



Red Hat Ceph Storage 5

운영 가이드

Red Hat Ceph Storage 운영 작업

Red Hat Ceph Storage 5 운영 가이드

Red Hat Ceph Storage 운영 작업

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

법적 공지

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

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

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

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

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

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

Linux[®] 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.

초록

이 문서에서는 Red Hat Ceph Storage 운영 작업을 수행하는 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.

차례

1장. CEPH 오케스트레이터 소개	5
1.1. CEPH ORCHESTRATOR 사용	5
2장. CEPH INTENDEDCESTRATOR를 사용한 서비스 관리	7
2.1. CEPH 오케스트레이터의 배치 사양	7
2.2. 명령줄 인터페이스를 사용하여 CEPH 데몬 배포	7
2.3. 명령줄 인터페이스를 사용하여 호스트 하위 집합에 CEPH 데몬 배포	9
2.4. CEPH ORCHESTRATOR의 서비스 사양	10
2.5. 서비스 사양을 사용하여 CEPH 데몬 배포	11
3장. CEPH INTENDEDCESTRATOR를 사용한 호스트 관리	14
3.1. 사전 요구 사항	14
3.2. CEPH ORCHESTRATOR를 사용하여 호스트 추가	14
3.3. CEPH ORCHESTRATOR를 사용하여 여러 호스트 추가	16
3.4. CEPH ORCHESTRATOR를 사용하여 호스트 나열	18
3.5. CEPHORCHESTRATOR를 사용하여 호스트에 레이블 추가	18
3.6. CEPH ORCHESTRATOR를 사용하여 호스트 제거	19
3.7. CEPH ORCHESTRATOR를 사용하여 유지 관리 모드로 호스트 배치	21
4장. CEPHORCHESTRATOR를 사용한 모니터 관리	23
4.1. CEPH 모니터	23
4.2. 모니터 선택 전략 구성	24
4.3. 명령줄 인터페이스를 사용하여 CEPH 모니터 데몬 배포	24
4.4. 서비스 사양을 사용하여 CEPH 모니터 데몬 배포	27
4.5. CEPH ORCHESTRATOR를 사용하여 특정 네트워크에 모니터 데몬 배포	28
4.6. CEPHORCHESTRATOR를 사용하여 MONITOR 데몬 제거	29
4.7. 비정상 스토리지 클러스터에서 CEPH MONITOR 제거	30
5장. CEPH INTENDEDCESTRATOR를 사용한 관리자 관리	33
5.1. 사전 요구 사항	33
5.2. CEPHORCHESTRATOR를 사용하여 MANAGER 데몬 배포	33
5.3. CEPHORCHESTRATOR를 사용하여 MANAGER 데몬 제거	34
5.4. CEPH MANAGER 밸런서 모듈 사용	35
5.5. CEPH MANAGER ALERTS 모듈 사용	38
5.6. CEPH 관리자 크래시 모듈 사용	41
6장. CEPH ORCHESTRATOR를 사용한 OSD 관리	46
6.1. CEPH OSD	46
6.2. CEPH OSD 노드 구성	46
6.3. OSD 메모리 자동 튜닝	46
6.4. CEPH OSD 배포를 위한 장치 나열	48
6.5. CEPH OSD 배포를 위한 ZAPPING 장치	50
6.6. 사용 가능한 모든 장치에 CEPH OSD 배포	51
6.7. 특정 장치 및 호스트에 CEPH OSD 배포	53
6.8. OSD 배포를 위한 고급 서비스 사양 및 필터	54
6.9. 고급 서비스 사양을 사용하여 CEPH OSD 배포	57
6.10. CEPHORCHESTRATOR를 사용하여 OSD 데몬 제거	61
6.11. CEPH ORCHESTRATOR를 사용하여 OSD 교체	63
6.12. CEPHORCHESTRATOR를 사용하여 OSD 제거 중지	64
6.13. CEPH ORCHESTRATOR를 사용하여 OSD 활성화	65
6.13.1. 데이터 마이그레이션 관찰	66
6.14. 배치 그룹 재계산	66

7장. CEPHORCHESTRATOR를 사용한 모니터링 스택 관리	68
7.1. CEPHORCHESTRATOR를 사용하여 모니터링 스택 배포	68
7.2. CEPHORCHESTRATOR를 사용하여 모니터링 스택 제거	71
8장. 기본 RED HAT CEPH STORAGE 클라이언트 설정	73
8.1. 클라이언트 시스템에서 파일 설정 구성	73
8.2. 클라이언트 시스템에서 인증 키 설정	73
9장. CEPHORCHESTRATOR를 사용한 MDS 서비스 관리	75
9.1. 사전 요구 사항	75
9.2. 명령줄 인터페이스를 사용하여 MDS 서비스 배포	75
9.3. 서비스 사양을 사용하여 MDS 서비스 배포	77
9.4. CEPHORCHESTRATOR를 사용하여 MDS 서비스 제거	79
10장. CEPHORCHESTRATOR를 사용한 CEPH 개체 게이트웨이 관리	82
10.1. 사전 요구 사항	82
10.2. 명령줄 인터페이스를 사용하여 CEPH OBJECT GATEWAY 배포	82
10.3. 서비스 사양을 사용하여 CEPH OBJECT GATEWAY 배포	85
10.4. CEPH ORCHESTRATOR를 사용하여 다중 사이트 CEPH OBJECT GATEWAY 배포	88
10.5. CEPHORCHESTRATOR를 사용하여 CEPH OBJECT GATEWAY 제거	93
11장. CEPH ORCHESTRATOR(LIMITED AVAILABILITY)를 사용한 NFS-GANESHA 게이트웨이 관리	95
11.1. 사전 요구 사항	95
11.2. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 생성	95
11.3. 명령줄 인터페이스를 사용하여 NFS-GANESHA 게이트웨이 배포	97
11.4. 서비스 사양을 사용하여 NFS-GANESHA 게이트웨이 배포	98
11.5. CEPHFS/NFS 서비스용 HA 구현(기술 프리뷰)	100
11.6. HA용 독립 실행형 CEPHFS/NFS 클러스터 업그레이드	103
11.7. 사양 파일을 사용하여 CEPHFS/NFS에 대한 HA 배포	107
11.8. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 업데이트	112
11.9. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 정보 보기	114
11.10. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 로그 가져오기	115
11.11. CEPH ORCHESTRATOR를 사용하여 사용자 지정 NFS-GANESHA 구성 설정	115
11.12. CEPHORCHESTRATOR를 사용하여 사용자 정의 NFS-GANESHA 구성 재설정	119
11.13. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 삭제	121
11.14. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 게이트웨이 제거	122
12장. CEPH ORCHESTRATOR (LIMITED AVAILABILITY)를 사용한 ISCSI 게이트웨이 관리	124
12.1. 사전 요구 사항	124
12.2. 명령줄 인터페이스를 사용하여 ISCSI 게이트웨이 배포	124
12.3. 서비스 사양을 사용하여 ISCSI 게이트웨이 배포	125
12.4. CEPHORCHESTRATOR를 사용하여 ISCSI 게이트웨이 제거	127
13장. SNMP 트랩 설정	129
13.1. 간단한 네트워크 관리 프로토콜	129
13.2. SNMPTRAPD 구성	130
13.3. SNMP 게이트웨이 배포	133
14장. 노드 오류 처리	138
14.1. 사전 요구 사항	139
14.2. 노드를 추가하거나 제거하기 전에 고려해야 할 사항	140
14.3. 성능 고려 사항	140
14.4. 노드 추가 또는 제거 권장 사항	141
14.5. CEPH OSD 노드 추가	143
14.6. CEPH OSD 노드 제거	146

14.7. 노드 오류 시뮬레이션	149
15장. 데이터 센터 오류 처리	152
15.1. 사전 요구 사항	152
15.2. 데이터 센터 오류 방지	152
15.3. 데이터 센터 오류 처리	153

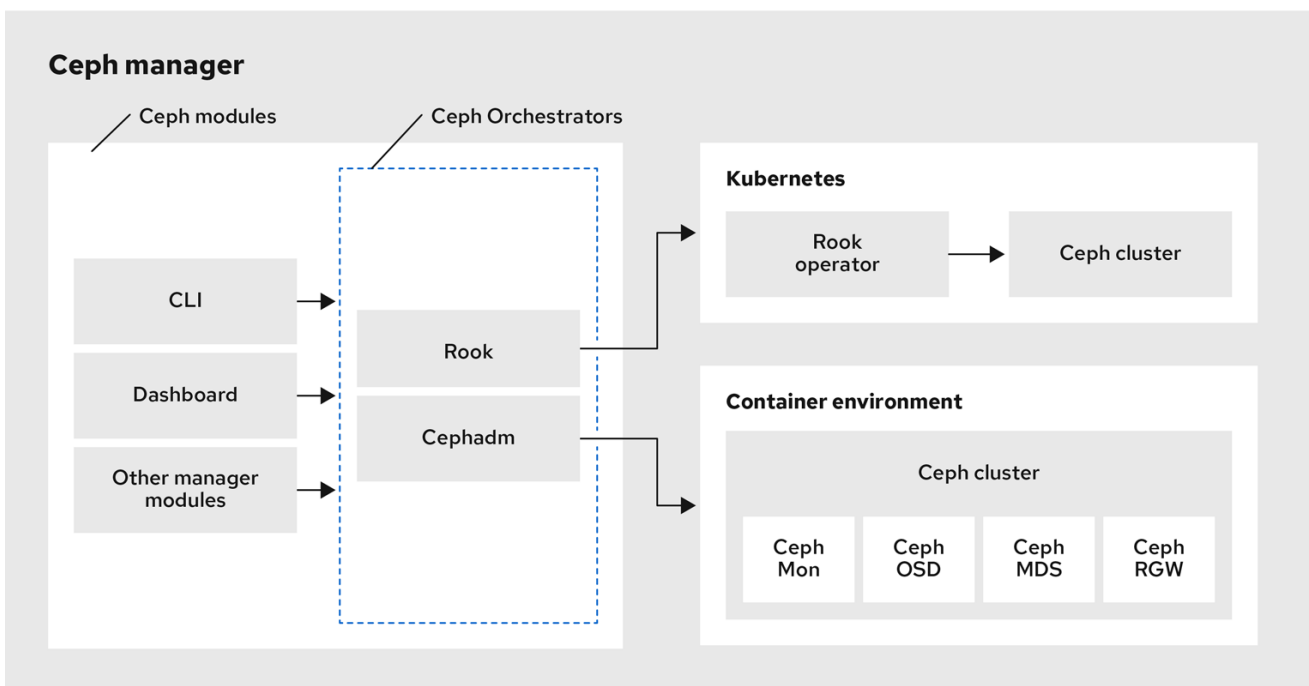
1장. CEPH 오케스트레이터 소개

스토리지 관리자는 Red Hat Ceph Storage 클러스터에서 장치를 검색하고 서비스를 생성하는 기능을 제공하는 Cephadm 유틸리티와 함께 Ceph Orchestrator를 사용할 수 있습니다.

1.1. CEPH ORCHESTRATOR 사용

Red Hat Ceph Storage Orchestrators는 Red Hat Ceph Storage 클러스터와 통합된 환경을 위한 Rook 및 Cephadm과 같은 배포 도구 간의 브리지 역할을 하는 관리자 모듈입니다. 또한 Ceph 명령줄 인터페이스 및 Ceph 대시보드와 통합됩니다.

다음은 CephOrchestrator의 워크플로 다이어그램입니다.



153_Ceph_0421

Red Hat Ceph Storage 오케스트레이터 유형

Red Hat Ceph Storage Orchestrators의 세 가지 주요 유형은 다음과 같습니다.

- **orchestrator CLI**: 이는 오케스트레이터에서 사용되는 일반적인 API이며 구현할 수 있는 명령 집합을 포함합니다. 또한 이러한 API는 외부 오케스트레이션 서비스를 사용하여 **ceph-mgr** 모듈을 오케스트레이션하는 일반 CLI(명령줄 인터페이스)를 제공합니다. 다음은 Ceph intendedchestrator와 함께 사용되는 용어입니다.
 - **호스트**: 이는 컨테이너 내부의 포트 이름, DNS 이름, 컨테이너 이름 또는 호스트 이름이 아닌 물리적 호스트의 호스트 이름입니다.
 - **서비스 유형**: 이는 nfs, mds, osd, mon, rgw, mgr, iscsi와 같은 서비스 유형입니다.
 - **서비스**: 모니터링 서비스, 관리자 서비스, OSD 서비스, Ceph Object Gateway 서비스 및 NFS 서비스와 같은 Ceph 스토리지 클러스터에서 제공하는 기능 서비스입니다.
 - **daemon**: Ceph Object Gateway 서비스와 같은 하나 이상의 호스트에 의해 배포된 서비스의 특정 인스턴스는 세 개의 다른 호스트에서 다른 Ceph Object Gateway 데몬을 실행할 수 있습니다.

- **Cephadm Orchestrator** - 이는 Rook 또는 Ansible과 같은 외부 툴에 의존하지 않고 SSH 연결을 설정하고 명시적 관리 명령을 실행하여 클러스터의 노드를 관리하는 Ceph Orchestrator 모듈입니다. 이 모듈은 day-one 및 day-two 작업을 위한 것입니다.

