Ansible** 플레이북을 실행합니다.

3.

보조 스토리지 클러스터에서 **Ansible** 플레이북을 실행합니다.



참고

클러스터가 배포되고 **Ceph Object Gateway**를 변경하는 경우 **--limit rgws** 옵션을 사용합니다.

- 

베어 메탈 배포:

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

- 

컨테이너 배포:

```
[user@ansible ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```

기본 및 보조 스토리지 클러스터에서 **Ansible** 플레이북을 실행하면 **Ceph Object Gateways**가 **active-active** 상태로 실행됩니다.

4.

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

구문

```
radosgw-admin sync status
```

### 5.7.3. 여러 영역 및 여러 인스턴스를 사용하여 다중 사이트 Ceph Object Gateway 구성

**Ceph-ansible**은 여러 **Ceph Object Gateway** 인스턴스가 있는 여러 스토리지 클러스터에서 여러 영역의 데이터를 미러링하도록 **Ceph Object Gateway**를 구성합니다.



#### 주의

**Ceph-ansible**은 단일 사이트 구성에서 이미 사용된 후에는 게이트웨이를 다중 사이트 설정으로 재구성할 수 없습니다. 이 구성을 수동으로 배포할 수 있습니다. **Red Hat** 지원팀에 문의하십시오.

#### 사전 요구 사항

- **Red Hat Ceph Storage** 클러스터가 실행 중인 2개입니다.
- 각 스토리지 클러스터에서는 두 개 이상의 **Object Gateway** 노드가 있습니다.
- **Ceph Object Gateway** 노드에서 **Red Hat Ceph Storage 설치 가이드의 Red Hat Ceph Storage 설치 요구 사항** 섹션에 나열된 작업을 수행합니다.
- 각 **Object Gateway** 노드에 대해 **Red Hat Ceph Storage 설치 가이드의 Ceph Object Gateway 설치** 섹션에서 1단계 6단계를 수행합니다.

#### 절차

1. 임의의 노드에서 영역 1과 2의 시스템 액세스 키와 시크릿 키를 생성하고 **multi-site-keys-realm-1.txt** 및 **multi-site-keys-realm-2.txt** 라는 파일에 각각 저장합니다.

```
# echo system_access_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 20 | head -n 1) > multi-site-keys-realm-1.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-realm-1.txt
```

```
# echo system_access_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 20 | head -n 1) >
multi-site-keys-realm-2.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-realm-2.txt
```

### 사이트-A 스토리지 클러스터

a.

**Ansible** 구성 디렉터리로 이동합니다.

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

b.

**group\_vars/all.yml** 파일을 열고 편집합니다. **rgw\_multisite** 행의 주석을 제거하고 **true** 로 설정합니다. **rgw\_multisite\_proto** 매개변수의 주석을 제거합니다.

```
rgw_multisite: true
rgw_multisite_proto: "http"
```

c.

**/usr/share/ceph-ansible** 에 **host\_vars** 디렉토리를 만듭니다.

```
[root@ansible ceph-ansible]# mkdir host_vars
```

d.

**site-A** 스토리지 클러스터의 각 **Object Gateway** 노드에 대해 **host\_vars** 에 파일을 만듭니다. 파일 이름은 **Ansible** 인벤토리 파일에서 사용된 이름과 동일해야 합니다. 예를 들어 오브젝트 게이트웨이 노드의 이름이 **rgw-site-a** 인 경우 **host\_vars/rgw-site-a** 파일을 만듭니다.

구문

```
touch host_vars/NODE_NAME
```

예제

```
[root@ansible ceph-ansible]# touch host_vars/rgw-site-a
```



## 참고

다중 사이트 구성에 사용되는 클러스터에 여러 **Ceph Object Gateway** 노드가 있는 경우 각 노드에 대해 별도의 파일을 생성합니다.

e.

첫 번째 영역에 여러 인스턴스를 생성하려면 파일을 편집하고 각 **Object Gateway** 노드의 모든 인스턴스에 구성 세부 정보를 추가합니다. 첫 번째 영역에 대한 **rgw\_instances** 아래의 항목을 업데이트하고 함께 다음 설정을 구성합니다. **ACCESS\_KEY\_1** 및 **SECRET\_KEY\_1**의 경우 **multi-site-keys-realm-1.txt** 파일에 저장된 임의의 문자열을 사용합니다.

## 구문

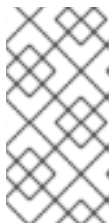
```
rgw_instances:
- instance_name: '_INSTANCE_NAME_1_'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonessecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: ZONE_NAME_1
  rgw_zonegroup: ZONE_GROUP_NAME_1
  rgw_realm: REALM_NAME_1
  rgw_zone_user: ZONE_USER_NAME_1
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
  system_access_key: ACCESS_KEY_1
  system_secret_key: SECRET_KEY_1
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
- instance_name: '_INSTANCE_NAME_2_'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonessecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: ZONE_NAME_1
  rgw_zonegroup: ZONE_GROUP_NAME_1
  rgw_realm: REALM_NAME_1
  rgw_zone_user: ZONE_USER_NAME_1
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
  system_access_key: ACCESS_KEY_1
  system_secret_key: SECRET_KEY_1
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
```

## 예제

```

rgw_instances:
  - instance_name: 'rgw0'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: paris
    rgw_zonegroup: idf
    rgw_realm: france
    rgw_zone_user: jacques.chirac
    rgw_zone_user_display_name: "Jacques Chirac"
    system_access_key: P9Eb6S8XNyo4dtZZUUMy
    system_secret_key: qqHCUtdNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: 8080
  - instance_name: 'rgw1'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: paris
    rgw_zonegroup: idf
    rgw_realm: france
    rgw_zone_user: jacques.chirac
    rgw_zone_user_display_name: "Jacques Chirac"
    system_access_key: P9Eb6S8XNyo4dtZZUUMy
    system_secret_key: qqHCUtdNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: 8080

```



## 참고

다음 단계를 건너뛰고 실행한 후 **site-B**의 모든 영역을 해당 영역에 보조로 구성한 후 **Ansible** 플레이북을 실행합니다.

f.

다른 영역에 대한 여러 인스턴스의 경우 **rgw\_instances**의 항목을 업데이트하는 것과 함께 다음 설정을 구성합니다. **ACCESS\_KEY\_2** 및 **SECRET\_KEY\_2**의 경우 **multi-site-keys-realm-2.txt** 파일에 저장된 임의의 문자열을 사용합니다.

구문



```

rgw_instances:
- instance_name: 'INSTANCE_NAME_1'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: ZONE_NAME_2
  rgw_zonegroup: ZONE_GROUP_NAME_2
  rgw_realm: REALM_NAME_2
  rgw_zone_user: ZONE_USER_NAME_2
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_2"
  system_access_key: ACCESS_KEY_2
  system_secret_key: SECRET_KEY_2
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
  endpoint:
  RGW_SITE_B_PRIMARY_HOSTNAME_ENDPOINT:RGW_SITE_B_PORT_NUMBER_1
- instance_name: 'INSTANCE_NAME_2'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: ZONE_NAME_2
  rgw_zonegroup: ZONE_GROUP_NAME_2
  rgw_realm: REALM_NAME_2
  rgw_zone_user: ZONE_USER_NAME_2
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_2"
  system_access_key: ACCESS_KEY_2
  system_secret_key: SECRET_KEY_2
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
  endpoint:
  RGW_SITE_B_PRIMARY_HOSTNAME_ENDPOINT:RGW_SITE_B_PORT_NUMBER_1

```

## 예제

```

rgw_instances:
- instance_name: 'rgw0'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: fairbanks
  rgw_zonegroup: alaska
  rgw_realm: usa
  rgw_zone_user: edward.lewis
  rgw_zone_user_display_name: "Edward Lewis"
  system_access_key: yu17wkvAx3B8Wyn08XoF
  system_secret_key: 5YZfaSUPqxSNIkZQQA3IBZ495hnlV6k2HAz710BY

```

```
radosgw_address: "{{ _radosgw_address }}"
radosgw_frontend_port: 8080
endpoint: http://rgw-site-b:8081
- instance_name: 'rgw1'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: fairbanks
  rgw_zonegroup: alaska
  rgw_realm: usa
  rgw_zone_user: edward.lewis
  rgw_zone_user_display_name: "Edward Lewis"
  system_access_key: yu17wkvAx3B8Wyn08XoF
  system_secret_key: 5YZfaSUPqxSNlkZQQA3IBZ495hnlV6k2HAz710BY
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: 8081
  endpoint: http://rgw-site-b:8081
```

g. **site-A** 스토리지 클러스터에서 **Ansible** 플레이북을 실행합니다.

- 베어 메탈 배포:

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

- 컨테이너 배포:

```
[user@ansible ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```

**사이트-B** 스토리지 클러스터

a. **Ceph-ansible** 구성 디렉터리로 이동합니다.

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

b. **group\_vars/all.yml** 파일을 열고 편집합니다. **rgw\_multisite** 행의 주석을 제거하고 **true** 로 설정합니다. **rgw\_multisite\_proto** 매개변수의 주석을 제거합니다.

```
rgw_multisite: true
rgw_multisite_proto: "http"
```

c.

`/usr/share/ceph-ansible` 에 `host_vars` 디렉토리를 만듭니다.

```
[root@ansible ceph-ansible]# mkdir host_vars
```

d.

**site-B** 스토리지 클러스터에서 각 **Object Gateway** 노드에 대해 `host_vars` 에 파일을 만듭니다. 파일 이름은 **Ansible** 인벤토리 파일에서 사용된 이름과 동일해야 합니다. 예를 들어 오브젝트 게이트웨이 노드의 이름이 `rgw-site-b` 인 경우 `host_vars/rgw-site-b` 파일을 만듭니다.

구문

```
touch host_vars/NODE_NAME
```

예제

```
[root@ansible ceph-ansible]# touch host_vars/rgw-site-b
```



참고

다중 사이트 구성에 사용되는 클러스터에 여러 **Ceph Object Gateway** 노드가 있는 경우 각 노드에 대해 파일을 생성합니다.

e.

첫 번째 영역에 여러 인스턴스를 생성하려면 파일을 편집하고 각 **Object Gateway** 노드의 모든 인스턴스에 구성 세부 정보를 추가합니다. 첫 번째 영역에 대한 `rgw_instances` 아래의 항목을 업데이트하고 함께 다음 설정을 구성합니다. `ACCESS_KEY_1` 및 `SECRET_KEY_1`의 경우 `multi-site-keys-realm-1.txt` 파일에 저장된 임의의 문자열을 사용합니다. `RGW_SITE_A_PRIECDHE_HOSTNAME_ENDPOINT`를 **site-A** 스토리지 클러스터의 **Object Gateway** 노드로 설정합니다.

구문

```

rgw_instances:
- instance_name: 'INSTANCE_NAME_1'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: ZONE_NAME_1
  rgw_zonegroup: ZONE_GROUP_NAME_1
  rgw_realm: REALM_NAME_1
  rgw_zone_user: ZONE_USER_NAME_1
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
  system_access_key: ACCESS_KEY_1
  system_secret_key: SECRET_KEY_1
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
  endpoint: RGW_SITE_A_HOSTNAME_ENDPOINT:RGW_SITE_A_PORT_NUMBER_1
- instance_name: '_INSTANCE_NAME_2_'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: ZONE_NAME_1
  rgw_zonegroup: ZONE_GROUP_NAME_1
  rgw_realm: REALM_NAME_1
  rgw_zone_user: ZONE_USER_NAME_1
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
  system_access_key: ACCESS_KEY_1
  system_secret_key: SECRET_KEY_1
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
  endpoint:
    RGW_SITE_A_PRIMARY_HOSTNAME_ENDPOINT:RGW_SITE_A_PORT_NUMBER_1

```

## 예제

```

rgw_instances:
- instance_name: 'rgw0'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: paris
  rgw_zonegroup: idf
  rgw_realm: france
  rgw_zone_user: jacques.chirac
  rgw_zone_user_display_name: "Jacques Chirac"
  system_access_key: P9Eb6S8XNyo4dtZZUUMy
  system_secret_key: qqHCUtdNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
  radosgw_address: "{{ _radosgw_address }}"

```

```

radosgw_frontend_port: 8080
endpoint: http://rgw-site-a:8080
- instance_name: 'rgw1'
  rgw_multisite: true
  rgw_zonemaster: false
  rgw_zonesecondary: true
  rgw_zonegroupmaster: false
  rgw_zone: paris
  rgw_zonegroup: idf
  rgw_realm: france
  rgw_zone_user: jacques.chirac
  rgw_zone_user_display_name: "Jacques Chirac"
  system_access_key: P9Eb6S8XNyo4dtZZUUMy
  system_secret_key: qqHCUtdNnpHq3PZRHW5un9I0bEBM812Uhow0XfB
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: 8081
  endpoint: http://rgw-site-a:8081

```

f.

다른 영역에 대한 여러 인스턴스의 경우 **rgw\_instances**의 항목을 업데이트하는 것과 함께 다음 설정을 구성합니다. **ACCESS\_KEY\_2** 및 **SECRET\_KEY\_2**의 경우 **multi-site-keys-realm-2.txt** 파일에 저장된 임의의 문자열을 사용합니다.

**RGW\_SITE\_A\_PRIECDHE\_HOSTNAME\_ENDPOINT**를 **site-A** 스토리지 클러스터의 **Object Gateway** 노드로 설정합니다.

구문

```

rgw_instances:
- instance_name: 'INSTANCE_NAME_1'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonesecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: ZONE_NAME_2
  rgw_zonegroup: ZONE_GROUP_NAME_2
  rgw_realm: REALM_NAME_2
  rgw_zone_user: ZONE_USER_NAME_2
  rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_2"
  system_access_key: ACCESS_KEY_2
  system_secret_key: SECRET_KEY_2
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: PORT_NUMBER_1
- instance_name: '_INSTANCE_NAME_2_'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonesecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: ZONE_NAME_2

```

```

rgw_zonegroup: ZONE_GROUP_NAME_2
rgw_realm: REALM_NAME_2
rgw_zone_user: ZONE_USER_NAME_2
rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_2"
system_access_key: ACCESS_KEY_2
system_secret_key: SECRET_KEY_2
radosgw_address: "{{ _radosgw_address }}"
radosgw_frontend_port: PORT_NUMBER_1

```

## 예제

```

rgw_instances:
- instance_name: 'rgw0'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonesecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: fairbanks
  rgw_zonegroup: alaska
  rgw_realm: usa
  rgw_zone_user: edward.lewis
  rgw_zone_user_display_name: "Edward Lewis"
  system_access_key: yu17wkvAx3B8Wyn08XoF
  system_secret_key: 5YZfaSUPqxSNIkZQQA3IBZ495hnlV6k2HAz710BY
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: 8080
- instance_name: 'rgw1'
  rgw_multisite: true
  rgw_zonemaster: true
  rgw_zonesecondary: false
  rgw_zonegroupmaster: true
  rgw_zone: fairbanks
  rgw_zonegroup: alaska
  rgw_realm: usa
  rgw_zone_user: edward.lewis
  rgw_zone_user_display_name: "Edward Lewis"
  system_access_key: yu17wkvAx3B8Wyn08XoF
  system_secret_key: 5YZfaSUPqxSNIkZQQA3IBZ495hnlV6k2HAz710BY
  radosgw_address: "{{ _radosgw_address }}"
  radosgw_frontend_port: 8081

```

g.

**site-B** 스토리지 클러스터에서 **Ansible** 플레이북을 실행합니다.

- **베어 메탈 배포:**

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

- **컨테이너 배포:**

```
[user@ansible ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```

**site-A** 의 다른 영역에 대해 **site-A** 스토리지 클러스터에서 **Ansible** 플레이북을 다시 실행합니다.

**site-A** 및 **site-B** 스토리지 클러스터에서 **Ansible** 플레이북을 실행하면 **Ceph Object Gateway**가 **active-active** 상태로 실행됩니다.

## 검증

1. **다중 사이트 Ceph Object Gateway** 구성을 확인합니다.
  - a. 각 사이트의 **Ceph Monitor** 및 **Object Gateway** 노드에서 **site-A** 및 **site-B**를 사용하여 **curl** 또는 다른 **HTTP** 클라이언트를 사용하여 **API**에 다른 사이트에서 액세스할 수 있는지 확인합니다.
  - b. 두 사이트 모두에서 **radosgw-admin sync status** 명령을 실행합니다.

## 구문

```
radosgw-admin sync status
radosgw-admin sync status --rgw -realm REALM_NAME 1
```

1

스토리지 클러스터의 각 노드에서 여러 영역에 이 옵션을 사용합니다.

## 예제

```
[user@ansible ceph-ansible]$ radosgw-admin sync status
```

```
[user@ansible ceph-ansible]$ radosgw-admin sync status --rgw -realm usa
```

## 5.8. 동일한 호스트에서 다른 하드웨어를 사용하여 OSD 배포

**Ansible의 device\_class** 기능을 사용하여 동일한 호스트에 **HD** 및 **SSD**와 같은 혼합 **OSD**를 배포할 수 있습니다.

## 사전 요구 사항

- 유효한 고객 서브스크립션.
- **Ansible** 관리 노드에 대한 루트 수준 액세스.
- **Red Hat Ceph Storage Tools** 및 **Ansible** 리포지토리를 활성화합니다.
- **Ansible** 애플리케이션과 함께 사용할 **ansible** 사용자 계정입니다.
- **OSD**가 배포됩니다.

## 절차

1. **group\_vars/mons.yml** 파일에 **crush\_rules** 를 만듭니다.

## 예제

```
crush_rule_config: true
crush_rule_hdd:
```



```

name: HDD
root: default
type: host
class: hdd
default: true
crush_rule_ssd:
name: SSD
root: default
type: host
class: ssd
default: true
crush_rules:
- "{{ crush_rule_hdd }}"
- "{{ crush_rule_ssd }}"
create_crush_tree: true

```



### 참고

클러스터에서 **SSD** 또는 **HD** 장치를 사용하지 않는 경우 해당 장치에 대한 **crush\_rules** 를 정의하지 마십시오.

2.

**group\_vars/clients.yml** 파일에 **crush\_rules** 를 생성한 상태에서 풀 을 생성합니다.

### 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 }}"

```

3.

인벤토리 파일을 샘플링하여 **OSD**에 루트를 할당합니다.

## 예제

```
[mons]
mon1

[osds]
osd1 osd_crush_location="{ 'root': 'default', 'rack': 'rack1', 'host': 'osd1' }"
osd2 osd_crush_location="{ 'root': 'default', 'rack': 'rack1', 'host': 'osd2' }"
osd3 osd_crush_location="{ 'root': 'default', 'rack': 'rack2', 'host': 'osd3' }"
osd4 osd_crush_location="{ 'root': 'default', 'rack': 'rack2', 'host': 'osd4' }"
osd5 devices=["/dev/sda", "/dev/sdb"] osd_crush_location="{ 'root': 'default', 'rack': 'rack3',
'host': 'osd5' }"
osd6 devices=["/dev/sda", "/dev/sdb"] osd_crush_location="{ 'root': 'default', 'rack': 'rack3',
'host': 'osd6' }"

[mgrs]
mgr1

[clients]
client1
```

4.

트리를 봅니다.

## 구문

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

## 예제

```
TYPE NAME
```

```
root default
  rack rack1
    host osd1
      osd.0
      osd.10
    host osd2
```

```

    osd.3
    osd.7
    osd.12
  rack rack2
    host osd3
    osd.1
    osd.6
    osd.11
  host osd4
    osd.4
    osd.9
    osd.13
  rack rack3
    host osd5
    osd.2
    osd.8
  host osd6
    osd.14
    osd.15

```

5.

풀을 검증합니다.

예제

```

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

pool: pool1
crush_rule: HDD

```

추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage 설치 가이드](#)에서 [Red Hat Ceph Storage](#) 클러스터 설치를 참조하십시오.
- 자세한 내용은 [Red Hat Ceph Storage Strategies Guide](#)의 [장치 클래스](#)를 참조하십시오.

## 5.9. NFS-GANESHA 게이트웨이 설치

**Ceph NFS Ganesha** 게이트웨이는 **Ceph** 개체 게이트웨이 상단에 구축된 **NFS** 인터페이스로, 파일 시스템 내의 파일을 **Ceph Object Storage**로 마이그레이션하기 위해 **Ceph Object Gateway**에 **POSIX** 파일 시스템 인터페이스를 제공합니다.

#### 사전 요구 사항

- **활성 + 클린 상태의 실행 중인 Ceph 스토리지 클러스터.**
- **Ceph Object Gateway를 실행하는 하나 이상의 노드입니다.**
- **NFS-Ganesha를 실행하기 전에 NFS-Ganesha를 실행할 호스트에서 실행 중인 커널 NFS 서비스 인스턴스를 비활성화합니다. 다른 NFS 인스턴스가 실행 중인 경우 NFS-Ganesha가 시작되지 않습니다.**

**\*암호 없는 SSH 액세스를 활성화합니다.**

- **NetNamespacebind 서비스가 실행 중인지 확인합니다.**

```
# systemctl start rpcbind
```



참고

**NetNamespacebind**를 제공하는 **NetNamespacebind** 패키지는 일반적으로 기본적으로 설치됩니다. 그렇지 않은 경우 먼저 패키지를 설치합니다.

- **nfs-service** 서비스가 실행 중인 경우 이를 중지하고 비활성화합니다.

```
# systemctl stop nfs-server.service
# systemctl disable nfs-server.service
```

#### 절차

**Ansible** 관리 노드에서 다음 작업을 수행합니다.

1. 샘플 파일에서 **nfss.yml** 파일을 생성합니다.

