



Red Hat Ceph Storage 5

설치 가이드

Red Hat Enterprise Linux에 Red Hat Ceph Storage 설치

Red Hat Ceph Storage 5 설치 가이드

Red Hat Enterprise Linux에 Red Hat Ceph Storage 설치

Legal Notice

Copyright © 2025 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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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.

Abstract

이 문서에서는 AMD64 및 Intel 64 아키텍처에서 실행되는 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 메시지에서 참조하십시오.

Table of Contents

1장. RED HAT CEPH STORAGE	4
2장. RED HAT CEPH STORAGE 고려 사항 및 권장 사항	6
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	11
2.5. OSD 호스트에서 RAID 컨트롤러 사용 고려 사항	12
2.6. CEPH를 실행할 때 LINUX 커널 튜닝 고려 사항	12
2.7. 코로케이션 작동 방식 및 이점	13
2.8. RED HAT CEPH STORAGE의 운영 체제 요구 사항	19
2.9. RED HAT CEPH STORAGE의 최소 하드웨어 고려 사항	20
2.10. 추가 리소스	22
3장. RED HAT CEPH STORAGE 설치	23
3.1. 사전 요구 사항	23
3.2. CEPHADM 유틸리티	24
3.3. CEPHADM의 작동 방식	25
3.4. CEPHADM-ANSIBLE PLAYBOOK	27
3.5. RED HAT CEPH STORAGE 노드를 CDN에 등록하고 서브스크립션 연결	28
3.6. ANSIBLE 인벤토리 위치 구성	32
3.7. RED HAT ENTERPRISE LINUX 9에서 ROOT 사용자로 SSH 로그인 활성화	35
3.8. SUDO 액세스를 사용하여 ANSIBLE 사용자 생성	37
3.9. SSH 구성	41
3.10. ANSIBLE에 대해 암호 없는 SSH 활성화	45
3.11. PREFLIGHT PLAYBOOK 실행	48
3.12. 새 스토리지 클러스터 부트스트랩	51
3.13. SSH 키 배포	90
3.14. CEPHADM 셸 시작	91
3.15. 클러스터 설치 확인	93
3.16. 호스트 추가	95
3.17. 호스트 레이블 지정	108
3.18. 모니터 서비스 추가	115
3.19. 관리 노드 설정	116
3.20. 관리자 서비스 추가	124
3.21. OSD 추가	126
3.22. 클라이언트 노드 배포	129
3.23. CEPH 스토리지 클러스터 제거	134
4장. CEPHADM-ANSIBLE 모듈을 사용하여 RED HAT CEPH STORAGE 클러스터 관리	138
4.1. CEPHADM-ANSIBLE 모듈	138
4.2. CEPHADM-ANSIBLE 모듈 옵션	139
4.3. CEPHADM_BOOTSTRAP 및 CEPHADM_REGISTRY_LOGIN 모듈을 사용하여 스토리지 클러스터 부트 스트랩	143
4.4. CEPH_ORCH_HOST 모듈을 사용하여 호스트 추가 또는 제거	147
4.5. CEPH_CONFIG 모듈을 사용하여 구성 옵션 설정	155
4.6. CEPH_ORCH_APPLY 모듈을 사용하여 서비스 사양 적용	158
4.7. CEPH_ORCH_DAEMON 모듈을 사용하여 CEPH 데몬 상태 관리	161
5장. 다음에 어떻게 해야 합니까?	165
부록 A. CEPH ANSIBLE과 CEPHADM 비교	166

부록 B. CEPHADM 명령	169
부록 C. CEPH 방화벽 포트	177

1장. RED HAT CEPH STORAGE

Red Hat Ceph Storage는 엔터프라이즈급 Ceph 스토리지 시스템 버전과 Ceph 관리 플랫폼, 배포 유틸리티 및 지원 서비스를 결합한 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.

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 Manager

Ceph Manager 데몬인 **ceph-mgr**은 Ceph Monitor 노드에서 실행되는 Ceph Monitor 데몬과 함께 존재하여 추가 서비스를 제공합니다. Ceph Manager는 Ceph Manager 모듈을 사용하여 다른 모니터링 및 관리 시스템을 위한 인터페이스를 제공합니다. Ceph Manager 데몬을 실행하려면 일반 스토리지 클러스터 작업을 수행해야 합니다.

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 노드로 실행할 수 있습니다. 이 노드 중 기본값은 3개이지만 프로덕션 스토리지 클러스터에서는 모드가 확장되지 않은 상태에서 더 나은 성능을 얻을 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 실행하고 비즈니스 요구 사항을 달성하는 데 매우 중요합니다.



참고

특정 사용 사례에 맞게 Red Hat Ceph Storage 클러스터를 계획하는 데 도움이 필요하십니까? 지원이 필요한 경우 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 오브젝트 스토리지를 제공하는 클라우드 플랫폼의 선도적인 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 있다. 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 RHEA 계층 구조를 생성할 때는 사용 사례와 성능 장단점에 비해 사용 가능한 비용을 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 3x와 비교하여 약 1.5x의 초기 데이터를 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 호스트에 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 스토리지 클러스터 내의 컨테이너에서 실행할 수 있습니다.

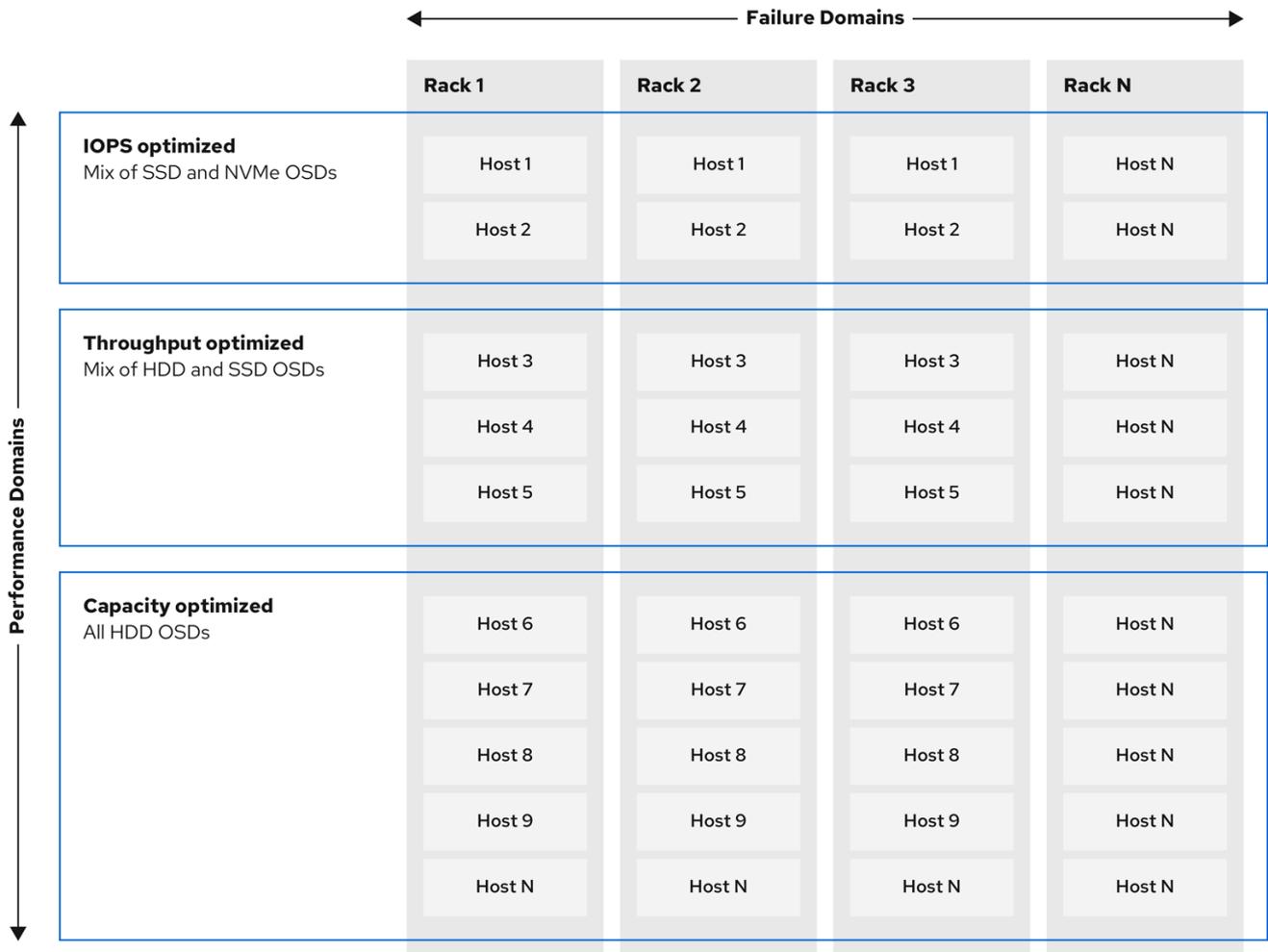
CRUSH 맵은 클러스터 리소스의 토폴로지를 설명하고 이 맵은 클라이언트 호스트와 클러스터 내의 Ceph 모니터 호스트에 모두 존재합니다. Ceph 클라이언트 및 Ceph OSD는 모두 CRUSH 맵과 CRUSH 알고리즘을 사용합니다. Ceph 클라이언트는 OSD와 직접 통신하여 중앙 집중식 개체 조회 및 잠재적인 성능 병목 현상을 제거합니다. CRUSH 맵과 해당 피어의 통신을 인식하면 OSD에서 복제, 백필 및 복구 기능을 처리할 수 있으므로 동적 오류 복구를 수행할 수 있습니다.

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.

- 하드 디스크 드라이브(HDD)는 일반적으로 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 집약적 워크로드에서는 SSD를 자주 사용합니다.

그림 2.1. 성능 및 오류 도메인



336_Ceph_0523

워크로드

Red Hat Ceph Storage는 세 가지 주요 워크로드에 최적화되어 있습니다.



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.

- **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 데이터를 제공하는 데 적합합니다. 처리량에 최적화된 배포에는 높은 대역폭 네트워킹 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

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, 네트워크 하드웨어 구성이 가격 대비 성능 요구 사항을 충족하기 위해 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, 백업, 복구 트래픽을 처리합니다. 최소한 **단일 10Gb/s** 이더넷 링크를 스토리지 하드웨어에 사용해야 하며 연결 및 처리량을 위해 10Gb/s 이더넷 링크를 추가할 수 있습니다.



중요

Red Hat은 복제된 풀에서 여러 기본으로 **osd_pool_default_size** 를 사용하여 공용 네트워크의 배수가 되도록 대역폭을 스토리지 클러스터 네트워크에 할당할 것을 권장합니다. 또한 별도의 네트워크 카드에서 공용 및 스토리지 클러스터 네트워크를 실행하는 것이 좋습니다.



중요

프로덕션 환경에서 Red Hat Ceph Storage 배포에는 10개의 Gb/s 이더넷을 사용하는 것이 좋습니다. 1Gb/s 이더넷 네트워크는 프로덕션 스토리지 클러스터에 적합하지 않습니다.

드라이브 장애가 발생하는 경우 1Gb/s 네트워크에서 1TB의 데이터를 복제하는 데 3시간이 걸리며 1Gb/s 네트워크에서 10TB의 데이터를 복제하는 데 30시간이 걸립니다. 10TB를 사용하는 것은 일반적인 드라이브 구성입니다. 반면 10Gb/s 이더넷 네트워크를 사용하면 1TB의 경우 복제 시간은 20분, 10TB의 경우 1시간입니다. Ceph OSD가 실패하면 스토리지 클러스터는 실패한 OSD와 동일한 장애 도메인 및 장치 클래스 내의 다른 OSD에 포함된 데이터를 복제하여 복구합니다.

랙과 같은 대규모 도메인에 장애가 발생하면 스토리지 클러스터가 훨씬 더 많은 대역폭을 사용한다는 것을 의미합니다. 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/s 이더넷 스위치에는 48개의 10GB/s 포트와 4개의 40GB/s 포트가 있습니다. 최대 처리량을 위해 스파인의 40Gb/s 포트를 사용합니다. 또는 다른 랙 및 스파인 라우터에 연결하기 위해 사용하지 않는 10Gb/s 포트를 4Gb/s 포트 40Gb/s 포트로 집계하는 것이 좋습니다. 또한 네트워크 인터페이스를 결합하는 데 LACP 모드 4를 사용하는 것이 좋습니다. 또한, 특히 백엔드 또는 클러스터 네트워크에서 최대 전송 단위(MTU)가 9000인 점보 프레임을 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. 프런트 사이드 네트워크에는 최소 10GB/s 이더넷을 사용합니다. 대규모 클러스터의 경우 백엔드 또는 클러스터 네트워크에 40Gb/s 이더넷을 사용하는 것이 좋습니다.



중요

네트워크 최적화를 위해 Red Hat은 대역폭당 CPU를 개선하고 차단되지 않는 네트워크 스위치 백 플레인 에 점보 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 환경의 모든 호스트 및 네트워킹 장치에서 MTU 값이 동일한지 확인합니다.

추가 리소스

- 자세한 내용은 *Red Hat Ceph Storage 구성 가이드의 사실 네트워크 구성* 섹션을 참조하십시오.
- 자세한 내용은 *Red Hat Ceph Storage 구성 가이드의 공용 네트워크 구성* 섹션을 참조하십시오.
- 자세한 내용은 *Red Hat Ceph Storage 구성 가이드의 여러 공용 네트워크 구성* 섹션을 참조하십시오.

2.5. OSD 호스트에서 RAID 컨트롤러 사용 고려 사항

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

- OSD 호스트에 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.

2.6. 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 호스트에 대해 조정이 설정되어 있는지 확인합니다. 추가 지침을 요구하는 Red Hat 지원 케이스를 열 수도 있습니다.

파일 디스크립터 증가

Ceph Object Gateway는 파일 디스크립터가 부족하면 중지될 수 있습니다. Ceph Object Gateway 호스트에서 `/etc/security/limits.conf` 파일을 수정하여 Ceph Object Gateway의 파일 디스크립터를 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 호스트에 `/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.

2.7. 코로케이션 작동 방식 및 이점

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

- 소규모의 총소유비용 (total cost of ownership)의 상당한 개선
- 최소 구성을 위해 6개의 호스트에서 3개로 감소
- 더 쉬운 업그레이드
- 리소스 분리 개선

코로케이션 작동 방식

Cephadm 오케스트레이터의 도움말을 사용하여 다음 목록에서 하나의 데몬을 하나 이상의 OSD 데몬 (ceph-osd)으로 배치할 수 있습니다.

- Ceph Monitor(**ceph-mon**) 및 Ceph Manager(**ceph-mgr**) 데몬
- Ceph Object Gateway(**nfs-ganesha**)용 NFS Ganesha(**nfs-ganesha**)
- RBD 미러 (**rbd-mirror**)
- Observability Stack (Grafana)

또한 Ceph Object Gateway(RGW) 및 Ceph File System(**ceph-mds**)의 경우 RBD 미러를 제외하고 위의 목록에서 OSD 데몬과 데몬을 함께 배치할 수 있습니다.



참고

지정된 노드에서 동일한 유형의 데몬을 배치하는 것은 지원되지 않습니다.



참고

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



참고

Red Hat은 **Ceph Object Gateway**를 **Ceph OSD** 컨테이너와 연결하여 성능을 향상시키는 것이 좋습니다.

위에서 공유된 공동 배치 규칙을 통해 다음 규칙을 준수하는 최소 클러스터 크기가 다음과 같습니다.

예시 1

1. 미디어: **Full Flash System (SSDs)**
2. 사용 사례: **블록(RBD) 및 파일(CephFS) 또는 오브젝트(Ceph Object Gateway)**
3. 노드 수: **3**
4. 복제 스키마: **2**

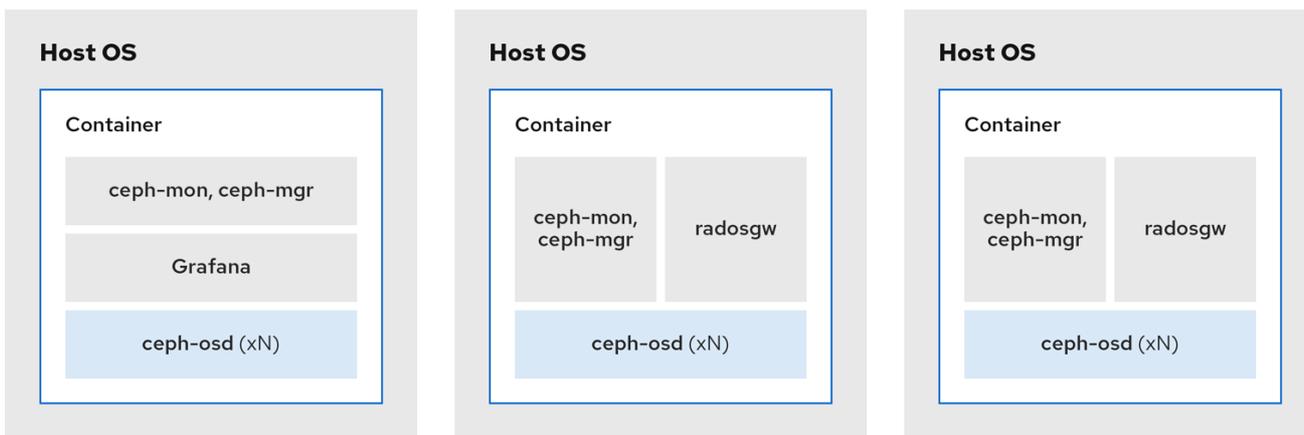
호스트	데몬	데몬	데몬
host1	OSD	모니터링/매니저	Grafana
host2	OSD	모니터링/매니저	RGW 또는 CephFS
host3	OSD	모니터링/매니저	RGW 또는 CephFS



참고

3개의 복제본이 있는 스토리지 클러스터의 최소 크기는 노드 4개입니다. 마찬가지로 두 개의 복제본이 있는 스토리지 클러스터의 크기는 3개의 노드 클러스터입니다. 성능이 저하된 상태의 클러스터의 연장된 기간을 방지하려면 클러스터에서 추가 노드를 사용하여 복제 요인에 대해 특정 수의 노드를 사용해야 합니다.

그림 2.2. 공동 배치 Daemons 예제 1



336_Ceph_0423

예시 2

1. 미디어: **SSD(Full Flash System)** 또는 회전 장치(HDD)
2. 사용 사례: 블록(RBD), 파일(CephFS), 오브젝트(Ceph Object Gateway)
3. 노드 수: **4**
4. 복제 스키마: **3**

호스트	데몬	데몬	데몬
host1	OSD	Grafana	CephFS
host2	OSD	모니터링/매니저	RGW
host3	OSD	모니터링/매니저	RGW

호스트	데몬	데몬	데몬
host4	OSD	모니터링/매니저	CephFS

그림 2.3. 공동 배치 Daemons 예제 2



336_Ceph_0423

예시 3

1. 미디어: **SSD(Full Flash System) 또는 회전 장치(HDD)**
2. 사용 사례: **Ceph Object Gateway용 블록(RBD), 오브젝트(Ceph Object Gateway) 및 NFS**
3. 노드 수: **4**
4. 복제 스키마: **3**

호스트	데몬	데몬	데몬
host1	OSD	Grafana	
host2	OSD	모니터링/매니저	RGW
host3	OSD	모니터링/매니저	RGW
host4	OSD	모니터링/매니저	NFS(RGW)

그림 2.4. 공동 배치 Daemons 예제 3



336_Ceph_0423

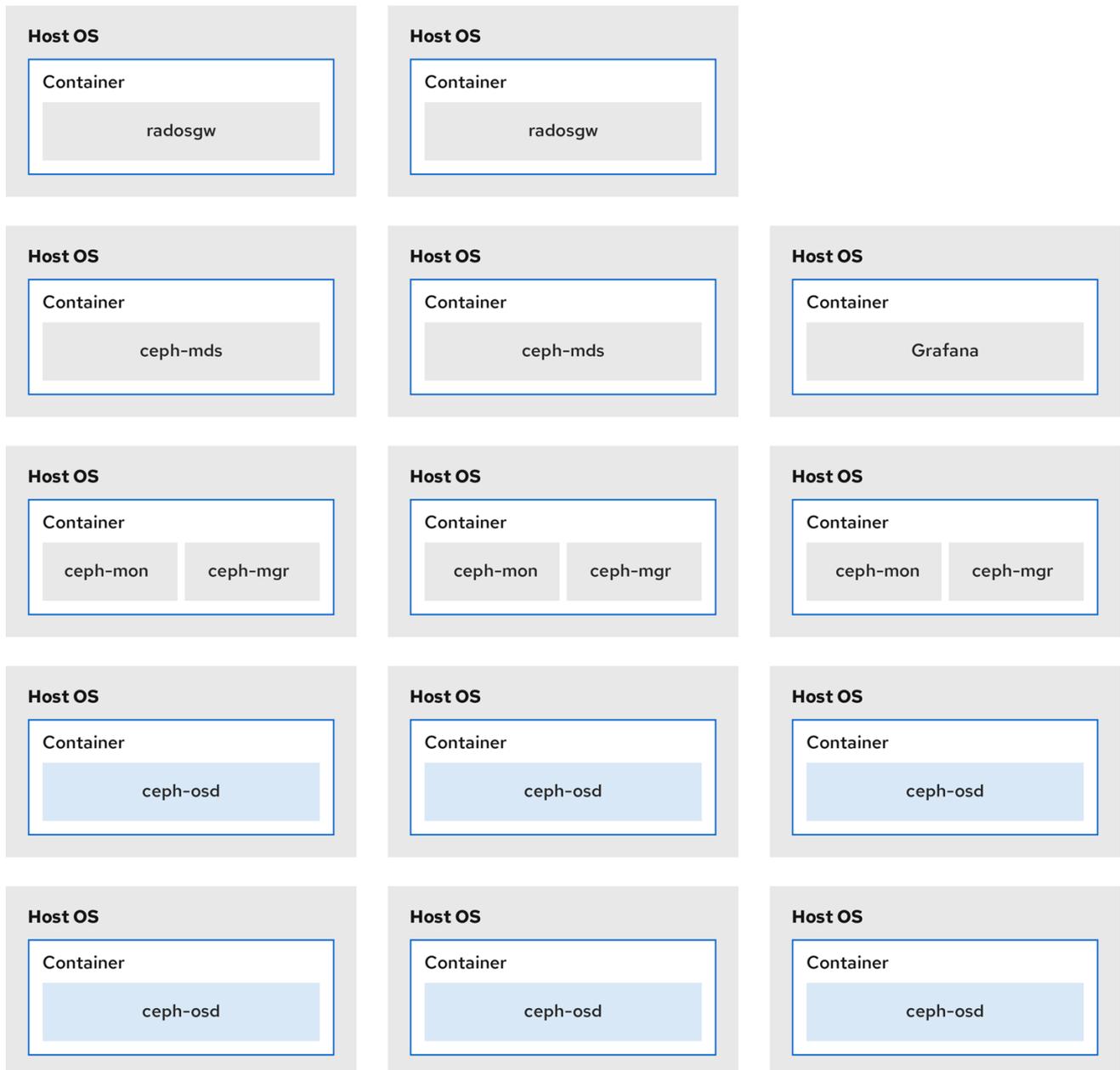
아래 다이어그램은 공동 배치 및 지정되지 않은 데몬이 있는 스토리지 클러스터의 차이점을 보여줍니다.

그림 2.5. 공동 배치된 데몬



336_Ceph_0423

그림 2.6. 공동배치되지 않은 데몬



108_Ceph_0720

2.8. RED HAT CEPH STORAGE의 운영 체제 요구 사항

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

Red Hat Ceph Storage 5 릴리스는 Red Hat Enterprise Linux 8.4 EUS 이상에서 지원됩니다.

Red Hat Ceph Storage 5는 컨테이너 기반 배포에서만 지원됩니다.

모든 노드에서 동일한 운영 체제 버전, 아키텍처 및 배포 유형을 사용합니다. 예를 들어, AMD64 및 Intel 64 아키텍처, Red Hat Enterprise Linux 8 운영 체제가 포함된 노드가 혼합되거나 컨테이너 기반 배포가 포함된 노드를 혼합하여 사용하지 마십시오.



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](#) 및 [Red Hat Enterprise Linux 8 Using SELinux Guide](#)를 참조하십시오.

추가 리소스

- [Red Hat Enterprise Linux 8에 대한 설명서.](#)

2.9. 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.



참고

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

표 2.1. 컨테이너

프로세스	기준	최소 권장 사항
ceph-osd-container	프로세서	OSD 컨테이너당 1x AMD64 또는 Intel 64 CPU CORE
	RAM	OSD 컨테이너당 최소 5GB의 RAM.
	노드 수	최소 3개의 노드가 필요합니다.