Cephadm Orchestrator를 사용하면 Ansible과 같은 배포 프레임워크를 활용하지 않고도 Ceph 스토리지 클러스터를 설치하는 것이 좋습니다. 관리자 데몬에 SSH 구성 및 키에 대한 액세스 권한을 제공하여 스토리지 장치의 인벤토리 생성, OSD 배포 및 교체, Ceph 데몬 시작 및 중지와 같은 관리 작업을 수행할 수 있는 모든 관리 작업을 수행할 수 있습니다. 또한 Cephadm Orchestrator는 공동 배치된 서비스를 독립적으로 업그레이드할 수 있도록 **systemd** 에서 관리하는 컨테이너 이미지를 배포합니다.

또한 이 오케스트레이터는 Ceph Monitor 및 Ceph Manager를 실행하는 최소 클러스터를 부트스트랩하는 명령을 포함하여 현재 호스트에서 컨테이너 이미지 기반 서비스 배포를 관리하는 데 필요한 모든 작업을 캡슐화하는 툴을 강조할 수 있습니다.

- **rook Orchestrator** - Rook은 Kubernetes 클러스터 내에서 실행되는 Ceph 스토리지 클러스터를 관리하기 위해 Kubernetes Rook Operator를 사용하는 오케스트레이션 도구입니다. rook 모듈은 Ceph의 Orchestrator 프레임워크와 Rook 간의 통합을 제공합니다. rook은 Kubernetes용 오픈 소스 클라우드 네이티브 스토리지 Operator입니다.

rook은 "operator" 모델을 따릅니다. 이 모델에서는 Ceph 스토리지 클러스터와 원하는 상태를 설명하기 위해 Kubernetes에 CRD(사용자 정의 리소스 정의) 오브젝트가 정의되고, rook Operator 데몬이 현재 클러스터 상태를 원하는 상태와 비교하여 원하는 상태로 전환하도록 하는 단계를 취합니다. Ceph의 원하는 상태를 설명하는 기본 오브젝트는 Ceph 스토리지 클러스터 CRD로, OSD에서 사용해야 하는 장치, 실행 중인 모니터 수, 사용할 Ceph 버전에 대한 정보가 포함됩니다.

rook은 RBD 풀, CephFS 파일 시스템 등을 설명하는 여러 다른 CRD를 정의합니다.

Rook Orchestrator 모듈은 **ceph-mgr** 데몬에서 실행되며 원하는 클러스터 상태를 설명하는 Kubernetes에서 Ceph 스토리지 클러스터를 변경하여 Ceph 오케스트레이션 API를 구현하는 글루스입니다. Rook 클러스터의 **ceph-mgr** 데몬은 Kubernetes Pod로 실행되고 있으므로 rook 모듈은 명시적 구성 없이 Kubernetes API에 연결할 수 있습니다.

2장. CEPH INTENDEDCESTRATOR를 사용한 서비스 관리

스토리지 관리자는 Red Hat Ceph Storage 클러스터를 설치한 후 Ceph Orchestrator를 사용하여 스토리지 클러스터에서 서비스를 모니터링하고 관리할 수 있습니다. 서비스는 함께 구성된 데몬 그룹입니다.

이 섹션에서는 다음 관리 정보를 다룹니다.

- [Ceph 오케스트레이터의 배치 사양.](#)
- [명령줄 인터페이스를 사용하여 Ceph 데몬 배포.](#)
- [명령줄 인터페이스를 사용하여 호스트 하위 세트에 Ceph 데몬을 배포합니다.](#)
- [Ceph 오케스트레이터의 서비스 사양.](#)
- [서비스 사양을 사용하여 Ceph 데몬 배포.](#)

2.1. CEPH 오케스트레이터의 배치 사양

Ceph Orchestrator를 사용하여 **osds,mons,mgrs,mds** 및 **rgw, iSCSI** 서비스를 배포할 수 있습니다. 배치 사양을 사용하여 서비스를 배포하는 것이 좋습니다. Ceph intendedchestrator를 사용하여 서비스를 배포하기 위해 배포해야 하는 데몬 수와 위치를 알아야 합니다. 배치 사양은 명령줄 인수로 또는 **yaml** 파일에서 서비스 사양으로 전달할 수 있습니다.

배치 사양을 사용하여 서비스를 배포하는 방법은 다음 두 가지가 있습니다.

- 명령줄 인터페이스에서 배치 사양을 직접 사용합니다. 예를 들어 호스트에 세 개의 모니터를 배포하려면 다음 명령을 실행하면 **host01,host02, host03**에 세 개의 모니터를 배포합니다.

예제

```
[ceph: root@host01 /]# ceph orch apply mon --placement="3 host01 host02 host03"
```

- YAML 파일에 배치 사양 사용. 예를 들어 모든 호스트에 **node-exporter**를 배포하려면 **yaml** 파일에서 다음을 지정할 수 있습니다.

예제

```
service_type: node-exporter
placement:
  host_pattern: '*'
```

2.2. 명령줄 인터페이스를 사용하여 CEPH 데몬 배포

Ceph Orchestrator를 사용하면 Ceph Manager, Ceph Monitor, Ceph OSD, 모니터링 스택 및 **ceph orch** 명령을 사용하여 데몬을 배포할 수 있습니다. 배치 사양은 orchestrator 명령을 사용하여 **--placement** 인수로 전달됩니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트가 스토리지 클러스터에 추가됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 다음 방법 중 하나를 사용하여 호스트에 데몬을 배포합니다.

- **방법 1:** 데몬 수 및 호스트 이름을 지정합니다.

구문

```
ceph orch apply SERVICE_NAME --placement="NUMBER_OF_DAEMONS  
HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

예제

```
[ceph: root@host01 /]# ceph orch apply mon --placement="3 host01 host02 host03"
```

- **방법 2:** 호스트에 레이블을 추가한 다음 라벨을 사용하여 데몬을 배포합니다.
 - a. 호스트에 라벨을 추가합니다.

구문

```
ceph orch host label add HOSTNAME_1 LABEL
```

예제

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

- b. 라벨을 사용하여 데몬을 배포합니다.

구문

```
ceph orch apply DAEMON_NAME label:LABEL
```

예제

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

- **방법 3:** 호스트에 레이블을 추가하고 **--placement** 인수를 사용하여 배포합니다.
 - a. 호스트에 라벨을 추가합니다.

구문

```
ceph orch host label add HOSTNAME_1 LABEL
```

예제

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

- b. 라벨 배치 사양을 사용하여 데몬을 배포합니다.

구문

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

예제

```
ceph orch apply mon --placement="label:mon"
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon  
[ceph: root@host01 /]# ceph orch ps --service_name=mon
```

추가 리소스

- *Red Hat Ceph Storage Operations Guide* [의 Ceph Orchestrator를 사용하여 호스트 추가](#) 섹션을 참조하십시오.

2.3. 명령줄 인터페이스를 사용하여 호스트 하위 집합에 CEPH 데몬 배포

--placement 옵션을 사용하여 호스트 서브 세트에 데몬을 배포할 수 있습니다. 배치 사양에 데몬을 배포할 호스트 이름으로 데몬 수를 지정할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. Ceph 데몬을 배포할 호스트를 나열합니다.

예제

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

3. 데몬을 배포합니다.

구문

```
ceph orch apply SERVICE_NAME --placement="NUMBER_OF_DAEMONS HOST_NAME_1
_HOST_NAME_2 HOST_NAME_3"
```

예제

```
ceph orch apply mgr --placement="2 host01 host02 host03"
```

이 예제에서는 **mgr** 데몬이 두 개의 호스트에만 배포됩니다.

검증

- 호스트를 나열합니다.

예제

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

추가 리소스

- Red Hat [Ceph Storage Operations Guide](#) 의 [Ceph Orchestrator](#) 를 사용하여 호스트 목록을 참조하십시오.

2.4. CEPH ORCHESTRATOR의 서비스 사양

서비스 사양은 Ceph 서비스를 배포하는 데 사용되는 서비스 속성 및 구성 설정을 지정하는 데이터 구조입니다. 다음은 서비스 사양을 지정하는 다중 문서 YAML 파일 **cluster.yaml** 의 예입니다.

예제

```
service_type: mon
placement:
  host_pattern: "mon*"
---
service_type: mgr
placement:
  host_pattern: "mgr*"
---
service_type: osd
```

```

service_id: default_drive_group
placement:
  host_pattern: "osd*"
data_devices:
  all: true

```

다음 목록은 서비스 사양의 속성이 다음과 같이 정의된 매개변수입니다.

- **service_type**: 서비스 유형:
 - mon, crash, mds, mgr, osd, rbd 또는 rbd-mirror와 같은 Ceph 서비스.
 - nfs 또는 rgw와 같은 Ceph 게이트웨이.
 - Alertmanager, Prometheus, Grafana 또는 Node-exporter와 같은 스택 모니터링.
 - 사용자 지정 컨테이너의 컨테이너입니다.
- **service_id**: 서비스의 고유 이름입니다.
- **배치**: 데몬을 배포할 위치와 방법을 정의하는 데 사용됩니다.
- **Unmanaged: true** 로 설정하면 Orchestrator가 이 서비스와 관련된 데몬을 배포하거나 제거하지 않습니다.

orchestrators의 상태 비저장 서비스

상태 비저장 서비스는 상태 정보를 사용할 필요가 없는 서비스입니다. 예를 들어, **rgw** 서비스를 시작하려면 서비스를 시작하거나 실행하는 데 추가 정보가 필요하지 않습니다. **rgw** 서비스는 기능을 제공하기 위해 이 상태에 대한 정보를 생성하지 않습니다. **rgw** 서비스가 시작되는 시기와 관계없이 상태는 동일합니다.

2.5. 서비스 사양을 사용하여 CEPH 데몬 배포

Ceph Orchestrator를 사용하면 Ceph Manager, Ceph Monitor, Ceph OSD, 모니터링 스택 등의 데몬을 YAML 파일의 서비스 사양을 사용하여 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.

절차

1. **yaml** 파일을 생성합니다.

예제

```
[root@host01 ~]# touch mon.yaml
```

2. 이 파일은 다음 두 가지 방법으로 구성할 수 있습니다.

- 배치 사양에 호스트 세부 정보를 포함하도록 파일을 편집합니다.

구문

```
service_type: SERVICE_NAME
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
```

예제

```
service_type: mon
placement:
  hosts:
    - host01
    - host02
    - host03
```

- 배치 사양에 라벨 세부 정보를 포함하도록 파일을 편집합니다.

구문

```
service_type: SERVICE_NAME
placement:
  label: "LABEL_1"
```

예제

```
service_type: mon
placement:
  label: "mon"
```

3. 컨테이너의 디렉터리에 YAML 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount mon.yaml:/var/lib/ceph/mon/mon.yaml
```

4. 디렉터리로 이동합니다.

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/mon/
```

5. 서비스 사양을 사용하여 Ceph 데몬을 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yaml
```

예제

```
[ceph: root@host01 mon]# ceph orch apply -i mon.yaml
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

추가 리소스

- Red Hat [Ceph Storage Operations Guide](#) 의 *Ceph Orchestrator*를 사용하여 호스트 목록을 참조하십시오.

3장. CEPH INTENDEDCESTRATOR를 사용한 호스트 관리

스토리지 관리자는 백엔드에서 Cephadm과 함께 Ceph Orchestrator를 사용하여 기존 Red Hat Ceph Storage 클러스터에서 호스트를 추가, 나열 및 제거할 수 있습니다.

호스트에 라벨을 추가할 수도 있습니다. 레이블은 자유 형식이며 구체적인 의미가 없습니다. 각 호스트에는 여러 개의 레이블이 있을 수 있습니다. 예를 들어, 모니터 데몬이 배포된 모든 호스트에 **mon** 레이블을 적용하고, 관리자 데몬이 배포된 모든 호스트에 대해 **mgr**, Ceph 개체 게이트웨이 **rgw** 등을 사용합니다.

스토리지 클러스터의 모든 호스트에 레이블을 지정하면 각 호스트에서 실행 중인 데몬을 빠르게 확인할 수 있으므로 시스템 관리 작업을 단순화할 수 있습니다. 또한 CephOrchestrator 또는 YAML 파일을 사용하여 특정 호스트 레이블이 있는 호스트에서 데몬을 배포하거나 제거할 수 있습니다.

이 섹션에서는 다음 관리 작업에 대해 설명합니다.

- [Ceph Orchestrator를 사용하여 호스트 추가.](#)
- [Ceph Orchestrator를 사용하여 여러 호스트 추가.](#)
- [Ceph Orchestrator를 사용하여 호스트 나열.](#)
- [Ceph Orchestrator를 사용하여 호스트에 레이블 추가.](#)
- [Ceph 오케스트레이터를 사용하여 호스트 제거.](#)
- [Ceph Orchestrator를 사용하여 유지 관리 모드로 호스트 배치.](#)

3.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 새 호스트의 IP 주소는 **/etc/hosts** 파일에서 업데이트해야 합니다.

3.2. CEPH ORCHESTRATOR를 사용하여 호스트 추가

백엔드의 Cephadm과 함께 Ceph Orchestrator를 사용하여 기존 Red Hat Ceph Storage 클러스터에 호스트를 추가할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준의 액세스.
- 노드를 CDN에 등록하고 서브스크립션을 연결합니다.
- 스토리지 클러스터의 모든 노드에 대해 **sudo** 및 암호 없는 **ssh** 액세스 권한을 가진 Ansible 사용자.

절차

1. Ceph 관리 노드에서 Cephadm 셸에 로그인합니다.

예제

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

- 클러스터의 공용 SSH 키를 폴더에 추출합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph cephadm get-pub-key > ~/ceph.pub
```

- Ceph 클러스터의 공용 SSH 키를 새 호스트의 root 사용자 **authorized_keys** 파일에 복사합니다.