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

2.

**[nfss]** 그룹의 `/etc/ansible/hosts` 파일에 게이트웨이 호스트를 추가하여 **Ansible**에 대한 그룹 멤버십을 식별합니다.

```
[nfss]
NFS_HOST_NAME_1
NFS_HOST_NAME_2
NFS_HOST_NAME[3..10]
```

호스트에 순차적인 이름 지정자가 있는 경우 범위 지정자(예: `[3..10]`)를 사용할 수 있습니다.

3.

**Ansible** 구성 디렉터리로 이동합니다.

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

4.

관리자 키를 **Ceph Object Gateway** 노드에 복사하려면 `/usr/share/ceph-ansible/group_vars/nfss.yml` 파일에서 `copy_admin_key` 설정의 주석을 제거하십시오.

```
copy_admin_key: true
```

5.

`/usr/share/ceph-ansible/group_vars/nfss.yml` 파일의 **FSAL(File System Abstraction Layer)** 섹션을 구성합니다. 내보내기 ID(`NUMERIC_EXPORT_ID`), S3 사용자 ID(`S3_USER`), S3 액세스 키(`ACCESS_KEY`) 및 시크릿 키(`SECRET_KEY`)를 제공합니다.

```
# FSAL RGW Config #

ceph_nfs_rgw_export_id: NUMERIC_EXPORT_ID
#ceph_nfs_rgw_pseudo_path: "/"
#ceph_nfs_rgw_protocols: "3,4"
#ceph_nfs_rgw_access_type: "RW"
ceph_nfs_rgw_user: "S3_USER"
ceph_nfs_rgw_access_key: "ACCESS_KEY"
ceph_nfs_rgw_secret_key: "SECRET_KEY"
```



주의

액세스 및 시크릿 키는 선택 사항이며 생성할 수 있습니다.

6.

**Ansible Playbook**을 실행합니다.

a.

베어 메탈 배포:

```
[ansible@admin ceph-ansible]$ ansible-playbook site.yml --limit nfss -i hosts
```

b.

컨테이너 배포:

```
[ansible@admin ceph-ansible]$ ansible-playbook site-container.yml --limit nfss -i hosts
```

추가 리소스

- [제한 옵션 이해](#)
- [오브젝트 게이트웨이 구성 및 관리 가이드](#)

### 5.10. 제한 옵션 이해

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

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

```
ansible-playbook site.yml|site-container.yml --limit osds|rgws|clients|mdss|nfss|iscsigws -i hosts
```

베어 메탈

예를 들어 베어 메탈에서 OSD만 재배포하려면 Ansible 사용자로 다음 명령을 실행합니다.

```
[ansible@ansible ceph-ansible]$ ansible-playbook site.yml --limit osds -i hosts
```

컨테이너

예를 들어 컨테이너에 OSD만 재배포하려면 Ansible 사용자로 다음 명령을 실행합니다.

```
[ansible@ansible ceph-ansible]$ ansible-playbook site-container.yml --limit osds -i hosts
```

### 5.11. 배치 그룹 자동 스케일러

PG(배치 그룹) 튜닝은 PG 계산기를 사용하여 `pg_num` 의 숫자 연결 수를 수동으로 연결하는 데 사용됩니다. Red Hat Ceph Storage 4.1부터는 `pg_autoscaler` Ceph 관리자 모듈을 활성화하면 PG 튜닝을 자동으로 수행할 수 있습니다. PG 자동 스케일러는 풀 단위로 구성되며 `pg_num` 은 2의 힘으로 확장됩니다. PG 자동 스케일러는 제안된 값이 실제 값의 3배 이상인 경우에만 `pg_num` 에 대한 변경을 제안합니다.

PG 자동 스케일러에는 세 가지 모드가 있습니다.

**warn**

새 풀과 기존 풀의 기본 모드입니다. 제안된 `pg_num` 값이 현재 `pg_num` 값과 너무 많이 다른 경우 상태 경고가 생성됩니다.

**on**

풀의 `pg_num` 이 자동으로 조정됩니다.

**off**

모든 풀에 대해 자동 스케일러를 해제할 수 있지만 스토리지 관리자는 풀의 `pg_num` 값을 수동으로 설정해야 합니다.

풀에 있는 PG 자동 스케일러를 활성화하면 `ceph osd 풀 autoscale-status` 명령을 실행하여 값 조정을 볼 수 있습니다. `autoscale-status` 명령은 풀의 현재 상태를 표시합니다. `autoscale-status` 열 설명은 다음과 같습니다.

**SIZE**

풀에 저장된 총 데이터 양(바이트)을 보고합니다. 이 크기에는 오브젝트 데이터 및 OMAP 데이터가 포함됩니다.

**TARGET SIZE**

스토리지 관리자가 제공하는 대로 예상 풀 크기를 보고합니다. 이 값은 풀의 이상적인 **PG** 수를 계산하는 데 사용됩니다.

**RATE**

복제 버킷에 대한 복제 요소 또는 삭제 코드된 풀의 비율입니다.

**RAW CAPACITY**

*pool*이 *mapped*하는 스토리지 장치의 원시 스토리지 용량은 **ArgoCD**에 따라 다릅니다.

**RATIO**

풀에서 소비되는 총 스토리지의 비율입니다.

## 대상 비율

스토리지 관리자가 제공하는 대로 풀에서 사용하는 총 스토리지 클러스터 공간의 일부를 지정하는 비율입니다.

**PG\_NUM**

풀의 현재 배치 그룹 수입니다.

**NEW PG\_NUM**

제안된 값입니다. 이 값은 설정되지 않을 수 있습니다.

**AUTOSCALE**

풀에 설정된 **PG** 자동 스케일러 모드입니다.

## 추가 리소스

- [배치 그룹 풀 계산기](#).

**5.11.1. 배치 그룹 자동 스케일러 구성**

**Red Hat Ceph Storage** 클러스터에서 새 풀에 대한 **PG** 자동 스케일러를 활성화하고 구성하도록 **Ceph Ansible**을 구성할 수 있습니다. 기본적으로 배치 그룹(**PG**) 자동 스케일러는 해제되어 있습니다.





## 중요

현재는 기존 **Red Hat Ceph Storage** 설치가 아닌 새로운 **Red Hat Ceph Storage** 배포에서 배치 그룹 자동 스케일러를 설정할 수 있습니다.

### 사전 요구 사항

- **Ansible** 관리 노드에 액세스
- **Ceph** 모니터 노드에 액세스합니다.

### 절차

1. **Ansible** 관리 노드에서 편집할 `group_vars/all.yml` 파일을 엽니다.
2. `pg_autoscale_mode` 옵션을 `True` 로 설정하고 새 또는 기존 풀의 `target_size_ratio` 값을 설정합니다.

### 예제

```

openstack_pools:
  - {"name": backups, "target_size_ratio": 0.1, "pg_autoscale_mode": True, "application": rbd}
  - {"name": volumes, "target_size_ratio": 0.5, "pg_autoscale_mode": True, "application": rbd}
  - {"name": vms, "target_size_ratio": 0.2, "pg_autoscale_mode": True, "application": rbd}
  - {"name": images, "target_size_ratio": 0.2, "pg_autoscale_mode": True, "application": rbd}

```



## 참고

`target_size_ratio` 값은 스토리지 클러스터의 다른 풀과 관련된 `weight` 백분율입니다.

3. `group_vars/all.yml` 파일에 변경 사항을 저장합니다.

4. 적절한 **Ansible** 플레이북을 실행합니다.

베어 메탈 배포

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

컨테이너 배포

```
[ansible@admin ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```

5. **Ansible** 플레이북이 완료되면 **Ceph Monitor** 노드에서 자동 스케일러 상태를 확인합니다.

```
[user@mon ~]$ ceph osd pool autoscale-status
```

## 5.12. 추가 리소스

- [Ansible 문서](#)

## 6장. 컨테이너화된 CEPH 데몬 배치

이 섹션에서는 다음을 설명합니다.

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

### 6.1. 코로케이션 작동 방식 및 이점

컨테이너화된 Ceph 데몬을 동일한 노드에서 배치할 수 있습니다. 다음은 일부 Ceph 서비스를 배치하면 다음과 같은 이점이 있습니다.

- 작은 규모로 TCO(total cost of ownership)가 크게 개선되었습니다.
- 최소 구성을 위해 6개의 노드에서 3개의 노드로 줄입니다.
- 더 쉽게 업그레이드할 수 있습니다.
- 리소스 격리 개선.

Knowledgebase 문서 [Red Hat Ceph Storage](#)를 참조하십시오. [Red Hat Ceph Storage](#) 클러스터의 데몬 배치에 대한 자세한 내용은 지원되는 [Configurations](#) 참조하십시오.

#### 코로케이션 작동 방식

Ansible 인벤토리 파일의 적절한 섹션에 동일한 노드를 추가하여 다음 목록에서 하나의 데몬을 OSD 데몬(ceph-osd)과 함께 배치할 수 있습니다.

- **Ceph 메타 데이터 서버 (ceph-mds)**
- **Ceph Monitor(ceph-mon) 및 Ceph Manager(ceph-mgr) 데몬**

- **NFS Ganesha(nfs-ganesha)**
- **RBD 미러 (rbd-mirror)**
- **iSCSI 게이트웨이(iscsigw)**

Red Hat Ceph Storage 4.2부터 메타데이터 서버(MDS)는 하나의 추가 스케일 아웃 데몬과 함께 배치할 수 있습니다.

또한 Ceph Object Gateway(radosgw) 또는 Grafana의 경우 RBD mirror.z를 제외하고 OSD 데몬과 위의 목록의 데몬과 함께 배치할 수 있습니다. 예를 들어 다음 노드는 유효한 5개의 노드 공동 배치 구성입니다.

노드	데몬	데몬	데몬
node1	OSD	모니터	Grafana
node2	OSD	모니터	RADOS 게이트웨이
node3	OSD	모니터	RADOS 게이트웨이
node4	OSD	메타 데이터 서버	
node5	OSD	메타 데이터 서버	

위 설정과 같이 5개의 노드 클러스터를 배포하려면 다음과 같이 **Ansible** 인벤토리 파일을 구성합니다.

공동 배치 데몬을 사용하는 **Ansible** 인벤토리 파일

```
[grafana-server]
node1

[mons]
node[1:3]

[mgrs]
node[1:3]
```

```
[osds]
node[1:5]
```

```
[rgws]
node[2:3]
```

```
[mdss]
node[4:5]
```



참고

**ceph-mon** 및 **ceph-mgr** 은 코로케이션을 위해 별도의 데몬 2개로 계산되지 않습니다.



참고

다른 데몬과 함께 **Grafana**를 배치하는 것은 **Cockpit** 기반 설치에서는 지원되지 않습니다. **ceph-ansible** 을 사용하여 스토리지 클러스터를 구성합니다.



참고

성능을 높이기 위해 **Ceph Object Gateway**를 OSD 컨테이너로 배치하는 것이 좋습니다. 추가 비용 없이 최고 성능을 달성하려면 **radosgw\_num\_instances**를 설정하여 두 개의 게이트웨이를 사용합니다. **2 group\_vars/all.yml**. 자세한 내용은 [Red Hat Ceph Storage RGW 배포 전략 및 크기 조정 지침](#)을 참조하십시오.

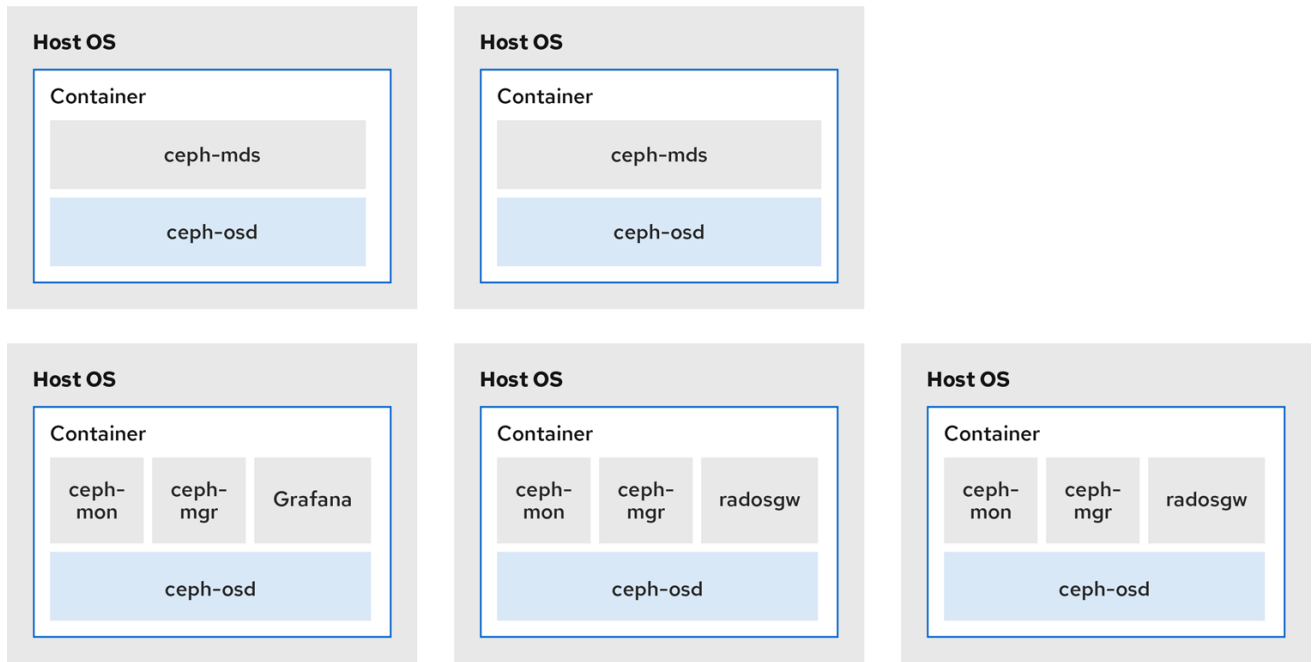


참고

**Grafana**를 다른 두 컨테이너와 함께 배치하려면 적절한 **CPU** 및 네트워크 리소스가 필요합니다. 리소스 소모가 발생하면 모니터만 **Grafana**를 공동 배치하고 리소스 소모가 계속 발생하는 경우 전용 노드에서 **Grafana**를 실행합니다.

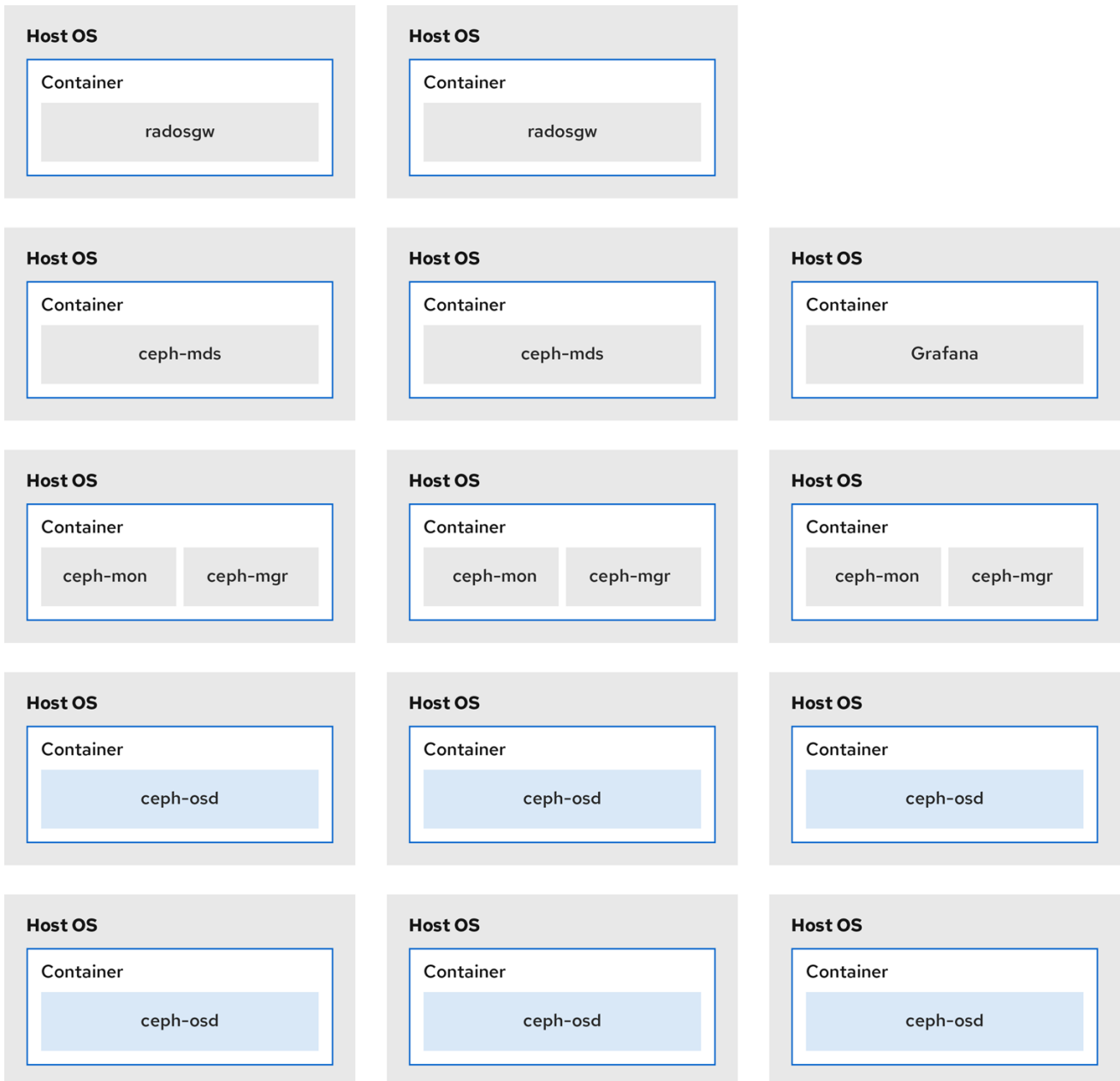
그림 6.1. “공동 배치된 데몬” 및 그림 6.2. “공동배치되지 않은 데몬” 이미지는 공동 배치된 데몬과 함께 클러스터의 차이점을 보여줍니다.

그림 6.1. 공동 배치된 데몬



108\_Ceph\_0720

그림 6.2. 공동배치되지 않은 데몬



108\_Ceph\_0720

동일한 노드에서 컨테이너화된 여러 **Ceph** 데몬을 함께 배치하면 **ceph-ansible Playbook**은 전용 **CPU** 및 **RAM** 리소스를 각 노드에 예약합니다. 기본적으로 **ceph-ansible**은 **Red Hat Ceph Storage Hardware Guide**의 권장 최소 하드웨어 장에 나열된 값을 사용합니다. 기본값을 변경하는 방법을 알아보려면 배치된 데몬에 대한 전용 리소스 설정 섹션을 참조하십시오.

## 6.2. 배치된 데몬을 위한 전용 리소스 설정

동일한 노드에서 두 개의 **Ceph** 데몬을 일치시키면 **ceph-ansible Playbook**은 각 데몬에 대해 **CPU** 및 **RAM** 리소스를 예약합니다. **ceph-ansible**에서 사용하는 기본값은 **Red Hat Ceph Storage Hardware Selection Guide**의 **Recommended Minimum Hardware** (최소 하드웨어 선택) 장에 나열되어 있습니다. 기본값을 변경하려면 **Ceph** 데몬을 배포할 때 필요한 매개변수를 설정합니다.

절차

1.

데몬의 기본 CPU 제한을 변경하려면 데몬을 배포할 때 적절한 `.yaml` 구성 파일에서 `ceph_daemon-type_docker_cpu_limit` 매개변수를 설정합니다. 자세한 내용은 다음 표를 참조하십시오.

데몬	매개변수	구성 파일
OSD	<code>ceph_osd_docker_cpu_limit</code>	<code>osds.yaml</code>
MDS	<code>ceph_mds_docker_cpu_limit</code>	<code>mdss.yaml</code>
RGW	<code>ceph_rgw_docker_cpu_limit</code>	<code>rgws.yaml</code>

예를 들어 Ceph 개체 게이트웨이의 기본 CPU 제한을 2로 변경하려면 다음과 같이 `/usr/share/ceph-ansible/group_vars/rgws.yaml` 파일을 편집합니다.