프로세스	기준	최소 권장 사항
	OS 디스크	호스트당 1x OS 디스크입니다.
	OSD 스토리지	OSD 컨테이너당 1x 스토리지 드라이브. OS 디스크와 공유할 수 없습니다.
	block.db	선택 사항이지만 Red Hat은 데몬당 1x SSD 또는 NVMe 또는 Optane 파티션 또는 lvm을 권장합니다. 오브젝트, 파일 및 혼합된 워크로드의 경우 BlueStore용 block.data 의 크기 조정은 블록 장치, Openstack cinder, Openstack cinder 워크로드용 block.data 의 1%입니다.
	block.wal	데몬당 1x SSD 또는 NVMe 또는 Optane 파티션 또는 논리 볼륨 옵션으로 사용할 수 있습니다. 작은 크기(예: 10GB)를 사용하고 block.db 장치보다 빠른 경우에만 사용합니다.
네트워크	2x 10GB 이더넷 NIC	ceph-mon-container
프로세서	mon-container당 1x AMD64 또는 Intel 64 CPU CORE	
RAM	mon-container 당 3GB	
디스크 공간	mon-container 당 10GB, 50GB 권장	
모니터 디스크	선택적으로 Monitor Stensdb 데이터를 위한 1x SSD 디스크	
네트워크	2x 1GB 이더넷 NIC, 10GB 권장	
Prometheus	/var/lib/ceph/ 디렉터리에서 /var/lib/ceph/ 디렉터리에서 20GB~50GB를 별도의 파일 시스템으로 생성하여 /var/ 디렉터리의 콘텐츠를 보호합니다.	ceph-mgr-container
프로세서	mgr-container 당 1x AMD64 또는 Intel 64 CPU CORE	

프로세스	기준	최소 권장 사항
RAM	mgr-container 당 3GB	
네트워크	2x 1GB 이더넷 NIC, 10GB 권장	ceph-radosgw-container
프로세서	radosgw-container당 1x AMD64 또는 Intel 64 CPU CORE	
RAM	데몬당 1GB	
디스크 공간	데몬당 5GB	
네트워크	1x 1GB 이더넷 NIC	
프로세서	mgs-container당 1x AMD64 또는 Intel 64 CPU CORE	ceph-mds-container
RAM	mds-container 당 3GB 이 수는 구성 가능한 MDS 캐시 크기에 따 라 다릅니다. RAM 요 구 사항은 일반적으로 mds_cache_mem ory_limit 구성 설정 에 설정된 양보다 두 배가 됩니다. 이는 전 체 시스템 메모리가 아니라 데몬의 메모리 이기도 합니다.	
디스크 공간	2GB의 mgs- container 와 가능한 디버그 로깅에 필요한 추가 공간을 고려할 때 20GB가 좋은 출발 점입니다.	

2.10. 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

스토리지 관리자는 **cephadm** 유틸리티를 사용하여 새 **Red Hat Ceph Storage** 클러스터를 배포할 수 있습니다.

cephadm 유틸리티는 **Ceph** 클러스터의 전체 라이프사이클을 관리합니다. 설치 및 관리 작업에는 다음 두 가지 유형의 작업이 포함됩니다.

- **Day One(1일차)** 작업에서는 단일 노드에서 실행되는 컨테이너화된 베어 메탈 **Ceph** 스토리지 클러스터를 설치하고 부트스트래핑해야 합니다. **Day One**에는 **Monitor** 및 **Manager** 데몬 배포 및 **Ceph OSD** 추가도 포함됩니다.
- **Day Two(2일차)** 작업에서는 **Ceph** 오케스트레이션 인터페이스, **cephadm orch** 또는 **Red Hat Ceph Storage** 대시보드를 사용하여 스토리지 클러스터에 다른 **Ceph** 서비스를 추가하여 스토리지 클러스터를 확장합니다.

3.1. 사전 요구 사항

- 인터넷이 연결된 **VM(가상 머신)** 또는 베어 메탈 서버를 하나 이상 실행
- **Red Hat Enterprise Linux 8.4 EUS** 이상.
- **Ansible 2.9** 이상
- 적절한 인타임먼트가 있는 유효한 **Red Hat** 서브스크립션.
- 모든 노드에 대한 루트 수준 액세스.
- **Red Hat Registry**에 액세스하기 위한 활성 **RHN(Red Hat Network)** 또는 서비스 계정.
- **iptables** 서비스를 새로 고침해도 클러스터에 문제가 발생하지 않도록 **iptables**에서 문제 해결 구성을 제거합니다. 예를 들어 **Red Hat Ceph Storage** 구성 가이드의 [기본 Ceph 포트에 대해](#)

구성된 방화벽 규칙 확인 섹션을 참조하십시오.

3.2. CEPHADM 유틸리티

cephadm 유틸리티는 **Ceph** 스토리지 클러스터를 배포 및 관리합니다. **CLI**(명령줄 인터페이스) 및 **Red Hat Ceph Storage** 대시보드 웹 인터페이스와 긴밀하게 통합되어 있어 두 환경에서 스토리지 클러스터를 관리할 수 있습니다. **cephadm** 은 **SSH**를 사용하여 **Ceph** 데몬 컨테이너를 추가, 제거 또는 업데이트하기 위해 **SSH**를 사용합니다. **Ansible** 또는 **Rook**과 같은 외부 구성 또는 오케스트레이션 툴에는 의존하지 않습니다.



참고

호스트에서 **preflight** 플레이북을 실행한 후 **cephadm** 유틸리티를 사용할 수 있습니다.

cephadm 유틸리티는 다음 두 가지 주요 구성 요소로 구성됩니다.

- **cephadm 셸**
- **cephadm 오케스트레이터**

cephadm 셸

cephadm 셸은 컨테이너 내에서 **bash 셸**을 시작합니다. 이를 통해 설치 및 부트 스트랩과 같은 "**Day One**" 클러스터 설정 작업을 수행하고 **ceph** 명령을 호출할 수 있습니다.

cephadm 셸을 호출하는 방법은 다음 두 가지가 있습니다.

- 시스템 프롬프트에서 **cephadm 셸**을 입력합니다.

예제

```
[root@host01 ~]# cephadm shell
[ceph: root@host01 /]# ceph -s
```

■

●

시스템 프롬프트에서 **cephadm shell** 과 실행할 명령을 입력합니다.

예제

```
[root@host01 ~]# cephadm shell ceph -s
```



참고

노드에 `/etc/ceph/` 의 구성 및 키링 파일이 포함된 경우 컨테이너 환경에서는 해당 파일의 값을 **cephadm** 셸의 기본값으로 사용합니다. 그러나 **Ceph Monitor** 노드에서 **cephadm** 셸을 실행하는 경우 **cephadm** 셸은 기본 구성을 사용하는 대신 **Ceph Monitor** 컨테이너의 기본 구성을 상속합니다.

cephadm 오케스트레이터

cephadm 오케스트레이터를 사용하면 스토리지 클러스터 확장 및 **Ceph** 데몬 및 서비스 프로비저닝과 같은 "두 가지" **Ceph** 기능을 수행할 수 있습니다. CLI(명령줄 인터페이스) 또는 웹 기반 **Red Hat Ceph Storage** 대시보드를 통해 **cephadm** 오케스트레이터를 사용할 수 있습니다. 오케스트레이터 명령은 **ceph orch** 형식을 사용합니다.

cephadm 스크립트는 **Ceph Manager**에서 사용하는 **Ceph** 오케스트레이션 모듈과 상호 작용합니다.

3.3. CEPHADM의 작동 방식

cephadm 명령은 **Red Hat Ceph Storage** 클러스터의 전체 라이프사이클을 관리합니다. **cephadm** 명령은 다음 작업을 수행할 수 있습니다.

●

새 **Red Hat Ceph Storage** 클러스터를 부트스트랩합니다.

- **Red Hat Ceph Storage CLI(명령줄 인터페이스)와 함께 작동하는 컨테이너화된 셸을 시작합니다.**
- 컨테이너화된 데몬 디버깅에 도움이 됩니다.

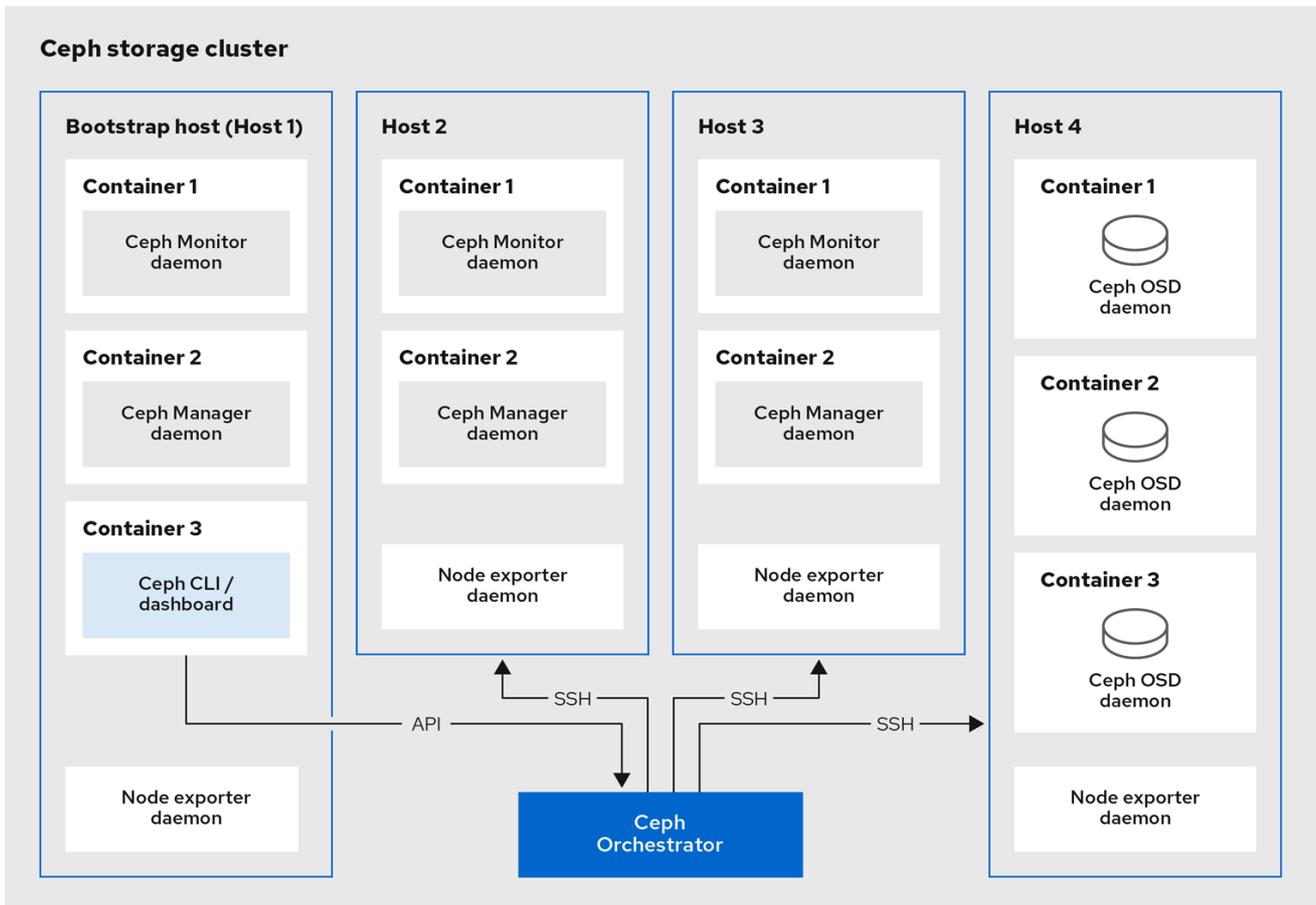
cephadm 명령은 **ssh** 를 사용하여 스토리지 클러스터의 노드와 통신합니다. 이를 통해 외부 툴을 사용하지 않고도 **Red Hat Ceph Storage** 컨테이너를 추가, 제거 또는 업데이트할 수 있습니다. 부트스트랩 프로세스 중에 **ssh** 키 쌍을 생성하거나 자체 **ssh** 키를 사용합니다.

cephadm 부트스트랩 프로세스는 하나의 **Ceph Monitor**와 하나의 **Ceph Manager**로 구성된 작은 스토리지 클러스터와 필요한 종속 항목을 모두 생성합니다. 그런 다음 오케스트레이터 **CLI** 또는 **Red Hat Ceph Storage** 대시보드를 사용하여 노드를 포함하도록 스토리지 클러스터를 확장하고 모든 **Red Hat Ceph Storage** 데몬 및 서비스를 프로비저닝합니다. **CLI** 또는 **Red Hat Ceph Storage** 대시보드 웹 인터페이스에서 관리 기능을 수행할 수 있습니다.



참고

cephadm 유틸리티는 **Red Hat Ceph Storage 5**의 새로운 기능입니다. 이전 버전의 **Red Hat Ceph Storage**는 지원되지 않습니다.



252_Ceph_0522

3.4. CEPHADM-ANSIBLE PLAYBOOK

cephadm-ansible 패키지는 **cephadm** 에서 다루지 않는 워크플로우를 간소화하는 **Ansible** 플레이북 컬렉션입니다. 설치 후 플레이북은 `/usr/share/cephadm-ansible/` 에 있습니다.



중요

Red Hat Enterprise Linux 9 이상에서는 **cephadm-ansible** 플레이북을 지원하지 않습니다.

cephadm-ansible 패키지에는 다음 플레이북이 포함되어 있습니다.

- **cephadm-preflight.yml**

- **cephadm-clients.yml**
- **cephadm-purge-cluster.yml**

cephadm-preflight Playbook

스토리지 클러스터를 부트스트랩하기 전에 **cephadm-preflight** 플레이북을 사용하여 호스트를 처음에 설정하고 새 노드 또는 클라이언트를 스토리지 클러스터에 추가합니다. 이 플레이북은 **Ceph** 리포지토리를 구성하고 **podman**, **lvm2**, **chrony**, **cephadm** 과 같은 일부 사전 요구 사항을 설치합니다.

cephadm-clients Playbook

cephadm-clients 플레이북을 사용하여 클라이언트 호스트를 설정합니다. 이 플레이북은 구성 및 인증 키 파일을 **Ceph** 클라이언트 그룹에 배포합니다.

cephadm-purge-cluster Playbook

cephadm-purge-cluster 플레이북을 사용하여 **Ceph** 클러스터를 제거합니다. 이 플레이북은 **cephadm**으로 관리되는 **Ceph** 클러스터를 제거합니다.

추가 리소스

- **cephadm-preflight** 플레이북에 대한 자세한 내용은 [Running the preflight playbook](#) 에서 참조하십시오.
- **cephadm-clients** 플레이북에 대한 자세한 내용은 [cephadm-clients 플레이북 실행](#) 을 참조하십시오.
- **cephadm-purge-cluster** 플레이북에 대한 자세한 내용은 [Ceph 스토리지 클러스터 정리를 참조](#) 하십시오.

3.5. RED HAT CEPH STORAGE 노드를 CDN에 등록하고 서브스크립션 연결

Red Hat Ceph Storage 5는 Red Hat Enterprise Linux 8.4 EUS 이상에서 지원됩니다.

사전 요구 사항

- 인터넷이 연결된 **VM(가상 머신)** 또는 **베어 메탈 서버**를 하나 이상 실행
- **Red Hat Enterprise Linux 8.4 EUS** 이상.
- 적절한 인타임먼트가 있는 유효한 **Red Hat** 서브스크립션.
- 모든 노드에 대한 루트 수준 액세스.

절차

1. 노드를 등록하고 메시지가 표시되면 **Red Hat** 고객 포털 인증 정보를 입력합니다.

구문

```
subscription-manager register
```

2. **CDN**에서 최신 서브스크립션 데이터를 가져옵니다.

구문

```
subscription-manager refresh
```

3. **Red Hat Ceph Storage**에 사용 가능한 모든 서브스크립션을 나열합니다.

구문

```
subscription-manager list --available --matches 'Red Hat Ceph Storage'
```

4.

적절한 서브스크립션을 식별하고 해당 풀 ID를 검색합니다.

5.

소프트웨어 인타이틀먼트에 대한 액세스 권한을 얻기 위해 풀 ID를 연결합니다. 이전 단계에서 확인한 풀 ID를 사용합니다.

구문

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

6.

기본 소프트웨어 리포지토리를 비활성화한 다음 해당 **Red Hat Enterprise Linux** 버전에서 서버 및 추가 리포지토리를 활성화합니다.

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
```

Red Hat Enterprise Linux 9

```
subscription-manager repos --disable=*
subscription-manager repos --enable=rhel-9-for-x86_64-baseos-rpms
subscription-manager repos --enable=rhel-9-for-x86_64-appstream-rpms
```

7.

Red Hat Enterprise Linux의 최신 패키지를 수신하도록 시스템을 업데이트합니다.

구문

```
# dnf update
```

8.

Red Hat Ceph Storage 5 콘텐츠를 구독하십시오. [Red Hat Satellite 6](#)으로 **Ceph** 등록 방법의 지침을 따르십시오.

9.

ceph-tools 리포지토리를 활성화합니다.

Red Hat Enterprise Linux 8

```
subscription-manager repos --enable=rhceph-5-tools-for-rhel-8-x86_64-rpms
```

Red Hat Enterprise Linux 9

```
subscription-manager repos --enable=rhceph-5-tools-for-rhel-9-x86_64-rpms
```

10. 클러스터에 추가하는 모든 노드에서 위의 단계를 반복합니다.

11. Red Hat Enterprise Linux 8에 **cephadm-ansible** 을 설치합니다.

예제

```
dnf install cephadm-ansible
```



중요

Red Hat Enterprise Linux 9의 경우 **cephadm-ansible** 이 지원되지 않으므로 이 단계를 건너뛰니다.

3.6. ANSIBLE 인벤토리 위치 구성

cephadm-ansible 스테이징 및 프로덕션 환경에 대한 인벤토리 위치 파일을 구성할 수 있습니다.

Ansible 인벤토리 호스트 파일에는 스토리지 클러스터의 일부인 모든 호스트가 포함됩니다. 인벤토리 호스트 파일에서 노드를 개별적으로 나열하거나 **[mons]**, **[osds]** 및 **[rgws]** 와 같은 그룹을 생성하여 인벤토리에 명확성을 제공하고 **--limit** 옵션을 사용하여 플레이북을 실행할 때 그룹 또는 노드를 쉽게 사용할 수 있습니다.



참고

클라이언트를 배포하는 경우 클라이언트 노드를 전용 **[clients]** 그룹에 정의해야 합니다.



중요

cephadm-ansible 은 지원되지 않으므로 Red Hat Enterprise Linux 9에 대해 이 단계를 건너뛰니다.

사전 요구 사항

- **Ansible** 관리 노드.
- **Ansible** 관리 노드에 대한 루트 수준 액세스.
- **cephadm-ansible** 패키지가 노드에 설치되어 있습니다.

절차

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

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

2. 선택 사항: 스테이징 및 프로덕션에 사용할 하위 디렉토리를 만듭니다.

```
[root@admin cephadm-ansible]# mkdir -p inventory/staging inventory/production
```

3. 선택 사항: **ansible.cfg** 파일을 편집하고 다음 행을 추가하여 기본 인벤토리 위치를 할당합니다.

```
[defaults]
inventory = ./inventory/staging
```

4. 선택 사항: 각 환경에 대한 인벤토리 **hosts** 파일을 생성합니다.

```
[root@admin cephadm-ansible]# touch inventory/staging/hosts
[root@admin cephadm-ansible]# touch inventory/production/hosts
```

5. 각 **hosts** 파일을 열고 편집하고 노드 및 **[admin]** 그룹을 추가합니다.

```
NODE_NAME_1
NODE_NAME_2

[admin]
ADMIN_NODE_NAME_1
```

-

NODE_NAME_1 및 ***NODE_NAME_2*** 를 모니터, OSD, MDS, 게이트웨이 노드와 같은 Ceph 노드로 바꿉니다.

- ***ADMIN_NODE_NAME_1*** 을 관리 인증 키가 저장된 노드의 이름으로 바꿉니다.

예제

```
host02  
host03  
host04  
  
[admin]  
host01
```



참고

ansible.cfg 파일의 인벤토리 위치를 **staging**으로 설정하는 경우 다음과 같이 스테이징 환경에서 플레이북을 실행해야 합니다.

구문

```
ansible-playbook -i inventory/staging/hosts PLAYBOOK.yml
```

프로덕션 환경에서 플레이북을 실행하려면 다음을 수행합니다.

구문

```
ansible-playbook -i inventory/production/hosts PLAYBOOK.yml
```

3.7. RED HAT ENTERPRISE LINUX 9에서 ROOT 사용자로 SSH 로그인 활성화

Red Hat Enterprise Linux 9는 `/etc/ssh/sshd_config` 파일에서 `PermitRootLogin` 매개변수가 `yes`로 설정되어 있어도 `root` 사용자로 SSH 로그인을 지원하지 않습니다. 다음과 같은 오류가 발생합니다.

예제

```
[root@host01 ~]# ssh root@myhostname
root@myhostname password:
Permission denied, please try again.
```

다음 방법 중 하나를 실행하여 **root** 사용자로 로그인을 활성화할 수 있습니다.

- **Red Hat Enterprise Linux 9**를 설치하는 동안 **root** 암호를 설정하는 동안 "암호로 **root** 로그인 허용" 플래그를 사용합니다.
- **Red Hat Enterprise Linux 9** 설치 후 **PermitRootLogin** 매개변수를 수동으로 설정합니다.

이 섹션에서는 **PermitRootLogin** 매개변수의 수동 설정에 대해 설명합니다.

사전 요구 사항

- 모든 노드에 대한 루트 수준 액세스.

절차

1. **etc/ssh/sshd_config** 파일을 열고 **PermitRootLogin** 을 **yes** 로 설정합니다.

예제

```
[root@admin ~]# echo 'PermitRootLogin yes' >> /etc/ssh/sshd_config.d/01-permitrootlogin.conf
```

2. **SSH** 서비스를 다시 시작하십시오.

예제

```
[root@admin ~]# systemctl restart sshd.service
```

3.

root 사용자로 노드에 로그인합니다.

구문

```
ssh root@HOST_NAME
```

HOST_NAME 을 Ceph 노드의 호스트 이름으로 바꿉니다.

예제

```
[root@admin ~]# ssh root@host01
```

메시지가 표시되면 **root** 암호를 입력합니다.

추가 리소스

-

자세한 내용은 [RHEL 9 서버 지식베이스 솔루션에서 ssh를 통해 root 사용자로 로그인할 수 없음](#)을 참조하십시오.

3.8. SUDO 액세스를 사용하여 ANSIBLE 사용자 생성

스토리지 클러스터의 모든 노드에서 암호가 없는 **root** 액세스 권한이 있는 **Ansible** 사용자를 생성하여 **cephadm-ansible** 플레이북을 실행할 수 있습니다. **Ansible** 사용자는 소프트웨어를 설치하고 암호를 묻지 않고 구성 파일을 생성할 수 있는 **root** 권한이 있는 사용자로 모든 **Red Hat Ceph Storage** 노드에 로그인할 수 있어야 합니다.



중요

Red Hat Enterprise Linux 9의 루트가 아닌 사용자인 경우 사용자 생성 단계에 따라 Red Hat Enterprise Linux 9에 대한 이 단계를 건너뛸 수 있습니다.

사전 요구 사항

- 모든 노드에 대한 루트 수준 액세스.
- Red Hat Enterprise Linux 9가 root 사용자로 로그인하려면 [Red Hat Enterprise Linux 9에서 SSH 로그인 활성화](#)를 참조하십시오.

절차

1. root 사용자로 노드에 로그인합니다.

구문

```
ssh root@HOST_NAME
```

HOST_NAME 을 Ceph 노드의 호스트 이름으로 바꿉니다.

예제

```
[root@admin ~]# ssh root@host01
```

메시지가 표시되면 root 암호를 입력합니다.

2. 새 **Ansible** 사용자를 생성합니다.

구문

```
adduser USER_NAME
```

USER_NAME 을 **Ansible** 사용자의 새 사용자 이름으로 바꿉니다.

예제

```
[root@host01 ~]# adduser ceph-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.

3. 이 사용자의 새 암호를 설정합니다.

구문

```
passwd USER_NAME
```

USER_NAME 을 **Ansible** 사용자의 새 사용자 이름으로 바꿉니다.

예제

```
[root@host01 ~]# passwd ceph-admin
```

메시지가 표시되면 새 암호를 두 번 입력합니다.

4.

새로 생성된 사용자에게 대해 **sudo** 액세스를 설정합니다.

구문

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

USER_NAME 을 **Ansible** 사용자의 새 사용자 이름으로 바꿉니다.

예제

```
[root@host01 ~]# cat << EOF >/etc/sudoers.d/ceph-admin
ceph-admin ALL = (root) NOPASSWD:ALL
EOF
```

5. 새 파일에 올바른 파일 권한을 할당합니다.

구문

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

USER_NAME 을 Ansible 사용자의 새 사용자 이름으로 바꿉니다.

예제

```
[root@host01 ~]# chmod 0440 /etc/sudoers.d/ceph-admin
```

6. 스토리지 클러스터의 모든 노드에서 위의 단계를 반복합니다.

추가 리소스

- 사용자 계정 생성에 대한 자세한 내용은 **Red Hat Enterprise Linux 8** 가이드의 *기본 시스템 설정* 구성 장에 있는 [사용자 계정 관리](#) 섹션을 참조하십시오.

3.9. SSH 구성

스토리지 관리자는 **Cephadm**을 사용하여 **SSH** 키를 사용하여 원격 호스트로 안전하게 인증할 수 있습니다. **SSH** 키는 원격 호스트에 연결하기 위해 모니터에 저장됩니다.

사전 요구 사항

- **Ansible** 관리 노드.

- **Ansible** 관리 노드에 대한 루트 수준 액세스.
- **cephadm-ansible** 패키지가 노드에 설치되어 있습니다.