구문

```
ssh-copy-id -f -i ~/PATH root@HOST_NAME_2
```

예제

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

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

예제

```
host01
host02
host03

[admin]
host00
```



참고

이전에 새 호스트를 Ansible 인벤토리 파일에 추가하고 호스트에서 preflight 플레이북을 실행한 경우 6단계로 건너뜁니다.

- limit** 옵션을 사용하여 preflight 플레이북을 실행합니다.

구문

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --limit NEWHOST
```

예제

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

preflight 플레이북에서는 새 호스트에 **podman, lvm2, chronyd, cephadm** 을 설치합니다. 설치가 완료되면 **cephadm** 이 **/usr/sbin/** 디렉터리에 있습니다.

6. Ceph 관리 노드에서 Cephadm 셸에 로그인합니다.

예제

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

7. **cephadm** orchestrator를 사용하여 스토리지 클러스터에 호스트를 추가합니다.

구문

```
ceph orch host add HOST_NAME IP_ADDRESS_OF_HOST
```

예제

```
[ceph: root@host01 /]# ceph orch host add host02 10.10.128.70
```

검증

- 호스트를 나열합니다.

예제

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

추가 리소스

- Red Hat [Ceph Storage Operations Guide](#) 의 [Ceph Orchestrator를 사용하여 호스트](#) 목록을 참조하십시오.
- **cephadm-preflight** 플레이북에 대한 자세한 내용은 [Red Hat Ceph Storage 설치 가이드](#) 의 [preflight 플레이북 실행](#) 섹션을 참조하십시오.
- [Red Hat Ceph Storage 설치 가이드](#)의 [CDN에 Red Hat Ceph Storage 노드 등록 및 서브스크립션 연결](#) 섹션을 참조하십시오.
- [Red Hat Ceph Storage 설치 가이드](#)의 [sudo 액세스로 Ansible 사용자 생성](#) 섹션을 참조하십시오.

3.3. CEPH ORCHESTRATOR를 사용하여 여러 호스트 추가

Ceph Orchestrator를 사용하여 YAML 파일 형식의 서비스 사양을 사용하여 Red Hat Ceph Storage 클러스터에 여러 호스트를 추가할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.

절차

1. **hosts.yaml** 파일을 생성합니다.

예제

```
[root@host01 ~]# touch hosts.yaml
```

2. 다음 세부 정보를 포함하도록 **hosts.yaml** 파일을 편집합니다.

예제

```
service_type: host
addr: host01
hostname: host01
labels:
- mon
- osd
- mgr
---
service_type: host
addr: host02
hostname: host02
labels:
- mon
- osd
- mgr
---
service_type: host
addr: host03
hostname: host03
labels:
- mon
- osd
```

3. 컨테이너의 디렉터리에 YAML 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount hosts.yaml:/var/lib/ceph/hosts.yaml
```

4. 디렉터리로 이동합니다.

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/
```

5. 서비스 사양을 사용하여 호스트를 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yaml
```

예제

```
[ceph: root@host01 hosts]# ceph orch apply -i hosts.yaml
```

검증

- 호스트를 나열합니다.

예제

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

추가 리소스

- Red Hat [Ceph Storage Operations Guide](#) 의 [Ceph Orchestrator](#)를 사용하여 호스트 목록을 참조하십시오.

3.4. CEPH ORCHESTRATOR를 사용하여 호스트 나열

Ceph Orchestrators를 사용하여 Ceph 클러스터 호스트를 나열할 수 있습니다.



참고

ceph orch 호스트 ls 명령의 출력에서 호스트 상태는 비어 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트가 스토리지 클러스터에 추가됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 클러스터의 호스트를 나열합니다.

예제

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

호스트 STATUS 가 예상되는 것을 확인할 수 있습니다.

3.5. CEPHORCHESTRATOR를 사용하여 호스트에 레이블 추가

Ceph Orchestrator를 사용하여 기존 Red Hat Ceph Storage 클러스터의 호스트에 레이블을 추가할 수 있습니다. 레이블의 몇 가지 예는 호스트에 배포된 서비스를 기반으로 **mgr**, **mon** 및 **osd**입니다.

cephadm에 특별한 의미가 있고 _로 시작하는 다음 호스트 레이블을 추가할 수도 있습니다.

- **_no_schedule**: 이 레이블을 사용하면 **cephadm** 이 호스트에 데몬을 예약하거나 배포할 수 없습니다. Ceph 데몬이 이미 포함된 기존 호스트에 추가하면 **cephadm** 이 자동으로 제거되지 않는 OSD를 제외하고 다른 곳에서 해당 데몬을 이동합니다. **_no_schedule** 레이블을 사용하여 호스트를 추가하면 데몬이 배포되지 않습니다. 호스트가 제거되기 전에 데몬이 드레인되면 해당 호스트에 **_no_schedule** 레이블이 설정됩니다.
- **_no_autotune_memory**: 이 레이블은 호스트에서 메모리를 자동 튜닝하지 않습니다. 해당 호스트의 하나 이상의 데몬에 대해 **osd_memory_target_autotune** 옵션 또는 기타 유사한 옵션이 활성화되어 있어도 데몬 메모리가 조정되지 않습니다.
- **_admin**: 기본적으로 **_admin** 레이블은 스토리지 클러스터의 부트스트랩 호스트에 적용되며 **client.admin** 키는 **ceph orch client-keyring {ls|set|rm}** 함수를 사용하여 해당 호스트에 배포되도록 설정됩니다. 이 레이블을 추가 호스트에 추가하면 일반적으로 **cephadm** 이 **/etc/ceph** 디렉터리에 구성 및 인증 키 파일을 배포합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 스토리지 클러스터에 호스트가 추가됨

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 호스트에 라벨을 추가합니다.

구문

```
ceph orch host label add HOST_NAME LABEL_NAME
```

예제

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

검증

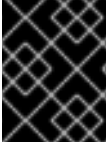
- 호스트를 나열합니다.

예제

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

3.6. CEPH ORCHESTRATOR를 사용하여 호스트 제거

Ceph Orchestrators를 사용하여 Ceph 클러스터 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 복사하십시오.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트가 스토리지 클러스터에 추가됩니다.
- 모든 서비스가 배포됩니다.
- Cephadm은 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 드레이닝합니다.

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 OSD가 해제되어 제거됩니다.

5. 스토리지 클러스터에서 모든 데몬이 제거되었는지 확인합니다.

구문

```
ceph orch ps HOSTNAME
```

예제

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

6. 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 운영 가이드](#)의 Ceph 오케스트레이터를 사용하여 호스트 추가 섹션을 참조하십시오.
- 자세한 내용은 Red Hat [Ceph Storage Operations Guide](#)의 CephOrchestrator를 사용하여 호스트 목록 섹션을 참조하십시오.

3.7. CEPH ORCHESTRATOR를 사용하여 유지 관리 모드로 호스트 배치

Ceph Orchestrator를 사용하여 유지 관리 모드에 호스트를 배치하거나 해제할 수 있습니다. 이렇게 하면 호스트의 모든 Ceph 데몬이 중지됩니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 클러스터에 추가된 호스트입니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 호스트를 유지 관리 모드로 배치하거나 유지보수 모드에서 배치할 수 있습니다.

- 호스트를 유지 관리 모드로 배치합니다.

구문

```
ceph orch host maintenance enter HOST_NAME [--force]
```

예제

```
[ceph: root@host01 /]# ceph orch host maintenance enter host02 --force
```

--force 플래그를 사용하면 경고를 우회할 수 있지만 경고는 하지 않습니다.

- 호스트를 유지 관리 모드에서 배치합니다.

구문

```
ceph orch host maintenance exit HOST_NAME
```

예제

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

검증

- 호스트를 나열합니다.

예제

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

4장. CEPHORCHESTRATOR를 사용한 모니터 관리

스토리지 관리자는 배치 사양을 사용하여 추가 모니터를 배포하고, 서비스 사양을 사용하여 모니터를 추가하고, 서브넷 구성에 모니터를 추가하고, 특정 호스트에 모니터를 추가할 수 있습니다. 이 외에도 Ceph Orchestrator를 사용하여 모니터를 제거할 수 있습니다.

기본적으로 Red Hat Ceph Storage 클러스터에는 다른 호스트에 배포된 3개 또는 5개의 모니터 데몬이 있습니다.

클러스터에 노드가 5개 이상인 경우 5개의 모니터를 배포할 것을 권장합니다.

Ceph는 클러스터가 확장될 때 데몬을 자동으로 배포하고 클러스터가 축소되면 데몬을 자동으로 모니터링합니다. 이러한 자동 확장 및 축소는 적절한 서브넷 구성에 따라 원활하게 실행됩니다.

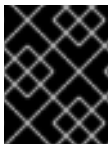
모니터 노드 또는 전체 클러스터가 단일 서브넷에 있는 경우 클러스터에 새 호스트를 추가하면 Cephadm에서 최대 5개의 모니터 데몬을 자동으로 추가합니다. Cephadm은 새 호스트에서 모니터 데몬을 자동으로 구성합니다. 새 호스트는 스토리지 클러스터에서 부트스트랩된 호스트와 동일한 서브넷에 있습니다.

또한 Cephadm은 스토리지 클러스터의 크기 변경에 따라 모니터를 배포 및 확장할 수 있습니다.

4.1. CEPH 모니터

Ceph 모니터는 스토리지 클러스터 맵의 마스터 복사본을 유지하는 경량 프로세스입니다. 모든 Ceph 클라이언트가 Ceph에 접속하여 스토리지 클러스터 맵의 현재 사본을 검색하여 클라이언트가 풀에 바인딩하고 데이터를 읽고 쓸 수 있습니다.

Ceph 모니터는 Paxos 프로토콜의 변형을 사용하여 맵과 스토리지 클러스터 전체의 기타 중요 정보에 대한 합의점을 구축합니다. Ceph는 Paxos의 특성으로 인해 쿼럼을 설정하기 위해 대부분의 모니터를 실행해야 하므로 합의점을 설정합니다.



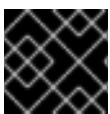
중요

프로덕션 클러스터에 대한 지원을 받으려면 별도의 호스트에 3개 이상의 모니터가 필요합니다.

홀수의 모니터 배포를 권장합니다. 홀수의 Ceph 모니터는 심지어 모니터 수보다 실패 가능성이 높습니다. 예를 들어, 2개 모니터 배포에서 쿼럼을 유지 관리하기 위해 Ceph는 3개의 모니터, 1개의 모니터, 1개의 모니터, 5개의 모니터가 있는 2개의 오류를 허용할 수 없습니다. 따라서 홀수의 숫자를 사용하는 것이 좋습니다. Ceph에서는 대부분의 모니터가 실행되고 있고 서로 통신할 수 있어야 하며, 3개 중 2개, 4개 중 3개 등에서 통신할 수 있어야 합니다.

멀티 노드 Ceph 스토리지 클러스터의 초기 배포를 위해 Red Hat은 세 개 이상의 모니터가 있는 경우 한 번에 2개 이상의 모니터를 늘려야 합니다.

Ceph 모니터는 경량이므로 OpenStack 노드와 동일한 호스트에서 실행할 수 있습니다. 그러나 Red Hat은 별도의 호스트에서 모니터를 실행하는 것이 좋습니다.



중요

Red Hat은 컨테이너화된 환경에서 Ceph 서비스를 조합하여만 지원합니다.

스토리지 클러스터에서 모니터를 제거하면 Ceph 모니터가 Paxos 프로토콜을 사용하여 마스터 스토리지 클러스터 맵에 대한 합의점을 설정한다고 고려하십시오. 쿼럼을 설정하기 위해 충분한 수의 Ceph 모니터가 있어야 합니다.

추가 리소스

- 지원되는 모든 Ceph 구성은 [Red Hat Ceph Storage 지원 구성 지식 베이스 문서](#)를 참조하십시오.

4.2. 모니터 선택 전략 구성

모니터 선택 전략은 순 분할을 식별하고 오류를 처리합니다. 선택 모니터 전략을 세 가지 모드로 구성할 수 있습니다.

1. **Classic** - 가장 낮은 순위의 모니터가 두 사이트 간의 투표 모듈에 따라 투표되는 기본 모드입니다.
2. **Disallow** - 이 모드를 사용하면 모니터를 허용하지 않은 것으로 표시할 수 있습니다. 이 경우 쿼럼에 참여하고 클라이언트에 서비스를 제공하지만 선택한 리더가 될 수는 없습니다. 이를 통해 허용하지 않은 리더 목록에 모니터를 추가할 수 있습니다. 모니터가 비활성화 목록에 있으면 항상 다른 모니터로 지연됩니다.
3. **연결** - 이 모드는 주로 네트워크 불일치를 해결하는 데 사용됩니다. 각 모니터에서 제공하는 활성 상태를 점검하는 ping을 기반으로 연결 점수를 평가하고, 가장 많이 연결되고 신뢰할 수 있는 모니터를 리더로 선택합니다. 이 모드는 순 분할을 처리하도록 설계되었으므로 클러스터가 여러 데이터 센터로 확장되거나 취약성이 발생할 수 있습니다. 이 모드에서는 연결 점수 등급을 통합하고 가장 좋은 점수로 모니터를 선택합니다. 특정 모니터가 리더가 되기를 원하는 경우 목록의 첫 번째 모니터가 0인 목록의 첫 번째 모니터가 되도록 선택 전략을 구성합니다.