```
ceph_rgw_docker_cpu_limit: 2
```

2.

OSD 데몬의 기본 RAM을 변경하려면 데몬을 배포할 때 `/usr/share/ceph-ansible/group_vars/all.yaml` 파일에서 `osd_memory_target` 을 설정합니다. 예를 들어 OSD RAM을 6GB로 제한하려면 다음을 수행합니다.

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



중요

하이퍼컨버지드 인프라(HCI) 구성에서 `osds.yaml` 구성 파일에서 `ceph_osd_docker_memory_limit` 매개변수를 사용하여 Docker 메모리 CGroup 제한을 변경할 수도 있습니다. 이 경우 `ceph_osd_docker_memory_limit` 를 `osd_memory_target` 보다 50% 더 높으므로 CGroup 제한이 HCI 구성에 대해 기본적으로 설정된 것보다 더 많은 `constraining`이 되도록 합니다. 예를 들어 `osd_memory_target` 이 6GB로 설정된 경우 `ceph_osd_docker_memory_limit` 를 9GB로 설정합니다.

```
ceph_osd_docker_memory_limit: 9g
```

추가 리소스



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

### 6.3. 추가 리소스

- [Red Hat Ceph Storage Hardware selection 가이드](#)

## 7장. RED HAT CEPH STORAGE 클러스터 업그레이드

스토리지 관리자는 **Red Hat Ceph Storage** 클러스터를 새 주요 버전 또는 새 마이너 버전으로 업그레이드하거나 비동기 업데이트를 현재 버전에만 적용할 수 있습니다. **rolling\_update.yml Ansible** 플레이북은 **Red Hat Ceph Storage**의 베어 메탈 또는 컨테이너화된 배포를 위해 업그레이드를 수행합니다. **Ansible**은 **Ceph** 노드를 다음 순서로 업그레이드합니다.

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

### 참고

**Red Hat Ceph Storage 3.1**부터 **Object Gateway** 및 고속 **NVMe** 기반 **SSD**(및 **SATA SSD**)를 사용할 때 성능을 위해 스토리지를 최적화하기 위해 새로운 **Ansible** 플레이북이 추가되었습니다. 플레이북은 저널과 버킷 인덱스를 **SSD**에 함께 배치하여 이 작업을 수행합니다. 이렇게 하면 한 장치에 모든 저널을 사용하는 것보다 성능이 향상됩니다. 이러한 플레이북은 **Ceph**를 설치할 때 사용하도록 설계되었습니다. 기존 **OSD**는 계속 작동하며 업그레이드하는 동안 추가 단계가 필요하지 않습니다. 이러한 방식으로 스토리지를 최적화하기 위해 **OSD**를 동시에 재구성하는 동안 **Ceph** 클러스터를 업그레이드할 수 없습니다. 저널 또는 버킷 인덱스에 다른 장치를 사용하려면 **OSD**를 다시 프로비저닝해야 합니다. 자세한 내용은 프로덕션 가이드의 **Ceph Object Gateway**에서 **LVM**에서 최적의 **NVMe** 사용을 참조하십시오.

**중요**

이전 지원 버전에서 Red Hat Ceph Storage 클러스터를 버전 4.2z2로 업그레이드할 때 모니터에서 비보안 `global_id` 회수를 허용하는 `HEALTH_WARN` 상태의 스토리지 클러스터에서 업그레이드가 완료됩니다. 이는 [CVE-2021-20288](#) 에서 사용할 수 있는 패치된 CVE 때문입니다. 이 문제는 Red Hat Ceph Storage 4.2z2의 CVE에서 해결되었습니다.

상태 경고를 음소거하는 권장 사항:

1. **`AUTH_INSECURE_GLOBAL_ID_RECLAIM`** 경고에 대한 `ceph` 상태 세부 정보 출력을 확인하여 업데이트되지 않는 클라이언트를 식별합니다.
2. 모든 클라이언트를 Red Hat Ceph Storage 4.2z2 릴리스로 업그레이드합니다.
3. 모든 클라이언트를 확인하고 **`AUTH_INSECURE_GLOBAL_ID_RECLAIM`** 경고가 더 이상 클라이언트에 표시되지 않으면 **`auth_allow_insecure_global_id_reclaim`** 을 `false` 로 설정합니다. 이 옵션을 `false` 로 설정하면 간헐적인 네트워크 중단이 모니터에 대한 연결을 중단한 후 패치되지 않은 클라이언트가 스토리지 클러스터에 다시 연결할 수 없거나 시간 초과 시 인증 티켓을 갱신할 수 없습니다. 기본값은 72시간입니다.

**구문**

```
ceph config set mon auth_allow_insecure_global_id_reclaim false
```

4. **`AUTH_INSECURE_GLOBAL_ID_RECLAIM`** 경고가 나열된 클라이언트가 없는지 확인합니다.

**중요**

`rolling_update.yml` 플레이북에는 동시에 업데이트할 노드 수를 조정하는 직렬 변수가 포함되어 있습니다. Red Hat은 Ansible이 클러스터 노드를 하나씩 업그레이드하도록 하는 기본값(1)을 사용하는 것이 좋습니다.



## 중요

언제든지 업그레이드가 실패하면 **ceph status** 명령을 사용하여 클러스터 상태를 확인하여 업그레이드 실패 원인을 파악합니다. 실패 이유와 해결 방법을 확실하지 않은 경우 [Red hat 지원팀에 문의](#)하십시오.



## 주의

**Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 다중 사이트 설정을 업그레이드하는 경우 다음 권장 사항 또는 기타 복제가 중단될 수 있습니다. **rolling\_update.yml** 을 실행하기 전에 **all.yml** 에서 **rgw\_multisite: false** 를 설정합니다. 업그레이드 후 **rgw\_multisite** 를 다시 활성화하지 마십시오. 업그레이드 후 새 게이트웨이를 추가해야 하는 경우에만 사용합니다. 버전 **3.3z5** 이상에서 **Red Hat Ceph Storage 3** 클러스터만 **Red Hat Ceph Storage 4**로 업그레이드합니다. **3.3z5** 이상으로 업데이트할 수 없는 경우 클러스터를 업그레이드하기 전에 사이트 간 동기화를 비활성화합니다. 동기화를 비활성화하려면 **rgw\_run\_sync\_thread = false** 를 설정하고 **RADOS Gateway** 데몬을 다시 시작합니다. 먼저 기본 클러스터를 업그레이드합니다. **Red Hat Ceph Storage 4.1** 이상으로 업그레이드 **3.3z5**와 관련이 있는 패키지 버전을 보려면 **Red Hat Ceph Storage 릴리스 및 해당 Ceph 패키지 버전은 무엇입니까?** 동기화를 비활성화하는 방법에 대한 자세한 내용은 **RGW 멀티 사이트 동기화**를 일시적으로 비활성화하는 방법을 참조하십시오.



## 주의

**Ceph Object Gateway**를 사용하고 **Red Hat Ceph Storage 3.x**에서 **Red Hat Ceph Storage 4.x**로 업그레이드하는 경우 프론트 엔드는 자동으로 **CivetWeb**에서 **Beast**로 변경됩니다. 이는 새 기본값입니다. 자세한 내용은 **오브젝트 게이트웨이 구성 및 관리 가이드의 구성**을 참조하십시오.



## 주의

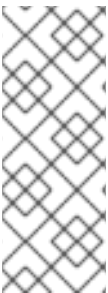
**RADOS 게이트웨이**를 사용하는 경우 **Ansible**은 프론트 엔드를 **CivetWeb**에서 **Beast**로 전환합니다. 이 과정에서 **RGW** 인스턴스 이름이 **rgw.HOSTNAME**에서 **rgw.HOSTNAME.rgw0**으로 변경됩니다. 이름 변경 **Ansible**은 **ceph.conf**의 기존 **RGW** 구성을 업데이트하지 않고 기본 구성을 추가하지 않고 기존 **CivetWeb** 기반 **RGW** 설정을 그대로 유지하지만 사용되지는 않습니다. 그런 다음 사용자 지정 **RGW** 구성 변경 사항이 손실되어 **RGW** 서비스가 중단될 수 있습니다. 업그레이드 전에 기존 **RGW** 구성을 **all.yml**의 **ceph\_conf\_overrides** 섹션에 추가하되 **.rgw0**을 추가하여 **RGW** 인스턴스 이름을 변경한 다음 **RGW** 서비스를 다시 시작합니다. 이렇게 하면 업그레이드 후 기본이 아닌 **RGW** 구성 변경 사항이 유지됩니다. **ceph\_conf\_overrides**에 대한 자세한 내용은 [Ceph 기본 설정](#) 닷어쓰기를 참조하십시오.

## 7.1. 지원되는 RED HAT CEPH STORAGE 업그레이드 시나리오

**Red Hat**은 다음과 같은 업그레이드 시나리오를 지원합니다.

베어 메탈 및 컨테이너화된 테이블을 읽고 특정 업그레이드 후 상태로 이동하기 위해 클러스터의 사전 업그레이드 상태를 확인합니다.

베어 메탈 또는 호스트 운영 체제가 주요 버전을 변경하지 않는 베어 메탈 및 컨테이너화된 업그레이드를 수행하려면 **ceph-ansible**을 사용합니다. **Red Hat Enterprise Linux 7**에서 **Red Hat Enterprise Linux 8**으로의 업그레이드는 **ceph-ansible**에서 지원되지 않습니다. **Red Hat Ceph Storage** 업그레이드의 일부로 베어 메탈 운영 체제를 **Red Hat Enterprise Linux 7.9**에서 **Red Hat Enterprise Linux 8.4**로 업그레이드하려면 **Red Hat Ceph Storage** 설치 가이드의 [Red Hat Ceph Storage 클러스터 및 운영 체제](#) 섹션을 수동으로 업그레이드하십시오.



## 참고

클러스터를 **Red Hat Ceph Storage 4**로 업그레이드하려면 **Red Hat Ceph Storage 3**의 최신 버전을 사용할 것을 권장합니다. 최신 **Red Hat Ceph Storage** 버전을 알아보려면 [Red Hat Ceph Storage 릴리스의 내용](#)을 참조하십시오. 자세한 내용은 지식베이스 문서입니다.

표 7.1. 베어 메탈 배포에 지원되는 업그레이드 시나리오

pre-upgrade 상태		post-upgrade 상태	
Red Hat Enterprise Linux 버전	Red Hat Ceph Storage 버전	Red Hat Enterprise Linux 버전	Red Hat Ceph Storage 버전
7.6	3.3	7.9	4.2
7.6	3.3	8.4	4.2
7.7	3.3	7.9	4.2
7.7	4.0	7.9	4.2
7.8	3.3	7.9	4.2
7.8	3.3	8.4	4.2
7.9	3.3	8.4	4.2
8.1	4.0	8.4	4.2
8.2	4.1	8.4	4.2
8.2	4.1	8.4	4.2
8.3	4.1	8.4	4.2

표 7.2. 컨테이너화된 배포에 지원되는 업그레이드 시나리오

pre-upgrade 상태			post-upgrade 상태		
호스트 Red Hat Enterprise Linux 버전	컨테이너 Red Hat Enterprise Linux 버전	Red Hat Ceph Storage 버전	호스트 Red Hat Enterprise Linux 버전	컨테이너 Red Hat Enterprise Linux 버전	Red Hat Ceph Storage 버전
7.6	7.8	3.3	7.9	8.4	4.2
7.7	7.8	3.3	7.9	8.4	4.2
7.7	8.1	4.0	7.9	8.4	4.2
7.8	7.8	3.3	7.9	8.4	4.2
8.1	8.1	4.0	8.4	8.4	4.2

pre-upgrade 상태			post-upgrade 상태		
8.2	8.2	4.1	8.4	8.4	4.2
8.3	8.3	4.1	8.4	8.4	4.2

## 7.2. 업그레이드 준비

**Red Hat Ceph Storage** 클러스터 업그레이드를 시작하기 전에 완료해야 할 사항이 몇 가지 있습니다. 이 단계는 하나 이상의 다른 항목에 지정하지 않는 한 **Red Hat Ceph Storage** 클러스터의 베어 메탈 및 컨테이너 배포에 모두 적용됩니다.



### 중요

최신 **Red Hat Ceph Storage 4** 버전으로만 업그레이드할 수 있습니다. 예를 들어 버전 4.1을 사용할 수 있는 경우 3에서 4.0으로 업그레이드할 수 없습니다. 4.1로 직접 이동해야 합니다.



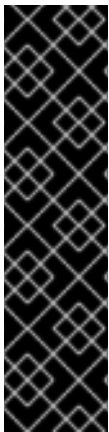
### 중요

**Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 업그레이드한 후 **FileStore** 오브젝트 저장소를 사용하는 경우 **BlueStore**로 마이그레이션해야 합니다.



### 중요

**ceph-ansible** 을 사용하여 **Red Hat Ceph Storage**를 업그레이드하는 동시에 **Red Hat Enterprise Linux 7**을 **Red Hat Enterprise Linux 8**로 업그레이드할 수는 없습니다. **Red Hat Enterprise Linux 7**을 그대로 유지해야 합니다. 운영 체제를 업그레이드하려면 **Red Hat Ceph Storage** 클러스터 및 운영 체제 수동 업그레이드를 참조하십시오.



### 중요

**Red Hat Ceph Storage 4.2z2** 이상 버전에서는 기본적으로 **bluefs\_buffered\_io** 옵션이 **True** 로 설정됩니다. 이 옵션을 사용하면 **BlueFS**에서 버퍼링된 읽기를 수행할 수 있으며 커널 페이지 캐시가 **MigsDB** 블록 읽기와 같은 읽기에 대한 보조 캐시 역할을 할 수 있습니다. 예를 들어, **OMAP** 반복 중에 모든 블록을 보관할 수 있을 만큼 충분히 크지 않은 경우 디스크 대신 페이지 캐시에서 해당 블록을 읽을 수 있습니다. 이로 인해 **osd\_memory\_target** 이 블록 캐시의 모든 항목을 유지하기에 너무 작으면 성능이 크게 향상될 수 있습니다. 현재 **bluefs\_buffered\_io** 를 활성화하고 시스템 수준 스위치를 비활성화하면 성능이 저하됩니다.

### 사전 요구 사항

- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.
- 스토리지 클러스터의 모든 노드의 시스템 클럭이 동기화됩니다. 모니터 노드가 동기화되지 않으면 업그레이드 프로세스가 제대로 완료되지 않을 수 있습니다.
- 버전 3에서 업그레이드하는 경우 버전 3 클러스터는 **Red Hat Ceph Storage 3의 최신 버전으로 업그레이드** 됩니다.
- 버전 4로 업그레이드하기 전에 **Prometheus** 노드 내보내기 서비스가 실행 중인 경우 서비스를 중지합니다.

### 예제

```
[root@mon ~]# systemctl stop prometheus-node-exporter.service
```



#### 중요

이는 **Red Hat Ceph Storage** 향후 릴리스에서 수정될 알려진 문제입니다. 이 문제에 대한 자세한 내용은 **Red Hat Knowledgebase 문서** 를 참조하십시오.



#### 참고

업그레이드 중 인터넷에 액세스할 수 없는 베어 메탈 또는 컨테이너 **Red Hat Ceph Storage** 클러스터 노드의 경우, **Red Hat Ceph Storage** 노드를 **CDN**에 등록하고 **Red Hat Ceph Storage** 설치 가이드 의 서브스크립션 연결 섹션의 절차에 따라 다음 절차를 따르십시오.

### 절차

1. 스토리지 클러스터의 모든 노드에 **root** 사용자로 로그인합니다.



**Ceph** 노드가 **Red Hat CDN(Content Delivery Network)**에 연결되어 있지 않은 경우 **ISO** 이미지를 사용하여 **Red Hat Ceph Storage**의 최신 버전으로 로컬 리포지토리를 업데이트하여 **Red Hat Ceph Storage**를 업그레이드할 수 있습니다.

3.

**Red Hat Ceph Storage**를 버전 3에서 버전 4로 업그레이드하는 경우 기존 **Ceph** 대시보드 설치를 제거합니다.

a.

**Ansible** 관리 노드에서 **cephmetrics-ansible** 디렉터리로 변경합니다.

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

b.

**purge.yml** 플레이북을 실행하여 기존 **Ceph** 대시보드 설치를 제거합니다.

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

4.

**Red Hat Ceph Storage**를 버전 3에서 버전 4로 업그레이드하는 경우 **Ansible** 관리 노드에서 **Ceph** 및 **Ansible** 리포지토리를 활성화합니다.

#### Red Hat Enterprise Linux 7

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms --enable=rhel-7-server-ansible-2.9-rpms
```

#### Red Hat Enterprise Linux 8

```
[root@admin ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
```

5.

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

**Red Hat Enterprise Linux 7**

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

**Red Hat Enterprise Linux 8**

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

6.

**infrastructure-playbooks/rolling\_update.yml** 플레이북을 편집하고 **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
```

각 **OSD** 노드에서 이러한 값은 **Ansible**이 최대 **25분** 동안 기다린 후 **30초**마다 스토리지 클러스터 상태를 확인하고 업그레이드 프로세스를 계속하기 전에 대기합니다.

**참고**

스토리지 클러스터의 사용된 스토리지 용량에 따라 **health\_osd\_check\_retries** 옵션 값을 **up** 또는 **down**으로 조정합니다. 예를 들어, **436TB** 중 **218TB**를 사용하는 경우 기본적으로 스토리지 용량의 **50%**를 사용하는 경우 **health\_osd\_check\_retries** 옵션을 **50** 으로 설정합니다.

7.

업그레이드하려는 스토리지 클러스터에 배타적 잠금 기능을 사용하는 **Ceph Block Device** 이미지가 포함된 경우 모든 **Ceph** 블록 장치 사용자에게 블랙리스트를 지정할 수 있는 권한이 있는지 확인하십시오.

```
ceph auth caps client.ID mon 'allow r, allow command "osd blacklist"' osd
'EXISTING_OSD_USER_CAPS'
```

8.

**Cockpit**을 사용하여 스토리지 클러스터를 원래 설치한 경우, **Cockpit**이 생성한 인벤토리 파일에 `/usr/share/ceph-ansible` 디렉터리에 심볼릭 링크를 만듭니다. `/usr/share/ansible-runner-service/inventory/hosts`:

a.

`/usr/share/ceph-ansible` 디렉토리로 변경합니다.

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

b.

심볼릭 링크를 만듭니다.

```
# ln -s /usr/share/ansible-runner-service/inventory/hosts hosts
```

9.

**ceph-ansible** 을 사용하여 클러스터를 업그레이드하려면 `etc/ansible/hosts` 디렉터리에 `hosts` 인벤토리 파일로 심볼릭 링크를 생성합니다.

```
# ln -s /etc/ansible/hosts hosts
```

10.

**Cockpit**을 사용하여 스토리지 클러스터를 원래 설치한 경우 **Cockpit** 생성된 **SSH** 키를 **Ansible** 사용자의 `~/.ssh` 디렉터리에 복사합니다.

a.

키를 복사합니다.

```
# cp /usr/share/ansible-runner-service/env/ssh_key.pub
/home/ANSIBLE_USERNAME/.ssh/id_rsa.pub
# cp /usr/share/ansible-runner-service/env/ssh_key
/home/ANSIBLE_USERNAME/.ssh/id_rsa
```

**ANSIBLE\_USERNAME** 을 일반적으로 `admin` 의 사용자 이름으로 바꿉니다.

예제

```
# cp /usr/share/ansible-runner-service/env/ssh_key.pub /home/admin/.ssh/id_rsa.pub
# cp /usr/share/ansible-runner-service/env/ssh_key /home/admin/.ssh/id_rsa
```

b.

키 파일에 적절한 소유자, 그룹 및 권한을 설정합니다.

```
# chown ANSIBLE_USERNAME:_ANSIBLE_USERNAME_
/home/ANSIBLE_USERNAME/.ssh/id_rsa.pub
# chown ANSIBLE_USERNAME:_ANSIBLE_USERNAME_
/home/ANSIBLE_USERNAME/.ssh/id_rsa
# chmod 644 /home/ANSIBLE_USERNAME/.ssh/id_rsa.pub
# chmod 600 /home/ANSIBLE_USERNAME/.ssh/id_rsa
```

**ANSIBLE\_USERNAME** 을 일반적으로 **admin** 의 사용자 이름으로 바꿉니다.

예제

```
# chown admin:admin /home/admin/.ssh/id_rsa.pub
# chown admin:admin /home/admin/.ssh/id_rsa
# chmod 644 /home/admin/.ssh/id_rsa.pub
# chmod 600 /home/admin/.ssh/id_rsa
```

추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage 리포지토리 활성화](#)를 참조하십시오.
- 클럭 동기화 및 클럭 불일치에 대한 자세한 내용은 [Red Hat Ceph Storage 문제 해결 가이드](#)의 [Clock Skew](#) 섹션을 참조하십시오.

### 7.3. ANSIBLE을 사용하여 스토리지 클러스터 업그레이드

**Ansible** 배포 툴을 사용하면 롤링 업그레이드를 수행하여 **Red Hat Ceph Storage** 클러스터를 업그레이드할 수 있습니다. 이러한 단계는 달리 명시되지 않는 한 베어 메탈 및 컨테이너 배포에 모두 적용됩니다.

사전 요구 사항

- **Ansible** 관리 노드에 대한 루트 수준 액세스.

- **ansible** 사용자 계정.

## 절차

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

### 예제

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

2. **Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 업그레이드하는 경우 **group\_vars/all.yml**, **group\_vars/osds.yml** 파일의 백업 사본, **group\_vars/clients.yml** 파일의 백업 사본을 만듭니다.