절차

1. **cephadm-ansible** 디렉터리로 이동합니다.
2. 새 **SSH** 키를 생성합니다.

예제

```
[ceph-admin@admin cephadm-ansible]$ ceph cephadm generate-key
```

3. **SSH** 키의 공개 부분을 검색합니다.

예제

```
[ceph-admin@admin cephadm-ansible]$ ceph cephadm get-pub-key
```

4. 현재 저장된 **SSH** 키를 삭제합니다.

예제

```
[ceph-admin@admin cephadm-ansible]$ ceph cephadm clear-key
```

5.

mgr 데몬을 다시 시작하여 구성을 다시 로드합니다.

예제

```
[ceph-admin@admin cephadm-ansible]$ ceph mgr fail
```

3.9.1. 다른 SSH 사용자 구성

스토리지 관리자는 암호를 묻지 않고 컨테이너 이미지를 다운로드하고, 컨테이너를 시작하고, 명령을 실행할 수 있는 충분한 권한으로 모든 **Ceph** 클러스터 노드에 로그인할 수 있는 루트가 아닌 **SSH** 사용자를 구성할 수 있습니다.



중요

루트가 아닌 **SSH** 사용자를 구성하기 전에 클러스터 **SSH** 키를 사용자의 **authorized_keys** 파일에 추가해야 하며 루트가 아닌 사용자에게는 **암호** 없는 **sudo** 액세스 권한이 있어야 합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ansible** 관리 노드.
- **Ansible** 관리 노드에 대한 루트 수준 액세스.
- **cephadm-ansible** 패키지가 노드에 설치되어 있습니다.

- 사용자의 **authorized_keys** 에 클러스터 **SSH** 키를 추가합니다.
- 루트가 아닌 사용자에게 대해 **암호 없는 sudo** 액세스를 활성화합니다.

절차

1. **cephadm-ansible** 디렉터리로 이동합니다.
2. 모든 **Cephadm** 작업을 수행할 사용자의 이름을 **Cephadm**에 제공합니다.

구문

```
[ceph-admin@admin cephadm-ansible]$ ceph cephadm set-user <user>
```

예제

```
[ceph-admin@admin cephadm-ansible]$ ceph cephadm set-user user
```

3. **SSH** 공개 키를 검색합니다.

구문

```
ceph cephadm get-pub-key > ~/ceph.pub
```

예제

```
[ceph-admin@admin cephadm-ansible]$ ceph cephadm get-pub-key > ~/ceph.pub
```

4.

SSH 키를 모든 호스트에 복사합니다.

구문

```
ssh-copy-id -f -i ~/ceph.pub USER@HOST
```

예제

```
[ceph-admin@admin cephadm-ansible]$ ssh-copy-id ceph-admin@host01
```

3.10. ANSIBLE에 대해 암호 없는 SSH 활성화

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



중요

Red Hat Enterprise Linux 9의 루트가 아닌 사용자인 경우 사용자 생성 단계에 따라 **Red Hat Enterprise Linux 9**에 대한 이 단계를 건너뛸 수 있습니다.

사전 요구 사항

- **Ansible** 관리 노드에 액세스
- 스토리지 클러스터의 모든 노드에 대한 **sudo** 액세스 권한이 있는 **Ansible** 사용자
- **Red Hat Enterprise Linux 9**가 **root** 사용자로 로그인하려면 **Red Hat Enterprise Linux 9**에서 **SSH 로그인 활성화**를 참조하십시오.

절차

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

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

2. 공개 키를 스토리지 클러스터의 모든 노드에 복사합니다.

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

USER_NAME 을 **Ansible** 사용자의 새 사용자 이름으로 바꿉니다. ***HOST_NAME*** 을 **Ceph** 노드의 호스트 이름으로 바꿉니다.

예제

```
[ceph-admin@admin ~]$ ssh-copy-id ceph-admin@host01
```

3. 사용자의 **SSH config** 파일을 생성합니다.

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

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

```
Host host01
  Hostname HOST_NAME
  User USER_NAME
Host host02
  Hostname HOST_NAME
  User USER_NAME
...
```

HOST_NAME 을 Ceph 노드의 호스트 이름으로 바꿉니다. ***USER_NAME*** 을 Ansible 사용자의 새 사용자 이름으로 바꿉니다.

예제

```
Host host01
  Hostname host01
  User ceph-admin
Host host02
  Hostname host02
  User ceph-admin
Host host03
  Hostname host03
  User ceph-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.

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

추가 리소스

- **ssh_config(5)** 매뉴얼 페이지.

-

OpenSSH를 사용하여 두 시스템 간에 보안 통신 사용을 참조하십시오.

3.11. PREFLIGHT PLAYBOOK 실행

이 **Ansible Playbook**은 **Ceph** 리포지토리를 구성하고 부트 스트랩을 위해 스토리지 클러스터를 준비합니다. **podman, lvm2, chrony, cephadm** 과 같은 몇 가지 사전 요구 사항도 설치합니다. **cephadm-ansible** 및 **cephadm-preflight.yml** 의 기본 위치는 **/usr/share/cephadm-ansible** 입니다.

preflight 플레이북은 **cephadm-ansible** 인벤토리 파일을 사용하여 스토리지 클러스터의 모든 관리자 및 노드를 식별합니다.



중요

cephadm-ansible 은 지원되지 않으므로 **Red Hat Enterprise Linux 9**에 대해 이 단계를 건너뜁니다.

인벤토리 파일의 기본 위치는 **/usr/share/cephadm-ansible/hosts** 입니다. 다음 예제는 일반적인 인벤토리 파일의 구조를 보여줍니다.

예제

```
host02
host03
host04

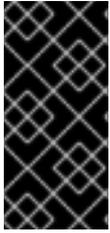
[admin]
host01
```

인벤토리 파일의 **[admin]** 그룹에는 관리자 인증 키가 저장된 노드의 이름이 포함되어 있습니다. 새 스토리지 클러스터에서 **[admin]** 그룹의 노드는 부트스트랩 노드가 됩니다. 클러스터를 부트 스트랩한 후 추가 관리 호스트를 추가하려면 **설치 가이드**에서 **관리 노드 설정**을 참조하십시오.



참고

초기 호스트를 부트스트랩하기 전에 **preflight** 플레이북을 실행합니다.



중요

비연결 설치를 수행하는 경우 **비연결 설치를 위한 preflight 플레이북 실행**을 참조하십시오.

사전 요구 사항

- **Ansible** 관리 노드에 대한 루트 수준 액세스.
- **sudo** 및 암호 없이 **ssh** 를 사용하는 **Ansible** 사용자는 스토리지 클러스터의 모든 노드에 액세스할 수 있습니다.



참고

아래 예에서 **host01**은 부트 스트랩 노드입니다.

절차

1. **/usr/share/cephadm-ansible** 디렉토리로 이동합니다.
2. **hosts** 파일을 열고 편집하고 노드를 추가합니다.

예제

```
host02
host03
host04

[admin]
host01
```

3. **preflight Playbook**을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars "ceph_origin=rhcs"
```

예제

```
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --extra-vars "ceph_origin=rhcs"
```

설치가 완료되면 **cephadm** 은 **/usr/sbin/** 디렉터리에 있습니다.

- 스토리지 클러스터에서 선택한 호스트 집합에서 **preflight** 플레이북을 실행하려면 **--limit** 옵션을 사용합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars "ceph_origin=rhcs" --limit GROUP_NAME|NODE_NAME
```

GROUP_NAME 을 인벤토리 파일의 그룹 이름으로 바꿉니다. **NODE_NAME** 을 인벤토

리 파일의 특정 노드 이름으로 바꿉니다.



참고

선택적으로 **[mons]**, **[osds]** 및 **[mgrs]** 와 같은 그룹 이름으로 인벤토리 파일의 노드를 그룹화할 수 있습니다. 그러나 관리 노드를 **[admin]** 그룹에 추가해야 하며 클라이언트가 **[clients]** 그룹에 추가해야 합니다.

예제

```
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml
--extra-vars "ceph_origin=rhcs" --limit clients
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml
--extra-vars "ceph_origin=rhcs" --limit host01
```

- **preflight** 플레이북을 실행하면 **cephadm-ansible** 이 클라이언트 노드에 **chrony** 및 **ceph-common** 을 자동으로 설치합니다.

preflight 플레이북은 **chrony** 를 설치하지만 단일 **NTP** 소스를 위해 구성합니다. 여러 소스를 구성하거나 연결이 끊긴 환경이 있는 경우 자세한 내용은 다음 설명서를 참조하십시오.

- [chrony를 어떻게 구성합니까?](#)
- [NTP 모범 사례.](#)
- [기본 chrony NTP 문제 해결.](#)

3.12. 새 스토리지 클러스터 부트스트랩

cephadm 유틸리티는 부트스트랩 프로세스 중에 다음 작업을 수행합니다.

- 로컬 노드에서 새 **Red Hat Ceph Storage** 클러스터의 **Ceph Monitor** 데몬과 **Ceph Manager** 데몬을 컨테이너로 설치 및 시작합니다.
- `/etc/ceph` 디렉토리를 생성합니다.
- 공개 키 사본을 **Red Hat Ceph Storage** 클러스터의 `/etc/ceph/ceph.pub` 에 작성하고 SSH 키를 루트 사용자의 `/root/.ssh/authorized_keys` 파일에 추가합니다.
- `_admin` 레이블을 부트스트랩 노드에 적용합니다.
- `/etc/ceph/ceph.conf` 에 새 클러스터와 통신하는 데 필요한 최소 구성 파일을 작성합니다.
- `client.admin` 관리 보안 키 사본을 `/etc/ceph/ceph.client.admin.keyring` 에 씁니다.
- **Prometheus**, **Grafana** 및 **node-exporter** 및 **alert-manager** 와 같은 기타 도구를 사용하여 기본 모니터링 스택을 배포합니다.



중요

비연결 설치를 수행하는 경우 [비연결 설치 수행](#)을 참조하십시오.



참고

새 스토리지 클러스터에서 실행하려는 기존 **Prometheus** 서비스가 있거나 **Rook**을 사용하여 **Ceph**를 실행하는 경우 `cephadm` 부트스트랩 명령과 함께 `--skip-monitoring-stack` 옵션을 사용합니다. 이 옵션은 나중에 수동으로 구성할 수 있도록 기본 모니터링 스택을 바이패스합니다.



중요

모니터링 스택을 배포하는 경우 *Red Hat Ceph Storage Operations* 가이드의 [Ceph Orchestrator](#)를 사용하여 [모니터링 스택 배포](#)를 참조하십시오.

**중요**

부트스트랩은 대시보드에 처음 로그인할 때 기본 사용자 이름 및 암호를 제공합니다. 부트스트랩은 로그인 후 암호를 변경해야 합니다.

**중요**

부트스트랩 프로세스를 시작하기 전에 사용하려는 컨테이너 이미지에 **cephadm** 과 동일한 **Red Hat Ceph Storage** 버전이 있는지 확인합니다. 두 버전이 일치하지 않으면 **Creating initial admin user** 단계에서 부트스트랩이 실패합니다.

**참고**

부트스트랩 프로세스를 시작하기 전에 registry.redhat.io 컨테이너 레지스트리의 사용자 이름과 암호를 생성해야 합니다. Red Hat 컨테이너 레지스트리 인증에 대한 자세한 내용은 지식 베이스 문서 [Red Hat Container Registry Authentication](#)을 참조하십시오.

사전 요구 사항

- 첫 번째 **Ceph Monitor** 컨테이너의 IP 주소이며, 이는 스토리지 클러스터의 첫 번째 노드의 IP 주소이기도 합니다.
- registry.redhat.io 에 대한 로그인 액세스
- `/var/lib/containers/` 용으로 최소 **10GB**의 여유 공간이 필요
- 모든 노드에 대한 루트 수준 액세스.

**참고**

스토리지 클러스터에 여러 네트워크와 인터페이스가 포함된 경우 스토리지 클러스터를 사용하는 모든 노드에서 액세스할 수 있는 네트워크를 선택해야 합니다.



참고

로컬 노드에서 **FQDN**(정규화된 도메인 이름)을 사용하는 경우 명령줄에서 **--allow-fqdn-hostname** 옵션을 **cephadm bootstrap**에 추가합니다.



중요

클러스터의 초기 모니터 노드가 될 노드에서 **cephadm bootstrap**을 실행합니다. **IP_ADDRESS** 옵션은 **cephadm bootstrap** 을 실행하는 데 사용 중인 노드의 IP 주소여야 합니다.



참고

IPV6 주소를 사용하여 스토리지 클러스터를 배포하려면 **--mon-ip IP_ADDRESS** 옵션에 **IPV6** 주소 형식을 사용합니다. 예: **cephadm 부트스트랩 --mon-ip 2620:52:0:880:225:90ff:febc:2536 --registry-json /etc/mylogin.json.**



중요

Red Hat Ceph Storage 5.1에서 **Ceph Object Gateway** 다중 사이트 구성은 여러 미해결 문제로 인해 지원되지 않습니다. 자세한 내용은 기술 자료 문서 [Red Hat Ceph Storage 5.1이 다중 사이트 구성을 지원하지 않음](#) 을 참조하십시오.

새로운 **Red Hat Ceph Storage** 클러스터를 부트 스트랩하는 동안 **--yes-i-know** 플래그를 사용하여 다중 사이트 회귀에 대한 경고를 출력합니다.



참고

Red Hat Ceph Storage 5.0z4를 새로 설치하려는 경우 부트스트랩 절차에 따라 지식베이스 문서인 [Red Hat Ceph Storage 4.2z4에서 5.0z4로 업그레이드 하는 방법](#)을 따르십시오.

절차

- 스토리지 클러스터를 부트스트랩합니다.

구문

```
cephadm bootstrap --cluster-network NETWORK_CIDR --mon-ip IP_ADDRESS --registry-url
registry.redhat.io --registry-username USER_NAME --registry-password PASSWORD --yes-
i-know
```

예제

```
[root@host01 ~]# cephadm bootstrap --cluster-network 10.10.128.0/24 --mon-ip
10.10.128.68 --registry-url registry.redhat.io --registry-username myuser1 --registry-
password mypassword1 --yes-i-know
```



참고

공용 네트워크를 통해 내부 클러스터 트래픽을 라우팅하려면 **--cluster-network NETWORK_CIDR** 옵션을 생략할 수 있습니다.

스크립트가 완료되는 데 몇 분이 걸립니다. 스크립트가 완료되면 **Red Hat Ceph Storage** 대시보드 URL에 인증 정보, **Ceph CLI**(명령줄 인터페이스)에 액세스하는 명령, **Telemetry**를 활성화하는 요청을 제공합니다.

예제

Ceph Dashboard is now available at:

```
URL: https://host01:8443/
User: admin
Password: i8nhu7zham
```

Enabling client.admin keyring and conf on hosts with "admin" label
You can access the Ceph CLI with:

```
sudo /usr/sbin/cephadm shell --fsid 266ee7a8-2a05-11eb-b846-5254002d4916 -c
/etc/ceph/ceph.conf -k /etc/ceph/ceph.client.admin.keyring
```

Please consider enabling telemetry to help improve Ceph:

```
ceph telemetry on
```

For more information see:

<https://docs.ceph.com/docs/master/mgr/telemetry/>

Bootstrap complete.

추가 리소스

- 권장 부트스트랩 명령 옵션에 대한 자세한 내용은 [권장 `cephadm` 부트스트랩 명령 옵션](#)을 참조하십시오.
- 부트스트랩 명령에 사용할 수 있는 옵션에 대한 자세한 내용은 부트스트랩 [명령 옵션](#)을 참조하십시오.
- 부트스트랩 프로세스의 로그인 인증 정보를 포함하도록 **JSON** 파일을 사용하는 방법에 대한 자세한 내용은 [JSON 파일을 사용하여 로그인 정보 보호](#)를 참조하십시오.

3.12.1. 권장되는 `cephadm` 부트스트랩 명령 옵션

`cephadm bootstrap` 명령에는 파일 위치 지정, `ssh` 설정 구성, 암호 설정, 다른 초기 구성 작업을 수행할 수 있는 여러 옵션이 있습니다.

`cephadm bootstrap`에 대한 기본 명령 옵션 세트를 사용하는 것이 좋습니다. 초기 클러스터가 가동되어 실행된 후 추가 옵션을 구성할 수 있습니다.

다음 예제에서는 권장 옵션을 지정하는 방법을 보여줍니다.

구문

```
cephadm bootstrap --ssh-user USER_NAME --mon-ip IP_ADDRESS --allow-fqdn-hostname --
registry-json REGISTRY_JSON
```

예제

```
[root@host01 ~]# cephadm bootstrap --ssh-user ceph-admin --mon-ip 10.10.128.68 --allow-fqdn-
hostname --registry-json /etc/mylogin.json
```

루트가 아닌 사용자는 [Creating an Ansible user with sudo access](#) 섹션 및 [Enabling passwordless SSH for Ansible](#) 섹션을 참조하십시오.

추가 리소스

- **--registry-json** 옵션에 대한 자세한 내용은 [JSON 파일을 사용하여 로그인 정보를 보호](#)를 참조하십시오.
- 사용 가능한 모든 **cephadm bootstrap** 옵션에 대한 자세한 내용은 [부트스트랩 명령 옵션](#)을 참조하십시오.
- 루트가 아닌 사용자로 스토리지 클러스터를 부트스트랩하는 방법에 대한 자세한 내용은 [루트가 아닌 사용자로 스토리지 클러스터 부팅](#)을 참조하십시오.

3.12.2. JSON 파일을 사용하여 로그인 정보 보호

스토리지 관리자는 **JSON** 파일에 로그인 및 암호 정보를 추가하도록 선택할 수 있으며 부트스트랩을 위해 **JSON** 파일을 참조하십시오. 이렇게 하면 로그인 인증 정보가 노출되지 않도록 보호됩니다.



참고

`cephadm --registry-login` 명령과 함께 **JSON** 파일을 사용할 수도 있습니다.

사전 요구 사항

- 첫 번째 **Ceph Monitor** 컨테이너의 **IP** 주소이며, 이는 스토리지 클러스터의 첫 번째 노드의 **IP** 주소이기도 합니다.
- **registry.redhat.io** 에 대한 로그인 액세스
- `/var/lib/containers/` 용으로 최소 **10GB**의 여유 공간이 필요
- 모든 노드에 대한 루트 수준 액세스.

절차

1. **JSON** 파일을 생성합니다. 이 예제에서 파일의 이름은 **mylogin.json** 입니다.

구문

```
{
  "url":"REGISTRY_URL",
  "username":"USER_NAME",
  "password":"PASSWORD"
}
```

예제

```
{
  "url":"registry.redhat.io",
  "username":"myuser1",
}
```

```
"password":"mypassword1"
}
```

2. 스토리지 클러스터를 부트스트랩합니다.

구문

```
cephadm bootstrap --mon-ip IP_ADDRESS --registry-json /etc/mylogin.json
```

예제

```
[root@host01 ~]# cephadm bootstrap --mon-ip 10.10.128.68 --registry-json /etc/mylogin.json
```

3.12.3. 서비스 구성 파일을 사용하여 스토리지 클러스터 부트스트랩

스토리지 클러스터를 부트스트랩하고 서비스 구성 파일을 사용하여 추가 호스트 및 데몬을 구성하려면 **cephadm bootstrap** 명령과 함께 **--apply-spec** 옵션을 사용합니다. 구성 파일은 배포하려는 서비스 유형, 배치 및 지정된 노드가 포함된 **.yaml** 파일입니다.



참고

다중 사이트 등의 애플리케이션에 기본이 아닌 영역 또는 영역을 사용하려면 구성 파일에 추가하고 **--apply-spec** 옵션을 사용하여 스토리지 클러스터를 부트스트랩한 후 **Ceph Object Gateway** 데몬을 구성합니다. 그러면 **Ceph Object Gateway** 데몬을 배포하기 전에 필요한 영역 또는 영역을 만들 수 있습니다. 자세한 내용은 [Red Hat Ceph Storage Operations 가이드](#)를 참조하십시오.



참고

Ceph iSCSI 게이트웨이, **NFS-Ganesha** 게이트웨이 또는 메타데이터 서버(**MDS**) 서비스를 배포하는 경우 스토리지 클러스터를 부트 스트랩한 후 구성합니다.

- **Ceph iSCSI** 게이트웨이 또는 **Ceph NFS-Ganesha** 게이트웨이를 배포하려면 먼저 **RADOS** 풀을 생성해야 합니다.
- **MDS** 서비스를 배포하려면 먼저 **CephFS** 볼륨을 생성해야 합니다.

자세한 내용은 [Red Hat Ceph Storage Operations 가이드](#)를 참조하십시오.



참고

Red Hat Ceph Storage 5.1부터 **--apply-spec** 옵션으로 부트스트랩 명령을 실행하는 경우 사양 파일에 부트스트랩 호스트의 **IP** 주소를 포함해야 합니다. 이렇게 하면 활성 **Ceph Manager**가 이미 실행 중인 부트스트랩 호스트를 다시 추가하는 동안 **IP** 주소를 루프백 주소로 확인할 수 없습니다.

부트스트랩 중에 **--apply spec** 옵션을 사용하지 않고 대신 호스트 다시 추가 및 실행 중인 활성 **Ceph Manager**가 포함된 다른 사양 파일과 함께 **ceph orch apply** 명령을 사용하는 경우, **addr** 필드를 명시적으로 제공해야 합니다. 부트 스트랩 후 모든 사양 파일을 적용하는 데 적용됩니다.

사전 요구 사항

- 실행 중인 **VM**(가상 머신) 또는 서버 중 하나 이상
- **Red Hat Enterprise Linux 8.4 EUS** 이상.
- 모든 노드에 대한 루트 수준 액세스.
- **registry.redhat.io** 에 대한 로그인 액세스

- 암호 없는 **ssh**는 스토리지 클러스터의 모든 호스트에 설정됩니다.
- **cephadm** 은 스토리지 클러스터의 첫 번째 모니터 노드가 될 노드에 설치됩니다.

절차

1. 부트스트랩 호스트에 로그인합니다.
2. 스토리지 클러스터에 대한 서비스 구성 **.yaml** 파일을 생성합니다. 예제 파일에서는 **cephadm** 부트스트랩 에 초기 호스트와 두 개의 추가 호스트를 구성하도록 지시합니다. 이 파일은 사용 가능한 모든 디스크에서 **OSD**를 만들도록 지정합니다.

예제

```

service_type: host
addr: host01
hostname: host01
---
service_type: host
addr: host02
hostname: host02
---
service_type: host
addr: host03
hostname: host03
---
service_type: host
addr: host04
hostname: host04
---
service_type: mon
placement:
  host_pattern: "host[0-2]"
---
service_type: osd
service_id: my_osds
placement:
  host_pattern: "host[1-3]"
data_devices:
  all: true

```

3.

--apply-spec 옵션을 사용하여 스토리지 클러스터를 부트스트랩합니다.

구문

```
cephadm bootstrap --apply-spec CONFIGURATION_FILE_NAME --mon-ip
MONITOR_IP_ADDRESS --ssh-private-key PRIVATE_KEY --ssh-public-key PUBLIC_KEY -
-registry-url registry.redhat.io --registry-username USER_NAME --registry-password
PASSWORD
```

예제

```
[root@host01 ~]# cephadm bootstrap --apply-spec initial-config.yaml --mon-ip 10.10.128.68 -
-ssh-private-key /home/ceph/.ssh/id_rsa --ssh-public-key /home/ceph/.ssh/id_rsa.pub --
registry-url registry.redhat.io --registry-username myuser1 --registry-password mypassword1
```

스크립트가 완료되는 데 몇 분이 걸립니다. 스크립트가 완료되면 **Red Hat Ceph Storage** 대시보드 **URL**에 인증 정보, **Ceph CLI**(명령줄 인터페이스)에 액세스하는 명령, **Telemetry**를 활성화하는 요청을 제공합니다.

4.

스토리지 클러스터가 가동되어 실행되면 추가 데몬 및 서비스 구성에 대한 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#)를 참조하십시오.

추가 리소스

•

부트스트랩 명령에 사용할 수 있는 옵션에 대한 자세한 내용은 [Bootstrap 명령 옵션](#)을 참조하십시오.

3.12.4. 루트가 아닌 사용자로 스토리지 클러스터 부트스트랩

부트스트랩 노드에서 루트가 아닌 사용자로 **Red Hat Ceph Storage** 클러스터를 부트스트랩하려면

cephadm bootstrap 명령과 함께 **--ssh-user** 옵션을 사용합니다. **--ssh-user**는 클러스터 노드에 대한 SSH 연결을 위한 사용자를 지정합니다.

루트가 아닌 사용자는 암호 없이 **sudo** 액세스 권한이 있어야 합니다. 자세한 내용은 [Creating an Ansible user with sudo access](#) 섹션 및 [Enabling password-less SSH for Ansible_](#) 섹션을 참조하십시오.

사전 요구 사항

- 스토리지 클러스터의 초기 모니터 노드에 대한 IP 주소이기도 한 첫 번째 Ceph Monitor 컨테이너의 IP 주소입니다
- **registry.redhat.io** 에 대한 로그인 액세스
- **/var/lib/containers/** 용으로 최소 **10GB**의 여유 공간이 필요
- **SSH** 공개 및 개인 키
- 부트 스트랩 노드에 대한 암호 없는 **sudo** 액세스

절차

1. 부트 스트랩 노드에서 **sudo**로 변경합니다.

구문