다른 모드의 기능이 필요하지 않은 한, 기존의 모드를 유지할 것을 권장합니다.

클러스터를 구성하기 전에 다음 명령에서 **election_strategy**를 **classic**, **disallow** 또는 **connectivity**로 변경합니다.

구문

```
ceph mon set election_strategy {classic|disallow|connectivity}
```

4.3. 명령줄 인터페이스를 사용하여 CEPH 모니터 데몬 배포

Ceph Orchestrator는 기본적으로 하나의 모니터 데몬을 배포합니다. 명령줄 인터페이스에서 **배치** 사양을 사용하여 추가 모니터 데몬을 배포할 수 있습니다. 다른 수의 모니터 데몬을 배포하려면 다른 번호를 지정합니다. 모니터 데몬을 배포해야 하는 호스트를 지정하지 않으면 Ceph Orchestrator가 호스트를 무작위로 선택하고 모니터 데몬을 배포합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. Ceph 모니터 데몬을 배포하는 네 가지 방법이 있습니다.

방법 1

- 배치 사양을 사용하여 호스트에 모니터를 배포합니다.



참고

특정 호스트에 배포할 **--placement** 옵션을 사용하는 것이 좋습니다.

구문

```
ceph orch apply mon --placement="HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

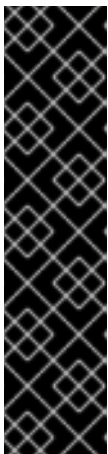
예제

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



참고

명령의 첫 번째 노드로 부트 스트랩 노드를 포함시켜야 합니다.



중요

ceph orch가 **mon supersedes**를 적용할 때 모니터를 개별적으로 추가하지 마십시오. 모든 호스트에 모니터를 추가하지 않습니다. 예를 들어 다음 명령을 실행하는 경우 첫 번째 명령은 **host01**에 모니터를 생성합니다. 그런 다음 두 번째 명령은 **host01**의 모니터를 대체하고 **host02**에서 모니터를 생성합니다. 그런 다음 세 번째 명령은 **host02**에서 모니터를 대체하고 **host03**에 모니터를 생성합니다. 결국 세 번째 호스트에서만 모니터가 있습니다.

```
# ceph orch apply mon host01
# ceph orch apply mon host02
# ceph orch apply mon host03
```

방법 2

- 배치 사양을 사용하여 라벨이 있는 특정 호스트에 특정 수의 모니터를 배포합니다.
 - 호스트에 라벨을 추가합니다.

구문

```
ceph orch host label add HOSTNAME_1 LABEL
```

예제

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

- 데몬을 배포합니다.

구문

```
ceph orch apply mon --placement="HOST_NAME_1:mon HOST_NAME_2:mon
HOST_NAME_3:mon"
```

예제

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

방법 3

- 배치 사양을 사용하여 특정 호스트에 특정 수의 모니터를 배포합니다.

구문

```
ceph orch apply mon --placement="NUMBER_OF_DAEMONS HOST_NAME_1
HOST_NAME_2 HOST_NAME_3"
```

예제

```
[ceph: root@host01 /]# ceph orch apply mon --placement="3 host01 host02 host03"
```

방법 4

- 스토리지 클러스터의 호스트에 무작위로 데몬을 배포합니다.

구문

```
ceph orch apply mon NUMBER_OF_DAEMONS
```

예제

```
[ceph: root@host01 /]# ceph orch apply mon 3
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

4.4. 서비스 사양을 사용하여 CEPH 모니터 데몬 배포

Ceph Orchestrator는 기본적으로 하나의 모니터 데몬을 배포합니다. YAML 형식 파일과 같이 서비스 사양을 사용하여 추가 모니터 데몬을 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.

절차

1. **mon.yaml** 파일을 생성합니다.

예제

```
[root@host01 ~]# touch mon.yaml
```

2. 다음 세부 사항을 포함하도록 **mon.yaml** 파일을 편집합니다.

구문

```
service_type: mon
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
```

예제

```
service_type: mon
placement:
  hosts:
    - host01
    - host02
```

3. 컨테이너의 디렉터리에 YAML 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount mon.yaml:/var/lib/ceph/mon/mon.yaml
```

4. 디렉터리로 이동합니다.

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/mon/
```

5. 모니터 데몬을 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yaml
```

예제

```
[ceph: root@host01 mon]# ceph orch apply -i mon.yaml
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

4.5. CEPH ORCHESTRATOR를 사용하여 특정 네트워크에 모니터 데몬 배포

Ceph Orchestrator는 기본적으로 하나의 모니터 데몬을 배포합니다. 각 모니터의 IP 주소 또는 CIDR 네트워크를 명시적으로 지정하고 각 모니터가 배치되는 위치를 제어할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 자동 모니터링 배포를 비활성화합니다.

예제

```
[ceph: root@host01 /]# ceph orch apply mon --unmanaged
```

3. 특정 네트워크의 호스트에 모니터를 배포합니다.

구문

```
ceph orch daemon add mon HOST_NAME_1:IP_OR_NETWORK
```

예제

```
[ceph: root@host01 /]# ceph orch daemon add mon host03:10.1.2.123
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

4.6. CEPHORCHESTRATOR를 사용하여 MONITOR 데몬 제거

호스트에서 모니터 데몬을 제거하려면 다른 호스트에 monitor 데몬을 재배포하면 됩니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 호스트에 배포된 하나 이상의 모니터 데몬.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. **ceph orch apply** 명령을 실행하여 필요한 모니터 데몬을 배포합니다.

구문

```
ceph orch apply mon "NUMBER_OF_DAEMONS HOST_NAME_1 HOST_NAME_3"
```

host02 에서 모니터 데몬을 제거하려면 다른 호스트에서 모니터를 재배포할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph orch apply mon "2 host01 host03"
```

검증

- 호스트, daemon 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage Operations 가이드의 명령행 인터페이스를 사용하여 Ceph 모니터 데몬 배포를 참조하십시오.](#)
- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 의 서비스 사양을 사용하여 Ceph 모니터 데몬 배포를 참조하십시오.

4.7. 비정상 스토리지 클러스터에서 CEPH MONITOR 제거

비정상 스토리지 클러스터에서 **ceph-mon** 데몬을 제거할 수 있습니다. 비정상적인 스토리지 클러스터는 **활성 상태 + 정리 상태**가 아닌 배치 그룹이 영구적으로 있는 클러스터입니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- Ceph Monitor 노드에 대한 루트 수준 액세스입니다.
- Ceph 모니터 노드가 하나 이상 실행되고 있습니다.

절차

1. 생존 모니터를 확인하고 호스트에 로그인합니다.

구문

```
ssh root@MONITOR_ID
```

예제

```
[root@admin ~]# ssh root@host00
```

2. 각 Ceph 모니터 호스트에 로그인하고 모든 Ceph 모니터를 중지합니다.

구문

```
cephadm unit --name DAEMON_NAME.HOSTNAME stop
```

예제

```
[root@host00 ~]# cephadm unit --name mon.host00 stop
```

3. 확장된 데몬 유지 관리에 적합한 환경을 설정하고 데몬을 대화식으로 실행합니다.

구문

```
cephadm shell --name DAEMON_NAME.HOSTNAME
```

예제

```
[root@host00 ~]# cephadm shell --name mon.host00
```

4. **monmap** 파일의 사본을 추출합니다.

구문

```
ceph-mon -i HOSTNAME --extract-monmap TEMP_PATH
```

예제

```
[ceph: root@host00 /]# ceph-mon -i host01 --extract-monmap /tmp/monmap
2022-01-05T11:13:24.440+0000 7f7603bd1700 -1 wrote monmap to /tmp/monmap
```

5. 비관리 Ceph 모니터를 제거합니다.

구문

```
monmaptool TEMPORARY_PATH --rm HOSTNAME
```

예제

```
[ceph: root@host00 /]# monmaptool /tmp/monmap --rm host01
```

6. 제거된 모니터와 함께 남아 있는 모니터를 Ceph 모니터에 삽입합니다.

구문

```
ceph-mon -i HOSTNAME --inject-monmap TEMP_PATH
```

예제

```
[ceph: root@host00 /]# ceph-mon -i host00 --inject-monmap /tmp/monmap
```

7. 생존 모니터만 시작합니다.

구문

```
cephadm unit --name DAEMON_NAME.HOSTNAME start
```

예제

```
[root@host00 ~]# cephadm unit --name mon.host00 start
```

8. 모니터가 퀵럼이 형성되는지 확인합니다.

예제

```
[ceph: root@host00 /]# ceph -s
```

9. 선택 사항: 삭제된 Ceph Monitor의 데이터 디렉터리를 **/var/lib/ceph/CLUSTER_FSID/mon.HOSTNAME** 디렉터리에 보관합니다.

5장. CEPH INTENDEDCESTRATOR를 사용한 관리자 관리

스토리지 관리자는 Ceph Orchestrator를 사용하여 추가 관리자 데몬을 배포할 수 있습니다. Cephadm은 부트스트랩 프로세스 중에 부트스트랩 노드에 관리자 데몬을 자동으로 설치합니다.

이 섹션에서는 다음 관리 작업에 대해 설명합니다.

- Ceph Orchestrator를 사용하여 manager 데몬 배포.
- Ceph Orchestrator를 사용하여 manager 데몬 제거.

5.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.

5.2. CEPHORCHESTRATOR를 사용하여 MANAGER 데몬 배포

Ceph Orchestrator는 기본적으로 두 개의 Manager 데몬을 배포합니다. 명령줄 인터페이스에서 **배치** 사양을 사용하여 추가 관리자 데몬을 배포할 수 있습니다. 다른 수의 관리자 데몬을 배포하려면 다른 번호를 지정합니다. Manager 데몬을 배포해야 하는 호스트를 지정하지 않으면 Ceph Orchestrator가 호스트를 무작위로 선택하고 Manager 데몬을 배포합니다.



참고

배포마다 배포에 최소 3개의 Ceph Manager가 있는지 확인합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. Manager 데몬을 두 가지 방법으로 배포할 수 있습니다.

방법 1

- 특정 호스트 세트에 배치 사양을 사용하여 관리자 데몬을 배포합니다.



참고

특정 호스트에 배포할 **--placement** 옵션을 사용하는 것이 좋습니다.

구문

```
ceph orch apply mgr --placement="HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

예제

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

방법 2

- 스토리지 클러스터의 호스트에 관리자 데몬을 무작위로 배포합니다.

구문

```
ceph orch apply mgr NUMBER_OF_DAEMONS
```

예제

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

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mgr
```

5.3. CEPHORCHESTRATOR를 사용하여 MANAGER 데몬 제거

호스트에서 manager 데몬을 제거하려면 다른 호스트에 데몬을 재배포하면 됩니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.
- 호스트에 배포된 관리자 데몬 1개 이상

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. **ceph orch apply** 명령을 실행하여 필요한 manager 데몬을 재배포합니다.

구문

```
ceph orch apply mgr "NUMBER_OF_DAEMONS HOST_NAME_1 HOST_NAME_3"
```

host02 에서 관리자 데몬을 제거하려면 다른 호스트에서 manager 데몬을 재배포할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph orch apply mgr "2 host01 host03"
```

검증

- 호스트, daemon 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mgr
```

추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage Operations 가이드](#) 의 Ceph Orchestrator를 사용하여 manager 데몬 배포를 참조하십시오.

5.4. CEPH MANAGER 밸런서 모듈 사용

이 밸런서는 자동으로 또는 감독된 방식으로 분산 배포를 수행하기 위해 OSD 간에 배치 그룹(PG)의 배치를 최적화하는 Ceph Manager(**ceph-mgr**)용 모듈입니다.

현재는 balancer 모듈을 비활성화할 수 없습니다. 구성을 사용자 지정하기만 해제할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.

절차

1. balancer 모듈이 활성화되었는지 확인합니다.

예제

```
[ceph: root@host01 /]# ceph mgr module enable balancer
```

2. balancer 모듈을 켭니다.

예제

```
[ceph: root@host01 /]# ceph balancer on
```

3. 기본 모드는 **crush-compat** 입니다. 모드는 다음을 사용하여 변경할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph balancer mode upmap
```

또는

예제

```
[ceph: root@host01 /]# ceph balancer mode crush-compat
```

상태

pod의 현재 상태는 다음을 사용하여 언제든지 확인할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph balancer status
```

자동 분산

기본적으로 balancer 모듈을 활성화하면 자동 분산이 사용됩니다.