```
[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 3**에서 **Red Hat Ceph Storage 4**로 업그레이드하는 경우 **group\_vars/all.yml.sample**, **group\_vars/osds.yml.sample** 및 **group\_vars/clients.yml.sample** 파일의 새 복사본을 생성하고 **group\_vars/all.yml** 파일 이름을 **group\_vars/all.yml**, **group\_vars/osds.yml**, **group\_vars/osds.yml**, **group\_vars/osds.yml.yml** 로 변경합니다. 그에 따라 열고 편집하여 이전에 백업한 사본에 변경 사항을 적용합니다.

```
[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. **group\_vars/osds.yml** 파일을 편집합니다. 다음 옵션을 추가하고 설정합니다.

```
nb_retry_wait_osd_up: 60
delay_wait_osd_up: 10
```



## 참고

이는 기본값입니다. 사용 사례에 따라 값을 수정할 수 있습니다.

5.

**Red Hat Ceph Storage 4**의 새 마이너 버전으로 업그레이드하는 경우 `group_vars/all.yml`의 `grafana_container_image` 값이 `group_vars/all.yml.sample` 과 동일한지 확인합니다. 동일하지 않은 경우 다음과 같이 편집합니다.

## 예제

```
grafana_container_image: registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:4
```



## 참고

표시된 이미지 경로는 **ceph-ansible** 버전 **4.0.23-1**에 포함되어 있습니다.

6.

샘플 파일에서 최신 `site.yml` 또는 `site-container.yml` 파일을 복사합니다.

a.

베어 메탈 배포의 경우:

```
[root@admin ceph-ansible]# cp site.yml.sample site.yml
```

b.

컨테이너 배포의 경우:

```
[root@admin ceph-ansible]# cp site-container.yml.sample site-container.yml
```

7.

`group_vars/all.yml` 파일을 열고 다음 옵션을 편집합니다.

a.

`fetch_directory` 옵션을 추가합니다.

```
fetch_directory: FULL_DIRECTORY_PATH
```

## 대체 버전

- **Ansible** 사용자의 홈 디렉터리와 같이 쓰기 가능한 위치가 있는 **FULL\_DIRECTORY\_PATH**.

b.

업그레이드하려는 클러스터에 **Ceph Object Gateway** 노드가 포함된 경우 **radosgw\_interface** 옵션을 추가합니다.

```
radosgw_interface: INTERFACE
```

## 대체 버전

- **Ceph Object Gateway** 노드가 수신 대기하는 인터페이스를 사용하여 인터넷 연결을 수행합니다.

c.

현재 설정에 **SSL** 인증서가 구성된 경우 다음을 편집해야 합니다.

```
radosgw_frontend_ssl_certificate: /etc/pki/ca-trust/extracted/CERTIFICATE_NAME
radosgw_frontend_port: 443
```

d.

기본 **OSD** 오브젝트 저장소는 **BlueStore**입니다. 기존 **OSD** 오브젝트 저장소를 유지하려면 **osd\_objectstore** 옵션을 **filestore** 로 명시적으로 설정해야 합니다.

```
osd_objectstore: filestore
```



## 참고

**osd\_objectstore** 옵션을 **filestore** 로 설정하면 **OSD**는 **BlueStore** 대신 **FileStore**를 사용합니다.



## 중요

**Red Hat Ceph Storage 4**부터 **FileStore**는 더 이상 사용되지 않는 기능입니다. **Red Hat**은 **FileStore OSD**를 **BlueStore OSD**로 마이그레이션하는 것이 좋습니다.

e.

**Red Hat Ceph Storage 4.1**부터 **/usr/share/ceph-ansible/group\_vars/all.yml** 에서

`dashboard_admin_password` 및 `grafana_admin_password` 의 주석 처리를 해제하거나 설정해야 합니다. 각각에 대해 보안 암호를 설정합니다. `dashboard_admin_user` 및 `grafana_admin_user` 의 사용자 지정 사용자 이름도 설정합니다.

f.

베어 메탈 및 컨테이너 배포 모두에 대해 다음을 수행합니다.

i.

`upgrade_ceph_packages` 옵션의 주석을 제거하고 `True` 로 설정합니다.

```
upgrade_ceph_packages: True
```

ii.

`ceph_rhcs_version` 옵션을 4 로 설정합니다.

```
ceph_rhcs_version: 4
```



참고

`ceph_rhcs_version` 옵션을 4 로 설정하면 최신 버전의 **Red Hat Ceph Storage 4**가 표시됩니다.

iii.

`ceph_docker_registry` 정보를 `all.yml` 에 추가합니다.

구문

```
ceph_docker_registry: registry.redhat.io
ceph_docker_registry_username: SERVICE_ACCOUNT_USER_NAME
ceph_docker_registry_password: TOKEN
```



참고

**Red Hat Registry Service** 계정이 없는 경우 레지스트리 서비스 계정 웹 페이지를 사용하여 생성합니다. 자세한 내용은 [Red Hat Container Registry Authentication Knowledgebase](#) 문서를 참조하십시오.





## 참고

**ceph\_docker\_registry\_username** 및 **ceph\_docker\_registry\_password** 매개변수에 서비스 계정을 사용하는 것 외에도 고객 포털 자격 증명을 사용하지만 보안을 보장하기 위해 **ceph\_docker\_registry\_password** 매개변수를 암호화할 수 있습니다. 자세한 내용은 [ansible-vault](#)를 사용하여 **Ansible 암호 변수 암호화**를 참조하십시오.

- g. 컨테이너 배포의 경우 다음을 수행합니다.

- i. **ceph\_docker\_image** 옵션을 변경하여 **Ceph 4** 컨테이너 버전을 가리킵니다.

```
ceph_docker_image: rhceph/rhceph-4-rhel8
```

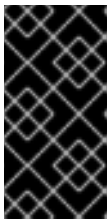
- ii. **rhceph/rhceph-4-rhel8**의 최신 버전을 가리키도록 **ceph\_docker\_image\_tag** 옵션을 변경합니다.

```
ceph_docker_image_tag: latest
```

8. **Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 업그레이드하는 경우 **Ansible** 인벤토리 파일을 기본적으로 열고 **[grafana-server]** 섹션에 **Ceph** 대시보드 노드 이름 또는 **IP** 주소를 추가합니다. 이 섹션이 없는 경우 노드 이름 또는 **IP** 주소와 함께 이 섹션을 추가합니다.

9. **Ansible** 사용자로 전환하거나 **Ansible** 사용자로 로그인한 다음 **rolling\_update.yml** 플레이북을 실행합니다.

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/rolling_update.yml
-i hosts
```



## 중요

**rolling\_update.yml** 플레이북에 **--limit Ansible** 옵션을 사용하는 것은 지원되지 않습니다.

10. **RBD** 미러링 데몬 노드의 **root** 사용자로 **rbd-mirror** 패키지를 수동으로 업그레이드합니다.

```
[root@rbd ~]# yum upgrade rbd-mirror
```

11.

**rbd-mirror** 데몬을 다시 시작합니다.

```
systemctl restart ceph-rbd-mirror@CLIENT_ID
```

12.

스토리지 클러스터의 상태를 확인합니다.

a.

베어 메탈 배포의 경우 **root** 사용자로 모니터 노드에 로그인하고 **Ceph status** 명령을 실행합니다.

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

b.

컨테이너 배포의 경우 **root** 사용자로 **Ceph Monitor** 노드에 로그인합니다.

i.

실행 중인 컨테이너를 모두 나열합니다.

#### **Red Hat Enterprise Linux 7**

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

#### **Red Hat Enterprise Linux 8**

```
[root@mon ~]# podman ps
```

ii.

상태 점검:

#### **Red Hat Enterprise Linux 7**

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

## Red Hat Enterprise Linux 8

```
[root@mon ~]# podman exec ceph-mon-MONITOR_NAME ceph -s
```

### 대체 버전

- 이전 단계에서 찾은 **Ceph Monitor** 컨테이너의 이름이 **MONITOR\_NAME** 입니다.

### 예제

```
[root@mon ~]# podman exec ceph-mon-mon01 ceph -s
```

13.

선택 사항: **Red Hat Ceph Storage 3.x**에서 **Red Hat Ceph Storage 4.x**로 업그레이드하는 경우 다음 상태 경고가 표시될 수 있습니다. **336 OSD**에서 탐지된 레거시 **BlueStore** 통계 보고 이는 최신 코드가 풀 통계를 다르게 계산하기 때문에 발생합니다.

**bluestore\_fsck\_quick\_fix\_on\_mount** 매개변수를 설정하여 이 문제를 해결할 수 있습니다.

a.

**bluestore\_fsck\_quick\_fix\_on\_mount** 를 **true** 로 설정합니다.

### 예제

```
[root@mon ~]# ceph config set osd bluestore_fsck_quick_fix_on_mount true
```

- b. **OSD가 중단된 동안 데이터 이동을 방지하려면 `noout` 및 `norebalance` 플래그를 설정합니다.**

예제

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

- c. **베어 메탈 배포의 경우 스토리지 클러스터의 모든 OSD 노드에서 `ceph-osd.target` 을 다시 시작합니다.**

예제

```
[root@osd ~]# systemctl restart ceph-osd.target
```

- d. **컨테이너화된 배포의 경우 개별 OSD를 하나씩 다시 시작하고 모든 배치 그룹이 `active+clean` 상태가 될 때까지 기다립니다.**

구문

```
systemctl restart ceph-osd@OSD_ID.service
```

예제

```
[root@osd ~]# systemctl restart ceph-osd@0.service
```

e.

모든 **OSD**가 복구되면 **nout** 및 **norebalance** 플래그를 설정 해제합니다.

예제

```
[root@mon ~]# ceph osd unset noout
[root@mon ~]# ceph osd unset norebalance
```

f.

모든 **OSD**가 복구되면 **Bluestore\_fsck\_quick\_fix\_on\_mount** 를 **false** 로 설정합니다.

예제

```
[root@mon ~]# ceph config set osd bluestore_fsck_quick_fix_on_mount false
```

g.

선택 사항: 베어 메탈 배포를 위한 대체 방법은 **OSD** 서비스를 중지하고, **ceph-bluestore-tool** 명령을 사용하여 **OSD**에서 복구 기능을 실행한 다음, **OSD** 서비스를 시작하는 것입니다.

i.

**OSD** 서비스를 중지합니다.

```
[root@osd ~]# systemctl stop ceph-osd.target
```

ii.

**OSD**에서 복구 기능을 실행하여 실제 **OSD ID**를 지정합니다.

구문

```
ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-OSDID repair
```

예제

```
[root@osd ~]# ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-2 repair
```

iii.

**OSD 서비스를 시작합니다.**

```
[root@osd ~]# systemctl start ceph-osd.target
```

14.

업그레이드가 완료되면 **Ansible** 플레이북을 실행하여 **FileStore OSD**를 **BlueStore OSD**로 마이그레이션할 수 있습니다.

구문

```
ansible-playbook infrastructure-playbooks/filestore-to-bluestore.yml --limit  
OSD_NODE_TO_MIGRATE
```

예제

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/filestore-to-  
bluestore.yml --limit osd01
```

마이그레이션이 완료되면 다음 하위 단계를 수행합니다.

- a. **group\_vars/osds.yml** 파일을 편집하려면 을 열고 **osd\_objectstore** 옵션을 **bluestore** 로 설정합니다. 예를 들면 다음과 같습니다.

```
osd_objectstore: bluestore
```

- b. **lvm\_volumes** 변수를 사용하는 경우 **journal** 및 **journal\_vg** 옵션을 각각 **db** 및 **db\_vg** 로 변경합니다. 예를 들면 다음과 같습니다.

이전

```
lvm_volumes:
- data: /dev/sdb
  journal: /dev/sdc1
- data: /dev/sdd
  journal: journal1
  journal_vg: journals
```

**Bluestore**로 변환 한 후

```
lvm_volumes:
- data: /dev/sdb
  db: /dev/sdc1
- data: /dev/sdd
  db: journal1
  db_vg: journals
```

15. **OpenStack** 환경에서 작업하는 경우 풀에 **RBD** 프로필 을 사용하도록 모든 **VDDK** 사용자를 업데이트합니다. 다음 명령은 **root** 사용자로 실행해야 합니다.

- a. **Glance** 사용자:

### 구문

```
ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd  
pool=GLANCE_POOL_NAME'
```

### 예제

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

b.

#### **Cinder 사용자:**

### 구문

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

c.

#### **OpenStack 일반 사용자:**



## 구문

```
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@mon ~]# 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 사용하여 이전 **CAPS** 설정이 캐시에서 삭제하고 새 **RBD** 프로필 설정을 적용할 수 있습니다.

16.

선택 사항: 클라이언트 노드에서 **Ceph** 클라이언트 측 라이브러리에 따라 애플리케이션을 다시 시작합니다.



## 참고

**QEMU** 또는 **KVM** 인스턴스를 실행하거나 전용 **QEMU** 또는 **KVM** 클라이언트를 사용하는 **OpenStack Nova** 컴퓨팅 노드를 업그레이드하는 경우 이 경우 인스턴스를 다시 시작하지 않기 때문에 **QEMU** 또는 **KVM** 인스턴스를 중지하고 시작합니다.

## 추가 리소스

- 자세한 내용은 [제한 옵션](#) 이해를 참조하십시오.

- 자세한 내용은 **Red Hat Ceph Storage** 관리 가이드에서 **FileStore**에서 개체 저장소를 마이그레이션하는 방법을 참조하십시오.
- 자세한 내용은 지식베이스 문서 **after a ceph-upgrade the cluster status reports 'Legacy BlueStore stats reporting detected'**.

#### 7.4. 명령줄 인터페이스를 사용하여 스토리지 클러스터 업그레이드

스토리지 클러스터가 실행되는 동안 **Red Hat Ceph Storage 3.3**에서 **Red Hat Ceph Storage 4**로 업그레이드할 수 있습니다. 이러한 버전의 중요한 차이점은 **Red Hat Ceph Storage 4**는 기본적으로 **msgr2** 프로토콜을 사용하므로 포트 **3300** 을 사용한다는 것입니다. 열려 있지 않으면 클러스터에서 **HEALTH\_WARN** 오류가 발생합니다.

스토리지 클러스터를 업그레이드할 때 고려해야 할 제약 조건은 다음과 같습니다.

- **Red Hat Ceph Storage 4**는 기본적으로 **msgr2** 프로토콜을 사용합니다. **Ceph Monitor** 노드에서 포트 **3300** 이 열려 있는지 확인합니다.
- **ceph-monitor** 데몬을 **Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 업그레이드한 후 **Red Hat Ceph Storage 3** **ceph-osd** 데몬은 **Red Hat Ceph Storage 4**로 업그레이드할 때까지 새 **OSD**를 생성할 수 없습니다.
- 업그레이드가 진행되는 동안 풀을 생성하지 마십시오.

#### 사전 요구 사항

- **Ceph Monitor, OSD, Object Gateway** 노드에 대한 루트 수준 액세스.

#### 절차

1. **Red Hat Ceph Storage 3**을 실행하는 동안 클러스터가 모든 **PG**에 대해 하나 이상의 전체 스크립을 완료했는지 확인합니다. 이렇게 하지 않으면 모니터 데몬이 시작 시 쿼럼에 대한 참여를 거부하고 작동하지 않게 유지할 수 있습니다. 클러스터가 모든 **PG** 중 하나 이상의 전체 스크립을 완료했는지 확인하려면 다음을 실행합니다.

```
# ceph osd dump | grep ^flags
```

**Red Hat Ceph Storage 3에서 Red Hat Ceph Storage 4로 업그레이드를 진행하려면 OSD 맵에 `recovery_deletes` 및 `purged_snapdirs` 플래그가 포함되어야 합니다.**

2. 클러스터가 정상이고 깨끗한 상태인지 확인합니다.

```
ceph health
HEALTH_OK
```

3. **`ceph-mon` 및 `ceph-manager` 를 실행하는 노드의 경우 다음을 실행합니다.**

```
# subscription-manager repos --enable=rhel-7-server-rhceph-4-mon-rpms
```

**Red Hat Ceph Storage 4 패키지가 활성화되면 각 `ceph-mon` 및 `ceph-manager` 노드에서 다음을 실행합니다.**

```
# firewall-cmd --add-port=3300/tcp
# firewall-cmd --add-port=3300/tcp --permanent
# yum update -y
# systemctl restart ceph-mon@<mon-hostname>
# systemctl restart ceph-mgr@<mgr-hostname>
```

**&lt;mon-hostname > 및 < mgr-hostname >을 대상 호스트의 호스트 이름으로 바꿉니다.**

4. **OSD를 업그레이드하기 전에 Ceph Monitor 노드에서 `noout` 및 `nodeep-scrub` 플래그를 설정하여 업그레이드 중에 OSD가 재조정되지 않도록 합니다.**

```
# ceph osd set noout
# ceph osd det nodeep-scrub
```

5. 각 OSD 노드에서 다음을 실행합니다.

```
# subscription-manager repos --enable=rhel-7-server-rhceph-4-osd-rpms
```

**Red Hat Ceph Storage 4 패키지가 활성화되면 OSD 노드를 업데이트합니다.**

```
# yum update -y
```

노드에서 실행되는 각 **OSD** 데몬에 대해 다음을 실행합니다.

```
# systemctl restart ceph-osd@<osd-num>
```

&lt;osd-num& gt;을 제시작할 **osd** 번호로 바꿉니다. 다음 **OSD** 노드로 진행하기 전에 노드의 모든 **OSD**가 다시 시작되었는지 확인합니다.

6.

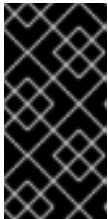
**ceph-disk** 로 배포된 스토리지 클러스터에 **OSD**가 있는 경우 **ceph-volume** 에 데몬을 시작하도록 지시합니다.

```
# ceph-volume simple scan
# ceph-volume simple activate --all
```

7.

**Failure**의 기능만 활성화합니다.

```
# ceph osd require-osd-release nautilus
```



중요

이 단계를 실행하지 않으면 **msgr2** 가 활성화된 후 **OSD**가 통신할 수 없습니다.

8.

모든 **OSD** 노드를 업그레이드한 후 **Ceph Monitor** 노드에서 **noout** 및 **nodeep-scrub** 플래그를 설정 해제합니다.

```
# ceph osd unset noout
# ceph osd unset nodeep-scrub
```

9.

기존 **NetNamespace** 버킷을 최신 버킷 유형 **straw2** 로 전환합니다.

```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

10.

**Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 업그레이드한 후 모든 데몬이 업데이트되면 다음 단계를 실행하십시오.

a.

**messenger v2** 프로토콜, **msgr2** 를 활성화합니다.

```
ceph mon enable-msgr2
```

그러면 이전 기본 포트 **6789**에 바인딩하는 모든 **Ceph** 모니터가 **3300**의 새 포트에 바인딩됩니다.

b.

모니터 상태를 확인합니다.

```
ceph mon dump
```



참고

**nautilus OSD**를 실행하면 **v2** 주소에 자동으로 바인딩되지 않습니다. 다시 시작해야 합니다.

11.

**Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 업그레이드된 각 호스트에 대해 모니터링 포트를 지정하지 않도록 **ceph.conf** 파일을 업데이트하거나 **v2** 및 **v1** 주소와 포트를 모두 참조합니다.

12.

**ceph.conf** 파일의 구성 옵션을 스토리지 클러스터의 구성 데이터베이스로 가져옵니다.

예제

```
[root@mon ~]# ceph config assimilate-conf -i /etc/ceph/ceph.conf
```

a.

스토리지 클러스터의 구성 데이터베이스를 확인합니다.

예제

```
[root@mon ~]# ceph config dump
```

b.

선택 사항: **Red Hat Ceph Storage 4**로 업그레이드한 후 각 호스트에 대해 최소 **ceph.conf** 파일을 생성합니다.

예제

```
[root@mon ~]# ceph config generate-minimal-conf > /etc/ceph/ceph.conf.new
[root@mon ~]# mv /etc/ceph/ceph.conf.new /etc/ceph/ceph.conf
```

13.

**Ceph Object Gateway** 노드에서 다음을 실행합니다.

```
# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
```

**Red Hat Ceph Storage 4** 패키지가 활성화되면 노드를 업데이트하고 **ceph-rgw** 데몬을 다시 시작하십시오.

```
# yum update -y
# systemctl restart ceph-rgw@<rgw-target>
```

& *It;rgw-target& gt;*을 재시작할 **rgw** 대상으로 바꿉니다.

14.

관리 노드의 경우 다음을 실행합니다.

```
# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
# yum update -y
```

15.

클러스터가 정상이고 깨끗한 상태인지 확인합니다.