```
su - SSH_USER_NAME
```

예제

```
[root@host01 ~]# su - ceph
Last login: Tue Sep 14 12:00:29 EST 2021 on pts/0
```

2.

부트 스트랩 노드에 대한 **SSH** 연결을 설정합니다.

예제

```
[ceph@host01 ~]# ssh host01
Last login: Tue Sep 14 12:03:29 EST 2021 on pts/0
```

3.

선택 사항: **cephadm bootstrap** 명령을 호출합니다.



참고

개인 키 및 공개 키는 선택 사항입니다. **SSH** 키가 이전에 생성되지 않은 경우 이 단계에서 생성할 수 있습니다.

--ssh-private-key 및 **--ssh-public-key** 옵션을 포함합니다.

구문

```
cephadm bootstrap --ssh-user USER_NAME --mon-ip IP_ADDRESS --ssh-private-key PRIVATE_KEY --ssh-public-key PUBLIC_KEY --registry-url registry.redhat.io --registry-username USER_NAME --registry-password PASSWORD
```

예제

```
cephadm bootstrap --ssh-user ceph-admin --mon-ip 10.10.128.68 --ssh-private-key
/home/ceph/.ssh/id_rsa --ssh-public-key /home/ceph/.ssh/id_rsa.pub --registry-url
registry.redhat.io --registry-username myuser1 --registry-password mypassword1
```

추가 리소스

- 사용 가능한 모든 **cephadm bootstrap** 옵션에 대한 자세한 내용은 [부트스트랩 명령 옵션](#)을 참조하십시오.
- **Ansible**을 사용하여 루트 없는 클러스터 부트스트랩을 자동화하는 방법에 대한 자세한 내용은 [Red Hat Ceph Storage 5 rootless deployment utilizing ansible ad-hoc commands](#)에서 참조하십시오.

3.12.5. 부트스트랩 명령 옵션

cephadm bootstrap 명령은 로컬 호스트에 **Ceph** 스토리지 클러스터를 부트스트랩합니다. 부트스트랩 노드에 **MON** 데몬과 **MGR** 데몬을 배포하고, 로컬 호스트에 모니터링 스택을 자동으로 배포한 다음 **ceph orch host add HOSTNAME**을 호출합니다.

다음 표에는 **cephadm bootstrap**에 사용할 수 있는 옵션이 나열되어 있습니다.

cephadm bootstrap 옵션	설명
<code>--config CONFIG_FILE, -c CONFIG_FILE</code>	<code>CONFIG_FILE</code> 은 부트스트랩 명령과 함께 사용할 ceph.conf 파일입니다.
<code>--cluster-network NETWORK_CIDR</code>	내부 클러스터 트래픽에 대해 <code>NETWORK_CIDR</code> 에서 정의한 서브넷을 사용합니다. CIDR 표기법으로 지정됩니다. 예를 들어 다음과 같습니다. 10.10.128.0/24 .
<code>--Mon-id MON_ID</code>	<code>MON_ID</code> 라는 호스트의 부트스트랩 기본값은 로컬 호스트입니다.
<code>--Mon-addrv MON_ADDRV</code>	Mon IP (예: [v2:localipaddr:3300,v1:localipaddr:6789])

cephadm bootstrap 옵션	설명
--Mon-ip <i>IP_ADDRESS</i>	cephadm bootstrap 을 실행하는 데 사용 중인 노드의 IP 주소입니다.
--mgr-id <i>MGR_ID</i>	MGR 노드를 설치해야 하는 호스트 ID입니다. 기본값: 임의로 생성됩니다.
--fsid <i>FSID</i>	클러스터 FSID.
--output-dir <i>OUTPUT_DIR</i>	이 디렉토리를 사용하여 config, 인증 키 및 pub 키 파일을 작성합니다.
--output-keyring <i>OUTPUT_KEYRING</i>	이 위치를 사용하여 새 클러스터 admin 및 mon 키가 있는 인증 키 파일을 작성합니다.
--output-config <i>OUTPUT_CONFIG</i>	이 위치를 사용하여 새 클러스터에 연결할 구성 파일을 작성합니다.
--output-pub-ssh-key <i>OUTPUT_PUB_SSH_KEY</i>	이 위치를 사용하여 클러스터에 대한 공용 SSH 키를 작성합니다.
--skip-ssh	로컬 호스트에서 ssh 키의 설정을 건너뜁니다.
--initial-dashboard-user <i>INITIAL_DASHBOARD_USER</i>	대시보드의 초기 사용자입니다.
--initial-dashboard-password <i>INITIAL_DASHBOARD_PASSWORD</i>	초기 대시보드 사용자의 초기 암호입니다.
--SSL-dashboard-port <i>SSL_DASHBOARD_PORT</i>	SSL을 사용하여 대시보드와 연결하는 데 사용되는 포트 번호입니다.
--dashboard-key <i>DASHBOARD_KEY</i>	대시보드 키.
--dashboard-crt <i>DASHBOARD_CRT</i>	대시보드 인증서.
--SSH-config <i>SSH_CONFIG</i>	SSH 구성.
--SSH-private-key <i>SSH_PRIVATE_KEY</i>	SSH 개인 키.
--SSH-public-key <i>SSH_PUBLIC_KEY</i>	SSH 공개 키.
--SSH-user <i>SSH_USER</i>	클러스터 호스트에 대한 SSH 연결 사용자를 설정합니다. 루트가 아닌 사용자에게 암호가 없는 sudo가 필요합니다.

cephadm bootstrap 옵션	설명
--skip-mon-network	부트 스트랩 mon ip를 기반으로 하는 mon public_network를 설정합니다.
--skip-dashboard	Ceph 대시보드를 활성화하지 마십시오.
--dashboard-password-noupdate	강제 대시보드 암호 변경을 비활성화합니다.
--no-minimize-config	구성 파일을 시뮬레이션하고 최소화하지 마십시오.
--skip-ping-check	mon IP가 ping할 수 있는지 확인하지 마십시오.
--skip-pull	부트 스트랩하기 전에 최신 이미지를 가져오지 마십시오.
--skip-firewalld	firewalld를 구성하지 마십시오.
--allow-overwrite	기존 -output-* config/keyring/ssh 파일의 덮어쓰기를 허용합니다.
--allow-fqdn-hostname	정규화된 호스트 이름을 허용합니다.
--skip-prepare-host	호스트를 준비하지 마십시오.
--orphan-initial-daemons	초기 mon, mgr 및 크래시 서비스 사양을 생성하지 마십시오.
--skip-monitoring-stack	모니터링 스택(prometheus, grafana, alertmanager, node-exporter)을 자동으로 프로비저닝하지 마십시오.
--apply-spec <i>APPLY_SPEC</i>	부트스트랩 후에 클러스터 사양 파일을 적용합니다 (ssh 키 복사, 호스트 추가 및 서비스 적용).
--registry-url <i>REGISTRY_URL</i>	로그인할 사용자 정의 레지스트리의 URL을 지정합니다. 예: registry.redhat.io .
--registry-username <i>REGISTRY_USERNAME</i>	사용자 정의 레지스트리에 대한 로그인 계정의 사용자 이름입니다.
--registry-password <i>REGISTRY_PASSWORD</i>	사용자 정의 레지스트리에 대한 로그인 계정의 암호입니다.
--registry-json <i>REGISTRY_JSON</i>	레지스트리 로그인 정보가 포함된 JSON 파일입니다.

추가 리소스

- **--skip-monitoring-stack** 옵션에 대한 자세한 내용은 [호스트 추가](#)를 참조하십시오.
- **registry-json** 옵션을 사용하여 레지스트리에 로그인하는 방법에 대한 자세한 내용은 **registry-login** 명령에 대한 도움말을 참조하십시오.
- **cephadm** 옵션에 대한 자세한 내용은 **cephadm** 에 대한 도움말을 참조하십시오.

3.12.6. 연결이 끊긴 설치를 위한 프라이빗 레지스트리 구성

연결이 끊긴 설치 절차를 사용하여 **cephadm** 을 설치하고 프라이빗 네트워크에 스토리지 클러스터를 부트스트랩할 수 있습니다. 연결이 끊긴 설치에서는 설치를 위해 프라이빗 레지스트리를 사용합니다. 배포 중에 **Red Hat Ceph Storage** 노드가 인터넷에 액세스할 수 없는 경우 다음 절차를 사용하십시오.

인증 및 자체 서명된 인증서를 사용하여 보안 프라이빗 레지스트리를 설정하려면 다음 절차를 따르십시오. 인터넷에 액세스할 수 있고 로컬 클러스터에 액세스할 수 있는 노드에서 다음 단계를 수행합니다.



참고

프로덕션에 비보안 레지스트리를 사용하는 것은 권장되지 않습니다.

사전 요구 사항

- 인터넷이 연결된 VM(가상 머신) 또는 서버가 하나 이상 실행되고 있습니다.
- **Red Hat Enterprise Linux 8.4 EUS** 이상.
- **registry.redhat.io** 에 대한 로그인 액세스
- 모든 노드에 대한 루트 수준 액세스.

절차

1. 공용 네트워크와 클러스터 노드에 모두 액세스할 수 있는 노드에 로그인합니다.
2. 노드를 등록하고 메시지가 표시되면 적절한 **Red Hat Customer Portal** 자격 증명을 입력합니다.

예제

```
[root@admin ~]# subscription-manager register
```

3. 최신 서브스크립션 데이터를 가져옵니다.

예제

```
[root@admin ~]# subscription-manager refresh
```

4. **Red Hat Ceph Storage**에 사용 가능한 모든 서브스크립션을 나열합니다.

예제

```
[root@admin ~]# subscription-manager list --available --all --matches="*Ceph*"
```

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

5.

서브스크립션을 첨부하여 소프트웨어 권한에 액세스할 수 있습니다.

구문

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

*POOL_ID*를 이전 단계에서 확인한 풀 ID로 교체합니다.

6.

기본 소프트웨어 리포지토리를 비활성화하고 서버와 추가 리포지토리를 활성화합니다.

Red Hat Enterprise Linux 8

```
[root@admin ~]# subscription-manager repos --disable=*  
[root@admin ~]# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms  
[root@admin ~]# subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
```

Red Hat Enterprise Linux 9

```
[root@admin ~]# subscription-manager repos --disable=*  
[root@admin ~]# subscription-manager repos --enable=rhel-9-for-x86_64-baseos-rpms  
[root@admin ~]# subscription-manager repos --enable=rhel-9-for-x86_64-appstream-rpms
```

7.

podman 및 **httpd-tools** 패키지를 설치합니다.

예제

```
[root@admin ~]# dnf install -y podman httpd-tools
```

8. 프라이빗 레지스트리의 폴더를 생성합니다.

예제

```
[root@admin ~]# mkdir -p /opt/registry/{auth,certs,data}
```

레지스트리는 **/opt/registry** 에 저장되고 디렉터리는 레지스트리를 실행하는 컨테이너에 마운트됩니다.

- **auth** 디렉터리는 레지스트리가 인증에 사용하는 **htpasswd** 파일을 저장합니다.
- **certs** 디렉터리는 레지스트리가 인증에 사용하는 인증서를 저장합니다.
- **data** 디렉터리는 레지스트리 이미지를 저장합니다.

9. 프라이빗 레지스트리에 액세스하기 위한 인증 정보를 생성합니다.

구문

```
htpasswd -bBc /opt/registry/auth/htpasswd PRIVATE_REGISTRY_USERNAME  
PRIVATE_REGISTRY_PASSWORD
```

- **b** 옵션은 명령줄의 암호를 제공합니다.
- **B** 옵션은 **Bcrypt** 암호화를 사용하여 암호를 저장합니다.
- **c** 옵션은 **htpasswd** 파일을 생성합니다.
- **PRIVATE_REGISTRY_USERNAME**을 프라이빗 레지스트리에 대해 생성할 사용자 이름으로 교체합니다.
- **PRIVATE_REGISTRY_PASSWORD**를 프라이빗 레지스트리 사용자 이름에 생성할 암호로 바꿉니다.

예제

```
[root@admin ~]# htpasswd -bBc /opt/registry/auth/htpasswd myregistryusername myregistrypassword1
```

10.

자체 서명된 인증서를 생성합니다.

구문

```
openssl req -newkey rsa:4096 -nodes -sha256 -keyout /opt/registry/certs/domain.key -x509 -days 365 -out /opt/registry/certs/domain.crt -addext "subjectAltName = DNS:LOCAL_NODE_FQDN"
```

• **LOCAL_NODE_FQDN**을 프라이빗 레지스트리 노드의 정규화된 호스트 이름으로 교체합니다.



참고

인증서의 각 옵션에 대한 메시지가 표시됩니다. **CN=** 값은 노드의 호스트 이름이며 **DNS** 또는 **/etc/hosts** 파일에서 확인할 수 있어야 합니다.

예제

```
[root@admin ~]# openssl req -newkey rsa:4096 -nodes -sha256 -keyout
/opt/registry/certs/domain.key -x509 -days 365 -out /opt/registry/certs/domain.crt -addext
"subjectAltName = DNS:admin.lab.redhat.com"
```



참고

자체 서명된 인증서를 생성할 때 적절한 **SAN**(주체 대체 이름)을 사용하여 인증서를 생성해야 합니다. 적절한 **SAN**이 포함되지 않은 인증서에 대해 **TLS** 확인이 필요한 **Podman** 명령은 다음 오류를 반환합니다. **x509: certificate relies on legacy Common Name field, use SANs or temporarily enable Common Name matching with GODEBUG=x509ignoreCN=0**

11.

skopeo가 파일 확장자 **.cert**인 **.cert**인 인증서를 찾을 수 있도록 **domain.cert**에 대한 심볼릭 링크를 만듭니다.

예제

```
[root@admin ~]# ln -s /opt/registry/certs/domain.crt /opt/registry/certs/domain.cert
```

12.

프라이빗 레지스트리 노드의 신뢰할 수 있는 목록에 인증서를 추가합니다.

구문

```
cp /opt/registry/certs/domain.crt /etc/pki/ca-trust/source/anchors/
update-ca-trust
trust list | grep -i "LOCAL_NODE_FQDN"
```

LOCAL_NODE_FQDN을 프라이빗 레지스트리 노드의 **FQDN**으로 바꿉니다.

예제

```
[root@admin ~]# cp /opt/registry/certs/domain.crt /etc/pki/ca-trust/source/anchors/
[root@admin ~]# update-ca-trust
[root@admin ~]# trust list | grep -i "admin.lab.redhat.com"

label: admin.lab.redhat.com
```

13.

인증서를 설치하기 위해 프라이빗 레지스트리에 액세스하는 노드에 복사하고 신뢰할 수 있는 목록을 업데이트합니다.

예제

```
[root@admin ~]# scp /opt/registry/certs/domain.crt root@host01:/etc/pki/ca-
trust/source/anchors/
[root@admin ~]# ssh root@host01
[root@host01 ~]# update-ca-trust
[root@host01 ~]# trust list | grep -i "admin.lab.redhat.com"

label: admin.lab.redhat.com
```

14.

로컬 보안 프라이빗 레지스트리를 시작합니다.

구문

```
[root@admin ~]# podman run --restart=always --name NAME_OF_CONTAINER \
-p 5000:5000 -v /opt/registry/data:/var/lib/registry:z \
-v /opt/registry/auth:/auth:z \
-v /opt/registry/certs:/certs:z \
-e "REGISTRY_AUTH=htpasswd" \
-e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
-e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
-e "REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt" \
-e "REGISTRY_HTTP_TLS_KEY=/certs/domain.key" \
-e REGISTRY_COMPATIBILITY_SCHEMA1_ENABLED=true \
-d registry:2
```

NAME_OF_CONTAINER 를 컨테이너에 할당할 이름으로 교체합니다.

예제

```
[root@admin ~]# podman run --restart=always --name myprivateregistry \
-p 5000:5000 -v /opt/registry/data:/var/lib/registry:z \
-v /opt/registry/auth:/auth:z \
-v /opt/registry/certs:/certs:z \
-e "REGISTRY_AUTH=htpasswd" \
-e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
-e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
-e "REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt" \
-e "REGISTRY_HTTP_TLS_KEY=/certs/domain.key" \
-e REGISTRY_COMPATIBILITY_SCHEMA1_ENABLED=true \
-d registry:2
```

이는 포트 5000에서 개인 레지스트리를 시작하고 레지스트리를 실행하는 컨테이너에 레지스트리 디렉터리의 볼륨을 마운트합니다.

15.

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

a.

/etc/containers/registries.conf 파일을 편집하려면 이를 열고 **registry.redhat.io**를 **unqualified-search-registries** 목록에 추가합니다.

예제

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

16.

Red Hat Customer Portal 인증 정보를 사용하여 **registry.redhat.io**에 로그인합니다.

구문

```
podman login registry.redhat.io
```

17.

Red Hat 고객 포털에서 프라이빗 레지스트리로 다음 **Red Hat Ceph Storage 5** 이미지, **Prometheus** 이미지 및 대시보드 이미지를 복사합니다.

표 3.1. 모니터링 스택의 사용자 정의 이미지 세부 정보

스택 구성 요소 모니터링	이미지 세부 정보
Prometheus	registry.redhat.io/openshift4/ose-prometheus:v4.10
Grafana	registry.redhat.io/rhceph/rhceph-5-dashboard-rhel8:latest

스택 구성 요소 모니터링	이미지 세부 정보
node-exporter	registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.10
AlertManager	registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.10
HAProxy	registry.redhat.io/rhceph/rhceph-haproxy-rhel8:latest
Keepalived	registry.redhat.io/rhceph/keepalived-rhel8:latest
SNMP Gateway	registry.redhat.io/rhceph/snmp-notifier-rhel8:latest

구문

```
podman run -v /CERTIFICATE_DIRECTORY_PATH:/certs:Z -v
/CERTIFICATE_DIRECTORY_PATH/domain.cert:/certs/domain.cert:Z --rm
registry.redhat.io/rhel8/skopeo:8.5-8 skopeo copy --remove-signatures --src-creds
RED_HAT_CUSTOMER_PORTAL_LOGIN:RED_HAT_CUSTOMER_PORTAL_PASSWORD
--dest-cert-dir=./certs/ --dest-creds
PRIVATE_REGISTRY_USERNAME:PRIVATE_REGISTRY_PASSWORD
docker://registry.redhat.io/SRC_IMAGE:SRC_TAG
docker://LOCAL_NODE_FQDN:5000/DST_IMAGE:DST_TAG
```

- **CERTIFICATE_DIRECTORY_PATH**를 자체 서명된 인증서의 디렉터리 경로로 바꿉니다.
- **RED_HAT_CUSTOMER_PORTAL_LOGIN** 및 **RED_HAT_CUSTOMER_PORTAL_PASSWORD**를 Red Hat Customer Portal 인증 정보로 교체합니다.
- **PRIVATE_REGISTRY_USERNAME** 및 **PRIVATE_REGISTRY_PASSWORD**를 프라이빗 레지스트리 인증 정보로 교체합니다.
- **SRC_IMAGE** 및 **SRC_TAG**를 registry.redhat.io에서 복사할 이미지의 이름 및 태그로 교체합니다.

- ***DST_IMAGE*** 및 ***DST_TAG***를 프라이빗 레지스트리에 복사할 이미지의 이름 및 태그로 바꿉니다.
- ***LOCAL_NODE_FQDN***을 프라이빗 레지스트리의 **FQDN**으로 바꿉니다.

예제

```
[root@admin ~]# podman run -v /opt/registry/certs:/certs:Z -v
/opt/registry/certs/domain.cert:/certs/domain.cert:Z --rm
registry.redhat.io/rhel8/skopeo:8.5-8 skopeo copy --remove-signatures --src-creds
myusername:mypassword1 --dest-cert-dir=./certs/ --dest-creds
myregistryusername:myregistrypassword1 docker://registry.redhat.io/rhceph/rhceph-5-
rhel8:latest docker://admin.lab.redhat.com:5000/rhceph/rhceph-5-rhel8:latest
```

```
[root@admin ~]# podman run -v /opt/registry/certs:/certs:Z -v
/opt/registry/certs/domain.cert:/certs/domain.cert:Z --rm
registry.redhat.io/rhel8/skopeo:8.5-8 skopeo copy --remove-signatures --src-creds
myusername:mypassword1 --dest-cert-dir=./certs/ --dest-creds
myregistryusername:myregistrypassword1 docker://registry.redhat.io/openshift4/ose-
prometheus-node-exporter:v4.10 docker://admin.lab.redhat.com:5000/openshift4/ose-
prometheus-node-exporter:v4.10
```

```
[root@admin ~]# podman run -v /opt/registry/certs:/certs:Z -v
/opt/registry/certs/domain.cert:/certs/domain.cert:Z --rm
registry.redhat.io/rhel8/skopeo:8.5-8 skopeo copy --remove-signatures --src-creds
myusername:mypassword1 --dest-cert-dir=./certs/ --dest-creds
myregistryusername:myregistrypassword1 docker://registry.redhat.io/rhceph/rhceph-5-
dashboard-rhel8:latest docker://admin.lab.redhat.com:5000/rhceph/rhceph-5-dashboard-
rhel8:latest
```

```
[root@admin ~]# podman run -v /opt/registry/certs:/certs:Z -v
/opt/registry/certs/domain.cert:/certs/domain.cert:Z --rm
registry.redhat.io/rhel8/skopeo:8.5-8 skopeo copy --remove-signatures --src-creds
myusername:mypassword1 --dest-cert-dir=./certs/ --dest-creds
myregistryusername:myregistrypassword1 docker://registry.redhat.io/openshift4/ose-
prometheus:v4.10 docker://admin.lab.redhat.com:5000/openshift4/ose-prometheus:v4.10
```

```
[root@admin ~]# podman run -v /opt/registry/certs:/certs:Z -v
/opt/registry/certs/domain.cert:/certs/domain.cert:Z --rm
registry.redhat.io/rhel8/skopeo:8.5-8 skopeo copy --remove-signatures --src-creds
myusername:mypassword1 --dest-cert-dir=./certs/ --dest-creds
myregistryusername:myregistrypassword1 docker://registry.redhat.io/openshift4/ose-
prometheus-alertmanager:v4.10 docker://admin.lab.redhat.com:5000/openshift4/ose-
prometheus-alertmanager:v4.10
```

18.

curl 명령을 사용하여 이미지가 로컬 레지스트리에 있는지 확인합니다.

구문

```
curl -u PRIVATE_REGISTRY_USERNAME:PRIVATE_REGISTRY_PASSWORD
https://LOCAL_NODE_FQDN:5000/v2/_catalog
```

예제

```
[root@admin ~]# curl -u myregistryusername:myregistrypassword1
https://admin.lab.redhat.com:5000/v2/_catalog

{"repositories":["openshift4/ose-prometheus","openshift4/ose-prometheus-
alertmanager","openshift4/ose-prometheus-node-exporter","rhceph/rhceph-5-dashboard-
rhel8","rhceph/rhceph-5-rhel8"]}
```

추가 리소스

- 다양한 이미지 **Ceph** 패키지 버전에 대한 자세한 내용은 지식 기반 솔루션의 [What are the Red Hat Ceph Storage releases and corresponding Ceph package versions?](#)을 참조하십시오.

3.12.7. 연결이 끊긴 설치를 위해 **preflight** Playbook 실행

cephadm-preflight.yml Ansible 플레이북을 사용하여 **Ceph** 리포지토리를 구성하고 부트스트랩을 위해 스토리지 클러스터를 준비합니다. **podman**, **lvm2**, **chrony**, **cephadm** 과 같은 몇 가지 사전 요구 사항도 설치합니다.



중요

cephadm-preflight Playbook은 지원되지 않으므로 **Red Hat Enterprise Linux 9**에 대해 이 단계를 건너뛰니다.

preflight 플레이북은 **cephadm-ansible** 인벤토리 **hosts** 파일을 사용하여 스토리지 클러스터의 모든 노드를 식별합니다. **cephadm-ansible**, **cephadm-preflight.yml**, 인벤토리 **hosts** 파일의 기본 위치는 **/usr/share/cephadm-ansible/**입니다.

다음 예제는 일반적인 인벤토리 파일의 구조를 보여줍니다.

예제

```
host02
host03
host04

[admin]
host01
```

인벤토리 파일의 **[admin]** 그룹에는 관리자 인증 키가 저장된 노드의 이름이 포함되어 있습니다.



참고

초기 호스트를 부트스트랩하기 전에 **preflight** 플레이북을 실행합니다.

사전 요구 사항

- **cephadm-ansible** 패키지가 **Ansible** 관리 노드에 설치되어 있습니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

- 암호 없는 **ssh**는 스토리지 클러스터의 모든 호스트에 설정됩니다.
- 다음 리포지토리가 활성화된 로컬 **YUM** 리포지토리 서버에 액세스하도록 구성된 노드.
 - **rhel-8-for-x86_64-baseos-rpms**
 - **rhel-8-for-x86_64-appstream-rpms**
 - **rhceph-5-tools-for-rhel-8-x86_64-rpms**



참고

로컬 **YUM** 리포지토리 설정에 대한 자세한 내용은 기술 자료 문서 [Creating a Local Repository and Sharing with Disconnected/Offline/Air-gapped Systems](#)를 참조하십시오.

절차

1. **Ansible** 관리 노드의 `/usr/share/cephadm-ansible` 디렉터리로 이동합니다.
2. **hosts** 파일을 열고 편집하고 노드를 추가합니다.
3. 로컬 **YUM** 리포지토리를 사용하려면 **ceph_origin** 매개변수가 **custom**으로 설정된 상태에서 **preflight** 플레이북을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars
"ceph_origin=custom" -e "custom_repo_url=CUSTOM_REPO_URL"
```

예제

```
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --
extra-vars "ceph_origin=custom" -e
"custom_repo_url=http://mycustomrepo.lab.redhat.com/x86_64/os/"
```

설치가 완료되면 **cephadm** 은 **/usr/sbin/** 디렉터리에 있습니다.