예제

```
[ceph: root@host01 /]# ceph balancer on
```

다음을 사용하여 밸런서를 다시 끌 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph balancer off
```

이 명령은 이전 클라이언트와 역호환되는 **crush-compat** 모드를 사용하며 시간이 지남에 따라 데이터 배포를 약간 변경하여 OSD를 동일하게 활용합니다.

제한

클러스터가 성능이 저하되고 OSD가 실패하여 시스템이 아직 복구되지 않은 경우와 같이 cluster가 저하된 경우는 Placement 배포에서 조정되지 않습니다.

클러스터가 정상이면 밸런서에서 잘못된 배치 또는 이동해야 하는 PC의 백분율이 기본적으로 5% 미만인 만큼 해당 변경 사항을 제한합니다. 이 백분율은 **target_max_misplaced_ratio** 설정을 사용하여 조정할 수 있습니다. 예를 들어 임계값을 7%로 늘리려면 다음을 수행합니다.

예제

```
[ceph: root@host01 /]# ceph config-key set mgr target_max_misplaced_ratio .07
```

감독된 최적화

밸런서 작업은 다음과 같은 몇 가지 단계로 나뉩니다.

1. 계획 수립.
2. 데이터 배포의 품질(현재 DASD 배포 또는 계획 실행 후 발생 시 발생)에 대한 데이터 배포 품질을 평가합니다.
3. 계획 실행.
 - 현재 배포를 평가하고 점수를 매기려면 다음을 수행합니다.

예제

```
[ceph: root@host01 /]# ceph balancer eval
```

- 단일 풀의 배포를 평가하려면 다음을 수행합니다.

구문

```
ceph balancer eval POOL_NAME
```

예제

```
[ceph: root@host01 /]# ceph balancer eval rbd
```

- 평가에 대한 세부 정보를 보려면 다음을 수행합니다.

예제

```
[ceph: root@host01 /]# ceph balancer eval-verbose ...
```

- 현재 구성된 모드를 사용하여 계획을 생성하려면 다음을 수행합니다.

구문

```
ceph balancer optimize PLAN_NAME
```

PLAN_NAME 을 사용자 지정 계획 이름으로 교체합니다.

예제

```
[ceph: root@host01 /]# ceph balancer optimize rbd_123
```

- 계획 내용을 보려면 다음을 수행합니다.

구문

```
ceph balancer show PLAN_NAME
```

예제

```
[ceph: root@host01 /]# ceph balancer show rbd_123
```

- 이전 계획을 삭제하려면 다음을 수행합니다.

구문

```
ceph balancer rm PLAN_NAME
```

예제

```
[ceph: root@host01 /]# ceph balancer rm rbd_123
```

- 현재 기록된 계획을 보려면 상태 명령을 사용합니다.

```
[ceph: root@host01 /]# ceph balancer status
```

- 계획을 실행한 후 발생하는 배포 품질을 계산하려면 다음을 수행합니다.

구문

```
ceph balancer eval PLAN_NAME
```

예제

```
[ceph: root@host01 /]# ceph balancer eval rbd_123
```

- 계획을 실행하려면 다음을 수행합니다.

구문

```
ceph balancer execute PLAN_NAME
```

예제

```
[ceph: root@host01 /]# ceph balancer execute rbd_123
```



참고

배포를 개선할 것으로 예상되는 경우에만 계획을 실행합니다. 실행 후 계획이 취소됩니다.

5.5. CEPH MANAGER ALERTS 모듈 사용

Ceph Manager alerts 모듈을 사용하여 Red Hat Ceph Storage 클러스터 상태에 대한 간단한 경고 메시지를 이메일로 보낼 수 있습니다.



참고

이 모듈은 강력한 모니터링 솔루션이 될 수 없습니다. Ceph 클러스터 자체의 일부로 실행된다는 사실은 **ceph-mgr** 데몬 실패 시 경고가 전송되지 않도록 기본적으로 제한됩니다. 그러나 이 모듈은 기존 모니터링 인프라가 없는 환경에 존재하는 독립 실행형 클러스터에 유용할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- Ceph Monitor 노드에 대한 루트 수준 액세스입니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. alerts 모듈을 활성화합니다.

예제

```
[ceph: root@host01 /]# ceph mgr module enable alerts
```

3. alerts 모듈이 활성화되었는지 확인합니다.

예제

```
[ceph: root@host01 /]# ceph mgr module ls | more
{
  "always_on_modules": [
    "balancer",
    "crash",
    "devicehealth",
    "orchestrator",
    "pg_autoscaler",
    "progress",
    "rbd_support",
    "status",
    "telemetry",
    "volumes"
  ],
  "enabled_modules": [
    "alerts",
    "cephadm",
    "dashboard",
    "iostat",
    "nfs",
```

```
"prometheus",
"restful"
]
```

4. SMTP(Simple Mail Transfer Protocol)를 구성합니다.

구문

```
ceph config set mgr mgr/alerts/smtp_host SMTP_SERVER
ceph config set mgr mgr/alerts/smtp_destination RECEIVER_EMAIL_ADDRESS
ceph config set mgr mgr/alerts/smtp_sender SENDER_EMAIL_ADDRESS
```

예제

```
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_host smtp.example.com
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_destination
example@example.com
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_sender
example2@example.com
```

5. 선택 사항: 기본적으로 alerts 모듈은 SSL 및 포트 465를 사용합니다. 이를 변경하려면 **smtp_ssl**을 **false**로 설정합니다.

구문

```
ceph config set mgr mgr/alerts/smtp_ssl false
ceph config set mgr mgr/alerts/smtp_port PORT_NUMBER
```

예제

```
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_ssl false
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_port 587
```

6. SMTP 서버에 인증합니다.

구문

```
ceph config set mgr mgr/alerts/smtp_user USERNAME
ceph config set mgr mgr/alerts/smtp_password PASSWORD
```

예제

```
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_user admin1234
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_password admin1234
```

7. 선택 사항: 기본적으로 SMTP **From** name은 **Ceph**입니다. 이를 변경하려면 **smtp_from_name** 매개 변수를 설정합니다.

구문

```
ceph config set mgr mgr/alerts/smtp_from_name CLUSTER_NAME
```

예제

```
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_from_name 'Ceph Cluster Test'
```

8. 선택 사항: 기본적으로 alerts 모듈은 스토리지 클러스터의 상태를 1분마다 확인하고 클러스터 상태가 변경될 때 메시지를 보냅니다. 빈도를 변경하려면 **interval** 매개변수를 설정합니다.

구문

```
ceph config set mgr mgr/alerts/interval INTERVAL
```

예제

```
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/interval "5m"
```

이 예에서는 간격이 5분으로 설정됩니다.

9. 선택 사항: 즉시 경고를 보냅니다.

예제

```
[ceph: root@host01 /]# ceph alerts send
```

추가 리소스

- Ceph 상태 메시지에 대한 자세한 내용은 [Red Hat Ceph Storage Troubleshooting Guide](#) 의 Ceph 클러스터 상태 메시지를 참조하십시오.

5.6. CEPH 관리자 크래시 모듈 사용

Ceph 관리자 크래시 모듈을 사용하면 데몬 크래시 덤프에 대한 정보를 수집하고 추가 분석을 위해 Red Hat Ceph Storage 클러스터에 저장할 수 있습니다.

기본적으로 데몬 크래시 덤프는 **/var/lib/ceph/crash** 에 덤프됩니다. 옵션 **크래시 dir** 을 사용하여 구성할 수 있습니다. 크래시 디렉터리의 이름은 시간, 날짜 및 임의로 생성된 UUID로 지정되며, 동일한 **crash_id** 와 함께 메타데이터 파일 **메타** 및 최근 로그 파일을 포함합니다.

ceph-crash.service 를 사용하여 이러한 충돌을 자동으로 제출하고 Ceph 모니터에서 유지할 수 있습니다. **ceph-crash.service** 는 crashdump 디렉터리를 감시하고 **ceph** 크래시 포스트 로 업로드합니다.

RECENT_CRASH heath 메시지는 Ceph 클러스터에서 가장 일반적인 상태 메시지 중 하나입니다. 이 상태 메시지는 하나 이상의 Ceph 데몬이 최근에 충돌했으며, 충돌이 아직 보관 또는 승인되지 않았음을 의미합니다. 이는 소프트웨어 버그, 오류가 발생한 디스크 등의 하드웨어 문제 또는 기타 일부 문제를 나타낼 수 있습니다. 옵션은 **mgr/crash/warn_recent_interval** 이 최근에 의미하는 시간을 제어합니다. 기본값은 2 주입니다. 다음 명령을 실행하여 경고를 비활성화할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph config set mgr/crash/warn_recent_interval 0
```

mgr/crash/retain_interval 옵션은 충돌 보고서를 자동으로 제거하기 전에 유지하려는 기간을 제어합니다. 이 옵션의 기본값은 1년입니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.

절차

1. crash 모듈이 활성화되었는지 확인합니다.

예제

```
[ceph: root@host01 /]# ceph mgr module ls | more
{
  "always_on_modules": [
    "balancer",
    "crash",
    "devicehealth",
    "orchestrator_cli",
    "progress",
    "rbd_support",
    "status",
    "volumes"
  ],
  "enabled_modules": [
    "dashboard",
    "pg_autoscaler",
    "prometheus"
  ]
}
```

2. 크래시 덤프를 저장합니다. 메타데이터 파일은 크래시 dir에 **메타** 형식으로 저장된 JSON Blob입니다. stdin에서 읽는 ceph 명령 **-i** 옵션을 호출할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph crash post -i meta
```

3. 새로운 및 아카이브의 모든 크래시 정보에 대한 타임스탬프 또는 UUID 충돌 ID를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph crash ls
```

4. 모든 새로운 충돌 정보에 대한 타임스탬프 또는 UUID 충돌 ID를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph crash ls-new
```

5. 모든 새로운 충돌 정보에 대한 타임스탬프 또는 UUID 충돌 ID를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph crash ls-new
```

6. 기간별로 그룹화된 저장된 충돌 정보 요약을 나열합니다.

예제

```
[ceph: root@host01 /]# ceph crash stat
8 crashes recorded
8 older than 1 days old:
2022-05-20T08:30:14.533316Z_4ea88673-8db6-4959-a8c6-0eea22d305c2
2022-05-20T08:30:14.590789Z_30a8bb92-2147-4e0f-a58b-a12c2c73d4f5
2022-05-20T08:34:42.278648Z_6a91a778-bce6-4ef3-a3fb-84c4276c8297
2022-05-20T08:34:42.801268Z_e5f25c74-c381-46b1-bee3-63d891f9fc2d
2022-05-20T08:34:42.803141Z_96adfc59-be3a-4a38-9981-e71ad3d55e47
2022-05-20T08:34:42.830416Z_e45ed474-550c-44b3-b9bb-283e3f4cc1fe
2022-05-24T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d
2022-05-24T19:58:44.315282Z_1847afbc-f8a9-45da-94e8-5aef0738954e
```

7. 저장된 충돌의 세부 정보를 확인합니다.

구문

```
ceph crash info CRASH_ID
```

예제

```
[ceph: root@host01 /]# ceph crash info 2022-05-24T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d
{
  "assert_condition": "session_map.sessions.empty()",
  "assert_file": "/builddir/build/BUILD/ceph-16.1.0-486-g324d7073/src/mon/Monitor.cc",
  "assert_func": "virtual Monitor::~~Monitor()",
  "assert_line": 287,
  "assert_msg": "/builddir/build/BUILD/ceph-16.1.0-486-g324d7073/src/mon/Monitor.cc: In function 'virtual Monitor::~~Monitor()' thread 7f67a1aeb700 time 2022-05-24T19:58:42.545485+0000\n/builddir/build/BUILD/ceph-16.1.0-486-g324d7073/src/mon/Monitor.cc: 287: FAILED\nceph_assert(session_map.sessions.empty())\n",
  "assert_thread_name": "ceph-mon",
  "backtrace": [
    "/lib64/libpthread.so.0(+0x12b30) [0x7f679678bb30]",
    "gsignal()",
    "abort()",
    "(ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x1a9) [0x7f6798c8d37b]",
    "/usr/lib64/ceph/libceph-common.so.2(+0x276544) [0x7f6798c8d544]",
    "(Monitor::~~Monitor()+0xe30) [0x561152ed3c80]",
    "(Monitor::~~Monitor()+0xd) [0x561152ed3cdd]",
    "main()",
    "__libc_start_main()",
    "_start()"
  ],
  "ceph_version": "16.2.8-65.el8cp",
  "crash_id": "2022-07-06T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d",
  "entity_name": "mon.ceph-adm4",
  "os_id": "rhel",
  "os_name": "Red Hat Enterprise Linux",
```

```

"os_version": "8.5 (Ootpa)",
"os_version_id": "8.5",
"process_name": "ceph-mon",
"stack_sig":
"957c21d558d0cba4cee9e8aaf9227b3b1b09738b8a4d2c9f4dc26d9233b0d511",
"timestamp": "2022-07-06T19:58:42.549073Z",
"utname_hostname": "host02",
"utname_machine": "x86_64",
"utname_release": "4.18.0-240.15.1.el8_3.x86_64",
"utname_sysname": "Linux",
"utname_version": "#1 SMP Wed Jul 06 03:12:15 EDT 2022"
}

```

8. KEEP 일보다 오래된 저장된 크래시 제거: 여기서 KEEP 는 정수여야 합니다.

구문

```
ceph crash prune KEEP
```

예제

```
[ceph: root@host01 /]# ceph crash prune 60
```

9. 충돌 보고서를 보관하여 더 이상 **RECENT_CRASH** 상태 점검으로 간주되지 않고 충돌 **ls-new** 출력에 표시되지 않도록 합니다. 이는 충돌 **ls** 에 나타납니다.