```
# ceph health
HEALTH_OK
```

16.

선택 사항: 클라이언트 노드에서 **Ceph** 클라이언트 측 라이브러리에 따라 애플리케이션을 다

시 시작합니다.

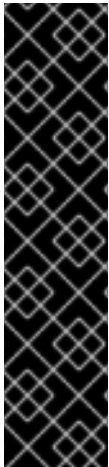


#### 참고

**QEMU** 또는 **KVM** 인스턴스를 실행하거나 전용 **QEMU** 또는 **KVM** 클라이언트를 사용하는 **OpenStack Nova** 컴퓨팅 노드를 업그레이드하는 경우 이 경우 인스턴스를 다시 시작하지 않기 때문에 **QEMU** 또는 **KVM** 인스턴스를 중지하고 시작합니다.

### 7.5. CEPH 파일 시스템 메타데이터 서버 노드를 수동으로 업그레이드

**Red Hat Enterprise Linux 7** 또는 **8**을 실행하는 **Red Hat Ceph Storage** 클러스터에서 **Ceph** 파일 시스템(**CephFS**) 메타데이터 서버(**MDS**) 소프트웨어를 수동으로 업그레이드할 수 있습니다.



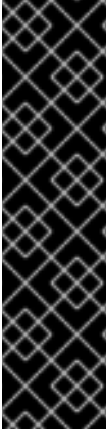
#### 중요

스토리지 클러스터를 업그레이드하기 전에 활성 **MDS** 순위 수를 파일 시스템별로 하나씩 줄입니다. 이렇게 하면 여러 **MDS** 간에 가능한 모든 버전 충돌이 발생하지 않습니다. 또한 업그레이드하기 전에 모든 **edge** 노드가 오프라인 상태로 전환하십시오.

**MDS** 클러스터에는 버전 관리 또는 파일 시스템 플래그가 내장되어 있지 않기 때문입니다. 이러한 기능을 사용하지 않으면 여러 **MDS**가 다른 버전의 **MDS** 소프트웨어를 사용하여 통신할 수 있으며 어설션 또는 기타 오류가 발생할 수 있습니다.

#### 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드는 **Red Hat Ceph Storage** 버전 **3.3z64** 또는 **4.1**을 사용합니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.



## 중요

기본 XFS 파일 시스템은 **ftype=1** 또는 **d\_type** 지원을 사용하여 포맷해야 합니다. **xfs\_info /var** 명령을 실행하여 **ftype** 이 1로 설정되어 있는지 확인합니다. **ftype** 값이 1이 아닌 경우 새 디스크를 연결하거나 볼륨을 만듭니다. 이 새 장치 상단에서 새 XFS 파일 시스템을 생성하여 **/var/lib/containers** 에 마운트합니다.

Red Hat Enterprise Linux 8.0부터 **mkfs.xfs** 는 기본적으로 **ftype=1** 을 활성화합니다.

## 절차

1. 활성 **MDS** 순위 수를 1로 줄입니다.

### 구문

```
ceph fs set FILE_SYSTEM_NAME max_mds 1
```

### 예제

```
[root@mds ~]# ceph fs set fs1 max_mds 1
```

2. 클러스터가 모든 **MDS** 순위를 중지할 때까지 기다립니다. 모든 **MDS**가 중지되면 순위 0만 활성화되어야 합니다. 나머지는 **Wait** 모드여야 합니다. 파일 시스템의 상태를 확인합니다.

```
[root@mds ~]# ceph status
```

3. **systemctl** 을 사용하여 모든 **MDS**를 오프라인으로 전환합니다.

```
[root@mds ~]# systemctl stop ceph-mds.target
```



4. 하나의 **MDS**만 온라인 상태인지 확인하고 파일 시스템의 순위 **0**이 있는지 확인합니다.

```
[root@mds ~]# ceph status
```

5. **RHEL 7의 Red Hat Ceph Storage 3**에서 업그레이드하는 경우 **Red Hat Ceph Storage 3** 툴 리포지토리를 비활성화하고 **Red Hat Ceph Storage 4** 툴 리포지토리를 활성화합니다.

```
[root@mds ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
[root@mds ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
```

6. 노드를 업데이트하고 **ceph-mds** 데몬을 다시 시작합니다.

```
[root@mds ~]# yum update -y
[root@mds ~]# systemctl restart ceph-mds.target
```

7. 대기 데몬에 대해 동일한 프로세스를 따릅니다. 툴 리포지토리를 비활성화 및 활성화한 다음 각 **Wait MDS**를 업그레이드 및 다시 시작합니다.

```
[root@mds ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
[root@mds ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
[root@mds ~]# yum update -y
[root@mds ~]# systemctl restart ceph-mds.target
```

8. **Wait**에서 **MDS**를 모두 다시 시작할 때 스토리지 클러스터에 대해 **max\_mds**의 이전 값을 복원하십시오.

구문

```
ceph fs set FILE_SYSTEM_NAME max_mds ORIGINAL_VALUE
```

예제

```
[root@mds ~]# ceph fs set fs1 max_mds 5
```

## 7.6. 추가 리소스

- **3.3z5와 관련이 있는 패키지 버전을 보려면 [Red Hat Ceph Storage 릴리스 및 해당 Ceph 패키지 버전은 무엇입니까?](#)**

## 8장. RED HAT CEPH STORAGE 클러스터 및 운영 체제를 수동으로 업그레이드

일반적으로 **ceph-ansible** 을 사용하면 **Red Hat Ceph Storage**와 **Red Hat Enterprise Linux**를 새로운 주요 릴리스로 업그레이드할 수 없습니다. 예를 들어 **Red Hat Enterprise Linux 7**에 있는 경우 **ceph-ansible** 을 사용하는 경우 해당 버전을 그대로 유지해야 합니다. 그러나 시스템 관리자는 수동으로 이 작업을 수행할 수 있습니다.

이 장은 **Red Hat Enterprise Linux 7.9**에서 실행되는 버전 4.1 또는 3.3z6에서 **Red Hat Enterprise Linux 8.4**에서 실행되는 버전 4.2의 **Red Hat Ceph Storage** 클러스터로 수동으로 업그레이드합니다.

### 중요

컨테이너화된 **Red Hat Ceph Storage** 클러스터를 버전 3.x 또는 4.x에서 버전 4.2로 업그레이드하려면 다음 세 가지 섹션, 지원되는 **Red Hat Ceph Storage** 업그레이드 시나리오, 업그레이드 준비, **Red Hat Ceph Storage** 설치 가이드에서 **Ansible**을 사용하여 스토리지 클러스터 업그레이드를 참조하십시오.

기존 **systemd** 템플릿을 마이그레이션하려면 **docker-to-podman** 플레이북을 실행합니다.

```
[user@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/docker-to-podman.yml -i hosts
```

여기서 **user** 는 **Ansible** 사용자입니다.

### 중요

노드가 둘 이상의 데몬과 함께 배치되면 이 장의 특정 섹션을 따라 노드에 배치된 데몬을 실행합니다. 예를 들어 **Ceph Monitor** 데몬과 **OSD** 데몬과 함께 배치된 노드:

**Ceph Monitor** 노드 및 운영 체제 수동 업그레이드 및 **Ceph OSD** 노드 및 해당 운영 체제를 수동으로 업그레이드하는 것을 참조하십시오.

### 중요

**Leapp** 업그레이드 유틸리티에서 **OSD** 암호화를 통한 업그레이드를 지원하지 않으므로 **Ceph OSD** 노드와 해당 운영 체제를 수동으로 업그레이드하면 암호화된 **OSD** 파티션에서 작동하지 않습니다.

### 8.1. 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드는 **Red Hat Enterprise Linux 7.9**를 실행하고 있습니다.
- 노드는 **Red Hat Ceph Storage** 버전 **3.3z6** 또는 **4.1**을 사용합니다.
- **Red Hat Enterprise Linux 8.3**의 설치 소스에 액세스합니다.

### 8.2. CEPH MONITOR 노드 및 해당 운영 체제 수동 업그레이드

시스템 관리자는 **Red Hat Ceph Storage** 클러스터 노드와 **Red Hat Enterprise Linux** 운영 체제를 새로운 주요 릴리스로 수동으로 **Ceph Monitor** 소프트웨어를 업그레이드할 수 있습니다.



#### 중요

한 번에 하나의 모니터 노드에서만 절차를 수행합니다. 클러스터 액세스 문제를 방지하려면 다음 노드를 진행하기 전에 현재 업그레이드된 모니터 노드가 정상적인 작업으로 반환되었는지 확인합니다.

#### 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드는 **Red Hat Enterprise Linux 7.9**를 실행하고 있습니다.
- 노드는 **Red Hat Ceph Storage** 버전 **3.3z6** 또는 **4.1**을 사용합니다.
- **Red Hat Enterprise Linux 8.3**의 설치 소스에 액세스합니다.

#### 절차

1. **모니터 서비스를 중지합니다.**

구문

```
systemctl stop ceph-mon@MONITOR_ID
```

**MONITOR\_ID** 를 모니터의 ID 번호로 바꿉니다.

2. **Red Hat Ceph Storage 3을 사용하는 경우 Red Hat Ceph Storage 3 리포지토리를 비활성화합니다.**

- a. **tools 리포지토리를 비활성화합니다.**

```
[root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
```

- b. **mon 저장소를 비활성화합니다.**

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

3. **Red Hat Ceph Storage 4를 사용하는 경우 Red Hat Ceph Storage 4 리포지토리를 비활성화합니다.**

- a. **tools 리포지토리를 비활성화합니다.**

```
[root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
```

- b. **mon 저장소를 비활성화합니다.**

```
[root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-mon-rpms
```

4. **st app 유틸리티를 설치합니다. [Red Hat Enterprise Linux 7](#)에서 [Red Hat Enterprise Linux](#)**

8로 업그레이드를 참조하십시오.

5. **stapp preupgrade** 검사를 통해 실행합니다. 명령줄의 **Assessing upgradability**를 참조하십시오.

6. `/etc/ssh/sshd_config` 에서 **PermitRootLogin yes** 를 설정합니다.

7. **OpenSSH SSH** 데몬을 다시 시작합니다.

```
[root@mon ~]# systemctl restart sshd.service
```

8. **Linux** 커널에서 **iSCSI** 모듈을 제거합니다.

```
[root@mon ~]# modprobe -r iscsi
```

9. **RHEL 7**에서 **RHEL 8**로의 업그레이드 수행에 따라 업그레이드를 수행합니다.

10. 노드를 재부팅합니다.

11. **Red Hat Enterprise Linux 8**용 **Red Hat Ceph Storage 4**용 리포지토리를 활성화합니다.

- a. **툴 리포지토리를 활성화합니다.**

```
[root@mon ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

- b. **mon** 저장소를 활성화합니다.

```
[root@mon ~]# subscription-manager repos --enable=rhceph-4-mon-for-rhel-8-x86_64-rpms
```

12. **ceph-mon** 패키지를 설치합니다.

```
[root@mon ~]# dnf install ceph-mon
```

13.

관리자 서비스가 모니터 서비스와 함께 배치되면 **ceph-mgr** 패키지를 설치합니다.

```
[root@mon ~]# dnf install ceph-mgr
```

14.

아직 업그레이드되지 않았거나 이미 이러한 파일이 복원된 노드에서 **ceph-client-admin.keyring** 및 **ceph.conf** 파일을 복원하십시오.

15.

기존 **NetNamespace** 버킷을 최신 버킷 유형 **straw2** 로 전환합니다.

```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

16.

**Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 업그레이드한 후 모든 데몬이 업데이트되면 다음 단계를 실행하십시오.

a.

**messenger v2** 프로토콜, **msgr2** 를 활성화합니다.

```
ceph mon enable-msgr2
```

그러면 이전 기본 포트 **6789**에 바인딩하는 모든 **Ceph** 모니터가 **3300**의 새 포트에 바인딩됩니다.



중요

추가 **Ceph Monitor** 구성을 수행하기 전에 모든 **Ceph** 모니터가 **Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 업그레이드되었는지 확인합니다.

b.

모니터 상태를 확인합니다.

```
ceph mon dump
```



참고

**nautilus OSD**를 실행하면 **v2** 주소에 자동으로 바인딩되지 않습니다. 다시 시작해야 합니다.

17.

**Red Hat Ceph Storage 3**에서 **Red Hat Ceph Storage 4**로 업그레이드된 각 호스트에 대해 모니터링 포트를 지정하지 않도록 **ceph.conf** 파일을 업데이트하거나 v2 및 v1 주소와 포트를 모두 참조합니다. **ceph.conf** 파일의 구성 옵션을 스토리지 클러스터의 구성 데이터베이스로 가져옵니다.

예제

```
[root@mon ~]# ceph config assimilate-conf -i /etc/ceph/ceph.conf
```

a.

스토리지 클러스터의 구성 데이터베이스를 확인합니다.

예제

```
[root@mon ~]# ceph config dump
```

b.

선택 사항: **Red Hat Ceph Storage 4**로 업그레이드한 후 각 호스트에 대해 최소 **ceph.conf** 파일을 생성합니다.

예제

```
[root@mon ~]# ceph config generate-minimal-conf > /etc/ceph/ceph.conf.new  
[root@mon ~]# mv /etc/ceph/ceph.conf.new /etc/ceph/ceph.conf
```

18.

**leveldb** 패키지를 설치합니다.

```
[root@mon ~]# dnf install leveldb
```



19. 모니터 서비스를 시작합니다.

```
[root@mon ~]# systemctl start ceph-mon.target
```

20. 관리자 서비스가 모니터 서비스와 함께 배치되면 관리자 서비스도 시작합니다.

```
[root@mon ~]# systemctl start ceph-mgr.target
```

21. 모니터 서비스가 백업되었으며 쿼럼에 있는지 확인합니다.

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

서비스 아래의 **mon**: 행에서 **node**가 쿼럼이 아닌 쿼럼에 나열되는지 확인합니다.

예제

```
mon: 3 daemons, quorum ceph4-mon,ceph4-mon2,ceph4-mon3 (age 2h)
```

22. 관리자 서비스가 모니터 서비스와 함께 배치되어 있는 경우 해당 서비스가 실행 중인지 확인합니다.

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

서비스 아래의 **mgr**: 행에서 관리자의 노드 이름을 찾습니다.

예제

```
mgr: ceph4-mon(active, since 2h), standbys: ceph4-mon3, ceph4-mon2
```

23.

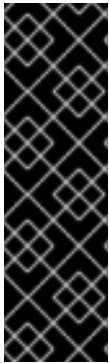
모두 업그레이드될 때까지 모든 모니터 노드에서 위의 단계를 반복합니다.

추가 리소스

- 자세한 내용은 설치 가이드의 [Red Hat Ceph Storage 클러스터 및 운영 체제 수동 업그레이드를 참조하십시오.](#)
- 자세한 내용은 [Red Hat Enterprise Linux 7에서 Red Hat Enterprise Linux 8으로 업그레이드를 참조하십시오.](#)

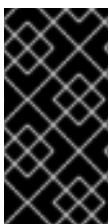
8.3. CEPH OSD 노드 및 운영 체제 수동 업그레이드

시스템 관리자는 [Red Hat Ceph Storage 클러스터 노드와 Red Hat Enterprise Linux 운영 체제를 새로운 주요 릴리스로 수동으로 업그레이드할 수 있습니다.](#)



중요

이 절차는 **Ceph 클러스터의 각 OSD 노드에 대해 수행해야 하지만 일반적으로 한 번에 하나의 OSD 노드만 수행해야 합니다. OSD 노드의 최대 하나의 실패 도메인은 병렬로 수행할 수 있습니다. 예를 들어 요청별 복제가 사용 중인 경우 하나의 전체 랙의 OSD 노드를 병렬로 업그레이드할 수 있습니다. 데이터 액세스 문제를 방지하려면 현재 OSD 노드의 OSD가 정상 작동으로 반환되고 다음 OSD를 진행하기 전에 모든 클러스터의 PG가 active+clean 상태에 있는지 확인합니다.**



중요

**Leapp upgrade 유틸리티는 OSD 암호화로 업그레이드를 지원하지 않기 때문에 암호화된 OSD 파티션에서는 이 절차가 작동하지 않습니다.**



중요

**OSD가 ceph-disk 를 사용하여 생성되었으며 여전히 ceph-disk 로 관리되는 경우 ceph-volume 을 사용하여 관리해야 합니다. 이 내용은 아래 선택적 단계에서 다룹니다.**

사전 요구 사항

- 실행 중인 [Red Hat Ceph Storage 클러스터.](#)

- 노드는 **Red Hat Enterprise Linux 7.9**를 실행하고 있습니다.
- 노드는 **Red Hat Ceph Storage 버전 3.3z6** 또는 **4.0**을 사용합니다.
- **Red Hat Enterprise Linux 8.3**의 설치 소스에 액세스합니다.

## 절차

1. 마이그레이션 중에 **OSD noout** 플래그가 표시되지 않도록 설정합니다.

```
ceph osd set noout
```

2. 클러스터에서 불필요한 로드를 방지하고 노드가 마이그레이션의 경우 데이터 재시프를 방지하려면 **OSD nobackfill, norecover, noscrub, nodeep-scrub** 플래그를 설정하여 **OSD nobackfill , norebalance, noscrub** 플래그를 설정합니다.

```
ceph osd set nobackfill
ceph osd set norecover
ceph osd set norebalance
ceph osd set noscrub
ceph osd set nodeep-scrub
```

3. 노드의 모든 **OSD** 프로세스를 정상적으로 종료합니다.

```
[root@mon ~]# systemctl stop ceph-osd.target
```

4. **Red Hat Ceph Storage 3**을 사용하는 경우 **Red Hat Ceph Storage 3** 리포지토리를 비활성화합니다.

- a. **tools** 리포지토리를 비활성화합니다.

```
[root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
```

- b. **osd** 저장소를 비활성화합니다.

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

5. **Red Hat Ceph Storage 4**를 사용하는 경우 **Red Hat Ceph Storage 4** 리포지토리를 비활성화합니다.
  - a. **tools** 리포지토리를 비활성화합니다.  

```
[root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
```
  - b. **osd** 저장소를 비활성화합니다.  

```
[root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-osd-rpms
```
6. **st app** 유틸리티를 설치합니다. [Red Hat Enterprise Linux 7에서 Red Hat Enterprise Linux 8로 업그레이드를 참조하십시오.](#)
7. **stapp preupgrade** 검사를 통해 실행합니다. 명령줄의 [Assessing upgradability](#)를 참조하십시오.
8. `/etc/ssh/sshd_config` 에서 **PermitRootLogin yes** 를 설정합니다.
9. **OpenSSH SSH** 데몬을 다시 시작합니다.  

```
[root@mon ~]# systemctl restart sshd.service
```
10. **Linux** 커널에서 **iSCSI** 모듈을 제거합니다.  

```
[root@mon ~]# modprobe -r iscsi
```
11. [RHEL 7에서 RHEL 8로의 업그레이드 수행에 따라 업그레이드를 수행합니다.](#)
12. 노드를 재부팅합니다.
13. **Red Hat Enterprise Linux 8용 Red Hat Ceph Storage 4용** 리포지토리를 활성화합니다.

- a. **툴 리포지토리를 활성화합니다.**

```
[root@mon ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

- b. **osd 리포지토리를 활성화합니다.**

```
[root@mon ~]# subscription-manager repos --enable=rhceph-4-osd-for-rhel-8-x86_64-rpms
```

14. **ceph-osd 패키지를 설치합니다.**

```
[root@mon ~]# dnf install ceph-osd
```

15. **leveldb 패키지를 설치합니다.**

```
[root@mon ~]# dnf install leveldb
```

16. 아직 업그레이드되지 않은 노드 또는 이미 해당 파일이 복원된 노드에서 **ceph.conf** 파일을 복원합니다.