4.

또는 **--limit** 옵션을 사용하여 스토리지 클러스터에서 선택한 호스트 집합에서 **preflight** 플레이북을 실행할 수 있습니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars
"ceph_origin=custom" -e "custom_repo_url=CUSTOM_REPO_URL" --limit
GROUP_NAME|NODE_NAME
```

GROUP_NAME 을 인벤토리 파일의 그룹 이름으로 바꿉니다. **NODE_NAME** 을 인벤토리 파일의 특정 노드 이름으로 바꿉니다.

예제

```
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --
extra-vars "ceph_origin=custom" -e
"custom_repo_url=http://mycustomrepo.lab.redhat.com/x86_64/os/" --limit clients
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --
extra-vars "ceph_origin=custom" -e
"custom_repo_url=http://mycustomrepo.lab.redhat.com/x86_64/os/" --limit host02
```

■



참고

preflight 플레이북을 실행하면 **cephadm-ansible** 이 클라이언트 노드에 **chrony** 및 **ceph-common** 을 자동으로 설치합니다.

3.12.8. 연결 해제된 설치 수행

설치를 수행하기 전에 **Red Hat** 레지스트리에 액세스할 수 있는 프록시 호스트에서 또는 이미지를 로컬 레지스트리에 복사하는 프록시 호스트에서 **Red Hat Ceph Storage** 컨테이너 이미지를 가져와야 합니다.



참고

로컬 레지스트리에서 로컬 레지스트리가 포함된 자체 서명된 인증서를 사용하는 경우 신뢰할 수 있는 루트 인증서를 부트스트랩 호스트에 추가했는지 확인하십시오. 자세한 내용은 [연결 해제된 설치에 대한 프라이빗 레지스트리 구성](#)을 참조하십시오.



참고

Red Hat Ceph Storage 5는 **Red Hat Enterprise Linux 8.4 EUS** 이상에서 지원됩니다.



중요

부트스트랩 프로세스를 시작하기 전에 사용하려는 컨테이너 이미지에 **cephadm**과 동일한 버전의 **Red Hat Ceph Storage**가 있는지 확인합니다. 두 버전이 일치하지 않으면 **Creating initial admin user** 단계에서 부트스트랩이 실패합니다.

사전 요구 사항

●

실행 중인 **VM**(가상 머신) 또는 서버 중 하나 이상

- 모든 노드에 대한 루트 수준 액세스.
- 암호 없는 **ssh**는 스토리지 클러스터의 모든 호스트에 설정됩니다.
- **preflight** 플레이북은 스토리지 클러스터의 부트스트랩 호스트에서 실행되었습니다. 자세한 내용은 [연결이 해제된 설치의 preflight Playbook 실행](#)을 참조하십시오.
- 프라이빗 레지스트리가 구성되었으며 부트스트랩 노드가 액세스할 수 있습니다. 자세한 내용은 [연결 해제된 설치에 대한 프라이빗 레지스트리 구성](#)을 참조하십시오.
- **Red Hat Ceph Storage** 컨테이너 이미지는 사용자 정의 레지스트리에 있습니다.

절차

1. 부트스트랩 호스트에 로그인합니다.
2. 스토리지 클러스터를 부트스트랩합니다.

구문

```
cephadm --image
PRIVATE_REGISTRY_NODE_FQDN:5000/CUSTOM_IMAGE_NAME:IMAGE_TAG
bootstrap --mon-ip IP_ADDRESS --registry-url PRIVATE_REGISTRY_NODE_FQDN:5000 --
registry-username PRIVATE_REGISTRY_USERNAME --registry-password
PRIVATE_REGISTRY_PASSWORD
```

- **PRIVATE_REGISTRY_NODE_FQDN**을 프라이빗 레지스트리의 정규화된 도메인 이름으로 교체합니다.
- **CUSTOM_IMAGE_NAME** 및 **IMAGE_TAG**를 프라이빗 레지스트리에 있는 **Red Hat**

Ceph Storage 컨테이너 이미지의 이름 및 태그로 교체합니다.

- **IP_ADDRESS**를 **cephadm bootstrap**을 실행하는 데 사용 중인 노드의 **IP** 주소로 바꿉니다.
- **PRIVATE_REGISTRY_USERNAME**을 프라이빗 레지스트리에 대해 생성할 사용자 이름으로 교체합니다.
- **PRIVATE_REGISTRY_PASSWORD**를 프라이빗 레지스트리 사용자 이름에 생성할 암호로 바꿉니다.

예제

```
[root@host01 ~]# cephadm --image admin.lab.redhat.com:5000/rhceph/rhceph-5-rhel8:latest bootstrap --mon-ip 10.10.128.68 --registry-url admin.lab.redhat.com:5000 --registry-username myregistryusername --registry-password myregistrypassword1
```

스크립트가 완료되는 데 몇 분이 걸립니다. 스크립트가 완료되면 **Red Hat Ceph Storage** 대시보드 **URL**에 인증 정보, **Ceph CLI**(명령줄 인터페이스)에 액세스하는 명령, **Telemetry**를 활성화하는 요청을 제공합니다.

Ceph Dashboard is now available at:

```
URL: https://host01:8443/
User: admin
Password: i8nhu7zham
```

Enabling client.admin keyring and conf on hosts with "admin" label
You can access the Ceph CLI with:

```
sudo /usr/sbin/cephadm shell --fsid 266ee7a8-2a05-11eb-b846-5254002d4916 -c /etc/ceph/ceph.conf -k /etc/ceph/ceph.client.admin.keyring
```

Please consider enabling telemetry to help improve Ceph:

```
ceph telemetry on
```

For more information see:

<https://docs.ceph.com/docs/master/mgr/telemetry/>

Bootstrap complete.

부트스트랩 프로세스가 완료된 후 [연결 해제된 설치의 사용자 지정 컨테이너 이미지 구성 변경](#)을 참조하십시오.

추가 리소스

- 스토리지 클러스터가 가동되어 실행되면 추가 데몬 및 서비스 구성에 대한 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#)를 참조하십시오.

3.12.9. 연결이 해제된 설치를 위한 사용자 정의 컨테이너 이미지 구성 변경

연결이 해제된 노드의 초기 부트스트랩을 수행한 후 스택 데몬 모니터링을 위한 사용자 지정 컨테이너 이미지를 지정해야 합니다. 노드가 기본 컨테이너 레지스트리에 액세스할 수 없으므로 스택 데몬 모니터링에 대한 기본 컨테이너 이미지를 덮어쓸 수 있습니다.



참고

구성을 변경하기 전에 초기 호스트의 부트스트랩 프로세스가 완료되었는지 확인합니다.

기본적으로 모니터링 스택 구성 요소는 기본 **Ceph** 이미지를 기반으로 배포됩니다. 스토리지 클러스터의 연결이 해제된 환경의 경우 사용 가능한 최신 모니터링 스택 구성 요소 이미지를 사용할 수 있습니다.

참고

사용자 지정 레지스트리를 사용하는 경우 **Ceph** 데몬을 추가하기 전에 새로 추가된 노드의 사용자 정의 레지스트리에 로그인해야 합니다.

구문

```
# ceph cephadm registry-login --registry-url CUSTOM_REGISTRY_NAME --
registry_username REGISTRY_USERNAME --registry_password
REGISTRY_PASSWORD
```

예제

```
# ceph cephadm registry-login --registry-url myregistry --registry_username
myregistryusername --registry_password myregistrypassword1
```

사전 요구 사항

- 실행 중인 **VM(가상 머신)** 또는 서버 중 하나 이상
- **Red Hat Enterprise Linux 8.4 EUS** 또는 **Red Hat Enterprise Linux 8.5**.
- 모든 노드에 대한 루트 수준 액세스.
- 암호 없는 **ssh**는 스토리지 클러스터의 모든 호스트에 설정됩니다.

절차

1.

ceph config 명령을 사용하여 사용자 정의 컨테이너 이미지를 설정합니다.

구문

```
ceph config set mgr mgr/cephadm/OPTION_NAME  
CUSTOM_REGISTRY_NAME/CONTAINER_NAME
```

OPTION_NAME에 대해 다음 옵션을 사용합니다.

```
container_image_prometheus  
container_image_grafana  
container_image_alertmanager  
container_image_node_exporter
```

예제

```
[root@host01 ~]# ceph config set mgr mgr/cephadm/container_image_prometheus  
myregistry/mycontainer  
[root@host01 ~]# ceph config set mgr mgr/cephadm/container_image_grafana  
myregistry/mycontainer  
[root@host01 ~]# ceph config set mgr mgr/cephadm/container_image_alertmanager  
myregistry/mycontainer  
[root@host01 ~]# ceph config set mgr mgr/cephadm/container_image_node_exporter  
myregistry/mycontainer
```

2.

node-exporter를 재배포합니다.

구문

```
ceph orch redeploy node-exporter
```



참고

배포되지 않는 서비스는 **ceph orch redeploy** 명령으로 재배포할 수 있습니다.



참고

사용자 지정 이미지를 설정하면 구성 이미지 이름 및 태그의 기본값이 재정의되지만 덮어쓰지 않습니다. 업데이트가 사용 가능할 때 기본값이 변경됩니다. 사용자 지정 이미지를 설정하면 자동 업데이트를 위해 사용자 지정 이미지를 설정한 구성 요소를 구성할 수 없습니다. 업데이트를 설치하려면 구성 이미지 이름과 태그를 수동으로 업데이트해야 합니다.

- 기본 구성을 사용하여 되돌리는 경우 사용자 정의 컨테이너 이미지를 재설정할 수 있습니다. **ceph config rm**을 사용하여 설정 옵션을 재설정합니다.

구문

```
ceph config rm mgr mgr/cephadm/OPTION_NAME
```

예제

```
ceph config rm mgr mgr/cephadm/container_image_prometheus
```

추가 리소스

- 연결이 해제된 설치를 수행하는 방법에 대한 자세한 내용은 연결이 [연결 해제된 설치 수행](#)을 참조하십시오.

3.13. SSH 키 배포

`cephadm-distribute-ssh-key.yml` 플레이북을 사용하여 키를 수동으로 생성하고 배포하는 대신 **SSH** 키를 배포할 수 있습니다. 플레이북은 인벤토리의 모든 호스트에 **SSH** 공개 키를 배포합니다.

Ansible 관리 노드에 **SSH** 키 쌍을 생성하고, 암호를 입력하라는 메시지 없이 **Ansible**이 노드에 액세스할 수 있도록 스토리지 클러스터의 각 노드에 공개 키를 배포할 수도 있습니다.

사전 요구 사항

- **Ansible**은 관리 노드에 설치됩니다.
- **Ansible** 관리 노드에 액세스
- 스토리지 클러스터의 모든 노드에 대한 **sudo** 액세스 권한이 있는 **Ansible** 사용자
- 부트스트랩이 완료되었습니다. *Red Hat Ceph Storage 설치 가이드*의 [새 스토리지 클러스터 부팅](#) 섹션을 참조하십시오.

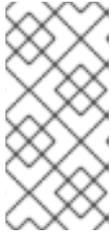
절차

1. **Ansible** 관리 노드에서 `/usr/share/cephadm-ansible` 디렉토리로 이동합니다.

예제

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

2. **Ansible** 관리 노드에서 **SSH** 키를 배포합니다. 선택적 `cephadm_pubkey_path` 매개변수는 **ansible** 컨트롤러 호스트에 있는 **SSH** 공개 키 파일의 전체 경로 이름입니다.



참고

`cephadm_pubkey_path` 를 지정하지 않으면 플레이북은 `cephadm get-pub-key` 명령에서 키를 가져옵니다. 즉, 최소 클러스터를 부트 스트랩했습니다.

구문

```
ansible-playbook -i INVENTORY_HOST_FILE cephadm-distribute-ssh-key.yml -e
cephadm_ssh_user=USER_NAME -e cephadm_pubkey_path= home/cephadm/ceph.key -e
admin_node=ADMIN_NODE_NAME_1
```

예제

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-distribute-ssh-
key.yml -e cephadm_ssh_user=ceph-admin -e
cephadm_pubkey_path=/home/cephadm/ceph.key -e admin_node=host01
```

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-distribute-ssh-
key.yml -e cephadm_ssh_user=ceph-admin -e admin_node=host01
```

3.14. CEPHADM 셸 시작

`cephadm shell` 명령은 모든 **Ceph** 패키지가 설치된 컨테이너에서 **bash** 셸을 시작합니다. 이를 통해 설치 및 부트 스트랩과 같은 "Day One" 클러스터 설정 작업을 수행하고 `ceph` 명령을 호출할 수 있습니다.

사전 요구 사항

- 설치 및 부트스트랩된 스토리지 클러스터입니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

절차

cephadm 셸을 시작하는 방법은 다음 두 가지가 있습니다.

- 시스템 프롬프트에서 **cephadm shell**을 입력합니다. 이 예제에서는 셸 내에서 **ceph -s** 명령을 호출합니다.

예제

```
[root@host01 ~]# cephadm shell
[ceph: root@host01 /]# ceph -s
```

- 시스템 프롬프트에서 **cephadm shell** 과 실행할 명령을 입력합니다.

예제

```
[root@host01 ~]# cephadm shell ceph -s

cluster:
  id: f64f341c-655d-11eb-8778-fa163e914bcc
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum host01,host02,host03 (age 94m)
  mgr: host01.lbnhug(active, since 59m), standbys: host02.rofgay, host03.ohipra
  mds: 1/1 daemons up, 1 standby
  osd: 18 osds: 18 up (since 10m), 18 in (since 10m)
  rgw: 4 daemons active (2 hosts, 1 zones)

data:
  volumes: 1/1 healthy
  pools: 8 pools, 225 pgs
  objects: 230 objects, 9.9 KiB
  usage: 271 MiB used, 269 GiB / 270 GiB avail
  pgs: 225 active+clean

io:
  client: 85 B/s rd, 0 op/s rd, 0 op/s wr
```



참고

노드에 `/etc/ceph/` 의 구성 및 키링 파일이 포함된 경우 컨테이너 환경에서는 해당 파일의 값을 `cephadm` 셸의 기본값으로 사용합니다. **MON** 노드에서 `cephadm` 셸을 실행하는 경우 `cephadm` 셸은 기본 구성을 사용하는 대신 **MON** 컨테이너의 기본 구성을 상속합니다.

3.15. 클러스터 설치 확인

클러스터 설치가 완료되면 **Red Hat Ceph Storage 5** 설치가 제대로 실행되고 있는지 확인할 수 있습니다.

스토리지 클러스터 설치를 `root` 사용자로 확인하는 방법은 다음 두 가지가 있습니다.

- `podman ps` 명령을 실행합니다.
- `cephadm shell ceph -s`를 실행합니다.

사전 요구 사항

- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

절차

- `podman ps` 명령을 실행합니다.

예제

```
[root@host01 ~]# podman ps
```



참고

Red Hat Ceph Storage 5에서 **systemd** 장치의 형식이 변경되었습니다. **NAMES** 열에서 단위 파일에 이제 **FSID**가 포함됩니다.

- **cephadm shell ceph -s** 명령을 실행합니다.

예제

```
[root@host01 ~]# cephadm shell ceph -s
cluster:
  id: f64f341c-655d-11eb-8778-fa163e914bcc
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum host01,host02,host03 (age 94m)
  mgr: host01.lbnhug(active, since 59m), standbys: host02.rofgay, host03.ohipra
  mds: 1/1 daemons up, 1 standby
  osd: 18 osds: 18 up (since 10m), 18 in (since 10m)
  rgw: 4 daemons active (2 hosts, 1 zones)

data:
  volumes: 1/1 healthy
  pools: 8 pools, 225 pgs
  objects: 230 objects, 9.9 KiB
  usage: 271 MiB used, 269 GiB / 270 GiB avail
  pgs: 225 active+clean

io:
  client: 85 B/s rd, 0 op/s rd, 0 op/s wr
```



참고

스토리지 클러스터의 상태는 호스트 및 데몬이 추가되지 않으므로 **HEALTH_WARN** 상태입니다.

3.16. 호스트 추가

Red Hat Ceph Storage 설치를 부트 스트랩하면 동일한 컨테이너 내에서 하나의 **Monitor** 데몬과 하나의 **Manager** 데몬으로 구성된 작동 중인 스토리지 클러스터가 생성됩니다. 스토리지 관리자는 스토리지 클러스터에 호스트를 추가하고 구성할 수 있습니다.

참고

- **Red Hat Enterprise Linux 8**의 경우 **preflight** 플레이북을 실행하면 **Ansible** 인벤토리 파일에 나열된 모든 호스트에 **podman, lvm2, chrony, cephadm**가 설치됩니다.
- **Red Hat Enterprise Linux 9**의 경우 **podman, lvm2, chrony, cephadm**을 수동으로 설치하고 **preflight** 플레이북이 지원되지 않으므로 **ansible** 플레이북을 실행하기 위한 단계를 건너뛰어야 합니다.
- 사용자 지정 레지스트리를 사용하는 경우 **Ceph** 데몬을 추가하기 전에 새로 추가된 노드의 사용자 정의 레지스트리에 로그인해야 합니다.

.Syntax

```
[source,subs="verbatim,quotes"]
```

```
----
```

```
# ceph cephadm registry-login --registry-url
_CUSTOM_REGISTRY_NAME_ --registry_username
_REGISTRY_USERNAME_ --registry_password
_REGISTRY_PASSWORD_
----
```

.Example

```
----
```

```
# ceph cephadm registry-login --registry-url myregistry --
registry_username myregistryusername --registry_password
myregistypassword1
----
```

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 스토리지 클러스터의 모든 노드에 대한 **sudo** 액세스 권한이 있는 루트 수준 또는 사용자입니다.

- 노드를 **CDN**에 등록하고 서브스크립션을 연결합니다.
- **sudo** 및 암호 없이 **ssh** 를 사용하는 **Ansible** 사용자는 스토리지 클러스터의 모든 노드에 액세스할 수 있습니다.

절차

+



참고

다음 절차에서는 표시된 대로 **root** 또는 사용자가 부트스트랩되는 사용자 이름을 사용합니다.

1.

admin 인증 키가 포함된 노드에서 스토리지 클러스터의 공개 **SSH** 키를 새 호스트의 **root** 사용자의 **authorized_keys** 파일에 설치합니다.

구문

```
ssh-copy-id -f -i /etc/ceph/ceph.pub user@NEWHOST
```

예제

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

2.

Ansible 관리 노드의 `/usr/share/cephadm-ansible` 디렉터리로 이동합니다.

예제

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

3.

Ansible 관리 노드에서 새 호스트를 **Ansible** 인벤토리 파일에 추가합니다. 파일의 기본 위치는 `/usr/share/cephadm-ansible/hosts` 입니다. 다음 예제는 일반적인 인벤토리 파일의 구조를 보여줍니다.

예제

```
[ceph-admin@admin ~]$ cat hosts
```

```
host02
host03
host04
```

```
[admin]
host01
```



참고

이전에 새 호스트를 **Ansible** 인벤토리 파일에 추가하고 호스트에서 **preflight** 플레이북을 실행한 경우 4 단계로 건너뛴니다.

4.

Red Hat Enterprise Linux 8의 `--limit` 옵션을 사용하여 사전 플레이북을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars "ceph_origin=rhcs"
--limit NEWHOST
```

예제

```
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --
extra-vars "ceph_origin=rhcs" --limit host02
```

preflight Playbook은 새 호스트에 **podman**, **lvm2**, **chrony**, **cephadm** 을 설치합니다. 설치가 완료되면 **cephadm**은 **/usr/sbin/** 디렉터리에 있습니다.

- **Red Hat Enterprise Linux 9**의 경우 **podman**, **lvm2**, **chrony**, **cephadm** 을 수동으로 설치합니다.

```
[root@host01 ~]# dnf install podman lvm2 chrony cephadm
```

5.

부트스트랩 노드에서 **cephadm orchestrator**를 사용하여 스토리지 클러스터에 새 호스트를 추가합니다.

구문

```
ceph orch host add NEWHOST
```

예제

```
[ceph: root@host01 /]# ceph orch host add host02
Added host 'host02' with addr '10.10.128.69'
[ceph: root@host01 /]# ceph orch host add host03
Added host 'host03' with addr '10.10.128.70'
```

6.

선택 사항: **preflight** 플레이북을 실행하기 전과 후에 **IP** 주소로 노드를 추가할 수도 있습니다. 스토리지 클러스터 환경에 **DNS**가 구성되지 않은 경우 호스트 이름과 함께 **IP** 주소로 호스트를 추가할 수 있습니다.

구문

```
ceph orch host add HOSTNAME IP_ADDRESS
```

예제

```
[ceph: root@host01 /]# ceph orch host add host02 10.10.128.69
Added host 'host02' with addr '10.10.128.69'
```

검증

•

스토리지 클러스터의 상태를 보고 새 호스트가 추가되었는지 확인합니다. **ceph orch host ls** 명령의 출력에서 호스트 **STATUS**가 비어 있습니다.

예제

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



추가 리소스

- [Red Hat Ceph Storage 설치 가이드의 CDN에 Red Hat Ceph Storage 노드 등록 및 서비스 크립션 연결](#) 섹션을 참조하십시오.
- [Red Hat Ceph Storage 설치 가이드의 sudo 액세스 권한으로 Ansible 사용자 생성](#) 섹션을 참조하십시오.

3.16.1. addr 옵션을 사용하여 호스트 확인

addr 옵션은 호스트에 연결하는 추가 방법을 제공합니다. 호스트의 **IP** 주소를 **addr** 옵션에 추가합니다. **ssh** 가 호스트 이름으로 연결할 수 없는 경우 **addr** 에 저장된 값을 사용하여 **IP** 주소로 호스트에 연결합니다.

사전 요구 사항

- 설치 및 부트스트랩된 스토리지 클러스터입니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

절차

cephadm 셸 내부에서 이 절차를 실행합니다.

1. **IP** 주소를 추가합니다.

구문

```
ceph orch host add HOSTNAME IP_ADDR
```

예제

```
[ceph: root@host01 /]# ceph orch host add host01 10.10.128.68
```

참고

호스트 이름으로 호스트를 추가하는 경우 해당 호스트가 IPv4 주소 대신 IPv6 주소로 추가되는 경우 `ceph orch host`를 사용하여 해당 호스트의 IP 주소를 지정합니다.

```
ceph orch host set-addr HOSTNAME IP_ADDR
```

추가한 호스트의 IPv6 형식에서 IPv4 형식으로 IP 주소를 변환하려면 다음 명령을 사용합니다.

```
ceph orch host set-addr HOSTNAME IPV4_ADDRESS
```

3.16.2. 여러 호스트 추가

YAML 파일을 사용하여 스토리지 클러스터에 동시에 여러 호스트를 추가합니다.

참고

호스트 컨테이너 내에 `hosts.yaml` 파일을 생성하거나 로컬 호스트에 파일을 생성한 다음 `cephadm` 셸을 사용하여 컨테이너 내에서 파일을 마운트합니다. `cephadm` 셸은 마운트된 파일을 `/mnt`에 자동으로 배치합니다. 로컬 호스트에서 직접 파일을 생성한 다음 `hosts.yaml` 파일을 마운트하지 않고 적용하면 `File does not exist` 오류가 표시될 수 있습니다.

사전 요구 사항

- 설치 및 부트스트랩된 스토리지 클러스터입니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

절차

1. 공개 **ssh** 키를 추가하려는 각 호스트에 복사합니다.
2. 텍스트 편집기를 사용하여 **hosts.yaml** 파일을 생성합니다.
3. 다음 예와 같이 **hosts.yaml** 파일에 호스트 설명을 추가합니다. 각 호스트에 배포하려는 데몬의 배치를 식별하는 레이블을 포함합니다. 각 호스트 설명을 세 개의 대시(---)로 분리합니다.

예제

```
service_type: host
addr:
hostname: host02
labels:
- mon
- osd
- mgr
---
service_type: host
addr:
hostname: host03
labels:
- mon
- osd
- mgr
---
service_type: host
addr:
hostname: host04
labels:
- mon
- osd
```

4. 호스트 컨테이너 내에 **hosts.yaml** 파일을 생성한 경우 **ceph orch apply** 명령을 호출합니다.