구문

```
ceph crash archive CRASH_ID
```

예제

```
[ceph: root@host01 /]# ceph crash archive 2022-05-24T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d
```

10. 모든 크래시 보고서를 보관합니다.

예제

```
[ceph: root@host01 /]# ceph crash archive-all
```

11. 크래시 덤프 제거:

구문

```
ceph crash rm CRASH_ID
```

예제

```
[ceph: root@host01 /]# ceph crash rm 2022-05-24T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d
```

추가 리소스

- [Ceph 상태 메시지에 대한 자세한 내용은 Red Hat Ceph Storage Troubleshooting Guide 의 Ceph 클러스터 상태 메시지를 참조하십시오.](#)

6 장. CEPH ORCHESTRATOR를 사용한 OSD 관리

스토리지 관리자는 Ceph Orchestrators를 사용하여 Red Hat Ceph Storage 클러스터의 OSD를 관리할 수 있습니다.

6.1. CEPH OSD

Red Hat Ceph Storage 클러스터가 가동되어 실행되면 런타임 시 스토리지 클러스터에 OSD를 추가할 수 있습니다.

Ceph OSD는 일반적으로 스토리지 드라이브 1대와 노드 내에서 연결된 저널을 위한 하나의 **ceph-osd** 데몬으로 구성됩니다. 노드에 여러 스토리지 드라이브가 있는 경우 각 드라이브에 대해 하나의 **ceph-osd** 데몬을 매핑합니다.

스토리지 용량의 상위에 도달하고 있는지 확인하기 위해 클러스터의 용량을 정기적으로 확인하는 것이 좋습니다. 스토리지 클러스터가 **전체 비율**에 도달하므로 하나 이상의 OSD를 추가하여 스토리지 클러스터의 용량을 확장합니다.

Red Hat Ceph Storage 클러스터의 크기를 줄이거나 하드웨어를 교체하려는 경우 런타임 시 OSD를 제거할 수도 있습니다. 노드에 스토리지 드라이브가 여러 개 있는 경우 해당 드라이브의 **ceph-osd** 데몬 중 하나를 제거해야 할 수도 있습니다. 일반적으로 스토리지 클러스터의 용량을 확인하여 용량의 상단에 도달했는지 확인하는 것이 좋습니다. 스토리지 클러스터가 **거의 전체 비율**에 있지 않은 OSD를 제거해야 합니다.



중요

OSD를 추가하기 전에 스토리지 클러스터가 **전체 비율**에 도달할 수 없도록 하십시오. 스토리지 클러스터가 **거의 전체 비율**에 도달한 후 발생하는 **OSD 오류**는 스토리지 클러스터가 **전체 비율**을 초과할 수 있습니다. Ceph는 스토리지 용량 문제를 해결할 때까지 데이터를 보호하기 위한 쓰기 액세스를 차단합니다. 먼저 **전체 비율**에 미치는 영향을 고려하지 않고 OSD를 제거하지 마십시오.

6.2. CEPH OSD 노드 구성

OSD를 사용할 풀의 스토리지 전략으로 Ceph OSD와 지원 하드웨어를 유사하게 구성합니다. Ceph에서는 일관된 성능 프로필을 위해 풀 간에 균일한 하드웨어를 선호합니다. 최상의 성능을 위해 동일한 유형 또는 크기의 드라이브가 있는 CRUSH 계층 구조를 고려하십시오.

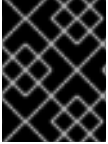
상당한 크기의 드라이브를 추가하는 경우 그에 따라 가중치를 조정합니다. OSD를 CRUSH 맵에 추가할 때 새 OSD의 가중치를 고려합니다. 하드 드라이브 용량은 연간 약 40% 증가하므로 최신 OSD 노드는 스토리지 클러스터의 이전 노드보다 더 큰 하드 드라이브를 사용할 수 있습니다. 즉, 가중치가 더 클 수 있습니다.

새 설치를 수행하기 전에 설치 [가이드](#)의 Red Hat Ceph Storage 설치 요구 사항 장을 검토하십시오.

6.3. OSD 메모리 자동 튜닝

OSD 데몬은 **osd_memory_target** 구성 옵션을 기반으로 메모리 사용량을 조정합니다. **osd_memory_target** 옵션은 시스템에서 사용 가능한 RAM에 따라 OSD 메모리를 설정합니다.

Red Hat Ceph Storage가 다른 서비스와 메모리를 공유하지 않는 전용 노드에 배포된 경우 **cephadm**은 총 RAM 용량 및 배포된 OSD 수에 따라 자동으로 OSD 소비를 조정합니다.



중요

기본적으로 Red Hat Ceph Storage 5.1에서 **osd_memory_target_autotune** 매개 변수가 **true** 로 설정됩니다.

구문

```
ceph config set osd osd_memory_target_autotune true
```

스토리지 클러스터가 Red Hat Ceph Storage 5.0으로 업그레이드되면 OSD 또는 OSD 교체와 같은 클러스터 유지 관리를 위해 **osd_memory_target_autotune** 매개 변수를 **true** 로 설정하여 시스템 메모리별로 osd 메모리를 자동 튜닝하는 것이 좋습니다.

Cephadm은 일부 **mgr/cephadm/autotune_memory_target_ratio** 로 시작합니다. 기본값은 시스템에서 총 RAM의 **0.7** 으로 설정하고, non-autotuned 데몬에서 사용하는 메모리를 제거하고, **osd_memory_target_autotune** 이 false 인 OSD에 대해 사용합니다.

osd_memory_target 매개 변수는 다음과 같이 계산됩니다.

구문

$$\text{osd_memory_target} = \text{TOTAL_RAM_OF_THE_OSD} * (1048576) * (\text{autotune_memory_target_ratio}) / \text{NUMBER_OF_OSDS_IN_THE_OSD_NODE} - (\text{SPACE_ALLOCATED_FOR_OTHER_DAEMONS})$$

SPACE_ALLOCATED_FOR_OTHER_DAEMONS 는 선택적으로 다음과 같은 데몬 공간 할당을 포함할 수 있습니다.

- Alertmanager: 1GB
- Grafana: 1GB
- Ceph Manager: 4GB
- Ceph 모니터: 2GB
- node-exporter: 1GB
- Prometheus: 1GB

예를 들어 노드에 OSD가 24개 있고 251GB RAM 공간이 있는 경우 **osd_memory_target** 은 **7860684936** 입니다.

최종 대상은 옵션이 있는 구성 데이터베이스에 반영됩니다. **MEM LIMIT** 열의 **ceph orch ps** 출력에서 각 데몬에서 사용하는 제한 및 현재 메모리를 볼 수 있습니다.

참고

Red Hat Ceph Storage 5.1에서 **osd_memory_target_autotune true**의 기본 설정은 컴퓨팅 및 Ceph 스토리지 서비스가 함께 배치되는 하이퍼컨버지드 인프라에 적합하지 않습니다. 하이퍼컨버지드 인프라에서 **autotune_memory_target_ratio**를 **0.2**로 설정하여 Ceph의 메모리 사용을 줄일 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph config set mgr
mgr/cephadm/autotune_memory_target_ratio 0.2
```

스토리지 클러스터에서 OSD의 특정 메모리 대상을 수동으로 설정할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph config set osd.123 osd_memory_target 7860684936
```

스토리지 클러스터에서 OSD 호스트에 대한 특정 메모리 대상을 수동으로 설정할 수 있습니다.

구문

```
ceph config set osd/host:HOSTNAME osd_memory_target TARGET_BYTES
```

예제

```
[ceph: root@host01 /]# ceph config set osd/host:host01 osd_memory_target 1000000000
```

참고

osd_memory_target_autotune을 활성화하면 기존 수동 OSD 메모리 대상 설정을 덮어 씁니다. **osd_memory_target_autotune** 옵션 또는 기타 유사한 옵션이 활성화된 경우에도 데몬 메모리가 튜닝되지 않도록 하려면 호스트에서 **_no_autotune_memory** 레이블을 설정합니다.

구문

```
ceph orch host label add HOSTNAME _no_autotune_memory
```

autotune 옵션을 비활성화하고 특정 메모리 대상을 설정하여 메모리 자동 튜닝에서 OSD를 제외할 수 있습니다.

예제

```
[ceph: root@host01 /]# ceph config set osd.123 osd_memory_target_autotune false
[ceph: root@host01 /]# ceph config set osd.123 osd_memory_target 16G
```

6.4. CEPH OSD 배포를 위한 장치 나열

Ceph intended orchestrator를 사용하여 OSD를 배포하기 전에 사용 가능한 장치 목록을 확인할 수 있습니다. 명령은 Cephadm에서 검색할 수 있는 장치 목록을 출력하는 데 사용됩니다. 다음 조건이 모두 충족되면 스토리지 장치를 사용할 수 있는 것으로 간주됩니다.

- 장치에 파티션이 없어야 합니다.
- 장치에 LVM 상태가 없어야 합니다.
- 장치를 마운트할 수 없습니다.
- 장치에 파일 시스템을 포함해서는 안 됩니다.
- 장치에는 Ceph BlueStore OSD가 포함되어 있지 않아야 합니다.
- 장치는 5GB보다 커야 합니다.



참고

Ceph는 사용할 수 없는 장치에서 OSD를 프로비저닝하지 않습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자 및 모니터 데몬이 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. OSD를 배포할 사용 가능한 장치를 나열합니다.

구문

```
ceph orch device ls [--hostname=HOSTNAME_1 HOSTNAME_2] [--wide] [--refresh]
```

예제

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

--wide 옵션을 사용하면 장치가 OSD로 사용할 수 없는 이유를 포함하여 장치와 관련된 모든 세부 정보를 제공합니다. 이 옵션은 NVMe 장치를 지원하지 않습니다.

3. 선택 사항: **ceph orch 장치 ls**의 출력에서 상태, Ident, Fault 필드를 활성화하려면 다음 명령을 실행합니다.



참고

이 필드는 **libstoragetekton 라이브러리**에서 지원되며 현재 SCSI, SAS 및 SATA 장치를 지원합니다.

- Cephadm 셸 외부의 root 사용자로, 하드웨어의 **libstorage tekton** 라이브러리의 호환성을 확인하여 예기치 않은 서비스 중단을 방지합니다.

예제

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

출력에서 각 SCSI VPD 0x83 ID 가 있는 Health Status (상태)가 꺼짐으로 표시됩니다.



참고

이 정보를 얻지 못하는 경우 필드를 활성화하면 장치의 심각한 동작이 발생할 수 있습니다.

- Cephadm 셸에 다시 로그인하고 **libstoragetekton** 지원을 활성화합니다.

예제

```
[root@host01 ~]# cephadm shell
[ceph: root@host01 /]# ceph config set mgr mgr/cephadm/device_enhanced_scan true
```

이 기능이 활성화되면 **ceph orch device ls** 가 Health 필드의 출력을 good 으로 제공합니다.

검증

- 장치를 나열합니다.

예제

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

6.5. CEPH OSD 배포를 위한 ZAPPING 장치

OSD를 배포하기 전에 사용 가능한 장치 목록을 확인해야 합니다. 장치에 사용 가능한 공간이 없는 경우 장치에서 데이터를 zapping하여 지울 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자 및 모니터 데몬이 배포됩니다.

절차

- Cephadm 셸에 로그인합니다.

예제

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

2. OSD를 배포할 사용 가능한 장치를 나열합니다.

구문

```
ceph orch device ls [--hostname=HOSTNAME_1 HOSTNAME_2] [--wide] [--refresh]
```

예제

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

3. 장치의 데이터를 지웁니다.

구문

```
ceph orch device zap HOSTNAME FILE_PATH --force
```

예제

```
[ceph: root@host01 /]# ceph orch device zap host02 /dev/sdb --force
```

검증

- 장치에서 공간을 사용할 수 있는지 확인합니다.

예제

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

Available (사용 가능)의 필드가 Yes (예)임을 확인할 수 있습니다.

추가 리소스

- 자세한 내용은 Red Hat [Ceph Storage Operations Guide](#)의 [Ceph OSD 배포 목록](#) 섹션을 참조하십시오.

6.6. 사용 가능한 모든 장치에 CEPH OSD 배포

사용 가능한 모든 장치에 모든 OSDs를 배포할 수 있습니다. Cephadm을 사용하면 Ceph Orchestrator가 사용 가능하고 사용되지 않는 스토리지 장치에서 OSD를 검색하고 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자 및 모니터 데몬이 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. OSD를 배포할 사용 가능한 장치를 나열합니다.

구문

```
ceph orch device ls [--hostname=HOSTNAME_1 HOSTNAME_2] [--wide] [--refresh]
```

예제

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

3. 사용 가능한 모든 장치에 OSD를 배포합니다.

예제

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

ceph orch apply의 영향은 영구적이므로 Orchestrator가 장치를 자동으로 찾고 클러스터에 추가한 다음 새 OSD를 생성합니다. 이는 다음과 같은 조건에서 수행됩니다.

- 새 디스크 또는 드라이브가 시스템에 추가됩니다.
 - 기존 디스크 또는 드라이브는 zapped입니다.
 - OSD가 제거되고 장치가 zapped됩니다.
- unmanaged** 매개변수를 사용하여 사용 가능한 모든 장치에서 OSD의 자동 생성을 비활성화할 수 있습니다.