17. **noout,nobackfill,norecover,norrebalance,noscrub** 및 **nodeep-scrub** 플래그를 설정 해제합니다.

```
# ceph osd unset noout
# ceph osd unset nobackfill
# ceph osd unset norecover
# ceph osd unset norebalance
# ceph osd unset noscrub
# ceph osd unset nodeep-scrub
```

18. 기존 **NetNamespace** 버킷을 최신 버킷 유형 **straw2** 로 전환합니다.

```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

19. 선택 사항: OSD가 **ceph-disk** 를 사용하여 생성되었으며 여전히 **ceph-disk** 로 관리되는 경우 **ceph-volume** 을 사용하여 관리해야 합니다.

- a. *각 오브젝트 스토리지 장치를 마운트합니다.*

구문

```
/dev/DRIVE /var/lib/ceph/osd/ceph-OSD_ID
```

*DRIVE* 를 스토리지 장치 이름 및 파티션 번호로 바꿉니다.

*OSD\_ID* 를 *OSD ID*로 교체합니다.

예제

```
[root@mon ~]# mount /dev/sdb1 /var/lib/ceph/osd/ceph-0
```

*ID\_NUMBER* 가 올바른지 확인합니다.

구문

```
cat /var/lib/ceph/osd/ceph-OSD_ID/whoami
```

*OSD\_ID* 를 *OSD ID*로 교체합니다.

예제

```
[root@mon ~]# cat /var/lib/ceph/osd/ceph-0/whoami
0
```

추가 오브젝트 저장소 장치에 대해 위의 단계를 반복합니다.

- b. 새로 마운트된 장치를 스캔합니다.

구문

```
ceph-volume simple scan /var/lib/ceph/osd/ceph-OSD_ID
```

**OSD\_ID** 를 **OSD ID**로 교체합니다.

예제

```
[root@mon ~]# ceph-volume simple scan /var/lib/ceph/osd/ceph-0
stderr: lsblk: /var/lib/ceph/osd/ceph-0: not a block device
stderr: lsblk: /var/lib/ceph/osd/ceph-0: not a block device
stderr: Unknown device, --name=, --path=, or absolute path in /dev/ or /sys expected.
Running command: /usr/sbin/cryptsetup status /dev/sdb1
--> OSD 0 got scanned and metadata persisted to file: /etc/ceph/osd/0-0c9917f7-fce8-42aa-bdec-8c2cf2d536ba.json
--> To take over management of this scanned OSD, and disable ceph-disk and udev, run:
--> ceph-volume simple activate 0 0c9917f7-fce8-42aa-bdec-8c2cf2d536ba
```

추가 오브젝트 저장소 장치에 대해 위의 단계를 반복합니다.

c.

장치를 활성화합니다.

구문

```
ceph-volume simple activate OSD_ID UUID
```

**OSD\_ID** 를 **OSD ID** 및 **UUID** 로 바꾸고 이전 스캔 출력에 표시된 **UUID**로 바꿉니다.

예제

```
[root@mon ~]# ceph-volume simple activate 0 0c9917f7-fce8-42aa-bdec-8c2cf2d536ba
Running command: /usr/bin/ln -snf /dev/sdb2 /var/lib/ceph/osd/ceph-0/journal
Running command: /usr/bin/chown -R ceph:ceph /dev/sdb2
Running command: /usr/bin/systemctl enable ceph-volume@simple-0-0c9917f7-fce8-42aa-bdec-8c2cf2d536ba
stderr: Created symlink /etc/systemd/system/multi-user.target.wants/ceph-volume@simple-0-0c9917f7-fce8-42aa-bdec-8c2cf2d536ba.service → /usr/lib/systemd/system/ceph-volume@.service.
Running command: /usr/bin/ln -sf /dev/null /etc/systemd/system/ceph-disk@.service
--> All ceph-disk systemd units have been disabled to prevent OSDs getting triggered by UDEV events
Running command: /usr/bin/systemctl enable --runtime ceph-osd@0
stderr: Created symlink /run/systemd/system/ceph-osd.target.wants/ceph-osd@0.service → /usr/lib/systemd/system/ceph-osd@.service.
Running command: /usr/bin/systemctl start ceph-osd@0
--> Successfully activated OSD 0 with FSID 0c9917f7-fce8-42aa-bdec-8c2cf2d536ba
```

추가 오브젝트 저장소 장치에 대해 위의 단계를 반복합니다.

20.

선택 사항: **OSD**가 **ceph-volume** 으로 생성되었으며 이전 단계를 완료하지 않은 경우 **OSD** 서비스를 지금 시작합니다.

```
[root@mon ~]# systemctl start ceph-osd.target
```



21.

**OSD를 활성화합니다.****BlueStore**

```
[root@mon ~]# ceph-volume lvm activate --all
```

22.

**OSD가 up 및 인지, active+clean 상태인지 확인합니다.**

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

**osd: 아래의 행에서 :, 모든 OSD가 실행 중인지 확인합니다.****예제**

```
osd: 3 osds: 3 up (since 8s), 3 in (since 3M)
```

23.

**모두 업그레이드될 때까지 모든 OSD 노드에서 위의 단계를 반복합니다.**

24.

**Red Hat Ceph Storage 3에서 업그레이드하는 경우 사전Nautilus OSD를 허용하지 않고 broken-only 기능을 활성화합니다.**

```
[root@mon ~]# ceph osd require-osd-release nautilus
```

**참고**

이 단계를 실행하지 않으면 msgsrv2가 활성화된 후 OSD가 통신할 수 없습니다.

25.

**Red Hat Ceph Storage 3에서 Red Hat Ceph Storage 4로 업그레이드한 후 모든 데몬이 업데이트되면 다음 단계를 실행하십시오.**

a.

**messenger v2 프로토콜, msgr2 를 활성화합니다.**

```
[root@mon ~]# ceph mon enable-msgr2
```

그러면 이전 기본 포트 **6789**에 바인딩하는 모든 **Ceph** 모니터가 **3300**의 새 포트에 바인딩됩니다.

b.

모든 노드에서 **ceph.conf** 파일의 구성 옵션을 스토리지 클러스터의 구성 데이터베이스로 가져옵니다.

예제

```
[root@mon ~]# ceph config assimilate-conf -i /etc/ceph/ceph.conf
```



참고

예를 들어 동일한 옵션 세트에 대해 다른 구성 값이 있는 경우 최종 결과는 파일이 동화되는 순서에 따라 달라집니다.

c.

스토리지 클러스터의 구성 데이터베이스를 확인합니다.

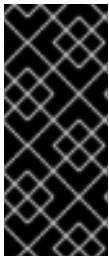
예제

```
[root@mon ~]# ceph config dump
```

- 자세한 내용은 설치 가이드의 **Red Hat Ceph Storage 클러스터 및 운영 체제 수동 업그레이드**를 참조하십시오.
- 자세한 내용은 **Red Hat Enterprise Linux 7에서 Red Hat Enterprise Linux 8으로 업그레이드**를 참조하십시오.

#### 8.4. CEPH OBJECT GATEWAY 노드 및 해당 운영 체제를 수동으로 업그레이드

시스템 관리자는 **Red Hat Ceph Storage 클러스터 노드와 Red Hat Enterprise Linux 운영 체제**를 새로운 주요 릴리스로 수동으로 업그레이드할 수 있습니다.



##### 중요

**Ceph 클러스터의 각 RGW 노드에 대해 이 절차를 수행해야 하지만, 한 번에 하나의 RGW 노드만 수행해야 합니다. 클라이언트 액세스 문제가 발생하지 않도록 다음 노드를 진행하기 전에 현재 업그레이드된 RGW를 일반 작업으로 반환했는지 확인합니다.**

##### 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage 클러스터**.
- 노드는 **Red Hat Enterprise Linux 7.9**를 실행하고 있습니다.
- 노드는 **Red Hat Ceph Storage 버전 3.3z6 또는 4.1**을 사용합니다.
- **Red Hat Enterprise Linux 8.3**의 설치 소스에 액세스합니다.

##### 절차

1. **Ceph Object Gateway 서비스를 중지합니다.**

```
# systemctl stop ceph-radosgw.target
```

2. **Red Hat Ceph Storage 3을 사용하는 경우 Red Hat Ceph Storage 3 툴 리포지토리를 비활성화합니다.**

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

3.

**Red Hat Ceph Storage 4**를 사용하는 경우 **Red Hat Ceph Storage 4** 툴 리포지토리를 비활성화합니다.

```
# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
```

4.

**st app** 유틸리티를 설치합니다. **Red Hat Enterprise Linux 7**에서 **Red Hat Enterprise Linux 8**로 업그레이드를 참조하십시오.

5.

**stapp preupgrade** 검사를 통해 실행합니다. 명령줄의 **Assessing upgradability**를 참조하십시오.

6.

`/etc/ssh/sshd_config` 에서 **PermitRootLogin yes** 를 설정합니다.

7.

**OpenSSH SSH** 데몬을 다시 시작합니다.

```
# systemctl restart sshd.service
```

8.

**Linux** 커널에서 **iSCSI** 모듈을 제거합니다.

```
# modprobe -r iscsi
```

9.

**RHEL 7**에서 **RHEL 8**로의 업그레이드 수행에 따라 업그레이드를 수행합니다.

10.

노드를 재부팅합니다.

11.

**Red Hat Enterprise Linux 8**용 **Red Hat Ceph Storage 4**용 툴 리포지토리를 활성화합니다.

```
# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

12.

**ceph-radosgw** 패키지를 설치합니다.

-

```
# dnf install ceph-radosgw
```

13.

선택 사항: 이 노드에 공동 배치된 모든 **Ceph** 서비스에 대한 패키지를 설치합니다. 필요한 경우 추가 **Ceph** 리포지토리를 활성화합니다.

14.

선택 사항: 다른 **Ceph** 서비스에 필요한 **leveldb** 패키지를 설치합니다.

```
# dnf install leveldb
```

15.

아직 업그레이드되지 않았거나 이미 이러한 파일이 복원된 노드에서 **ceph-client-admin.keyring** 및 **ceph.conf** 파일을 복원하십시오.

16.

**RGW** 서비스를 시작합니다.

```
# systemctl start ceph-radosgw.target
```

17.

기존 **NetNamespace** 버킷을 최신 버킷 유형 **straw2** 로 전환합니다.

```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

18.

데몬이 활성 상태인지 확인합니다.

```
# ceph -s
```

서비스 아래에는 **rgw:** 행이 있습니다. .

예제

```
rgw: 1 daemon active (jb-ceph4-rgw.rgw0)
```

19.

모두 업그레이드될 때까지 모든 **Ceph Object Gateway** 노드에서 위의 단계를 반복합니다.

#### 추가 리소스

- 자세한 내용은 설치 가이드의 **Red Hat Ceph Storage** 클러스터 및 운영 체제 수동 업그레이드를 참조하십시오.
- 자세한 내용은 **Red Hat Enterprise Linux 7**에서 **Red Hat Enterprise Linux 8**으로 업그레이드를 참조하십시오.

### 8.5. CEPH 대시보드 노드 및 운영 체제를 수동으로 업그레이드

시스템 관리자는 **Red Hat Ceph Storage** 클러스터 노드와 **Red Hat Enterprise Linux** 운영 체제를 새로운 주요 릴리스로 수동으로 업그레이드할 수 있습니다.

#### 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드는 **Red Hat Enterprise Linux 7.9**를 실행하고 있습니다.
- 노드는 **Red Hat Ceph Storage** 버전 **3.3z6** 또는 **4.1**을 실행하고 있습니다.
- **Red Hat Enterprise Linux 8.3**의 설치 소스에 액세스합니다.

#### 절차

1. 클러스터에서 기존 대시보드를 설치 제거합니다.
  - a. `/usr/share/cephmetrics-ansible` 디렉토리로 변경합니다.

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

b.

**purge.yml Ansible 플레이북을 실행합니다.**

```
# ansible-playbook -v purge.yml
```

2.

**Red Hat Ceph Storage 3을 사용하는 경우 Red Hat Ceph Storage 3 툴 리포지토리를 비활성화합니다.**

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

3.

**Red Hat Ceph Storage 4를 사용하는 경우 Red Hat Ceph Storage 4 툴 리포지토리를 비활성화합니다.**

```
# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
```

4.

**st app** 유틸리티를 설치합니다. [Red Hat Enterprise Linux 7에서 Red Hat Enterprise Linux 8로 업그레이드를 참조하십시오.](#)

5.

**stapp preupgrade** 검사를 통해 실행합니다. [명령줄의 Assessing upgradability를 참조하십시오.](#)

6.

**/etc/ssh/sshd\_config** 에서 **PermitRootLogin yes** 를 설정합니다.

7.

**OpenSSH SSH** 데몬을 다시 시작합니다.

```
# systemctl restart sshd.service
```

8.

**Linux** 커널에서 **iSCSI** 모듈을 제거합니다.

```
# modprobe -r iscsi
```

9.

[RHEL 7에서 RHEL 8로의 업그레이드 수행에 따라 업그레이드를 수행합니다.](#)

10.

노드를 재부팅합니다.

11.

**Red Hat Enterprise Linux 8용 Red Hat Ceph Storage 4용 툴 리포지토리를 활성화합니다.**

```
# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

12.

**Ansible 리포지토리를 활성화합니다.**

```
# subscription-manager repos --enable=ansible-2.9-for-rhel-8-x86_64-rpms
```

13.

클러스터를 관리하도록 **ceph-anible**을 구성합니다. 대시보드를 설치합니다. 사전 요구 사항을 포함하여 **Ansible**을 사용하여 **Red Hat Ceph Storage** 설치의 지침을 따르십시오.

14.

위의 절차의 일부로 **ansible-playbook site.yml** 을 실행하면 대시보드의 **URL**이 출력됩니다. **URL** 검색 및 대시보드에 액세스하는 방법에 대한 자세한 내용은 대시보드 가이드의 **Ansible**을 사용하여 대시보드 설치를 참조하십시오.

#### 추가 리소스

- 자세한 내용은 설치 가이드의 **Red Hat Ceph Storage** 클러스터 및 운영 체제 수동 업그레이드를 참조하십시오.
- 자세한 내용은 **Red Hat Enterprise Linux 7**에서 **Red Hat Enterprise Linux 8**으로 업그레이드를 참조하십시오.
- 자세한 내용은 대시보드 가이드에서 **Ansible**을 사용하여 대시보드 설치를 참조하십시오.

#### 8.6. CEPH ANSIBLE 노드 수동 업그레이드 및 설정 재구성

**Red Hat Ceph Storage** 클러스터 노드 및 **Red Hat Enterprise Linux** 운영 체제에서 **Ceph Ansible** 소프트웨어를 새 주요 릴리스로 수동으로 업그레이드합니다. 이 절차는 지정하지 않는 한 베어 메탈 및 컨테이너 배포에 모두 적용됩니다.



#### 중요

**Ceph Ansible** 노드에서 **hostOS**를 업그레이드하기 전에 **group\_vars** 및 **hosts** 파일의 백업을 수행합니다. **Ceph Ansible** 노드를 재구성하기 전에 생성된 백업을 사용합니다.



## 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드는 **Red Hat Enterprise Linux 7.9**를 실행하고 있습니다.
- 노드는 **Red Hat Ceph Storage** 버전 **3.3z6** 또는 **4.1**을 실행하고 있습니다.
- **Red Hat Enterprise Linux 8.3**의 설치 소스에 액세스합니다.

## 절차

1. **Red Hat Enterprise Linux 8용 Red Hat Ceph Storage 4용** 툴 리포지토리를 활성화합니다.

```
[root@dashboard ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

2. **Ansible** 리포지토리를 활성화합니다.

```
[root@dashboard ~]# subscription-manager repos --enable=ansible-2.9-for-rhel-8-x86_64-rpms
```

3. 스토리지 클러스터를 관리하도록 **ceph-ansible** 을 구성합니다. 대시보드를 설치합니다. 사전 요구 사항을 포함하여 **Ansible을 사용하여 Red Hat Ceph Storage** 설치의 지침을 따르십시오.

4. 위의 절차의 일부로 **ansible-playbook site.yml** 을 실행하면 대시보드의 **URL**이 출력됩니다. **URL** 검색 및 대시보드에 액세스하는 방법에 대한 자세한 내용은 대시보드 가이드의 **Ansible을 사용하여** 대시보드 설치를 참조하십시오.

## 추가 리소스

- 자세한 내용은 설치 가이드의 **Red Hat Ceph Storage 클러스터 및 운영 체제 수동 업그레이드**를 참조하십시오.
- 자세한 내용은 **Red Hat Enterprise Linux 7에서 Red Hat Enterprise Linux 8으로 업그레이드**를 참조하십시오.

- 자세한 내용은 [대시보드 가이드](#)에서 **Ansible**을 사용하여 대시보드 설치를 참조하십시오.

### 8.7. CEPH FILE SYSTEM 메타데이터 서버 노드 및 운영 체제를 수동으로 업그레이드

**Red Hat Ceph Storage** 클러스터와 **Red Hat Enterprise Linux** 운영 체제를 새로운 메이저 릴리스로 수동으로 **Ceph File System (CephFS)** 메타데이터 서버(MDS) 소프트웨어를 업그레이드할 수 있습니다.



#### 중요

스토리지 클러스터를 업그레이드하기 전에 활성 **MDS** 순위 수를 파일 시스템별로 하나씩 줄입니다. 이렇게 하면 여러 **MDS** 간에 가능한 모든 버전 충돌이 발생하지 않습니다. 또한 업그레이드하기 전에 모든 **edge** 노드가 오프라인 상태로 전환하십시오.

**MDS** 클러스터에는 버전 관리 또는 파일 시스템 플래그가 내장되어 있지 않기 때문입니다. 이러한 기능을 사용하지 않으면 여러 **MDS**가 다른 버전의 **MDS** 소프트웨어를 사용하여 통신할 수 있으며 어설션 또는 기타 오류가 발생할 수 있습니다.

#### 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드는 **Red Hat Enterprise Linux 7.9**를 실행하고 있습니다.
- 노드는 **Red Hat Ceph Storage** 버전 **3.3z6** 또는 **4.1**을 사용합니다.
- **Red Hat Enterprise Linux 8.3**의 설치 소스에 액세스합니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

중요

기본 XFS 파일 시스템은 **ftype=1** 또는 **d\_type** 지원을 사용하여 포맷해야 합니다. **xfs\_info /var** 명령을 실행하여 **ftype** 이 1로 설정되어 있는지 확인합니다. **ftype** 값이 1이 아닌 경우 새 디스크를 연결하거나 볼륨을 만듭니다. 이 새 장치 상단에서 새 XFS 파일 시스템을 생성하여 **/var/lib/containers** 에 마운트합니다.

Red Hat Enterprise Linux 8부터 **mkfs.xfs** 는 기본적으로 **ftype=1** 을 활성화합니다.

## 절차

1. 활성 MDS 순위 수를 1로 줄입니다.

## 구문

```
ceph fs set FILE_SYSTEM_NAME max_mds 1
```

## 예제

```
[root@mds ~]# ceph fs set fs1 max_mds 1
```

2. 클러스터가 모든 MDS 순위를 중지할 때까지 기다립니다. 모든 MDS가 중지되면 순위 0만 활성화되어야 합니다. 나머지는 **Wait** 모드여야 합니다. 파일 시스템의 상태를 확인합니다.

```
[root@mds ~]# ceph status
```

3. **systemctl** 을 사용하여 모든 MDS를 오프라인으로 전환합니다.

```
[root@mds ~]# systemctl stop ceph-mds.target
```

4. 하나의 **MDS**만 온라인 상태인지 확인하고 파일 시스템의 순위 **0**이 있는지 확인합니다.

```
[root@mds ~]# ceph status
```

5. 운영 체제 버전의 툴 리포지토리를 비활성화합니다.

- a. **RHEL 7의 Red Hat Ceph Storage 3**에서 업그레이드하는 경우 **Red Hat Ceph Storage 3** 툴 리포지토리를 비활성화합니다.

```
[root@mds ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
```

- b. **Red Hat Ceph Storage 4**를 사용하는 경우 **Red Hat Ceph Storage 4** 툴 리포지토리를 비활성화합니다.

```
[root@mds ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
```

6. **st app** 유틸리티를 설치합니다. **become p**에 대한 자세한 내용은 [Red Hat Enterprise Linux 7에서 Red Hat Enterprise Linux 8로 업그레이드를 참조하십시오.](#)

7. **stapp preupgrade** 검사를 통해 실행합니다. 자세한 내용은 [명령줄에서 Upgradability를 참조하십시오.](#)

8. **/etc/ssh/sshd\_config** 를 편집하고 **PermitRootLogin** 을 **yes** 로 설정합니다.

9. **OpenSSH SSH** 데몬을 다시 시작합니다.

```
[root@mds ~]# systemctl restart sshd.service
```

10. **Linux** 커널에서 **iSCSI** 모듈을 제거합니다.

```
[root@mds ~]# modprobe -r iscsi
```

11. 업그레이드를 수행합니다. **RHEL 7**에서 **RHEL 8**로의 업그레이드 수행을 참조하십시오.

12.

**MDS** 노드를 재부팅합니다.

13.

**Red Hat Enterprise Linux 8용 Red Hat Ceph Storage 4용** 틀 리포지토리를 활성화합니다.

```
[root@mds ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

14.

**ceph-mds** 패키지를 설치합니다.

```
[root@mds ~]# dnf install ceph-mds -y
```

15.

**선택 사항:** 이 노드에 공동 배치된 모든 **Ceph** 서비스에 대한 패키지를 설치합니다. 필요한 경우 추가 **Ceph** 리포지토리를 활성화합니다.

16.

**선택 사항:** 다른 **Ceph** 서비스에 필요한 **leveldb** 패키지를 설치합니다.

```
[root@mds ~]# dnf install leveldb
```

17.

아직 업그레이드되지 않은 노드에서 **ceph-client-admin.keyring** 및 **ceph.conf** 파일을 복원하거나 이미 해당 파일이 복원된 노드에서 복원하십시오.

18.

기존 **NetNamespace** 버킷을 최신 버킷 유형 **straw2** 로 전환합니다.

```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

19.

**MDS** 서비스를 시작합니다.