예제

```
[root@host01 ~]# ceph orch apply -i hosts.yaml
Added host 'host02' with addr '10.10.128.69'
Added host 'host03' with addr '10.10.128.70'
Added host 'host04' with addr '10.10.128.71'
```

5. 로컬 호스트에서 직접 **hosts.yaml** 파일을 생성한 경우 **cephadm 셸**을 사용하여 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount hosts.yaml -- ceph orch apply -i /mnt/hosts.yaml
```

6. 호스트 및 해당 라벨 목록을 표시합니다.

예제

```
[root@host01 ~]# ceph orch host ls
HOST   ADDR   LABELS   STATUS
host02 host02  mon osd mgr
host03 host03  mon osd mgr
host04 host04  mon osd
```



참고

호스트가 온라인에 있고 정상적으로 작동하는 경우 해당 상태는 비어 있습니다. 오프라인 호스트는 **OFFLINE** 상태를 표시하며 유지 관리 모드의 호스트는 **MAINTENANCE** 상태를 표시합니다.

3.16.3. 연결 해제된 배포에서 호스트 추가

프라이빗 IP를 통해 프라이빗 네트워크에서 스토리지 클러스터를 실행 중이고 호스트 도메인 이름 서버(DNS)에 연결할 수 없는 경우 스토리지 클러스터에 추가할 각 호스트의 호스트 이름과 IP 주소를 모두 포함해야 합니다.

사전 요구 사항

- 실행 중인 스토리지 클러스터입니다.
- 스토리지 클러스터의 모든 호스트에 대한 루트 수준 액세스.

절차

1. **cephadm** 셸을 호출합니다.

구문

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

2. 호스트를 추가합니다.

구문

```
ceph orch host add HOST_NAME HOST_ADDRESS
```

예제

```
[ceph: root@host01 /]# ceph orch host add host03 10.10.128.70
```

3.16.4. 호스트 제거

Ceph Orchestrators로 **Ceph** 클러스터의 호스트를 제거할 수 있습니다. 모든 데몬은 작업이 완료될 때까지 데몬 또는 클러스터를 배포할 수 없도록 **_no_schedule** 레이블을 추가하는 **drain** 옵션으로 제거됩니다.



중요

부트스트랩 호스트를 제거하는 경우 호스트를 제거하기 전에 관리자 인증 키와 구성 파일을 스토리지 클러스터의 다른 호스트에 복사해야 합니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 모든 노드에 대한 루트 수준 액세스.
- 호스트는 스토리지 클러스터에 추가됩니다.
- 모든 서비스가 배포됩니다.
- **Cephadm**은 서비스를 제거해야 하는 노드에 배포됩니다.

절차

1. **Cephadm 셸에 로그인합니다.**

예제

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

2. **호스트 세부 정보를 가져옵니다.**

예제

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

3. **호스트의 모든 데몬을 트레이닝합니다.**

구문

```
ceph orch host drain HOSTNAME
```

예제

```
[ceph: root@host01 /]# ceph orch host drain host02
```

`_no_schedule` 레이블은 배포를 차단하는 호스트에 자동으로 적용됩니다.

4. **OSD 제거 상태를 확인합니다.**

예제

```
[ceph: root@host01 /]# ceph orch osd rm status
```

OSD에 배치 그룹(**PG**)이 없는 경우 스토리지 클러스터에서 **OSD**가 해제되어 제거됩니다.

5. 스토리지 클러스터에서 모든 데몬이 제거되었는지 확인합니다.

구문

```
ceph orch ps HOSTNAME
```

예제

```
[ceph: root@host01 /]# ceph orch ps host02
```

o.

호스트를 제거합니다.

구문

```
ceph orch host rm HOSTNAME
```

예제

```
[ceph: root@host01 /]# ceph orch host rm host02
```

추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#)의 [Ceph Orchestrator](#) 섹션을 사용하여 호스트 추가를 참조하십시오.
- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#)의 [Ceph Orchestrator](#) 섹션을 사용하여 호스트 목록을 참조하십시오.

3.17. 호스트 레이블 지정

Ceph 오케스트레이터는 호스트에 레이블 할당을 지원합니다. 레이블은 자유형이며 특정 의미가 없습니다. 즉, `mon,monitor,mycluster_monitor` 또는 다른 텍스트 문자열을 사용할 수 있습니다. 각 호스트에는 여러 개의 레이블이 있을 수 있습니다.

예를 들어, Ceph Monitor 데몬을 배포하려는 모든 호스트에 `mon` 레이블을 적용하고, Ceph Manager 데몬을 배포하려는 모든 호스트에 `mgr`, Ceph Object Gateway 데몬의 경우 `rgw` 를 적용합니다.

스토리지 클러스터의 모든 호스트에 레이블을 지정하면 각 호스트에서 실행되는 데몬을 빠르게 식별할

수 있으므로 시스템 관리 작업을 단순화하는 데 도움이 됩니다. 또한 **Ceph** 오케스트레이터 또는 **YAML** 파일을 사용하여 특정 호스트 레이블이 있는 호스트에서 데몬을 배포하거나 제거할 수 있습니다.

사전 요구 사항

- 설치 및 부트스트랩된 스토리지 클러스터입니다.

3.17.1. 호스트에 레이블 추가

Ceph 오케스트레이터를 사용하여 호스트에 레이블을 추가할 수 있습니다. 각 호스트에는 여러 개의 레이블이 있을 수 있습니다. 레이블을 사용하여 데몬 배치를 지정할 수 있습니다.

사전 요구 사항

- 설치 및 부트스트랩된 스토리지 클러스터입니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.
- 호스트는 스토리지 클러스터에 추가됩니다.

절차

1. **cephadm** 셸을 시작합니다.

```
[root@host01 ~]# cephadm shell
[ceph: root@host01 /]#
```

2. 호스트에 레이블을 추가합니다.

구문

```
ceph orch host label add HOSTNAME LABEL
```

예제

```
[ceph: root@host01 /]# ceph orch host label add host02 mon
```

검증

- 호스트를 나열합니다.

예제

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

3.17.2. 호스트에서 레이블 제거

Ceph 오케스트레이터를 사용하여 호스트에서 레이블을 제거할 수 있습니다.

사전 요구 사항

- 설치 및 부트스트랩된 스토리지 클러스터입니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

절차

1. **cephadm** 셸을 시작합니다.

```
[root@host01 ~]# cephadm shell
[ceph: root@host01 /]#
```

2.

ceph 오케스트레이터를 사용하여 호스트에서 레이블을 제거합니다.

구문

```
ceph orch host label rm HOSTNAME LABEL
```

예제

```
[ceph: root@host01 /]# ceph orch host label rm host02 mon
```

검증

- 호스트를 나열합니다.

예제

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

3.17.3. 호스트 레이블을 사용하여 특정 호스트에 데몬 배포

호스트 레이블을 사용하여 특정 호스트에 데몬을 배포할 수 있습니다. 호스트 레이블을 사용하여 특정 호스트에 데몬을 배포하는 방법은 다음 두 가지가 있습니다.

- 명령줄에서 **--placement** 옵션을 사용합니다.
- **YAML** 파일을 사용합니다.

사전 요구 사항

- 설치 및 부트스트랩된 스토리지 클러스터입니다.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

절차

1. **Cephadm 셸**에 로그인합니다.

예제

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

2. 현재 호스트 및 레이블을 나열합니다.

예제

```
[ceph: root@host01 /]# ceph orch host ls
HOST   ADDR   LABELS      STATUS
host01      _admin,mon,osd,mgr
host02      mon,osd,mgr,mylabel
```

- 방법 1: 명령줄에서 데몬을 배포하려면 **--placement** 옵션을 사용합니다.

구문

```
ceph orch apply DAEMON --placement="label:LABEL"
```

예제

```
[ceph: root@host01 /]# ceph orch apply prometheus --placement="label:mylabel"
```

- 메서드 2 **YAML** 파일의 특정 호스트 레이블에 데몬을 할당하려면 **YAML** 파일에 서비스 유형 및 라벨을 지정합니다.

- a. **placement.yml** 파일을 생성합니다.

예제

```
[ceph: root@host01 /]# vi placement.yml
```

- b. **placement.yml** 파일에 서비스 유형 및 라벨을 지정합니다.

예제

```
service_type: prometheus
placement:
  label: "mylabel"
```

C.

데몬 배치 파일을 적용합니다.

구문

```
ceph orch apply -i FILENAME
```

예제

```
[ceph: root@host01 /]# ceph orch apply -i placement.yml
Scheduled prometheus update...
```

검증

- 데몬 상태를 나열합니다.

구문

```
ceph orch ps --daemon_type=DAEMON_NAME
```

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=prometheus
NAME          HOST PORTS STATUS   REFRESHED AGE MEM USE MEM LIM
VERSION IMAGE ID CONTAINER ID
prometheus.host02 host02 *:9095 running (2h) 8m ago 2h 85.3M - 2.22.2
ac25aac5d567 ad8c7593d7c0
```

3.18. 모니터 서비스 추가

일반적인 **Red Hat Ceph Storage** 스토리지 클러스터에는 다른 호스트에 배포된 모니터 데몬이 3개 또는 5개 이상 있습니다. 스토리지 클러스터에 5개 이상의 호스트가 있는 경우 5개의 모니터 노드를 배포하는 것이 좋습니다.



참고

방화벽의 경우 자세한 내용은 [Red Hat Ceph Storage 구성 가이드의 Ceph Monitor 노드 방화벽 설정](#)을 참조하십시오.



참고

부트스트랩 노드는 스토리지 클러스터의 초기 모니터입니다. 배포하려는 호스트 목록에 부트스트랩 노드를 포함해야 합니다.



참고

둘 이상의 특정 호스트에 **Monitor** 서비스를 적용하려면 동일한 `ceph orch apply` 명령 내의 모든 호스트 이름을 지정해야 합니다. `ceph orch apply mon --placement host1`을 지정한 다음 `ceph orch apply mon --placement host2`를 지정하면 두 번째 명령은 `host1`에서 **Monitor** 서비스를 제거하고 **Monitor** 서비스를 `host2`에 적용합니다.

모니터 노드 또는 전체 클러스터가 단일 서브넷에 있는 경우 클러스터에 새 호스트를 추가할 때 최대 5개의 모니터 데몬을 자동으로 추가합니다. `cephadm`은 새 호스트에 **Monitor** 데몬을 자동으로 구성합니다. 새 호스트는 스토리지 클러스터의 첫 번째(**bootstrap**) 호스트와 동일한 서브넷에 있습니다. `cephadm`은 스토리지 클러스터 크기의 변화에 따라 모니터를 배포하고 확장할 수 있습니다.

사전 요구 사항

- 스토리지 클러스터의 모든 호스트에 대한 루트 수준 액세스.
- 실행 중인 스토리지 클러스터입니다.

절차

1. 스토리지 클러스터의 임의의 호스트 5개에 5개의 모니터 데몬을 적용합니다.

```
ceph orch apply mon 5
```

2. 자동 모니터 배포를 비활성화합니다.

```
ceph orch apply mon --unmanaged
```

추가 리소스

- 스토리지 클러스터에 **Ceph** 모니터를 배포하는 방법에 대한 자세한 내용은 [Red Hat Ceph Storage 운영 가이드의 명령줄 인터페이스를 사용하여 Ceph Monitor 데몬 배포](#) 섹션을 참조하십시오.

3.19. 관리 노드 설정

관리 노드를 사용하여 스토리지 클러스터를 관리합니다.

관리 노드에는 클러스터 구성 파일과 관리자 인증 키가 모두 포함되어 있습니다. 이러한 두 파일은 모두 `/etc/ceph` 디렉터리에 저장되며 스토리지 클러스터 이름을 접두사로 사용합니다.

예를 들어 기본 `ceph` 클러스터 이름은 `ceph` 입니다. 기본 이름을 사용하는 클러스터에서 관리자 인증 키의 이름은 `/etc/ceph/ceph.client.admin.keyring` 입니다. 해당 클러스터 구성 파일의 이름은 `/etc/ceph/ceph.conf` 입니다.

스토리지 클러스터에서 추가 호스트를 `admin` 노드로 설정하려면 관리자 노드로 지정할 호스트에 `_admin` 레이블을 적용합니다.



참고

기본적으로 `_admin` 레이블을 노드에 적용한 후 `cephadm` 은 `ceph.conf` 및 `client.admin` 인증 키 파일을 해당 노드에 복사합니다. `cephadm bootstrap` 명령을 사용하여 `--skip-admin-label` 옵션을 지정하지 않으면 `_admin` 레이블이 부트스트랩 노드에 자동으로 적용됩니다.

사전 요구 사항

- `cephadm` 이 설치된 실행 중인 스토리지 클러스터.
- 스토리지 클러스터에는 모니터 및 관리자 노드가 실행 중입니다.
- 클러스터의 모든 노드에 대한 루트 수준 액세스.

절차

1. `ceph orch host ls` 를 사용하여 스토리지 클러스터의 호스트를 확인합니다.

예제

```
[root@host01 ~]# ceph orch host ls
HOST ADDR LABELS STATUS
host01    mon,mgr,_admin
host02    mon
host03    mon,mgr
host04
host05
host06
```

2. `_admin` 레이블을 사용하여 스토리지 클러스터에서 관리 호스트를 지정합니다. 최상의 결과를 얻으려면 이 호스트에 **Monitor** 및 **Manager** 데몬이 모두 실행되고 있어야 합니다.

구문

```
ceph orch host label add HOSTNAME_admin
```

예제

```
[root@host01 ~]# ceph orch host label add host03 _admin
```

3.

관리자 호스트에 **_admin** 레이블이 있는지 확인합니다.

예제

```
[root@host01 ~]# ceph orch host ls
HOST ADDR LABELS STATUS
host01    mon,mgr,_admin
host02    mon
host03    mon,mgr,_admin
host04
host05
host06
```

4.

관리 노드에 로그인하여 스토리지 클러스터를 관리합니다.

3.19.1. 호스트 레이블을 사용하여 Ceph 모니터 노드 배포

일반적인 Red Hat Ceph Storage 스토리지 클러스터에는 서로 다른 호스트에 배포된 3개 또는 5개의 Ceph Monitor 데몬이 있습니다. 스토리지 클러스터에 5개 이상의 호스트가 있는 경우 5개의 Ceph

Monitor 노드를 배포하는 것이 좋습니다.

Ceph Monitor 노드 또는 전체 클러스터가 단일 서브넷에 있는 경우 **cephadm** 은 클러스터에 새 노드를 추가할 때 최대 5개의 **Ceph Monitor** 데몬을 자동으로 추가합니다. **cephadm** 은 새 노드에 **Ceph Monitor** 데몬을 자동으로 구성합니다. 새 노드는 스토리지 클러스터의 첫 번째(**bootstrap**) 노드와 동일한 서브넷에 있습니다. **cephadm** 은 모니터를 배포 및 스케일링하여 스토리지 클러스터 크기에 대응합니다.



참고

호스트 레이블을 사용하여 **Ceph Monitor** 노드가 포함된 호스트를 식별합니다.

사전 요구 사항

- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.
- 실행 중인 스토리지 클러스터입니다.

절차

1. 호스트에 **mon** 레이블을 할당합니다.

구문

```
ceph orch host label add HOSTNAME mon
```

예제

```
[ceph: root@host01 /]# ceph orch host label add host02 mon
[ceph: root@host01 /]# ceph orch host label add host03 mon
```

2.

현재 호스트 및 라벨을 표시합니다.

구문

```
ceph orch host ls
```

예제

```
[ceph: root@host01 /]# ceph orch host ls
HOST ADDR LABELS STATUS
host01    mon,mgr,_admin
host02    mon
host03    mon
host04
host05
host06
```

-

host 레이블을 기반으로 **Ceph Monitor** 데몬을 배포합니다.

구문

```
ceph orch apply mon label:mon
```

- 특정 호스트 세트에 **Ceph Monitor** 데몬을 배포합니다.

구문

```
ceph orch apply mon HOSTNAME1,HOSTNAME2,HOSTNAME3
```

예제

```
[ceph: root@host01 /]# ceph orch apply mon host01,host02,host03
```



참고

배포하려는 호스트 목록에 부트스트랩 노드를 포함해야 합니다.

3.19.2. IP 주소 또는 네트워크 이름으로 Ceph Monitor 노드 추가

일반적인 **Red Hat Ceph Storage** 스토리지 클러스터에는 다른 호스트에 배포된 모니터 데몬이 3개 또는 5개 이상 있습니다. 스토리지 클러스터에 5개 이상의 호스트가 있는 경우 5개의 모니터 노드를 배포하는 것이 좋습니다.

모니터 노드 또는 전체 클러스터가 단일 서브넷에 있는 경우 클러스터에 새 노드를 추가할 때 **cephadm** 에서 최대 5개의 모니터 데몬을 자동으로 추가합니다. 새 노드에서 모니터 데몬을 구성할 필요가 없습니다. 새 노드는 스토리지 클러스터의 첫 번째 노드와 동일한 서브넷에 있습니다. 스토리지 클러스터의 첫 번째 노드는 부트 스트랩 노드입니다. **cephadm** 은 모니터도 배포 및 스케일링하여 스토리지 클러스터의 크기에 해당할 수 있습니다.

사전 요구 사항

- 스토리지 클러스터의 모든 노드에 대한 루트 수준 액세스.

- 실행 중인 스토리지 클러스터입니다.

절차

1. 각각의 추가 **Ceph** 모니터 노드를 배포하려면 다음을 수행합니다.

구문

```
ceph orch apply mon NODE:IP_ADDRESS_OR_NETWORK_NAME
[NODE:IP_ADDRESS_OR_NETWORK_NAME...]
```

예제

```
[ceph: root@host01 /]# ceph orch apply mon host02:10.10.128.69 host03:mynetwork
```

3.19.3. 호스트에서 admin 레이블 제거

Ceph 오케스트레이터를 사용하여 호스트에서 **admin** 레이블을 제거할 수 있습니다.

사전 요구 사항

- **cephadm** 이 설치되어 부트스트랩된 실행 중인 스토리지 클러스터
- 스토리지 클러스터에는 모니터 및 관리자 노드가 실행 중입니다.
- 클러스터의 모든 노드에 대한 루트 수준 액세스.

절차

1. **ceph orch** 호스트 **ls** 를 사용하여 호스트를 확인하고 스토리지 클러스터에서 관리 호스트를 확인합니다.

예제

```
[root@host01 ~]# ceph orch host ls
HOST ADDR LABELS STATUS
host01    mon,mgr,_admin
host02    mon
host03    mon,mgr,_admin
host04
host05
host06
```

2. **Cephadm 셸**에 로그인합니다.

예제

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

3. **ceph** 오케스트레이터를 사용하여 호스트에서 **admin** 레이블을 제거합니다.

구문

```
ceph orch host label rm HOSTNAME LABEL
```

예제

```
[ceph: root@host01 /]# ceph orch host label rm host03 _admin
```

4.

관리자 호스트에 `_admin` 레이블이 있는지 확인합니다.

예제

```
[root@host01 ~]# ceph orch host ls
HOST ADDR LABELS STATUS
host01      mon,mgr,_admin
host02      mon
host03      mon,mgr
host04
host05
host06
```

중요

노드에서 `admin` 레이블을 제거한 후 해당 노드에서 `ceph.conf` 및 `client.admin` 키링 파일을 삭제해야 합니다. 또한 노드는 `[admin] ansible` 인벤토리 파일에서 제거해야 합니다.

3.20. 관리자 서비스 추가

`cephadm` 은 부트 스트랩 프로세스 중에 부트 스트랩 노드에 **Manager** 데몬을 자동으로 설치합니다. **Ceph** 오케스트레이터를 사용하여 추가 **Manager** 데몬을 배포합니다.

Ceph 오케스트레이터는 기본적으로 두 개의 **Manager** 데몬을 배포합니다. 다른 수의 **Manager** 데몬을 배포하려면 다른 수를 지정합니다. **Manager** 데몬을 배포해야 하는 호스트를 지정하지 않으면 **Ceph** 오케스트레이터가 호스트를 임의로 선택하고 **Manager** 데몬을 배포합니다.



참고

Manager 데몬을 둘 이상의 특정 호스트에 적용하려면 동일한 `ceph orch apply` 명령 내의 모든 호스트 이름을 지정해야 합니다. `ceph orch apply mgr --placement host1` 을 지정하고 `ceph orch apply mgr --placement host2` 를 지정하는 경우 두 번째 명령은 `host1`의 **Manager** 데몬을 제거하고 **Manager** 데몬을 `host2`에 적용합니다.

Red Hat은 `--placement` 옵션을 사용하여 특정 호스트에 배포하는 것이 좋습니다.

사전 요구 사항

- 실행 중인 스토리지 클러스터입니다.

절차

- 특정 수의 **Manager** 데몬을 임의로 선택한 호스트에 적용하려는 경우 다음을 수행합니다.

구문

```
ceph orch apply mgr NUMBER_OF_DAEMONS
```

예제

```
[ceph: root@host01 ~]# ceph orch apply mgr 3
```

- 스토리지 클러스터의 특정 호스트에 **Manager** 데몬을 추가하려면 다음을 수행합니다.

구문

```
ceph orch apply mgr --placement "HOSTNAME1 HOSTNAME2 HOSTNAME3"
```

예제

```
[ceph: root@host01 /]# ceph orch apply mgr --placement "host02 host03 host04"
```

3.21. OSD 추가

Cephadm은 사용할 수 없는 장치에 **OSD**를 프로비저닝하지 않습니다. 다음 조건을 모두 충족하는 경우 스토리지 장치는 사용 가능한 것으로 간주됩니다.

- 장치에 파티션이 없어야 합니다.
- 장치를 마운트해서는 안 됩니다.
- 장치에 파일 시스템이 포함되어서는 안 됩니다.
- 장치에 **Ceph BlueStore OSD**가 포함되어 있지 않아야 합니다.
- 장치가 **5GB**보다 커야 합니다.

**중요**

기본적으로 `osd_memory_target_autotune` 매개변수는 **Red Hat Ceph Storage 5.1** 에서 `true` 로 설정됩니다. OSD 메모리 튜닝에 대한 자세한 내용은 *Red Hat Ceph Storage Operations Guide* 의 **OSD 메모리 자동 튜닝** 섹션을 참조하십시오.

**중요**

Red Hat Ceph Storage 5.0에서 알려진 버그로 인해 사전 생성된 LVM 디스크는 **OSD, DB 또는 WAL** 배포에서는 지원되지 않습니다.

Red Hat Ceph Storage 5.1부터 **DB 및 WAL** 장치를 포함한 **OSD** 배포에 미리 생성된 LVM 디스크가 지원됩니다.

사전 요구 사항

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

절차

1. **OSD**를 배포하는 데 사용 가능한 장치를 나열합니다.

구문

```
ceph orch device ls [--hostname=HOSTNAME1 HOSTNAME2] [--wide] [--refresh]
```

예제

```
[ceph: root@host01 /]# ceph orch device ls --wide --refresh
```

2.

특정 호스트 또는 사용 가능한 모든 장치에 **OSD**를 배포할 수 있습니다.

•

특정 호스트의 특정 장치에서 **OSD**를 생성하려면 다음을 수행합니다.

구문

```
ceph orch daemon add osd HOSTNAME:DEVICE_PATH
```

예제

```
[ceph: root@host01 /]# ceph orch daemon add osd host02:/dev/sdb
```

•

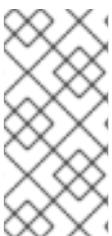
사용 가능한 및 사용되지 않는 장치에 **OSD**를 배포하려면 **--all-available-devices** 옵션을 사용합니다.

예제

```
[ceph: root@host01 /]# ceph orch apply osd --all-available-devices
```

참고

이 명령은 공동 배치된 **WAL** 및 **DB** 장치를 생성합니다. 배치되지 않은 장치를 생성하려면 이 명령을 사용하지 마십시오.



추가 리소스

- OSD의 드라이브 사양에 대한 자세한 내용은 *Red Hat Ceph Storage Operations Guide*의 [OSD 배포를 위한 고급 서비스 사양 및 필터](#) 섹션을 참조하십시오.
- 장치의 데이터를 지우는 **zapping** 장치에 대한 자세한 내용은 *Red Hat Ceph Storage Operations Guide*의 [Ceph OSD 배포용 장치](#) 섹션을 참조하십시오.

3.22. 클라이언트 노드 배포

스토리지 관리자는 `cephadm-preflight.yml` 및 `cephadm-clients.yml` 플레이북을 실행하여 클라이언트 노드를 배포할 수 있습니다. `cephadm-preflight.yml` 플레이북은 Ceph 리포지토리를 구성하고 부트 스트랩을 위해 스토리지 클러스터를 준비합니다. `podman`, `lvm2`, `chrony`, `cephadm` 과 같은 몇 가지 사전 요구 사항도 설치합니다.



중요

`cephadm-preflight` Playbook은 지원되지 않으므로 Red Hat Enterprise Linux 9에 대해 이 단계를 건너뛵니다.

`cephadm-clients.yml` Playbook은 구성 및 인증 파일 배포를 Ceph 클라이언트 그룹에 처리합니다.



참고

Ceph 클러스터를 업그레이드한 후 `cephadm-ansible` 플레이북을 사용하지 않는 경우 클라이언트 노드에서 `ceph-common` 패키지 및 클라이언트 라이브러리를 업그레이드해야 합니다. 자세한 내용은 *Red Hat Ceph Storage 업그레이드 가이드*의 [Red Hat Ceph Storage 클러스터 업그레이드](#) 섹션을 참조하십시오.

사전 요구 사항

- Ansible 관리 노드에 대한 루트 수준 액세스.
- 스토리지 클러스터의 모든 노드에 대한 `sudo` 및 암호 없는 SSH 액세스 권한이 있는 Ansible 사용자.

- **cephadm-ansible** 패키지 설치.
- **[admin]** 그룹은 관리 인증 키가 **/etc/ceph/ceph.client.admin.keyring** 에 있는 노드를 사용하여 인벤토리 파일에 정의됩니다.

절차

1. **Ansible** 사용자로 **Ansible** 관리 노드의 **/usr/share/cephadm-ansible** 디렉터리로 이동합니다.

예제

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

2. **hosts** 인벤토리 파일을 열고 편집하고 **[clients]** 그룹 및 클라이언트를 인벤토리에 추가합니다.