예제

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

매개 변수 **--unmanaged**를 **true**로 설정하면 OSD 생성이 비활성화되고 새 OSD 서비스를 적용하면 변경 사항이 없습니다.



참고

ceph orch 데몬 **add** 명령은 새 OSD를 생성하지만 OSD 서비스는 추가하지 않습니다.

검증

- 서비스를 나열합니다.

예제

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

- 노드 및 장치의 세부 정보를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

추가 리소스

- Red Hat [Ceph Storage Operations Guide](#) 의 [Ceph OSD 배포 목록](#) 섹션을 참조하십시오.

6.7. 특정 장치 및 호스트에 CEPH OSD 배포

Ceph Orchestrator를 사용하여 특정 장치 및 호스트에 모든 Ceph OSD를 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자 및 모니터 데몬이 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. OSD를 배포할 사용 가능한 장치를 나열합니다.

구문

```
ceph orch device ls [--hostname=HOSTNAME_1 HOSTNAME_2] [--wide] [--refresh]
```

예제

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

3. 특정 장치 및 호스트에 OSD를 배포합니다.

구문

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

예제

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

LVM 계층이 없는 원시 물리적 장치에 ODS를 배포하려면 **--method raw** 옵션을 사용합니다.

구문

```
ceph orch daemon add osd --method raw HOSTNAME:DEVICE_PATH
```

예제

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



참고

별도의 DB 또는 WAL 장치가 있는 경우 DB 또는 WAL 장치에 대한 블록 비율은 **1:1** 이어야 합니다.

검증

- 서비스를 나열합니다.

예제

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

- 노드 및 장치의 세부 정보를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

```
ceph orch ps --service_name=SERVICE_NAME
```

예제

```
[ceph: root@host01 /]# ceph orch ps --service_name=osd
```

추가 리소스

- Red Hat [Ceph Storage Operations Guide](#) 의 [Ceph OSD 배포 목록](#) 섹션을 참조하십시오.

6.8. OSD 배포를 위한 고급 서비스 사양 및 필터

유형의 서비스 사양 OSD는 디스크의 속성을 사용하여 클러스터 레이아웃을 설명하는 방법입니다. 이는 사용자에게 장치 이름 및 경로의 세부 사항을 모르는 상태에서 필요한 구성으로 OSD로 전환해야 하는 디스크를 Ceph에 알리는 추상적인 방법을 제공합니다. 각 장치 및 각 호스트에 대해 **yaml** 파일 또는 **json** 파일을 정의합니다.

OSD 사양에 대한 일반 설정

- **service_type:** 'osd': OSDS를 생성해야 합니다.
- **service_id:** 원하는 서비스 이름 또는 ID를 사용합니다. OSD 세트는 사양 파일을 사용하여 생성됩니다. 이 이름은 모든 OSD를 함께 관리하고 오케스트레이터 서비스를 나타내는 데 사용됩니다.
- **배치:** OSD를 배포해야 하는 호스트를 정의하는 데 사용됩니다.
다음 옵션에서 사용할 수 있습니다.
 - **host_octets:** '*' - 호스트를 선택하는 데 사용되는 호스트 이름 패턴입니다.
 - **Label:** 'osd_host' - OSD를 배포해야 하는 호스트에서 사용되는 레이블입니다.
 - **hosts:** 'host01', 'host02' - OSD를 배포해야 하는 호스트 이름의 명시적 목록입니다.
- **장치 선택:** OSD가 생성되는 장치입니다. 이를 통해 OSD를 다른 장치에서 분리할 수 있습니다. 세 가지 구성 요소가 있는 BlueStore OSD만 생성할 수 있습니다.
 - OSD 데이터: 모든 OSD 데이터를 포함합니다.
 - 월: bluestore 내부 저널 또는 쓰기 로그
 - DB: bluestore 내부 메타데이터
- **data_devices:** OSD를 배포할 장치를 정의합니다. 이 경우 OSD는 배치된 스키마에서 생성됩니다. 필터를 사용하여 장치 및 폴더를 선택할 수 있습니다.
- **wal_devices:** WAL OSD에 사용되는 장치를 정의합니다. 필터를 사용하여 장치 및 폴더를 선택할 수 있습니다.
- **db_devices:** DB OSD의 장치를 정의합니다. 필터를 사용하여 장치 및 폴더를 선택할 수 있습니다.
- **encrypted:** **True** 또는 **False**로 설정할 수 있는 OSD에 대한 정보를 암호화하는 선택적 매개변수
- **Unmanaged:** 기본적으로 False로 설정된 선택적 매개변수입니다. 오케스트레이터가 OSD 서비스를 관리하지 않으려면 True로 설정할 수 있습니다.
- **block_wal_size:** 사용자 정의 값 (바이트)입니다. User-defined value, in bytes.
- **block_db_size:** 사용자 정의 값 (바이트)입니다. User-defined value, in bytes.
- **osds_per_device:** 장치당 두 개 이상의 OSD를 배포하기 위한 사용자 정의 값입니다.
- **method:** OSD가 LVM 계층으로 생성되는지 여부를 지정하는 선택적 매개 변수입니다. LVM 계층을 포함하지 않는 원시 물리적 장치에 OSD를 생성하려면 **raw**로 설정합니다. 별도의 DB 또는 WAL 장치가 있는 경우 DB 또는 WAL 장치에 대한 블록 비율은 **1:1이어야 합니다**.

장치 지정을 위한 필터

필터는 **data_devices**, **wal_devices** 및 **db_devices** 매개변수와 함께 사용됩니다.

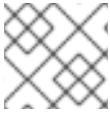
필터 이름	설명	구문	예제
-------	----	----	----

모델	특정 디스크를 대상으로 지정합니다. lsblk -o NAME,FSTYPE,LABEL,MOUNTPOINT,SIZE,MODEL 명령 또는 smartctl -i /dev/sdX 를 실행하여 모델에 대한 세부 정보를 얻을 수 있습니다.	모델 번호: <i>DISK_MODEL_NAME</i>	모델 번호: MC-55-44-XZ
vendor	대상별 디스크	vendor: <i>DISK_VENDOR_NAME</i>	vendor: 공급 업체 Cs
크기 사양	정확한 크기의 디스크 포함	크기: <i>정확한</i>	크기: '10G'
크기 사양	범위 내에 있는 디스크 크기 포함	크기: <i>LOW:HIGH</i>	크기: '10G:40G'
크기 사양	크기보다 작거나 같은 디스크 포함	크기: <i>:HIGH</i>	크기: ':10G'
크기 사양	크기보다 크거나 같은 디스크 포함	크기: <i>아래:</i>	크기: '40G:'
rotational	디스크의 회전 특성입니다. 1은 회전되지 않은 모든 디스크와 일치하며 0은 암호화되지 않은 모든 디스크와 일치합니다. 회전 = 0인 경우 OSD는 SSD 또는 NVME로 구성됩니다. rotational=1인 경우 OSD는 HDD로 구성됩니다.	rotational: 0 또는 1	rotational: 0
all	사용 가능한 모든 디스크를 고려하십시오.	모두: true	모두: true
제한자	유효한 필터를 지정했지만 일치하는 디스크의 양을 제한하려면 'limit' 지시문을 사용할 수 있습니다. 마지막 수단으로만 사용해야 합니다.	제한: <i>NUMBER</i>	제한: 2



참고

동일한 호스트에서 인식되지 않는 구성 요소가 있는 OSD를 생성하려면 사용되는 다양한 유형의 장치를 지정해야 하며 장치가 동일한 호스트에 있어야 합니다.



참고

OSD 배포에 사용되는 장치는 **libstorage** IdentityProvider에서 지원해야 합니다.

추가 리소스

- Red Hat [Ceph Storage Operations Guide](#) 의 고급 사양을 사용하여 Ceph OSD 배포 섹션을 참조하십시오.
- **libstorageostree**에 대한 자세한 내용은 Red Hat [Ceph Storage Operations Guide](#) 의 [Listing devices for Ceph OSD](#) 배포를 참조하십시오.

6.9. 고급 서비스 사양을 사용하여 CEPH OSD 배포

OSD 유형의 서비스 사양은 디스크의 속성을 사용하여 클러스터 레이아웃을 설명하는 방법입니다. 이는 사용자에게 장치 이름 및 경로의 세부 사항을 모르는 상태에서 필요한 구성으로 OSD로 전환해야 하는 디스크를 Ceph에 알리는 추상적인 방법을 제공합니다.

yaml 파일 또는 **json** 파일을 정의하여 각 장치 및 각 호스트의 OSD를 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자 및 모니터 데몬이 배포됩니다.

절차

1. 모니터 노드에서 **osd_spec.yaml** 파일을 생성합니다.

예제

```
[root@host01 ~]# touch osd_spec.yaml
```

2. 다음 세부 정보를 포함하도록 **osd_spec.yaml** 파일을 편집합니다.

구문

```
service_type: osd
service_id: SERVICE_ID
placement:
  host_pattern: '*' # optional
data_devices: # optional
  model: DISK_MODEL_NAME # optional
paths:
  - /DEVICE_PATH
osds_per_device: NUMBER_OF_DEVICES # optional
db_devices: # optional
size: # optional
all: true # optional
```

```
paths:
- /DEVICE_PATH
encrypted: true
```

- a. 간단한 시나리오: 이 경우 모든 노드에 동일한 설정이 적용됩니다.

예제

```
service_type: osd
service_id: osd_spec_default
placement:
  host_pattern: '*'
data_devices:
  all: true
paths:
- /dev/sdb
encrypted: true
```

예제

```
service_type: osd
service_id: osd_spec_default
placement:
  host_pattern: '*'
data_devices:
  size: '80G'
db_devices:
  size: '40G:'
paths:
- /dev/sdc
```

- b. 간단한 시나리오: 이 경우 모든 노드는 LVM 계층 없이 원시 모드로 생성된 OSD 장치와 동일한 설정을 갖습니다.

예제

```
service_type: osd
service_id: all-available-devices
encrypted: "true"
method: raw
placement:
  host_pattern: "*"
data_devices:
  all: "true"
```

- c. 고급 시나리오: 이렇게 하면 모든 HDD를 **data_devices** 로 사용하고 두 SSD가 전용 DB 또는 WAL 장치로 할당되어 원하는 레이아웃이 생성됩니다. 나머지 SSD는 NVMEs 벤더가 전용 DB 또는 WAL 장치로 할당된 **data_devices** 입니다.

예제

```
service_type: osd
service_id: osd_spec_hdd
placement:
```

```

    host_pattern: '*'
    data_devices:
      rotational: 0
    db_devices:
      model: Model-name
      limit: 2
    ---
    service_type: osd
    service_id: osd_spec_ssd
    placement:
      host_pattern: '*'
    data_devices:
      model: Model-name
    db_devices:
      vendor: Vendor-name

```

- d. 공유하지 않은 노드가 있는 고급 시나리오는 다음과 같습니다. 이는 `host_hiera` 키에 따라 다른 호스트에 다른 OSD 사양을 적용합니다.

예제

```

    service_type: osd
    service_id: osd_spec_node_one_to_five
    placement:
      host_pattern: 'node[1-5]'
    data_devices:
      rotational: 1
    db_devices:
      rotational: 0
    ---
    service_type: osd
    service_id: osd_spec_six_to_ten
    placement:
      host_pattern: 'node[6-10]'
    data_devices:
      model: Model-name
    db_devices:
      model: Model-name

```

- e. 전용 WAL 및 DB 장치가 있는 고급 시나리오:

예제

```

    service_type: osd
    service_id: osd_using_paths
    placement:
      hosts:
        - host01
        - host02
    data_devices:
      paths:
        - /dev/sdb
    db_devices:
      paths:
        - /dev/sdc

```

```
wal_devices:
  paths:
    - /dev/sdd
```

- f. 장치당 여러 OSD가 있는 고급 시나리오:

예제

```
service_type: osd
service_id: multiple_osds
placement:
  hosts:
    - host01
    - host02
osds_per_device: 4
data_devices:
  paths:
    - /dev/sdb
```

3. 컨테이너의 디렉터리에 YAML 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount osd_spec.yaml:/var/lib/ceph/osd/osd_spec.yaml
```

4. 디렉터리로 이동합니다.

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/osd/
```

5. OSD를 배포하기 전에 예행을 실행합니다.



참고

이 단계에서는 데몬을 배포하지 않고 배포를 미리 봅니다.