```
[root@mds ~]# systemctl restart ceph-mds.target
```

20.

데몬이 활성 상태인지 확인합니다.

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

21.

대기 데몬에 대해 동일한 프로세스를 따릅니다.

22.

**Wait**에서 모든 **MDS**를 다시 시작한 후 클러스터의 **max\_mds** 값을 복원하십시오.

구문

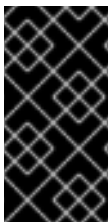
```
ceph fs set FILE_SYSTEM_NAME max_mds ORIGINAL_VALUE
```

예제

```
[root@mds ~]# ceph fs set fs1 max_mds 5
```

## 8.8. OSD 노드에서 운영 체제 업그레이드 실패에서 복구

시스템 관리자가 수동으로 **Ceph OSD** 노드 및 해당 운영 체제를 업그레이드 할 때 오류가 발생하면 다음 절차를 사용하여 오류를 복구할 수 있습니다. 이 절차에서는 노드에 **Red Hat Enterprise Linux 8.4**를 새로 설치하고, 노드가 삭제된 동안 **OSD**에 쓰기 외에 데이터를 다시 채우지 않고 **OSD**를 복구할 수 있습니다.



중요

**OSD**를 백업한 미디어 또는 해당 **wal.db** 또는 **block.db** 데이터베이스를 사용하지 마십시오.

### 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 업그레이드에 실패한 **OSD** 노드입니다.
- **Red Hat Enterprise Linux 8.4**의 설치 소스에 액세스합니다.

## 절차

1. 오류가 발생한 노드에서 **Red Hat Enterprise Linux 8.4**의 표준 설치를 수행하고 **Red Hat Enterprise Linux** 리포지토리를 활성화합니다.

- **표준 RHEL 설치 수행**

2. **Red Hat Enterprise Linux 8용 Red Hat Ceph Storage 4용 리포지토리를 활성화합니다.**

- a. **툴 리포지토리를 활성화합니다.**

```
# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

- b. **osd 리포지토리를 활성화합니다.**

```
# subscription-manager repos --enable=rhceph-4-osd-for-rhel-8-x86_64-rpms
```

3. **ceph-osd 패키지를 설치합니다.**

```
# dnf install ceph-osd
```

4. 아직 업그레이드되지 않았거나 해당 파일이 복원된 노드에서 **ceph.conf** 파일을 **/etc/ceph** 로 복원합니다.

5. **OSD 서비스를 시작합니다.**

```
# systemctl start ceph-osd.target
```

6. **오브젝트 저장소 장치를 활성화합니다.**

```
ceph-volume lvm activate --all
```

7. **OSD 및 클러스터 백필 쓰기가 복구된 OSD에 대한 쓰기를 확인합니다.**

```
# ceph -w
```

■

모든 PG가 **active+clean** 상태에 있을 때까지 출력을 모니터링합니다.

#### 추가 리소스

- 자세한 내용은 설치 가이드의 **Red Hat Ceph Storage** 클러스터 및 운영 체제 수동 업그레이드를 참조하십시오.
- 자세한 내용은 **Red Hat Enterprise Linux 7**에서 **Red Hat Enterprise Linux 8**으로 업그레이드를 참조하십시오.

#### 8.9. 추가 리소스

- 운영 체제를 새로운 주요 릴리스로 업그레이드할 필요가 없는 경우 **Red Hat Ceph Storage 클러스터** 업그레이드를 참조하십시오.



## 9장. 다음에 어떻게 해야 하나요?

이는 **Red Hat Ceph Storage**가 최신 데이터 센터의 스토리지 요구 사항을 충족하기 위해 수행할 수 있는 시작일 뿐입니다. 다양한 항목에 대한 자세한 내용은 다음과 같습니다. **Here are links to more information on a variety of topics:**

- 성능 벤치마킹 및 성능 카운터에 대한 액세스, **Red Hat Ceph Storage 4 관리 가이드의 성능** 장을 참조하십시오.
- 스냅샷 생성 및 관리, **Red Hat Ceph Storage 4용 블록 장치 가이드의 스냅샷** 장을 참조하십시오.
- **Red Hat Ceph Storage** 클러스터 확장은 **Red Hat Ceph Storage 4용 운영 가이드의 스토리지 클러스터 크기 관리** 장을 참조하십시오.
- **Ceph** 블록 장치 미러링은 **Red Hat Ceph Storage 4용 블록 장치 가이드의 블록 장치 미러링** 장을 참조하십시오.
- 프로세스 관리 장은 **Red Hat Ceph Storage 4용 관리 가이드의 프로세스 관리** 장을 참조하십시오.
- 조정 가능한 매개변수는 **Red Hat Ceph Storage 4의 구성 가이드** 를 참조하십시오.
- **Ceph**를 **OpenStack**의 백엔드 스토리지로 사용하는 경우 **Red Hat OpenStack Platform**용 스토리지 가이드의 **백엔드** 섹션을 참조하십시오.
- **Ceph** 대시보드를 사용하여 **Red Hat Ceph Storage** 클러스터의 상태 및 용량을 모니터링합니다. 자세한 내용은 **대시보드 가이드** 를 참조하십시오.

## 부록 A. 문제 해결

## A.1. ANSIBLE은 예상보다 적은 장치를 감지하므로 설치를 중지합니다.

**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
```

이는 다음을 의미합니다.

`/usr/share/ceph-ansible/group_vars/osds.yml` 파일에서 `osd_auto_discovery` 매개변수를 `true` 로 설정하면 **Ansible**에서 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 오류가 발생하면 **Ansible**에서 실패한 장치를 감지하지 못하고 전체 설치 프로세스를 중지합니다.

예:

1. **OSD** 노드 3개(`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 `host2, /dev/sdb` 및 `/dev/sdc`(이전의 `/dev/sdd`)에 두 개의 디스크만 사용되도록 예상합니다.
4. **Ansible**은 설치 프로세스를 중지하고 위의 오류 메시지를 반환합니다.

문제를 해결하려면 다음을 수행합니다.

`/etc/ansible/hosts` 파일에서 오류가 발생한 디스크(위 예의 `host 2`)를 사용하여 `OSD` 노드에서 사용하는 장치를 지정합니다.

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

자세한 내용은 [5장. Ansible을 사용하여 Red Hat Ceph Storage 설치](#) 을 참조하십시오.

## 부록 B. 명령줄 인터페이스를 사용하여 CEPH 소프트웨어 설치

스토리지 관리자는 **Red Hat Ceph Storage** 소프트웨어의 다양한 구성 요소를 수동으로 설치하도록 선택할 수 있습니다.

### B.1. 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

- **활성 + 클린 상태의 실행 중인 Ceph 스토리지 클러스터.**

#### 절차

1. 클라이언트 노드에서 **Red Hat Ceph Storage 4** 툴 리포지토리를 활성화합니다.

```
[root@gateway ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

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

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

3. 초기 모니터 노드에서 **Ceph** 구성 파일, 이 경우 **ceph.conf** 를 복사하고 관리 인증 키를 클라이언트 노드에 복사합니다.

## 구문

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

&lt;client\_host\_name& gt;을 클라이언트 노드의 호스트 이름으로 바꿉니다.

## B.2. RED HAT CEPH STORAGE 수동 설치

## 중요

**Red Hat**은 수동으로 배포된 클러스터를 지원하거나 테스트하지 않습니다. 따라서 **Red Hat**은 **Red Hat Ceph Storage 4**를 사용하여 새 클러스터를 배포하는 데 **Ansible**을 사용하는 것이 좋습니다. 자세한 내용은 [5장. Ansible을 사용하여 Red Hat Ceph Storage 설치](#)을 참조하십시오.

**YUM**과 같은 명령줄 유틸리티를 사용하여 수동으로 배포된 클러스터를 업그레이드할 수 있지만 **Red Hat**은 이 방법을 지원하거나 테스트하지 않습니다.

모든 **Ceph** 클러스터에는 모니터가 한 개 이상 필요하며 클러스터에 저장된 오브젝트의 복사본만큼 적어도 많은 **OSD**가 필요합니다. **Red Hat**은 프로덕션 환경에 세 개의 모니터를 사용하고 최소 **3**개의 **OSD**(오브젝트 스토리지 장치)를 사용하는 것이 좋습니다.

초기 모니터를 부트 스트랩하는 것은 **Ceph** 스토리지 클러스터를 배포의 첫 번째 단계입니다. **Ceph** 모

니터 배포는 다음과 같은 전체 클러스터에 대한 중요한 기준도 설정합니다.

- 풀의 복제본 수
- OSD당 배치 그룹 수
- 하트비트 간격
- 모든 인증 요구 사항

이러한 값은 기본적으로 설정되므로 프로덕션에 맞게 클러스터를 설정할 때 이를 아는 것이 유용합니다.

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

- 초기 [모니터 노드부트스트랩](#)
- [OSD\(오브젝트 스토리지 장치\) 노드추가](#)

### Bootstrapping 모니터링

모니터를 부트 스트랩하려면 **Ceph** 스토리지 클러스터를 확장하여 다음과 같은 데이터가 필요합니다.

#### 고유 식별자

파일 시스템 식별자(**fsid**)는 클러스터의 고유 식별자입니다. **fsid** 는 **Ceph** 스토리지 클러스터가 **Ceph** 파일 시스템에 기본적으로 사용될 때 원래 사용되었습니다. 이제 **Ceph**에서 기본 인터페이스, 블록 장치 및 오브젝트 스토리지 게이트웨이 인터페이스도 지원하므로 **fsid** 는 약간의 잘못된 사용자입니다.

#### 모니터 이름

클러스터 내의 각 모니터 인스턴스에는 고유한 이름이 있습니다. 일반적으로 **Ceph Monitor** 이름은 노드 이름입니다. **Red Hat**은 노드당 하나의 **Ceph Monitor**를 권장하며 **Ceph Monitor** 데몬을 사용하여 **Ceph OSD** 데몬을 함께 배치하지 않습니다. 짧은 노드 이름을 검색하려면 **hostname -s** 명령을 사용합니다.

## 모니터 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.
- 호스트 이름과 해당 **IP** 주소가 한 개 이상 있어야 합니다.

## 키 삭제 모니터링

모니터는 시크릿 키를 사용하여 서로 통신합니다. 모니터 보안 키를 사용하여 인증 키를 생성하고 초기 모니터를 부트스트래핑할 때 제공해야 합니다.

## 관리자 키링

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

관련 요구 사항은 **Ceph** 구성 파일을 생성하는 것을 의미하지 않습니다. 그러나 **Red Hat**은 **Ceph** 구성 파일을 생성하고 **fsid**, **mon initial members** 및 **mon** 호스트 설정으로 채우는 것이 좋습니다.

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

초기 모니터를 부트스트랩하려면 다음 단계를 수행합니다.

1. **Red Hat Ceph Storage 4 Monitor** 리포지토리를 활성화합니다.

```
[root@monitor ~]# subscription-manager repos --enable=rhceph-4-mon-for-rhel-8-x86_64-rpms
```

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

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

3.

루트 로서 `/etc/ceph/` 디렉터리에 **Ceph** 구성 파일을 만듭니다.

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

4.

루트 로서 클러스터의 고유 식별자를 생성하고 **Ceph** 구성 파일의 `[global]` 섹션에 고유 식별자를 추가합니다.

```
# 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.

루트 로서 초기 모니터를 **Ceph** 구성 파일에 추가합니다.

구문

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

예제

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

7.

`root` 로 초기 모니터의 IP 주소를 **Ceph** 구성 파일에 추가합니다.



## 구문

```
# echo "mon host = <ip-address>[,<ip-address>]" >> /etc/ceph/ceph.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 4**의 구성 가이드의 **바인딩** 섹션을 참조하십시오.

8.

**root** 로 클러스터의 인증 키를 생성하고 **Monitor** 시크릿 키를 생성합니다.

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

9.

**root** 사용자로 관리자 인증 키를 생성하고 **ceph.client.admin.keyring** 사용자를 생성하고 사용자를 인증 키에 추가합니다.

## 구문

```
# ceph-authtool --create-keyring /etc/ceph/ceph.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.

**root** 로 **ceph.client.admin.keyring** 키를 **ceph.mon.keyring** 에 추가합니다.

```
# 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. 초기 모니터의 노드 이름, IP 주소, fsid 를 사용하여 지정하고 이를 **/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 모니터 노드에서 **root** 로 기본 데이터 디렉토리를 생성합니다.

구문

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

예제

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

13.

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

구문

```
# ceph-mon --mkfs -i <monitor_host_name> --monmap /tmp/monmap --keyring  
/tmp/ceph.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 4**의 [구성 가이드](#)를 참조하십시오. 다음 **Ceph** 구성 파일의 예는 가장 일반적인 구성 설정 중 일부를 나열합니다.

#### 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** 로 완료된 파일을 생성합니다.

#### 구문

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

#### 예제

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

16.

루트로서 새로 생성된 디렉토리 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** 키링 파일은 각각 **glance** 및 **cinder** 가 소유해야 합니다. 예를 들어 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.

root 로 초기 모니터 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
```

18.

**root** 로 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 Monitors**를 추가하려면 **Red Hat Ceph Storage 4 관리 가이드**의 **모니터 추가** 섹션을 참조하십시오.

## OSD Bootstrapping

초기 모니터가 실행되면 **OSD**(오브젝트 스토리지 장치)를 추가할 수 있습니다. **OSD**가 충분한 **OSD**가 개체 복사본 수를 처리할 때까지 클러스터는 **활성 + 클린** 상태에 도달할 수 없습니다.

개체의 기본 복사본 수는 **3**입니다. 최소 **3**개의 **OSD** 노드가 필요합니다. 그러나 오브젝트 사본을 두 개만 사용하므로 **OSD** 노드를 두 개 추가하면 **Ceph** 구성 파일에서 **osd** 풀 기본 크기 및 **osd** 풀 기본 크기 설정을 업데이트합니다.

자세한 내용은 **Red Hat Ceph Storage 4용 구성 가이드**의 **OSD 구성 참조** 섹션을 참조하십시오.

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

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

1. **Red Hat Ceph Storage 4 OSD** 리포지토리를 활성화합니다.

```
[root@osd ~]# subscription-manager repos --enable=rhceph-4-osd-for-rhel-8-x86_64-rpms
```

2. 루트로서 **Ceph OSD** 노드에 **ceph-osd** 패키지를 설치합니다.

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

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

**OSD에 대한 UUID(Universally Unique Identifier)를 생성합니다.**

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

5.

**root 로 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**의 기본 디렉터리를 생성합니다.

## 구문

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

## 예제

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

7. 루트 로서 **OSD**로 사용할 드라이브를 준비하여 방금 만든 디렉터리에 마운트합니다. **Ceph** 데이터 및 저널용 파티션을 생성합니다. 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/ceph-<osd_id>
# echo "<path_to_partition> /var/lib/ceph/osd/ceph-<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 디렉토리를 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
```

9. 루트로 **OSD** 인증 키를 등록합니다.

구문

```
# ceph auth add osd.<osd_id> osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-
<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** 로 **OSD** 노드를 **ArgoCD** 맵에 추가합니다.

### 구문

```
# ceph osd crush add-bucket <host_name> host
```

### 예제

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

11.

**root** 로 **OSD** 노드를 기본 **ArgoCD** 트리 아래에 배치합니다.

### 구문

```
# ceph osd crush move <host_name> root=default
```

## 예제

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

12.

**root** 로 **OSD** 디스크를 **ArgoCD** 맵에 추가합니다.

## 구문

```
# ceph 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

**DASD** 맵을 분리하고 장치 목록에 **OSD**를 추가할 수도 있습니다. **OSD** 노드를 버킷으로 추가한 다음, 장치를 **OSD** 노드에서 항목으로 추가하고, **OSD**를 가중치를 할당하고, **ArgoCD** 맵을 다시 컴파일하고, **NetNamespace** 맵을 설정합니다. 자세한 내용은 **Red Hat Ceph Storage 4용 스토리지 전략 가이드**의 **ArgoCD 맵 편집** 섹션을 참조하십시오.

13.

**루트**로서 새로 생성된 디렉토리 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.

**OSD** 노드는 **Ceph** 스토리지 클러스터 구성에 있습니다. 그러나 **OSD** 데몬은 **down** 및 **in** 입니다. 데이터 수신을 시작하기 전에 새 **OSD**가 가동 중이어야 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 이 데몬이 시작되어 있습니다.

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 4**의 관리 가이드에 있는 **OSD 추가** 섹션을 참조하십시오.

### B.3. 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 않으면 **Ceph Manager**를 수동으로 설치할 수 있습니다. **Red Hat**은 **Ceph Manager**와 **Ceph Monitor** 데몬을 동일한 노드에 배치하는 것이 좋습니다.

사전 요구 사항

- 작동 중인 **Red Hat Ceph Storage** 클러스터

- **root 또는 sudo 액세스**
- **rhceph-4-mon-for-rhel-8-x86\_64-rpms 리포지토리 활성화**
- **방화벽을 사용하는 경우 공용 네트워크에서 포트 6800-7300 을 엽니다.**

## 절차

**ceph-mgr** 이 배포되거나 **root** 사용자 또는 **sudo** 유틸리티로 노드에서 다음 명령을 사용합니다.

1. **ceph-mgr** 패키지를 설치합니다.

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

2. **/var/lib/ceph/mgr/ceph-hostname/** 디렉터리를 생성합니다.

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

**ceph-mgr** 데몬이 배포되는 노드의 호스트 이름으로 **hostname** 을 바꿉니다. 예를 들면 다음과 같습니다.

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

3. 새로 생성된 디렉터리에서 **ceph-mgr** 데몬의 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
```

**ceph-mgr** 이 배포될 노드의 호스트 이름으로 호스트 이름을 바꿉니다. 예를 들면 다음과 같습니다.

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

- 7. **ceph-mgr** 데몬이 성공적으로 시작되었는지 확인합니다.

```
ceph -s
```

출력에는 **services:** 섹션 아래에 다음과 유사한 행이 포함됩니다.

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

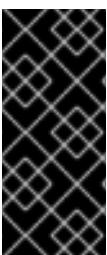
- 8. 더 많은 **ceph-mgr** 데몬을 설치하여 현재 활성화 데몬이 실패하는 경우 활성화 상태가 되는 데몬으로 서비스를 제공합니다.

추가 리소스

- [Red Hat Ceph Storage 설치 요구사항](#)

**B.4. 수동으로 CEPH 블록 장치 설치**

다음 절차에서는 썬 프로비저닝 및 크기 조정 가능한 **Ceph** 블록 장치를 설치하고 마운트하는 방법을 보여줍니다.



중요

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



## 사전 요구 사항

- **B.1절. “Ceph 명령줄 인터페이스 설치”** 섹션에 나열된 작업을 수행해야 합니다.
- **QEMU**를 사용하는 **VM(가상 머신)**의 백엔드로 **Ceph** 블록 장치를 사용하는 경우 기본 파일 설명자를 늘립니다. 자세한 내용은 **Ceph - VM이 RBD 디스크 지식베이스에 대량의 데이터를 전송할 때 중단되는 내용을 참조하십시오.**

## 절차

1. **OSD 노드의 파일에 대한 전체 권한이 있는 client.rbd** 라는 **Ceph** 블록 장치 사용자를 생성하고 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 있도록 허용하려는 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 4 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
```

**< image\_name >**, **< image\_size >**, **< pool\_name >**을 지정합니다. 예를 들면 다음과 같습니다.

```
$ rbd create image1 --size 4G --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(qcowd)** 클라이언트를 사용하는 경우 블록 장치 이미지를 매핑하지 못할 수 있습니다.

이 문제를 해결하려면 지원되지 않는 기능을 비활성화합니다. 이렇게 하려면 다음 옵션 중 하나를 사용합니다.

- 지원되지 않는 기능을 동적으로 비활성화합니다.

```
rbd feature disable <image_name> <feature_name>
```

예를 들어 다음과 같습니다.

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

- **`rbd create`** 명령과 함께 **`--image-feature`** 계층 **`ing`** 옵션을 사용하여 새로 생성된 블록 장치 이미지에서만 계층화 할 수 있습니다.
- **Ceph** 구성 파일에서 기능을 기본값으로 비활성화합니다.

```
rbd_default_features = 1
```

자세한 내용은 **Red Hat Ceph Storage 4** 릴리스 노트의 **알려진 문제** 장을 참조하십시오.

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

3.

새로 생성된 이미지를 블록 장치에 매핑합니다.

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

예를 들어 다음과 같습니다.

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

4.

파일 시스템을 생성하여 블록 장치를 사용합니다.

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

폴 이름과 이미지 이름을 지정합니다. 예를 들면 다음과 같습니다.

```
# mkfs.ext4 /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 들어 다음과 같습니다.

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

### 추가 리소스

- **Red Hat Ceph Storage 4용 블록 장치 가이드**입니다.

## B.5. 수동으로 CEPH OBJECT GATEWAY 설치

**RADOS 게이트웨이**라고도 하는 **Ceph 오브젝트 게이트웨이**는 **librados API** 상단에 구축된 오브젝트 스토리지 인터페이스로, **Ceph 스토리지 클러스터에 RESTful 게이트웨이**를 제공합니다.

### 사전 요구 사항

- **활성 + 클린 상태의 실행 중인 Ceph 스토리지 클러스터.**
- **3장. Red Hat Ceph Storage 설치 요구사항**에 나열된 작업을 수행합니다.

### 절차

1. **Red Hat Ceph Storage 4 툴 리포지토리를 활성화**합니다.

```
[root@gateway ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-debug-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 파일**을 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 새 **Object Gateway** 노드와 **Ceph** 스토리지 클러스터의 다른 모든 노드에 복사합니다.

구문

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

예제

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

c.

**ceph.client.admin.keyring** 파일을 새 **Object Gateway** 노드에 복사합니다.

구문

```
# scp /etc/ceph/ceph.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 만듭니다.

```
# 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/ceph-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 제공하는 경우 읽기 기능을 제공해야 합니다. 그러나 모니터 쓰기 기능을 제공하는 것은 선택 사항입니다. 제공하는 경우 **Ceph Object Gateway**에서 풀을 자동으로 생성할 수 있습니다.