예제

```
host02  
host03  
host04  
  
[clients]  
client01  
client02  
client03  
  
[admin]  
host01
```

3.

cephadm-preflight.yml 플레이북을 실행하여 클라이언트에 사전 요구 사항을 설치합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --limit
CLIENT_GROUP_NAME|CLIENT_NODE_NAME
```

예제

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

4.

cephadm-clients.yml 플레이북을 실행하여 인증 키 및 **Ceph** 구성 파일을 클라이언트 세트에 배포합니다.

a.

사용자 지정 대상 인증 키 이름을 사용하여 인증 키를 복사하려면 다음을 수행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-clients.yml --extra-vars
'{"fsid":"FSID","keyring":"KEYRING_PATH","client_group":"CLIENT_GROUP_NAME","con
fig":"CEPH_CONFIGURATION_PATH","keyring_dest":"KEYRING_DESTINATION_PATH
"}'
```

•

VDD K_FILE 을 Ansible 인벤토리 파일 이름으로 바꿉니다.

- **FSID** 를 클러스터의 **FSID**로 바꿉니다.
- **KEYRING_PATH** 를 클라이언트에 복사하려는 관리 호스트의 인증 키의 전체 경로 이름으로 바꿉니다.
- 선택 사항: **CLIENT_GROUP_NAME** 을 클라이언트가 설정할 **Ansible** 그룹 이름으로 교체합니다.
- 선택 사항: **CEPH_CONFIGURATION_PATH** 를 관리 노드의 **Ceph** 구성 파일의 전체 경로로 바꿉니다.
- 선택 사항: **KEYRING_DESTINATION_PATH** 를 인증 키가 복사될 대상의 전체 경로 이름으로 바꿉니다.



참고

플레이북을 실행할 때 **conf** 옵션을 사용하여 구성 파일을 지정하지 않으면 플레이북에서 최소 구성 파일을 생성하고 배포합니다. 기본적으로 생성된 파일은 **/etc/ceph/ceph.conf** 에 있습니다.

예제

```
[ceph-admin@host01 cephadm-ansible]$ ansible-playbook -i hosts cephadm-clients.yml --extra-vars '{"fsid":"266ee7a8-2a05-11eb-b846-5254002d4916","client_group":"clients","keyring":"/etc/ceph/ceph.client.admin.keyring","conf":"/etc/ceph/ceph.conf","keyring_dest":"/etc/ceph/custom.name.ceph.keyring"}
```

- b. **ceph.keyring** 의 기본 대상 키링으로 인증 키를 복사하고 기본 클라이언트 그룹을 사용하여 인증 키를 복사하려면 다음을 수행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-clients.yml --extra-vars
'{"fsid":"FSID","keyring":"KEYRING_PATH","conf":"CEPH_CONFIGURATION_PATH"}
```

예제

```
[ceph-admin@host01 cephadm-ansible]$ ansible-playbook -i hosts cephadm-clients.yml
--extra-vars '{"fsid":"266ee7a8-2a05-11eb-b846-
5254002d4916","keyring":"/etc/ceph/ceph.client.admin.keyring","conf":"/etc/ceph/ceph.conf"
}'
```

검증

- 클라이언트 노드에 로그인하고 인증 키 및 구성 파일이 있는지 확인합니다.

예제

```
[user@client01 ~]# ls -l /etc/ceph/
-rw-----. 1 ceph ceph 151 Jul 11 12:23 custom.name.ceph.keyring
-rw-----. 1 ceph ceph 151 Jul 11 12:23 ceph.keyring
-rw-----. 1 ceph ceph 269 Jul 11 12:23 ceph.conf
```

추가 리소스

- 관리자 키에 대한 자세한 내용은 *Red Hat Ceph Storage 관리 가이드의 Ceph 사용자 관리 섹션을 참조하십시오.*
-

cephadm-preflight 플레이북에 대한 자세한 내용은 *Red Hat Ceph Storage 설치 가이드*의 *Running the preflight Playbook* 섹션을 참조하십시오.

3.23. CEPH 스토리지 클러스터 제거

Ceph 스토리지 클러스터를 삭제해도 서버의 이전 배포에서 남아 있는 데이터 또는 연결이 모두 지워집니다. Red Hat Enterprise Linux 8의 경우 이 Ansible 스크립트는 스토리지 클러스터의 모든 호스트에서 스크립트에 전달된 FSID에 속하는 모든 데몬, 로그 및 데이터를 제거합니다. Red Hat Enterprise Linux 9의 경우 Ansible이 지원되지 않으므로 **cephadm rm-cluster** 명령을 사용하십시오.

For Red Hat Enterprise Linux 8

Ansible 인벤토리 파일은 클러스터의 모든 호스트와 Ceph 스토리지 클러스터에서 각 호스트가 수행하는 역할을 나열합니다. 인벤토리 파일의 기본 위치는 **/usr/share/cephadm-ansible/hosts** 이지만 이 파일은 어디에서나 배치할 수 있습니다.



중요

이 프로세스는 **cephadm** 바이너리가 스토리지 클러스터의 모든 호스트에 설치된 경우에만 작동합니다.

다음 예제는 인벤토리 파일의 구조를 보여줍니다.

예제

```
host02
host03
host04

[admin]
host01

[clients]
client01
client02
client03
```

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- **Ansible 2.9** 이상이 부트스트랩 노드에 설치됩니다.
- **Ansible** 관리 노드에 대한 루트 수준 액세스.
- **sudo** 및 암호 없이 **ssh** 를 사용하는 **Ansible** 사용자는 스토리지 클러스터의 모든 노드에 액세스할 수 있습니다.
- **[admin]** 그룹은 관리 인증 키가 **/etc/ceph/ceph.client.admin.keyring** 에 있는 노드를 사용하여 인벤토리 파일에 정의됩니다.

절차

- 부트 스트랩 노드에서 **Ansible** 사용자로 제거 스크립트를 실행합니다.

구문

```
ansible-playbook -i hosts cephadm-purge-cluster.yml -e fsid=FSID -vvv
```

예제

```
[ceph-admin@host01 cephadm-ansible]$ ansible-playbook -i hosts cephadm-purge-cluster.yml -e fsid=a6ca415a-cde7-11eb-a41a-002590fc2544 -vvv
```



참고

제거 중에 디스크 장치를 **zap**하려면 추가 **extra-var(-e ceph_origin=rhcs)**이 필요합니다.

스크립트가 완료되면 모든 **OSD** 디스크를 포함한 전체 스토리지 클러스터가 클러스터의 모든 호스트에서 제거됩니다.

For Red Hat Enterprise Linux 9

사전 요구 사항

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

절차

1. 새 데몬 배포를 방지하기 위해 모든 오케스트레이션 작업을 중지하려면 **cephadm** 을 비활성화합니다.

예제

```
[ceph: root#host01 /]# ceph mgr module disable cephadm
```

2. 클러스터의 **FSID**를 가져옵니다.

예제



```
[ceph: root#host01 /]# ceph fsid
```

3.

cephadm 셸을 종료합니다.

예제

```
[ceph: root@host01 /]# exit
```

4.

클러스터의 모든 호스트에서 **Ceph** 데몬을 삭제합니다.

구문

```
cephadm rm-cluster --force --zap-osds --fsid FSID
```

예제

```
[root@host01 ~]# cephadm rm-cluster --force --zap-osds --fsid a6ca415a-cde7-11eb-a41a-002590fc2544
```

4장. CEPHADM-ANSIBLE 모듈을 사용하여 RED HAT CEPH STORAGE 클러스터 관리

스토리지 관리자는 **Ansible** 플레이북에서 **cephadm-ansible** 모듈을 사용하여 **Red Hat Ceph Storage** 클러스터를 관리할 수 있습니다. **cephadm-ansible** 패키지는 **cephadm** 호출을 래핑하여 클러스터를 관리하기 위해 고유한 **Ansible** 플레이북을 작성할 수 있는 여러 모듈을 제공합니다.



참고

현재 **cephadm-ansible** 모듈은 가장 중요한 작업만 지원합니다. **cephadm-ansible** 모듈에서 다루지 않는 작업은 플레이북에서 **command** 또는 **shell Ansible** 모듈을 사용하여 완료해야 합니다.

4.1. CEPHADM-ANSIBLE 모듈

cephadm-ansible 모듈은 **cephadm** 및 **ceph orch** 명령에 래퍼를 제공하여 **Ansible** 플레이북 작성을 간소화하는 모듈 컬렉션입니다. 모듈을 사용하여 하나 이상의 모듈을 사용하여 클러스터를 관리하기 위해 고유한 **Ansible** 플레이북을 작성할 수 있습니다.

cephadm-ansible 패키지에는 다음 모듈이 포함되어 있습니다.

- **cephadm_bootstrap**
- **ceph_orch_host**
- **ceph_config**
- **ceph_orch_apply**
- **ceph_orch_daemon**
- **cephadm_registry_login**

4.2. CEPHADM-ANSIBLE 모듈 옵션

다음 표에는 **cephadm-ansible** 모듈에 사용할 수 있는 옵션이 나열되어 있습니다. 필요에 따라 나열된 옵션을 **Ansible** 플레이북에서 모듈을 사용할 때 설정해야 합니다. 기본값인 **true** 로 나열된 옵션은 모듈을 사용할 때 옵션이 자동으로 설정되므로 플레이북에 지정할 필요가 없음을 나타냅니다. 예를 들어 **cephadm_bootstrap** 모듈의 경우 **dashboard: false** 를 설정하지 않으면 **Ceph** 대시보드가 설치됩니다.

표 4.1. **cephadm_bootstrap** 모듈에 사용 가능한 옵션.

cephadm_bootstrap	설명	필수 항목	기본값
mon_ip	Ceph Monitor IP 주소.	true	
image	Ceph 컨테이너 이미지.	false	
docker	podman 대신 docker 를 사용하십시오.	false	
fsid	Ceph FSID를 정의합니다.	false	
pull	Ceph 컨테이너 이미지를 가져옵니다.	false	true
대시보드	Ceph 대시보드를 배포합니다.	false	true
dashboard_user	특정 Ceph 대시보드 사용자를 지정합니다.	false	
dashboard_password	Ceph 대시보드 암호.	false	
모니터링	모니터링 스택을 배포합니다.	false	true
firewalld	firewalld를 사용하여 방화벽 규칙을 관리합니다.	false	true
allow_overwrite	기존 --output-config, --output-keyring 또는 --output-pub-ssh-key 파일을 덮어쓸 수 있습니다.	false	false
registry_url	사용자 정의 레지스트리의 URL입니다.	false	

cephadm_bootstrap	설명	필수 항목	기본값
registry_username	사용자 정의 레지스트리의 사용자 이름.	false	
registry_password	사용자 정의 레지스트리의 암호입니다.	false	
registry_json	사용자 정의 레지스트리 로그인 정보가 있는 JSON 파일입니다.	false	
ssh_user	호스트에서 cephadm ssh에 사용할 SSH 사용자입니다.	false	
ssh_config	cephadm SSH 클라이언트의 SSH 구성 파일 경로입니다.	false	
allow_fqdn_hostname	FQDN(정규화된 도메인 이름)인 호스트 이름을 허용합니다.	false	false
cluster_network	클러스터 복제, 복구 및 하트비트에 사용할 서브넷입니다.	false	

표 4.2. ceph_orch_host 모듈에 사용 가능한 옵션입니다.

ceph_orch_host	설명	필수 항목	기본값
fsid	와 상호 작용할 Ceph 클러스터의 FSID입니다.	false	
image	사용할 Ceph 컨테이너 이미지입니다.	false	
name	추가, 제거 또는 업데이트 할 호스트의 이름입니다.	true	
주소	호스트의 IP 주소입니다.	상태가 있는 경우 True입니다.	
set_admin_label	지정된 호스트에서 _admin 레이블을 설정합니다.	false	false

ceph_orch_host	설명	필수 항목	기본값
labels	호스트에 적용할 라벨 목록입니다.	false	[]
state	present 로 설정하면 name 에 지정된 이름이 있는지 확인합니다. absent 로 설정하면 이름 에 지정된 호스트가 제거됩니다. drain 으로 설정하면 이름 에 지정된 호스트에서 모든 데몬을 제거하도록 예약합니다.	false	present

표 4.3. ceph_config 모듈에 사용 가능한 옵션

ceph_config	설명	필수 항목	기본값
fsid	와 상호 작용할 Ceph 클러스터의 FSID입니다.	false	
image	사용할 Ceph 컨테이너 이미지입니다.	false	
작업	옵션 에 지정된 매개 변수를 설정하거나 가져올지 여부입니다.	false	set
who	구성을 로 설정하는 데몬은 무엇입니까.	true	
옵션	설정하거나 가져올 매개 변수의 이름입니다.	true	
value	설정할 매개 변수의 값입니다.	action이 설정된 경우 true	

표 4.4. ceph_orch_apply 모듈에 사용 가능한 옵션입니다.

ceph_orch_apply	설명	필수 항목
fsid	와 상호 작용할 Ceph 클러스터의 FSID입니다.	false
image	사용할 Ceph 컨테이너 이미지입니다.	false

ceph_orch_apply	설명	필수 항목
spec	적용할 서비스 사양입니다.	true

표 4.5. ceph_orch_daemon 모듈에 사용할 수 있는 옵션입니다.

ceph_orch_daemon	설명	필수 항목
fsid	와 상호 작용할 Ceph 클러스터의 FSID입니다.	false
image	사용할 Ceph 컨테이너 이미지입니다.	false
state	이름에 지정된 서비스의 원하는 상태입니다.	true 가 시작되면 서비스가 시작되었는지 확인합니다. 중지 되면 서비스가 중지되었는지 확인합니다. 를 다시 시작하면 서비스를 다시 시작합니다.
daemon_id	서비스의 ID입니다.	true
daemon_type	서비스 유형입니다.	true

표 4.6. cephadm_registry_login 모듈에 사용 가능한 옵션

cephadm_registry_login	설명	필수 항목	기본값
state	레지스트리에서 로그인 또는 로그아웃.	false	login
docker	podman 대신 docker 를 사용하십시오.	false	
registry_url	사용자 정의 레지스트리의 URL입니다.	false	
registry_username	사용자 정의 레지스트리의 사용자 이름.	상태가 로그인 인 경우 True 입니다.	

cephadm_registry_login	설명	필수 항목	기본값
registry_password	사용자 정의 레지스트리의 암호입니다.	상태가 로그인인 경우 True 입니다.	
registry_json	JSON 파일의 경로입니다. 이 작업을 실행하기 전에 이 파일이 원격 호스트에 있어야 합니다. 이 옵션은 현재 지원되지 않습니다.		

4.3. CEPHADM_BOOTSTRAP 및 CEPHADM_REGISTRY_LOGIN 모듈을 사용하여 스토리지 클러스터 부트 스트랩

스토리지 관리자는 Ansible 플레이북에서 `cephadm_bootstrap` 및 `cephadm_registry_login` 모듈을 사용하여 Ansible을 사용하여 스토리지 클러스터를 부트스트랩할 수 있습니다.

사전 요구 사항

- 첫 번째 **Ceph Monitor** 컨테이너의 IP 주소이며, 이는 스토리지 클러스터의 첫 번째 노드의 IP 주소이기도 합니다.
- `registry.redhat.io`에 대한 로그인 액세스
- `/var/lib/containers/` 용으로 최소 **10GB**의 여유 공간이 필요
- **Red Hat Enterprise Linux 8.4 EUS** 또는 **Red Hat Enterprise Linux 8.5**.
- Ansible 관리 노드에 `cephadm-ansible` 패키지 설치
- 암호 없는 **SSH**는 스토리지 클러스터의 모든 호스트에 설정됩니다.
- 호스트는 **CDN**에 등록됩니다.

절차

1. **Ansible** 관리 노드에 로그인합니다.
2. **Ansible** 관리 노드에서 `/usr/share/cephadm-ansible` 디렉토리로 이동합니다.

예제

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

3. 호스트 파일을 생성하고 호스트, 레이블을 추가하고, 스토리지 클러스터에서 첫 번째 호스트의 IP 주소를 모니터링합니다.

구문

```
sudo vi INVENTORY_FILE

HOST1 labels=["LABEL1", 'LABEL2']
HOST2 labels=["LABEL1", 'LABEL2']
HOST3 labels=["LABEL1"]

[admin]
ADMIN_HOST monitor_address=MONITOR_IP_ADDRESS labels=["ADMIN_LABEL',
'LABEL1', 'LABEL2']
```

예제

```
[ceph-admin@admin cephadm-ansible]$ sudo vi hosts

host02 labels=["mon', 'mgr']"
host03 labels=["mon', 'mgr']"
host04 labels=["osd"]"
```

```

host05 labels=["osd]"
host06 labels=["osd]"

[admin]
host01 monitor_address=10.10.128.68 labels=["_admin', 'mon', 'mgr']"

```

4.

preflight Playbook을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars "ceph_origin=rhcs"
```

예제

```
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --extra-vars "ceph_origin=rhcs"
```

5.

클러스터를 부트스트랩할 플레이북을 생성합니다.

구문

```

sudo vi PLAYBOOK_FILENAME.yml

---
- name: NAME_OF_PLAY
  hosts: BOOTSTRAP_HOST
  become: USE_ELEVATED_PRIVILEGES
  gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
  tasks:
    -name: NAME_OF_TASK

```

```

cephadm_registry_login:
  state: STATE
  registry_url: REGISTRY_URL
  registry_username: REGISTRY_USER_NAME
  registry_password: REGISTRY_PASSWORD

- name: NAME_OF_TASK
  cephadm_bootstrap:
    mon_ip: "{{ monitor_address }}"
    dashboard_user: DASHBOARD_USER
    dashboard_password: DASHBOARD_PASSWORD
    allow_fqdn_hostname: ALLOW_FQDN_HOSTNAME
    cluster_network: NETWORK_CIDR

```

예제

```
[ceph-admin@admin cephadm-ansible]$ sudo vi bootstrap.yml
```

```

---
- name: bootstrap the cluster
  hosts: host01
  become: true
  gather_facts: false
  tasks:
    - name: login to registry
      cephadm_registry_login:
        state: login
        registry_url: registry.redhat.io
        registry_username: user1
        registry_password: mypassword1

    - name: bootstrap initial cluster
      cephadm_bootstrap:
        mon_ip: "{{ monitor_address }}"
        dashboard_user: mydashboarduser
        dashboard_password: mydashboardpassword
        allow_fqdn_hostname: true
        cluster_network: 10.10.128.0/28

```

6.

플레이북을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE PLAYBOOK_FILENAME.yml -vvv
```

예제

```
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts bootstrap.yml -vvv
```

검증

- 플레이북을 실행한 후 **Ansible** 출력을 검토합니다.

4.4. CEPH_ORCH_HOST 모듈을 사용하여 호스트 추가 또는 제거

스토리지 관리자는 **Ansible** 플레이북에서 **ceph_orch_host** 모듈을 사용하여 스토리지 클러스터에서 호스트를 추가하고 제거할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 노드를 **CDN**에 등록하고 서브스크립션을 연결합니다.
- 스토리지 클러스터의 모든 노드에 대한 **sudo** 및 암호 없는 **SSH** 액세스 권한이 있는 **Ansible** 사용자.
- **Ansible** 관리 노드에 **cephadm-ansible** 패키지 설치

- 새 호스트에는 스토리지 클러스터의 공용 **SSH** 키가 있습니다. 스토리지 클러스터의 공용 **SSH** 키를 새 호스트에 복사하는 방법에 대한 자세한 내용은 *Red Hat Ceph Storage 설치 가이드*의 [호스트 추가](#)를 참조하십시오.

절차

- 다음 절차에 따라 새 호스트를 클러스터에 추가합니다.

 - Ansible** 관리 노드에 로그인합니다.
 - Ansible** 관리 노드에서 `/usr/share/cephadm-ansible` 디렉토리로 이동합니다.

예제

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

- 새 호스트 및 레이블을 **Ansible** 인벤토리 파일에 추가합니다.

구문

```
sudo vi INVENTORY_FILE

NEW_HOST1 labels=["LABEL1", 'LABEL2']
NEW_HOST2 labels=["LABEL1", 'LABEL2']
NEW_HOST3 labels=["LABEL1"]

[admin]
ADMIN_HOST monitor_address=MONITOR_IP_ADDRESS labels=["ADMIN_LABEL',
'LABEL1', 'LABEL2']
```

예제

```
[ceph-admin@admin cephadm-ansible]$ sudo vi hosts

host02 labels=["mon', 'mgr']"
host03 labels=["mon', 'mgr']"
host04 labels=["osd']"
host05 labels=["osd']"
host06 labels=["osd']"

[admin]
host01 monitor_address= 10.10.128.68 labels=["_admin', 'mon', 'mgr']"
```



참고

이전에 새 호스트를 **Ansible** 인벤토리 파일에 추가하고 호스트에서 **preflight** 플레이북을 실행한 경우 **3** 단계로 건너뛸니다.

d.

--limit 옵션을 사용하여 **preflight** 플레이북을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars
"ceph_origin=rhcs" --limit NEWHOST
```

예제

```
[ceph-admin@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml
--extra-vars "ceph_origin=rhcs" --limit host02
```

preflight Playbook은 새 호스트에 **podman**, **lvm2**, **chrony**, **cephadm** 을 설치합니다. 설치가 완료되면 **cephadm**은 **/usr/sbin/** 디렉터리에 있습니다.

- e. 새 호스트를 클러스터에 추가하는 플레이북을 생성합니다.

구문

```
sudo vi PLAYBOOK_FILENAME.yml

---
- name: PLAY_NAME
  hosts: HOSTS_OR_HOST_GROUPS
  become: USE_ELEVATED_PRIVILEGES
  gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
  tasks:
    - name: NAME_OF_TASK
      ceph_orch_host:
        name: "{{ ansible_facts['hostname'] }}"
        address: "{{ ansible_facts['default_ipv4']['address'] }}"
        labels: "{{ labels }}"
      delegate_to: HOST_TO_DELEGATE_TASK_TO

    - name: NAME_OF_TASK
      when: inventory_hostname in groups['admin']
      ansible.builtin.shell:
        cmd: CEPH_COMMAND_TO_RUN
      register: REGISTER_NAME

    - name: NAME_OF_TASK
      when: inventory_hostname in groups['admin']
      debug:
        msg: "{{ REGISTER_NAME.stdout }}"
```



참고

기본적으로 **Ansible**은 플레이북의 **hosts** 줄과 일치하는 호스트에서 모든 작업을 실행합니다. **ceph orch** 명령은 관리자 인증 키 및 **Ceph** 구성 파일이 포함된 호스트에서 실행해야 합니다. **delegate_to** 키워드를 사용하여 클러스터에서 관리 호스트를 지정합니다.

예제

```
[ceph-admin@admin cephadm-ansible]$ sudo vi add-hosts.yml
---
- name: add additional hosts to the cluster
  hosts: all
  become: true
  gather_facts: true
  tasks:
    - name: add hosts to the cluster
      ceph_orch_host:
        name: "{{ ansible_facts['hostname'] }}"
        address: "{{ ansible_facts['default_ipv4']['address'] }}"
        labels: "{{ labels }}"
        delegate_to: host01

    - name: list hosts in the cluster
      when: inventory_hostname in groups['admin']
      ansible.builtin.shell:
        cmd: ceph orch host ls
      register: host_list

    - name: print current list of hosts
      when: inventory_hostname in groups['admin']
      debug:
        msg: "{{ host_list.stdout }}"
```

이 예제에서 플레이북은 새 호스트를 클러스터에 추가하고 현재 호스트 목록을 표시합니다.

f.

플레이북을 실행하여 클러스터에 추가 호스트를 추가합니다.

구문

```
ansible-playbook -i INVENTORY_FILE PLAYBOOK_FILENAME.yml
```

예제

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

2.

다음 절차에 따라 클러스터에서 호스트를 제거합니다.

a.

Ansible 관리 노드에 로그인합니다.

b.

Ansible 관리 노드에서 `/usr/share/cephadm-ansible` 디렉토리로 이동합니다.

예제

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

c.

클러스터에서 호스트 또는 호스트를 제거하는 플레이북을 생성합니다.

구문

```
sudo vi PLAYBOOK_FILENAME.yml
```

```

---
- name: NAME_OF_PLAY
  hosts: ADMIN_HOST
  become: USE_ELEVATED_PRIVILEGES
  gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
  tasks:
    - name: NAME_OF_TASK
      ceph_orch_host:
        name: HOST_TO_REMOVE
        state: STATE

    - name: NAME_OF_TASK
      ceph_orch_host:
        name: HOST_TO_REMOVE
        state: STATE
      retries: NUMBER_OF_RETRIES
      delay: DELAY
      until: CONTINUE_UNTIL
      register: REGISTER_NAME

    - name: NAME_OF_TASK
      ansible.builtin.shell:
        cmd: ceph orch host ls
      register: REGISTER_NAME

    - name: NAME_OF_TASK
      debug:
        msg: "{{ REGISTER_NAME.stdout }}"

```

예제

```
[ceph-admin@admin cephadm-ansible]$ sudo vi remove-hosts.yml
```

```

---
- name: remove host
  hosts: host01
  become: true
  gather_facts: true
  tasks:
    - name: drain host07
      ceph_orch_host:
        name: host07
        state: drain

    - name: remove host from the cluster
      ceph_orch_host:
        name: host07
        state: absent
      retries: 20

```

```

delay: 1
until: result is succeeded
register: result

- name: list hosts in the cluster
  ansible.builtin.shell:
    cmd: ceph orch host ls
  register: host_list

- name: print current list of hosts
  debug:
    msg: "{{ host_list.stdout }}"

```

이 예제에서 플레이북 작업은 **host07**의 모든 데몬을 트레이닝하고, 클러스터에서 호스트를 제거하고 현재 호스트 목록을 표시합니다.

d.