이러한 경우 풀에 적절한 수의 배치 그룹을 지정해야 합니다. 그렇지 않으면 게이트웨이는 기본 번호를 사용하며 사용자의 요구에 가장 적합하지 않을 수 있습니다. 자세한 내용은 풀 계산기당 **Ceph PG(배치 그룹)** 를 참조하십시오.

6.

**Object Gateway** 노드에서 완료된 파일을 만듭니다.

```
# 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.

**Object Gateway** 노드에서 **TCP 포트 8080**을 엽니다.

```
# firewall-cmd --zone=public --add-port=8080/tcp
# firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

9.

**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
```

**Ceph Object Gateway**가 설치되면 모니터에 쓰기 기능이 설정된 경우 **Ceph Object Gateway**가 자동으로 풀을 생성합니다. 풀을 수동으로 생성하는 방법에 대한 자세한 내용은 스토리지 전략 가이드의 풀

장을 참조하십시오.

추가 리소스

- [Red Hat Ceph Storage 4 개체 게이트웨이 구성 및 관리 가이드](#)



## 부록 C. ANSIBLE 인벤토리 위치 구성

옵션으로 **ceph-ansible** 스테이징 및 프로덕션 환경에 대한 인벤토리 위치 파일을 구성할 수 있습니다.

## 사전 요구 사항

- **Ansible** 관리 노드.
- **Ansible** 관리 노드에 대한 루트 수준 액세스.
- **ceph-ansible** 패키지가 노드에 설치되어 있습니다.

## 절차

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

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

2. 스테이징 및 프로덕션에 사용할 하위 디렉토리를 만듭니다.

```
[ansible@admin ceph-ansible]$ mkdir -p inventory/staging inventory/production
```

3. **ansible.cfg** 파일을 편집하고 다음 행을 추가합니다.

```
[defaults]
inventory = ./inventory/staging # Assign a default inventory directory
```

4. 각 환경에 대한 인벤토리 **'hosts'** 파일을 생성합니다.

```
[ansible@admin ceph-ansible]$ touch inventory/staging/hosts
[ansible@admin ceph-ansible]$ touch inventory/production/hosts
```

- a. 각 호스트 파일을 열고 편집하고 **[mons]** 섹션에서 **Ceph Monitor** 노드를 추가합니다.

```
[mons]
MONITOR_NODE_NAME_1
```

```
MONITOR_NODE_NAME_1
MONITOR_NODE_NAME_1
```

### 예제

```
[mons]
mon-stage-node1
mon-stage-node2
mon-stage-node3
```



### 참고

기본적으로 플레이북은 스테이징 환경에서 실행됩니다. 프로덕션 환경에서 플레이북을 실행하려면 다음을 수행합니다.

```
[ansible@admin ceph-ansible]$ ansible-playbook -i inventory/production
playbook.yml
```

### 추가 리소스

- **ceph-ansible** 패키지 설치에 대한 자세한 내용은 [Red Hat Storage Cluster 설치](#)를 참조하십시오.

## 부록 D. CEPH 기본 설정 덮어쓰기

**Ansible** 구성 파일에 달리 지정하지 않는 한 **Ceph**는 기본 설정을 사용합니다.

**Ansible**에서 **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.

**Ansible**은 **Ceph** 구성 파일과 동일한 섹션을 지원합니다. `[global]`, `[mon]`, `[osd]`, `[mds]`, `[rgw]` 등. 특정 **Ceph Object Gateway** 인스턴스와 같은 특정 인스턴스를 재정의할 수도 있습니다. 예를 들어 다음과 같습니다.

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

ceph_conf_overrides:
  client.rgw.server601.rgw1:
    rgw_enable_ops_log: true
    log_file: /var/log/ceph/ceph-rgw-rgw1.log
```

## 참고

`ceph_conf_overrides` 설정에서 변수를 키로 사용하지 마십시오. 특정 구성 값을 재정의하려는 섹션에 대한 호스트의 절대 레이블을 전달해야 합니다.

## 참고

**Ansible**에는 **Ceph** 구성 파일의 특정 섹션을 참조할 때 괄호가 포함되지 않습니다. 섹션 및 설정 이름은 콜론으로 종료됩니다.

## 중요

**Ceph** 구성 파일에 두 개의 충돌하는 클러스터 네트워크가 발생할 수 있으므로 **CONFIG OVERRIDE** 섹션에서 `cluster_network` 매개변수를 사용하여 클러스터 네트워크를 설정하지 마십시오.

클러스터 네트워크를 설정하려면 **CEPH CONFIGURATION** 섹션에서 `cluster_network` 매개 변수를 사용합니다. 자세한 내용은 [Red Hat Ceph Storage 설치 가이드](#)에 [Red Hat Ceph Storage 클러스터 설치를 참조하십시오](#).

## 부록 E. 기존 CEPH 클러스터를 ANSIBLE로 가져오기

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

1. 버전 1.3에서 버전 2로 수동으로 업그레이드한 후 관리 노드에 **Ansible**을 설치 및 구성합니다.

2. **Ansible** 관리 노드에 클러스터의 모든 **Ceph** 노드에 대한 암호 없는 **ssh** 액세스 권한이 있는지 확인합니다. 자세한 내용은 3.9절. “**Ansible에 대해 암호 없는 SSH 활성화**”를 참조하십시오.

3. 루트 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** 로 `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. `group_vars/all.yml` 에서 검색된 **fsid** 를 설정합니다.

8. `/etc/ansible/hosts` 에서 **Ceph** 호스트를 포함하도록 **Ansible** 인벤토리를 수정합니다. `[mons]` 섹션에 모니터를 추가하여 `[rgws]` 섹션의 `[osds]` 섹션 및 게이트웨이 아래에 **OSD**를 추가하여 **Ansible**에 대한 역할을 식별합니다.

9. `all.yml` 파일의 `[global]`, `[osd]`, `[mon]`, `[client]` 섹션에 사용된 원래 `ceph.conf` 옵션으로 `ceph_conf_overrides` 가 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/  
# ansible-playbook infrastructure-playbooks/take-over-existing-cluster.yml -u <username> -i  
hosts
```

## 부록 F. ANSIBLE에서 배포한 스토리지 클러스터 제거

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



## 주의

Ceph 스토리지 클러스터를 제거한 후 OSD의 모든 데이터가 영구적으로 손실됩니다.

## 사전 요구 사항

- **Ansible** 관리 노드에 대한 루트 수준 액세스.
- **ansible** 사용자 계정에 액세스합니다.
- 베어 메탈 배포의 경우:
  - `/usr/share/ceph-ansible/group-vars/osds.yml` 파일의 `osd_auto_discovery` 옵션이 `true` 로 설정된 경우 **Ansible**은 스토리지 클러스터를 제거하지 못합니다. 따라서 `osd_auto_discovery` 를 주석으로 처리하고 `osds.yml` 파일에 OSD 장치를 선언합니다.
- **ansible** 사용자 계정에서 `/var/log/ansible/ansible.log` 파일에 쓸 수 있는지 확인합니다.

## 절차

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

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

2. **ansible** 사용자로 `purge` 플레이북을 실행합니다.

a.

베어 메탈 배포의 경우 **purge-cluster.yml** 플레이북을 사용하여 **Ceph** 스토리지 클러스터를 삭제합니다.

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/purge-cluster.yml
```

b.

컨테이너 배포의 경우:

i.

**purge-docker-cluster.yml** 플레이북을 사용하여 **Ceph** 스토리지 클러스터를 삭제합니다.

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/purge-docker-cluster.yml
```



참고

이 **Playbook**은 **Ceph Ansible** 플레이북에서 생성한 모든 패키지, 컨테이너, 구성 파일 및 모든 데이터를 제거합니다.

ii.

기본값(/etc/ansible/hosts) 이외의 다른 인벤토리 파일을 지정하려면 **-i** 매개변수를 사용합니다.

구문

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/purge-docker-cluster.yml -i INVENTORY_FILE
```

대체 버전

**inventory** 파일의 경로를 사용하여 **inventory\_FILE**.

예제

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

iii.

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

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

iv.

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

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

#### 추가 리소스

- 자세한 내용은 [OSD Ansible 설정](#)을 참조하십시오.



## 부록 G. ANSIBLE을 사용하여 CEPH 대시보드 삭제

대시보드를 더 이상 설치하지 않으려면 **purge-dashboard.yml** 플레이북을 사용하여 대시보드를 제거합니다. 대시보드 또는 해당 구성 요소와 관련된 문제를 해결할 때 대시보드를 제거할 수도 있습니다.

### 사전 요구 사항

- **Red Hat Ceph Storage 4.3 이상.**
- **최신 버전의 Red Hat Ceph Storage와 함께 Ceph-ansible이 제공됩니다.**
- **스토리지 클러스터의 모든 노드에 대한 sudo 수준 액세스.**

### 절차

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

### 예제

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

3. **Ansible purge-dashboard.yml 플레이북을 실행하고 메시지가 표시되면 yes 를 입력하여 대시보드 삭제를 확인합니다.**

### 예제

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/purge-dashboard.yml -i hosts -vvvv
```

## 검증

- **ceph mgr services** 명령을 실행하여 대시보드가 더 이상 실행되지 않는지 확인합니다.

## 구문

```
ceph mgr services
```

대시보드 **URL**이 표시되지 않습니다.

## 추가 리소스

- 대시보드를 설치하려면 **Red Hat Ceph Storage 대시보드 가이드의 Ansible**을 사용하여 대시보드 설치를 참조하십시오.

## 부록 H. ANSIBLE-VAULT를 사용하여 ANSIBLE 암호 변수 암호화

**ansible-vault** 를 사용하여 암호를 저장하는 데 사용되는 **Ansible** 변수를 암호화할 수 있으므로 일반 텍스트로 읽을 수 없습니다. 예를 들어 **group\_vars/all.yml** 에서 **ceph\_docker\_registry\_username** 및 **ceph\_docker\_registry\_password** 변수는 서비스 계정 자격 증명 또는 고객 포털 자격 증명으로 설정할 수 있습니다. 서비스 계정은 공유하도록 설계되었지만 고객 포털 암호를 보호해야 합니다. **ceph\_docker\_registry\_password** 를 암호화하는 것 외에도 **dashboard\_admin\_password** 및 **grafana\_admin\_password** 를 암호화할 수도 있습니다.

### 사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ansible** 관리 노드에 액세스

### 절차

1. **Ansible** 관리 노드에 로그인합니다.
2. **/usr/share/ceph-ansible/** 디렉토리로 변경합니다.

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

3. **ansible-vault** 를 실행하고 새 자격 증명 모음 암호를 생성합니다.

### 예제

```
[admin@admin ceph-ansible]$ ansible-vault encrypt_string --stdin-name
'ceph_docker_registry_password_vault'
New Vault password:
```

4. 자격 증명 모음 암호를 다시 입력하여 확인합니다.

### 예제

```
[admin@admin ceph-ansible]$ ansible-vault encrypt_string --stdin-name
'ceph_docker_registry_password_vault'
New Vault password:
Confirm New Vault password:
```

5.

암호화할 암호를 입력한 다음 **CTRL+D**를 두 번 입력하여 항목을 완료합니다.

구문

```
ansible-vault encrypt_string --stdin-name 'ceph_docker_registry_password_vault'
New Vault password:
Confirm New Vault password:
Reading plaintext input from stdin. (ctrl-d to end input)
PASSWORD
```

**DASD**를 암호 로 교체하십시오.

예제

```
[admin@admin ceph-ansible]$ ansible-vault encrypt_string --stdin-name
'ceph_docker_registry_password_vault'
New Vault password:
Confirm New Vault password:
Reading plaintext input from stdin. (ctrl-d to end input)
SecurePassword
```

암호를 입력한 후 **Enter**를 입력하지 마십시오. 그렇지 않으면 암호화된 문자열에 암호의 일부로 새 행을 포함합니다.

6.

**ceph\_docker\_registry\_password\_vault: !vault |** 로 시작하는 출력을 기록한 후 다음 단계에서 사용될 몇 가지 숫자 행으로 끝납니다.

예제

```
[admin@admin ceph-ansible]$ ansible-vault encrypt_string --stdin-name
'ceph_docker_registry_password_vault'
New Vault password:
Confirm New Vault password:
Reading plaintext input from stdin. (ctrl-d to end input)
SecurePasswordceph_docker_registry_password_vault: !vault |
    $ANSIBLE_VAULT;1.1;AES256

3838363964616665613032666666332626438363439303738363763313264373530323761653
06234

3161386334616632653530383231316631636462363761660a3733383733346634343638653
56633

663839633230333036623337653839383536306234333465653635346364346436343364306
43438

6134306662646365370a3431353166333038306535656337363034666362613263613337666
13462
    39353365343137323163343937636464663534383234326531666139376561663532
Encryption successful
```

필요한 출력은 공백이나 새 줄 없이 암호가 즉시 시작됩니다.

7.

**group\_vars/all.yml** 편집을 위해 를 열고 위의 출력을 파일에 붙여넣습니다.

예제

```
ceph_docker_registry_password_vault: !vault |
    $ANSIBLE_VAULT;1.1;AES256

3838363964616665613032666666332626438363439303738363763313264373530323761653
06234

3161386334616632653530383231316631636462363761660a3733383733346634343638653
```

56633

66383963323033303662333765383938353630623433346565363534636434643634336430643438

6134306662646365370a343135316633303830653565633736303466636261326361333766613462

39353365343137323163343937636464663534383234326531666139376561663532

- 8. 다음과 같이 암호화된 암호 아래에 행을 추가합니다.

예제

```
ceph_docker_registry_password: "{{ ceph_docker_registry_password_vault }}"
```



참고

**Ansible**의 버그가 **Ansible** 변수에 직접 자격 증명 모음 값을 할당할 때 문자열 유형을 중단하는 버그로 인해 두 개의 변수를 사용해야 합니다.

- 9. **ansible-playbook** 을 실행할 때 **vault** 암호를 요청하도록 **Ansible**을 구성합니다.

- a. `/usr/share/ceph-ansible/ansible.cfg` 편집을 위해 를 열고 **[defaults]** 섹션에 다음 행을 추가합니다.

```
ask_vault_pass = True
```

- b. 필요한 경우 **ansible-playbook**을 실행할 때마다 **--ask-vault-pass** 를 전달할 수 있습니다.

예제

```
[admin@admin ceph-ansible]$ ansible-playbook -v site.yml --ask-vault-pass
```

10.

**site.yml** 또는 **site-container.yml** 을 다시 실행하여 암호화된 암호와 관련된 오류가 없는지 확인합니다.

예제

```
[admin@admin ceph-ansible]$ ansible-playbook -v site.yml -i hosts --ask-vault-pass
```

**-i hosts** 옵션은 **/etc/ansible/hosts** 의 기본 **Ansible** 인벤토리 위치를 사용하지 않는 경우에만 필요합니다.

추가 리소스

- [Red Hat Container Registry Authentication](#)의 서비스 계정 정보 참조

부록 I. 일반 ANSIBLE 설정

다음은 가장 일반적인 구성 가능한 Ansible 매개 변수입니다. 배포 방법에 따라 베어 메탈 또는 컨테이너의 두 가지 매개변수 세트가 있습니다.



참고

이는 사용 가능한 모든 Ansible 매개 변수의 전체 목록이 아닙니다.

베어 메탈 및 컨테이너 설정

**monitor\_interface**

Ceph 모니터 노드가 수신 대기하는 인터페이스입니다.

값

사용자 정의

필수 항목

있음

참고

monitor\_\* 매개변수 중 하나 이상에 값을 할당해야 합니다.

**monitor\_address**

Ceph 모니터 노드에서도 수신하는 주소입니다.

값

사용자 정의

필수 항목

있음

참고

monitor\_\* 매개변수 중 하나 이상에 값을 할당해야 합니다.



**monitor\_address\_block**

**Ceph** 공용 네트워크의 서브넷입니다.

값

사용자 정의

필수 항목

있음

참고

노드의 IP 주소가 알 수 없지만 서브넷이 알려진 경우 사용합니다. **monitor\_\*** 매개변수 중 하나 이상에 값을 할당해야 합니다.

**ip\_version**

값

ipv6

필수 항목

예. IPv6 주소 사용 시

**public\_network**

IPv6를 사용하는 경우 **Ceph** 공용 네트워크의 IP 주소 및 넷마스크 또는 해당 IPv6 주소.

값

사용자 정의

필수 항목

있음

참고

자세한 내용은 [Red Hat Ceph Storage](#)에 대한 네트워크 구성 확인을 참조하십시오.

**cluster\_network**

IPv6를 사용하는 경우 **Ceph 클러스터 네트워크의 IP 주소 및 넷마스크 또는 해당 IPv6 주소.**

값

사용자 정의

필수 항목

없음

참고

자세한 내용은 [Red Hat Ceph Storage에 대한 네트워크 구성 확인을 참조하십시오.](#)

### `configure_firewall`

**Ansible에서 적절한 방화벽 규칙을 구성하려고 합니다.**

값

**true 또는 false**

필수 항목

없음

### 베어 메탈-특정 설정

#### `ceph_origin`

값

리포지토리 또는 디스트로 또는 로컬

필수 항목

있음

참고

리포지토리 값은 새 리포지토리를 통해 **Ceph**가 설치됨을 의미합니다. **distro** 값은 별도의 리포지토리 파일이 추가되지 않으며 **Linux** 배포에 포함된 모든 **Ceph** 버전을 얻을 수 있음을 의미합니다. **local** 값은 **Ceph** 바이너리가 로컬 시스템에서 복사됨을 의미합니다.

#### `ceph_repository_type`

값

**CDN 또는 iso**

필수 항목

있음

**ceph\_rhcs\_version**

값

4

필수 항목

있음

**ceph\_rhcs\_iso\_path**

*ISO 이미지의 전체 경로입니다.*

값

*사용자 정의*

필수 항목

*예, ceph\_repository\_type 이 iso 로 설정된 경우 .*

**컨테이너- 특정 설정**

**ceph\_docker\_image**

값

*로컬 Docker 레지스트리를 사용하는 경우 rhceph/rhceph-4-rhel8 또는 cephimageinlocalreg.*

필수 항목

있음

**ceph\_docker\_image\_tag**

값

**rhceph/rhceph-4-rhel8** 의 최신 버전 또는 로컬 레지스트리 구성 중에 지정된 **customtag** 입니다.

필수 항목

있음

#### **containerized\_deployment**

값

**true**

필수 항목

있음

#### **ceph\_docker\_registry**

값

**registry.redhat.io**, 또는 **LOCAL\_FQDN\_NODE\_NAME** (로컬 Docker 레지스트리를 사용하는 경우).

필수 항목

있음

## 부록 J. OSD ANSIBLE 설정

다음은 가장 일반적인 구성 가능한 **OSD Ansible** 매개 변수입니다.

### **osd\_auto\_discovery**

**OSD**로 사용할 빈 장치를 자동으로 찾습니다.

값

**false**

필수 항목

없음

참고

장치에서는 사용할 수 없습니다. **purge-docker-cluster.yml** 또는 **purge-cluster.yml** 과 함께 사용할 수 없습니다. 이러한 플레이북을 사용하려면 **osd\_auto\_discovery** 를 주석 처리하고 장치를 사용하여 **OSD** 장치를 선언합니다.

### **devices**

**Ceph** 데이터가 저장되는 장치 목록입니다.

값

사용자 정의

필수 항목

예, 장치 목록을 지정하는 경우

참고

**osd\_auto\_discovery** 설정을 사용할 때 사용할 수 없습니다. **devices** 옵션을 사용하는 경우 **ceph-volume lvm** 배치 모드에서는 최적화된 **OSD** 구성을 생성합니다.

### **dmccrypt**

**OSD**를 암호화하려면 다음을 수행합니다.

값

**true**

필수 항목

없음

참고

기본값은 **false**입니다.

## **lvm\_volumes**

**FileStore** 또는 **BlueStore** 사전 목록입니다.

값

사용자 정의

필수 항목

예, **devices** 매개변수를 사용하여 스토리지 장치를 정의하지 않은 경우

참고

각 사전에는 데이터 **journal** 및 **data\_vg** 키가 포함되어야 합니다. 모든 논리 볼륨 또는 볼륨 그룹은 전체 경로가 아닌 이름이어야 합니다. 데이터 및 저널 키는 논리 볼륨(LV) 또는 파티션일 수 있지만 여러 데이터 LV에 저널을 사용하지 않습니다. **data\_vg** 키는 **data** LV를 포함하는 볼륨 그룹이어야 합니다. 필요한 경우 **journal\_vg** 키를 사용하여 저널 LV가 포함된 볼륨 그룹을 지정할 수 있습니다.

## **osds\_per\_device**

장치별로 생성할 OSD 수입니다.

값

사용자 정의

필수 항목

없음

참고

기본값은 **1**입니다.

## **osd\_objectstore**

**OSD의 Ceph 오브젝트 저장소 유형입니다.**

**값**

**bluestore 또는 filestore**

**필수 항목**

**없음**

**참고**

**기본값은 bluestore 입니다. 업그레이드에 필요합니다.**