Playbook을 실행하여 클러스터에서 호스트를 제거합니다.

구문

```
ansible-playbook -i INVENTORY_FILE PLAYBOOK_FILENAME.yml
```

예제

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

검증

-

클러스터의 현재 호스트 목록을 표시하는 **Ansible** 작업 출력을 검토합니다.

예제

```

TASK [print current hosts]
*****
Friday 24 June 2022 14:52:40 -0400 (0:00:03.365)    0:02:31.702 *****
ok: [host01] =>
msg: |-
  HOST  ADDR      LABELS  STATUS
  host01 10.10.128.68  _admin mon mgr
  host02 10.10.128.69  mon mgr
  host03 10.10.128.70  mon mgr
  host04 10.10.128.71  osd
  host05 10.10.128.72  osd
  host06 10.10.128.73  osd

```

4.5. CEPH_CONFIG 모듈을 사용하여 구성 옵션 설정

스토리지 관리자는 `ceph_config` 모듈을 사용하여 **Red Hat Ceph Storage** 구성 옵션을 설정하거나 가져올 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 스토리지 클러스터의 모든 노드에 대한 **sudo** 및 암호 없는 **SSH** 액세스 권한이 있는 **Ansible** 사용자.
- **Ansible** 관리 노드에 **cephadm-ansible** 패키지 설치
- **Ansible** 인벤토리 파일에는 클러스터와 관리 호스트가 포함되어 있습니다.

절차

1. **Ansible** 관리 노드에 로그인합니다.

2.

Ansible 관리 노드에서 `/usr/share/cephadm-ansible` 디렉토리로 이동합니다.

예제

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

3.

구성 변경 사항으로 플레이북을 생성합니다.

구문

```
sudo vi PLAYBOOK_FILENAME.yml

---
- name: PLAY_NAME
  hosts: ADMIN_HOST
  become: USE_ELEVATED_PRIVILEGES
  gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
  tasks:
    - name: NAME_OF_TASK
      ceph_config:
        action: GET_OR_SET
        who: DAEMON_TO_SET_CONFIGURATION_TO
        option: CEPH_CONFIGURATION_OPTION
        value: VALUE_OF_PARAMETER_TO_SET

    - name: NAME_OF_TASK
      ceph_config:
        action: GET_OR_SET
        who: DAEMON_TO_SET_CONFIGURATION_TO
        option: CEPH_CONFIGURATION_OPTION
        register: REGISTER_NAME

    - name: NAME_OF_TASK
      debug:
        msg: "MESSAGE_TO_DISPLAY{{ REGISTER_NAME.stdout }}"
```

예제

```
[ceph-admin@admin cephadm-ansible]$ sudo vi change_configuration.yml

---
- name: set pool delete
  hosts: host01
  become: true
  gather_facts: false
  tasks:
    - name: set the allow pool delete option
      ceph_config:
        action: set
        who: mon
        option: mon_allow_pool_delete
        value: true

    - name: get the allow pool delete setting
      ceph_config:
        action: get
        who: mon
        option: mon_allow_pool_delete
      register: verify_mon_allow_pool_delete

    - name: print current mon_allow_pool_delete setting
      debug:
        msg: "the value of 'mon_allow_pool_delete' is {{ verify_mon_allow_pool_delete.stdout }}"
```

이 예제에서 플레이북은 먼저 **mon_allow_pool_delete** 옵션을 **false** 로 설정합니다. 그러면 **Playbook**이 현재 **mon_allow_pool_delete** 설정을 가져와서 **Ansible** 출력에 값을 표시합니다.

4.

플레이북을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE _PLAYBOOK_FILENAME.yml
```

예제

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

검증

- 플레이북 작업의 출력을 검토합니다.

예제

```
TASK [print current mon_allow_pool_delete setting]
*****
Wednesday 29 June 2022 13:51:41 -0400 (0:00:05.523)    0:00:17.953 *****
ok: [host01] =>
  msg: the value of 'mon_allow_pool_delete' is true
```

추가 리소스

- 구성 옵션에 대한 자세한 내용은 [Red Hat Ceph Storage Configuration Guide](#) 를 참조하십시오.

4.6. CEPH_ORCH_APPLY 모듈을 사용하여 서비스 사양 적용

스토리지 관리자는 **Ansible** 플레이북의 **ceph_orch_apply** 모듈을 사용하여 스토리지 클러스터에 서비스 사양을 적용할 수 있습니다. 서비스 사양은 **Ceph** 서비스를 배포하는 데 사용되는 서비스 속성 및 구성 설정을 지정하는 데이터 구조입니다. 서비스 사양을 사용하여 **mon**, **크래시**, **m****ds**, **m****gr**, **o****sd**, **r****db** 또는 **rbd-mirror** 와 같은 **Ceph** 서비스 유형을 배포할 수 있습니다.

사전 요구 사항

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

- 스토리지 클러스터의 모든 노드에 대한 **sudo** 및 암호 없는 **SSH** 액세스 권한이 있는 **Ansible** 사용자.
- **Ansible** 관리 노드에 **cephadm-ansible** 패키지 설치
- **Ansible** 인벤토리 파일에는 클러스터와 관리 호스트가 포함되어 있습니다.

절차

1. **Ansible** 관리 노드에 로그인합니다.
2. **Ansible** 관리 노드에서 **/usr/share/cephadm-ansible** 디렉토리로 이동합니다.

예제

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

3. 서비스 사양으로 플레이북을 생성합니다.

구문

```
sudo vi PLAYBOOK_FILENAME.yml

---
- name: PLAY_NAME
  hosts: HOSTS_OR_HOST_GROUPS
  become: USE_ELEVATED_PRIVILEGES
  gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
  tasks:
    - name: NAME_OF_TASK
      ceph_orch_apply:
        spec: |
```

```

service_type: SERVICE_TYPE
service_id: UNIQUE_NAME_OF_SERVICE
placement:
  host_pattern: 'HOST_PATTERN_TO_SELECT_HOSTS'
  label: LABEL
spec:
  SPECIFICATION_OPTIONS:

```

예제

```
[ceph-admin@admin cephadm-ansible]$ sudo vi deploy_osd_service.yml
```

```

---
- name: deploy osd service
  hosts: host01
  become: true
  gather_facts: true
  tasks:
    - name: apply osd spec
      ceph_orch_apply:
        spec: |
          service_type: osd
          service_id: osd
          placement:
            host_pattern: '*'
            label: osd
          spec:
            data_devices:
              all: true

```

이 예제에서 플레이북은 **osd** 레이블이 있는 모든 호스트에 **Ceph OSD** 서비스를 배포합니다.

4.

플레이북을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE_PLAYBOOK_FILENAME.yml
```

예제

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

검증

- 플레이북 작업의 출력을 검토합니다.

추가 리소스

- 서비스 사양 옵션에 대한 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 를 참조하십시오.

4.7. CEPH_ORCH_DAEMON 모듈을 사용하여 CEPH 데몬 상태 관리

스토리지 관리자는 **Ansible** 플레이북에서 `ceph_orch_daemon` 모듈을 사용하여 호스트에서 **Ceph** 데몬을 시작, 중지 및 재시작할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 스토리지 클러스터의 모든 노드에 대한 **sudo** 및 암호 없는 **SSH** 액세스 권한이 있는 **Ansible** 사용자.
- **Ansible** 관리 노드에 `cephadm-ansible` 패키지 설치

- **Ansible** 인벤토리 파일에는 클러스터와 관리 호스트가 포함되어 있습니다.

절차

1. **Ansible** 관리 노드에 로그인합니다.
2. **Ansible** 관리 노드에서 `/usr/share/cephadm-ansible` 디렉토리로 이동합니다.

예제

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

3. 데몬 상태 변경 사항을 사용하여 플레이북을 생성합니다.

구문

```
sudo vi PLAYBOOK_FILENAME.yml

---
- name: PLAY_NAME
  hosts: ADMIN_HOST
  become: USE_ELEVATED_PRIVILEGES
  gather_facts: GATHER_FACTS_ABOUT_REMOTE_HOSTS
  tasks:
    - name: NAME_OF_TASK
      ceph_orch_daemon:
        state: STATE_OF_SERVICE
        daemon_id: DAEMON_ID
        daemon_type: TYPE_OF_SERVICE
```

예제

```
[ceph-admin@admin cephadm-ansible]$ sudo vi restart_services.yml
```

```
---
- name: start and stop services
  hosts: host01
  become: true
  gather_facts: false
  tasks:
    - name: start osd.0
      ceph_orch_daemon:
        state: started
        daemon_id: 0
        daemon_type: osd

    - name: stop mon.host02
      ceph_orch_daemon:
        state: stopped
        daemon_id: host02
        daemon_type: mon
```

이 예제에서 플레이북은 ID가 0 인 OSD를 시작하고, host02 ID가 있는 Ceph Monitor를 중지합니다.

4.

플레이북을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE _PLAYBOOK_FILENAME.yml
```

예제

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

검증

- 플레이북 작업의 출력을 검토합니다.

5장. 다음에 어떻게 해야 하나요?

스토리지 관리자는 **Red Hat Ceph Storage 5**를 설치 및 구성한 후 스토리지 클러스터에 대해 "두 번" 작업을 수행할 준비가 되었습니다. 이러한 작업에는 메타데이터 서버(MDS) 및 오브젝트 게이트웨이(RGW) 추가 및 iSCSI 및 NFS와 같은 서비스 구성이 포함됩니다.

`cephadm orchestrator`를 사용하여 "Day two" 작업을 수행하는 방법에 대한 자세한 내용은 [Red Hat Ceph Storage 5 운영 가이드](#)를 참조하십시오.

"두 가지" 작업에서 **Ceph Object Gateway**를 배포, 구성 및 관리하려면 [Red Hat Ceph Storage 5 Object Gateway 가이드](#)를 참조하십시오.

부록 A. CEPH ANSIBLE과 CEPHADM 비교

Red Hat Ceph Storage 5에는 스토리지 클러스터 컨테이너화된 배포를 위한 새로운 배포 도구인 Cephadm이 도입되었습니다.

이 표는 Cephadm과 Ceph-Ansible 플레이북을 사용하여 1일 차 및 2일차 작업을 위해 Ceph 클러스터의 컨테이너화된 배포를 관리합니다.

표 A.1. 1일 운영

설명	Ceph-Ansible	Cephadm
Red Hat Ceph Storage 클러스터 설치	site-container.yml 플레이북을 실행합니다.	cephadm bootstrap 명령을 실행하여 admin 노드에 클러스터를 부트스트랩합니다.
호스트 추가	Ceph Ansible 인벤토리를 사용합니다.	ceph orch 호스트를 실행하고 HOST_NAME 을 추가하여 클러스터에 호스트를 추가합니다.
모니터 추가	add-mon.yml 플레이북을 실행합니다.	ceph orch apply mon 명령을 실행합니다.
관리자 추가	site-container.yml 플레이북을 실행합니다.	ceph orch apply mgr 명령을 실행합니다.
OSD 추가	add-osd.yml 플레이북을 실행합니다.	ceph orch apply osd 명령을 실행하여 사용 가능한 모든 장치 또는 특정 호스트에 OSD를 추가합니다.
특정 장치에 OSD 추가	osd.yml 파일에서 장치를 선택한 다음 add-osd.yml 플레이북을 실행합니다.	osd.yml 파일의 data_devices 아래에 있는 paths 필터를 선택한 다음 ceph orch apply -i Files _NAME.yml 명령을 실행합니다.
MDS 추가	site-container.yml 플레이북을 실행합니다.	ceph orch apply Files SYSTEM_NAME 명령을 실행하여 MDS를 추가합니다.
Ceph Object Gateway 추가	site-container.yml 플레이북을 실행합니다.	ceph orch apply rgw 명령을 실행하여 Ceph Object Gateway를 추가합니다.

표 A.2. 2일 차 작업

설명	Ceph-Ansible	Cephadm
호스트 제거	Ansible 인벤토리를 사용합니다.	ceph orch 호스트 rm HOST_NAME 을 실행하여 호스트를 제거합니다.
모니터 제거	shrink-mon.yml 플레이북을 실행합니다.	ceph orch가 mon 을 실행하여 다른 모니터를 재배포합니다.
관리자 제거	shrink-mon.yml 플레이북을 실행합니다.	ceph orch 를 실행하여 다른 관리자를 재배포하기 위해 mgr 을 적용합니다.
OSD 제거	shrink-osd.yml 플레이북을 실행합니다.	ceph orch osd rm OSD_ID 를 실행하여 OSD를 제거합니다.
MDS 제거	shrink-mds.yml 플레이북을 실행합니다.	ceph orch rm SERVICE_NAME 을 실행하여 특정 서비스를 제거합니다.
NFS 프로토콜을 통해 Ceph 파일 시스템 내보내기.	Red Hat Ceph Storage 4에서는 지원되지 않습니다.	ceph nfs export create 명령을 실행합니다.
Ceph Object Gateway 배포	site-container.yml 플레이북을 실행합니다.	ceph orch가 rgw SERVICE_NAME 을 실행하여 Ceph Object Gateway 서비스를 배포합니다.
Ceph Object Gateway 제거	shrink-rgw.yml 플레이북을 실행합니다.	ceph orch rm SERVICE_NAME 을 실행하여 특정 서비스를 제거합니다.
iSCSI 게이트웨이 배포	site-container.yml 플레이북을 실행합니다.	ceph orch 를 실행하여 iSCSI 게이트웨이를 배포하도록 iscsi 를 적용합니다.
블록 장치 미러링	site-container.yml 플레이북을 실행합니다.	ceph orch apply rbd-mirror 명령을 실행합니다.
Red Hat Ceph Storage의 마이너 버전 업그레이드	infrastructure-playbooks/rolling_update.yml 플레이북을 실행합니다.	ceph orch upgrade start 명령을 실행합니다.
Red Hat Ceph Storage 4에서 Red Hat Ceph Storage 5로 업그레이드	infrastructure-playbooks/rolling_update.yml 플레이북을 실행합니다.	Cephadm을 사용한 업그레이드는 지원되지 않습니다.

설명	Ceph-Ansible	Cephadm
모니터링 스택 배포	설치 중에 all.yml 파일을 편집합니다.	서비스를 지정한 후 ceph orch apply -i ____ .yml 을 실행합니다.

추가 리소스

- **Ceph Orchestrator** 사용에 대한 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 를 참조하십시오.

부록 B. CEPHADM 명령

cephadm은 **Cephadm Orchestrator**의 로컬 호스트를 관리하는 명령줄 도구입니다. 현재 호스트의 상태를 조사하고 수정하는 명령을 제공합니다.

일부 명령은 일반적으로 디버깅에 사용됩니다.



참고

모든 호스트에 **cephadm**이 필요하지는 않지만 특정 데몬을 조사할 때 유용합니다. **cephadm-ansible-preflight** 플레이북에서는 모든 호스트에 **cephadm**을 설치하고 **cephadm-ansible purge Playbook**을 모든 호스트에 설치해야 제대로 작동하려면 **cephadm**을 설치해야 합니다.

adopt

설명

업그레이드된 스토리지 클러스터 데몬을 변환하여 **cephadm**을 실행합니다.

구문

```
cephadm adopt [-h] --name DAEMON_NAME --style STYLE [--cluster CLUSTER] --legacy-dir
[LEGACY_DIR] --config-json CONFIG_JSON [--skip-firewall] [--skip-pull]
```

예제

```
[root@host01 ~]# cephadm adopt --style=legacy --name prometheus.host02
```

Ceph-volume

설명

이 명령은 특정 호스트의 모든 장치를 나열하는 데 사용됩니다. 컨테이너 내에서 **ceph-volume** 명령을 실행하여 플러그형 툴을 사용하여 **lvm** 또는 물리 디스크와 같은 다른 장치 기술로 **OSD**를 배포하고, **OSD**를 준비, 활성화 및 시작하는 예측 가능하고 강력한 방법을 따릅니다.

구문

```
cephadm ceph-volume inventory/simple/raw/lvm [-h] [--fsid FSID] [--config-json
CONFIG_JSON] [--config CONFIG, -c CONFIG] [--keyring KEYRING, -k KEYRING]
```

예제

■

```
[root@nhost01 ~]# cephadm ceph-volume inventory --fsid f64f341c-655d-11eb-8778-
fa163e914bcc
```

check-host

설명

Ceph 클러스터에 적합한 호스트 구성을 확인합니다.

구문

```
cephadm check-host [--expect-hostname HOSTNAME]
```

예제

```
[root@host01 ~]# cephadm check-host --expect-hostname host02
```

deploy

설명

로컬 호스트에 데몬을 배포합니다.

구문

```
cephadm shell deploy DAEMON_TYPE [-h] [--name DAEMON_NAME] [--fsid FSID] [--config
CONFIG, -c CONFIG] [--config-json CONFIG_JSON] [--keyring KEYRING] [--key KEY] [--osd-
fsid OSD_FSID] [--skip-firewalld] [--tcp-ports TCP_PORTS] [--reconfig] [--allow-pttrace] [--
memory-request MEMORY_REQUEST] [--memory-limit MEMORY_LIMIT] [--meta-json
META_JSON]
```

예제

```
[root@host01 ~]# cephadm shell deploy mon --fsid f64f341c-655d-11eb-8778-fa163e914bcc
```

Enter

설명

실행 중인 데몬 컨테이너 내에서 대화형 셸을 실행합니다.

구문

```
cephadm enter [-h] [--fsid FSID] --name NAME [command [command ...]]
```

예제

```
[root@host01 ~]# cephadm enter --name 52c611f2h1d9
```

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

help

설명

cephadm 에서 지원하는 모든 명령을 확인합니다.

구문

```
cephadm help
```

예제

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

install

설명

패키지를 설치합니다.

구문

```
cephadm install PACKAGES
```

예제

```
[root@host01 ~]# cephadm install ceph-common ceph-osd
```

inspect-image

설명

로컬 **Ceph** 컨테이너 이미지를 검사합니다.

구문

```
cephadm --image IMAGE_ID inspect-image
```

예제

```
[root@host01 ~]# cephadm --image
13ea90216d0be03003d12d7869f72ad9de5cec9e54a27fd308e01e467c0d4a0a inspect-image
```

list-networks

설명

IP 네트워크를 나열합니다.

구문

```
cephadm list-networks
```

예제

```
[root@host01 ~]# cephadm list-networks
```

ls

설명

호스트의 **cephadm** 에 알려진 데몬 인스턴스를 나열합니다. 명령에 **--no-detail** 을 사용하면 데몬 이름, **fsid**, 스타일 및 데몬당 **systemd** 단위에 대한 세부 정보를 제공하는 명령에 **--no-detail** 을 사용할 수 있습니다. **--legacy-dir** 옵션을 사용하여 데몬을 검색할 기존 기본 디렉터리를 지정할 수 있습니다.

구문

```
cephadm ls [--no-detail] [--legacy-dir LEGACY_DIR]
```

예제

```
[root@host01 ~]# cephadm ls --no-detail
```

logs

설명

데몬 컨테이너의 **journald** 로그를 출력합니다. **journalctl** 명령과 유사합니다.

구문

```
cephadm logs [--fsid FSID] --name DAEMON_NAME
cephadm logs [--fsid FSID] --name DAEMON_NAME -- -n NUMBER # Last N lines
cephadm logs [--fsid FSID] --name DAEMON_NAME -- -f # Follow the logs
```

예제

```
[root@host01 ~]# cephadm logs --fsid 57bddb48-ee04-11eb-9962-001a4a000672 --name
osd.8
[root@host01 ~]# cephadm logs --fsid 57bddb48-ee04-11eb-9962-001a4a000672 --name
osd.8 -- -n 20
[root@host01 ~]# cephadm logs --fsid 57bddb48-ee04-11eb-9962-001a4a000672 --name
osd.8 -- -f
```

■

prepare-host

설명

cephadm 용으로 호스트를 준비합니다.

구문

```
cephadm prepare-host [--expect-hostname HOSTNAME]
```

예제

```
[root@host01 ~]# cephadm prepare-host
[root@host01 ~]# cephadm prepare-host --expect-hostname host01
```

pull

설명

Ceph 이미지를 가져옵니다.

구문

```
cephadm [-h] [--image IMAGE_ID] pull
```

예제

```
[root@host01 ~]# cephadm --image
13ea90216d0be03003d12d7869f72ad9de5cec9e54a27fd308e01e467c0d4a0a pull
```

registry-login

설명

인증된 레지스트리에 대한 **cephadm** 로그인 정보를 제공합니다. **Cephadm**은 호출 호스트를 해당 레지스트리에 기록하려고 합니다.

구문

```
cephadm registry-login --registry-url [REGISTRY_URL] --registry-username [USERNAME] --
registry-password [PASSWORD] [--fsid FSID] [--registry-json JSON_FILE]
```

예제

```
[root@host01 ~]# cephadm registry-login --registry-url registry.redhat.io --registry-username
myuser1 --registry-password mypassword1
```

다음과 같이 포맷된 로그인 정보가 포함된 **JSON** 레지스트리 파일을 사용할 수도 있습니다.

구문

```
cat REGISTRY_FILE

{
  "url": "REGISTRY_URL",
  "username": "REGISTRY_USERNAME",
  "password": "REGISTRY_PASSWORD"
}
```

예제

```
[root@host01 ~]# cat registry_file

{
  "url": "registry.redhat.io",
  "username": "myuser",
  "password": "mypass"
}

[root@host01 ~]# cephadm registry-login -i registry_file
```

rm-daemon

설명

특정 데몬 인스턴스를 제거합니다. 호스트에서 **cephadm rm-daemon** 명령을 직접 실행하는 경우, 명령이 데몬을 제거하더라도 **cephadm mgr** 모듈에서 데몬이 누락되었음을 확인하고 재배포합니다. 이 명령은 문제가 있으며 실험적 용도 및 디버깅에만 사용해야 합니다.

구문

```
cephadm rm-daemon [--fsid FSID] [--name DAEMON_NAME] [--force ] [--force-delete-data]
```

예제

```
[root@host01 ~]# cephadm rm-daemon --fsid f64f341c-655d-11eb-8778-fa163e914bcc --name osd.8
```

rm-cluster

설명

특정 호스트가 실행되는 특정 호스트의 스토리지 클러스터에서 모든 데몬을 제거합니다. **rm-daemon** 과 유사하게 이러한 방식으로 데몬을 제거하고 **Ceph Orchestrator**가 일시 중지되지 않

고 일부 데몬이 관리되지 않는 서비스에 속하는 경우 **cephadm orchestrator**는 해당 데몬을 다시 배포합니다.

구문

```
cephadm rm-cluster [--fsid FSID] [--force]
```

예제

```
[root@host01 ~]# cephadm rm-cluster --fsid f64f341c-655d-11eb-8778-fa163e914bcc
```

rm-repo

설명

패키지 리포지토리 구성을 제거합니다. 이는 주로 **Red Hat Ceph Storage**의 연결이 끊어진 설치에 사용됩니다.

구문

```
cephadm rm-repo [-h]
```

예제

```
[root@host01 ~]# cephadm rm-repo
```

run

설명

컨테이너에서 **Ceph** 데몬을 **foreground**에서 실행합니다.

구문

```
cephadm run [--fsid FSID] --name DAEMON_NAME
```

예제

```
[root@host01 ~]# cephadm run --fsid f64f341c-655d-11eb-8778-fa163e914bcc --name osd.8
```

shell

설명

유추되거나 지정된 **Ceph** 클러스터에서 **Ceph** 명령에 액세스하여 대화형 셸을 실행합니다. **cephadm shell** 명령을 사용하여 셸을 입력하고 셸 내에서 모든 오케스트레이터 명령을 실행할 수

있습니다.

구문

```
cephadm shell [--fsid FSID] [--name DAEMON_NAME, -n DAEMON_NAME] [--config CONFIG, -c CONFIG] [--mount MOUNT, -m MOUNT] [--keyring KEYRING, -k KEYRING] [--env ENV, -e ENV]
```

예제

```
[root@host01 ~]# cephadm shell -- ceph orch ls
[root@host01 ~]# cephadm shell
```

단위

설명

이 작업을 사용하여 데몬을 시작, 중지, 다시 시작, 활성화 및 비활성화합니다. 데몬의 **systemd** 장치에서 작동합니다.

구문

```
cephadm unit [--fsid FSID] --name DAEMON_NAME start/stop/restart/enable/disable
```

예제

```
[root@host01 ~]# cephadm unit --fsid f64f341c-655d-11eb-8778-fa163e914bcc --name osd.8
start
```

version

설명

스토리지 클러스터의 버전을 제공합니다.

구문

```
cephadm version
```

예제

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

부록 C. CEPH 방화벽 포트

이는 Red Hat Ceph Storage의 다양한 구성 요소에서 사용하는 일반적인 방화벽 포트입니다.

포트	components
5000	프라이빗 레지스트리.
6800-7100	Ceph OSD.
3300	Ceph Monitor 데몬에 연결된 Ceph 클라이언트 및 Ceph 데몬. 이 포트가 우선합니다.
6789	Ceph Monitor 데몬에 연결된 Ceph 클라이언트 및 Ceph 데몬. 포트 3300이 실패한 경우 이 포트가 고려됩니다.