예제

```
[ceph: root@host01 osd]# ceph orch apply -i osd_spec.yaml --dry-run
```

6. 서비스 사양을 사용하여 OSD를 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yml
```

예제

```
[ceph: root@host01 osd]# ceph orch apply -i osd_spec.yaml
```

검증

- 서비스를 나열합니다.

예제

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

- 노드 및 장치의 세부 정보를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

추가 리소스

- Red Hat Ceph Storage Operations Guide 의 [OSD 배포를 위한 고급 서비스 사양 및 필터](#) 를 참조하십시오.

6.10. CEPHORCHESTRATOR를 사용하여 OSD 데몬 제거

Cephadm을 사용하여 클러스터에서 OSD를 제거할 수 있습니다.

클러스터에서 OSD를 제거하려면 다음 두 가지 단계를 수행해야 합니다.

1. 클러스터에서 모든 배치 그룹(PG)을 비웁니다.
2. 클러스터에서 PG-free OSD를 제거합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- Ceph Monitor, Ceph Manager 및 Ceph OSD 데몬은 스토리지 클러스터에 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 장치 및 OSD를 제거해야 하는 노드를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

3. OSD를 제거합니다.

구문

```
ceph orch osd rm OSD_ID [--replace] [--force]
```

예제

```
[ceph: root@host01 /]# ceph orch osd rm 0
```



참고

--replace 와 같은 옵션 없이 스토리지 클러스터에서 OSD를 제거하면 스토리지 클러스터에서 장치가 완전히 제거됩니다. OSD 배포에 동일한 장치를 사용하려면 스토리지 클러스터에 추가하기 전에 먼저 장치를 zap해야 합니다.

4. 선택 사항: 특정 노드에서 여러 OSD를 제거하려면 다음 명령을 실행합니다.

구문

```
ceph orch osd rm OSD_ID OSD_ID
```

예제

```
[ceph: root@host01 /]# ceph orch osd rm 2 5
```

5. OSD 제거 상태를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph orch osd rm status
OSD_ID HOST  STATE  PGS  REPLACE FORCE DRAIN_STARTED_AT
2  host01 draining 124  False False 2021-09-07 16:26:07.142980
5  host01 draining 107  False False 2021-09-07 16:26:08.330371
```

OSD에 PG가 남아 있지 않으면 클러스터에서 해제되고 제거됩니다.

검증

- Ceph OSD가 제거된 장치 및 노드의 세부 정보를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

추가 리소스

- 자세한 내용은 Red Hat [Ceph Storage Operations Guide](#)의 사용 가능한 모든 장치에 Ceph OSD 배포 섹션을 참조하십시오.
- 자세한 내용은 Red Hat [Ceph Storage Operations 가이드](#)의 특정 장치 및 호스트에 Ceph OSD 배포 섹션을 참조하십시오.

- 장치의 공간 정리에 대한 자세한 내용은 [Red Hat Ceph Storage Operations 가이드의 Ceph OSD 배포용 Zapping](#) 장치 섹션을 참조하십시오.

6.11. CEPH ORCHESTRATOR를 사용하여 OSD 교체

디스크가 실패하면 물리적 스토리지 장치를 교체하고 동일한 OSD ID를 재사용하여 CRUSH 맵을 재구성할 필요가 없습니다.

ceph orch rm 명령을 사용하여 OSD ID를 유지하면서 클러스터에서 OSD를 교체할 수 있습니다. OSD는 CRUSH 계층에서 영구적으로 제거되지는 않지만 삭제된 플래그에 **할당됩니다**. 이 플래그는 다음 OSD 배포에서 재사용할 수 있는 OSD ID를 결정하는 데 사용됩니다. 삭제된 플래그는 다음 OSD 배포에서 재사용되는 OSD ID를 결정하는 데 사용됩니다.

배포에 OSD 사양을 사용하는 경우 새로 추가된 디스크에는 교체된 해당 항목의 OSD ID가 할당됩니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모니터, 관리자 및 OSD 데몬이 스토리지 클러스터에 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. OSD를 교체해야 하는 장치 및 노드를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

3. OSD를 교체합니다.

구문

```
ceph orch osd rm OSD_ID --replace
```

예제

```
[ceph: root@host01 /]# ceph orch osd rm 0 --replace
```

4. OSD 교체 상태를 확인합니다.

예제

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

검증

- Ceph OSD가 교체되는 장치 및 노드의 세부 정보를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

동일한 호스트에서 실행 중인 ID와 동일한 ID가 있는 OSD가 표시됩니다.

추가 리소스

- 자세한 내용은 Red Hat [Ceph Storage Operations Guide](#)의 사용 가능한 모든 장치에 Ceph OSD 배포 섹션을 참조하십시오.
- 자세한 내용은 Red Hat [Ceph Storage Operations 가이드](#)의 특정 장치 및 호스트에 Ceph OSD 배포 섹션을 참조하십시오.

6.12. CEPHORCHESTRATOR를 사용하여 OSD 제거 중지

제거를 위해 대기 중인 OSD만 제거를 중지할 수 있습니다. 이렇게 하면 OSD의 초기 상태가 재설정되고 제거 대기열이 해제됩니다.

OSD가 제거 프로세스에 있으면 프로세스를 중지할 수 없습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모니터, 관리자 및 OSD 데몬이 클러스터에 배포됩니다.
- 시작된 OSD 프로세스를 제거합니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 장치 및 OSD가 제거된 노드를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

3. 대기 중인 OSD의 제거를 중지합니다.

구문

```
ceph orch osd rm stop OSD_ID
```

예제

```
[ceph: root@host01 /]# ceph orch osd rm stop 0
```

4. OSD 제거 상태를 확인합니다.

예제

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

검증

- 제거를 위해 Ceph OSD가 대기열에 있는 장치 및 노드의 세부 정보를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

추가 리소스

- 자세한 내용은 Red Hat [Ceph Storage Operations Guide](#)의 [Ceph Orchestrator를 사용하여 OSD 데몬 제거](#)를 참조하십시오.

6.13. CEPH ORCHESTRATOR를 사용하여 OSD 활성화

호스트의 운영 체제를 재설치한 경우 클러스터에서 OSD를 활성화할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모니터, 관리자 및 OSD 데몬이 스토리지 클러스터에 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 호스트의 운영 체제를 다시 설치한 후 OSD를 활성화합니다.

구문

```
ceph cephadm osd activate HOSTNAME
```

예제

```
[ceph: root@host01 /]# ceph cephadm osd activate host03
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

```
ceph orch ps --service_name=SERVICE_NAME
```

예제

```
[ceph: root@host01 /]# ceph orch ps --service_name=osd
```

6.13.1. 데이터 마이그레이션 관찰

OSD를 CRUSH 맵에 추가하거나 제거하면 Ceph에서 배치 그룹을 새 OSD 또는 기존 OSD로 마이그레이션하여 데이터 리밸런싱을 시작합니다. **ceph-w** 명령을 사용하여 데이터 마이그레이션을 확인할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 최근에 OSD를 추가하거나 제거했습니다.

절차

1. 데이터 마이그레이션을 관찰하려면 다음을 수행합니다.

예제

```
[ceph: root@host01 /]# ceph -w
```

2. 마이그레이션이 완료되면 배치 그룹 상태가 **active+clean** 에서 **활성**, 일부 **저하된 개체** 및 마지막으로 **활성+clean** 으로 변경되는 것을 봅니다.
3. 유틸리티를 종료하려면 **Ctrl + C** 를 누릅니다.

6.14. 배치 그룹 재계산

배치 그룹(PG)은 사용 가능한 OSD에서 풀 데이터의 분배를 정의합니다. 배치 그룹은 지정된 중복 알고리즘을 기반으로 구축되어 사용할 수 있습니다. 3방향 복제의 경우 중복은 세 개의 다른 OSD를 사용하도록 정의됩니다. Erasure-coded 풀의 경우 사용할 OSD 수는 청크 수로 정의됩니다.

풀을 정의할 때 배치 그룹의 수는 사용 가능한 모든 OSD에 걸쳐 데이터가 분산되는 세분화 등급을 정의합니다. 크기가 클수록 용량 부하의 동등화가 더 클 수 있습니다. 그러나 데이터를 재구성하는 경우 배치 그룹을 처리하는 것도 중요하므로 신중하게 선택하는 것이 중요합니다. 계산 작업을 지원하기 위해 툴을 사용하여 민첩한 환경을 생성할 수 있습니다.

스토리지 클러스터의 라이프사이클 동안 풀은 초기에 예상되는 한계보다 증가할 수 있습니다. 드라이브 수가 늘어나면 재계산이 권장됩니다. OSD 당 배치 그룹 수는 약 100개여야 합니다. 스토리지 클러스터에 OSD를 더 추가하면 OSD 당 DC 수가 시간이 지남에 따라 줄어듭니다. 스토리지 클러스터에서 처음 120개의 드라이브로 시작하고 풀의 `pg_num` 을 4000개로 설정하면 OSD 당 100개의 Placements가 되며, 3개의 복제 요소가 적용됩니다. 시간이 지남에 따라 OSD 수를 10배로 늘릴 경우 OSD 당 DC 수는 10개로 줄어듭니다. OSD 당 배치 수가 적지 않게 분산된 용량이기 때문에 풀당 PPP를 조정하는 것이 좋습니다.

배치 그룹의 수를 조정하는 작업은 온라인으로 수행할 수 있습니다. 재계산은 PPP 숫자를 재계산하는 것뿐만 아니라 데이터 재배포가 포함될 것이며, 이는 긴 프로세스가 될 것입니다. 그러나 데이터 가용성은 언제든지 유지됩니다.

실패한 OSD에 있는 모든 DC의 재구성이 즉시 시작되므로 OSD 당 VDO 수가 매우 많으면 안 됩니다. 많은 IOPS가 적시에 재구성을 수행해야 하므로 사용할 수 없을 수 있습니다. 이로 인해 깊은 I/O 대기열과 스토리지 클러스터를 사용할 수 없게 되거나 복구 시간이 길어집니다.

추가 리소스

- 지정된 사용 사례에 따른 값을 계산하려면 DASD [계산기](#) 를 참조하십시오.
- 자세한 내용은 Red Hat Ceph Storage Strategies Guide(Red Hat Ceph 스토리지 전략 가이드)의 [코드 풀](#) 장을 참조하십시오.

7장. CEPHORCHESTRATOR를 사용한 모니터링 스택 관리

스토리지 관리자는 백엔드에서 Cephadm과 함께 Ceph Orchestrator를 사용하여 모니터링 및 경고 스택을 배포할 수 있습니다. 모니터링 스택은 Prometheus, Prometheus exporters, Prometheus Alertmanager 및 Grafana로 구성됩니다. 사용자는 Cephadm을 사용하여 YAML 구성 파일에 이러한 서비스를 정의하거나 명령줄 인터페이스를 사용하여 배포할 수 있습니다. 동일한 유형의 여러 서비스가 배포되면고가용성 설정이 배포됩니다. 노드 내보내기는 이 규칙에 대한 예외입니다.



참고

Red Hat Ceph Storage 5.0은 Prometheus, Grafana, Alertmanager 및 node-exporter와 같은 모니터링 서비스를 배포하기 위한 사용자 지정 이미지를 지원하지 않습니다.

Cephadm을 사용하여 다음 모니터링 서비스를 배포할 수 있습니다.

- Prometheus는 모니터링 및 경고 툴킷입니다. 사전 정의된 임계값에 도달한 경우 Prometheus 내보내기에서 제공한 데이터를 수집하고 사전 구성된 경고를 실행합니다. Prometheus manager 모듈은 **ceph-mgr**의 컬렉션 지점에서 Ceph 성능 카운터를 전달할 수 있는 Prometheus 내보내기를 제공합니다.
데몬을 제공하는 지표와 같은 스크랩 대상을 포함한 Prometheus 구성은 Cephadm에 의해 자동으로 설정됩니다. 또한 Cephadm은 기본 경고 목록(예: 상태 오류, 10% OSD 다운 또는 pgs 비활성)을 배포합니다.
- Alertmanager는 Prometheus 서버에서 전송한 경고를 처리합니다. 이는 경고를 중복, 그룹 및 라우팅하여 올바른 수신자에게 라우팅합니다. 기본적으로 Ceph 대시보드는 자동으로 수신자로 구성됩니다. Alertmanager는 Prometheus 서버에서 보낸 경고를 처리합니다. Alertmanager를 사용하여 경고를 음소거할 수 있지만 Ceph 대시보드를 사용하여 음소거할 수도 있습니다.
- Grafana는 시각화 및 경고 소프트웨어입니다. Grafana의 경고 기능은 이 모니터링 스택에서 사용되지 않습니다. 경고의 경우 Alertmanager가 사용됩니다.
기본적으로 Grafana에 대한 트래픽은 TLS로 암호화됩니다. 고유한 TLS 인증서를 제공하거나 자체 서명된 인증서를 사용할 수 있습니다. Grafana를 배포하기 전에 사용자 정의 인증서가 구성되지 않은 경우 Grafana에 대해 자체 서명된 인증서가 자동으로 생성되고 구성됩니다. Grafana의 사용자 지정 인증서는 다음 명령을 사용하여 구성할 수 있습니다.

구문

```
ceph config-key set mgr/cephadm/grafana_key -i
PRESENT_WORKING_DIRECTORY/key.pem
ceph config-key set mgr/cephadm/grafana_cert -i
PRESENT_WORKING_DIRECTORY/certificate.pem
```

노드 내보내기는 설치된 노드에 대한 데이터를 제공하는 Prometheus의 내보내기제입니다. 모든 노드에 노드 내보내기를 설치하는 것이 좋습니다. 이 작업은 node-exporter 서비스 유형의 **monitoring.yml** 파일을 사용하여 수행할 수 있습니다.

7.1. CEPHORCHESTRATOR를 사용하여 모니터링 스택 배포

모니터링 스택은 Prometheus, Prometheus exporters, Prometheus Alertmanager 및 Grafana로 구성됩니다. Ceph 대시보드는 이러한 구성 요소를 사용하여 클러스터 사용량 및 성능에 대한 세부 지표를 저장하고 시각화합니다.

YAML 파일 형식의 서비스 사양을 사용하여 모니터링 스택을 배포할 수 있습니다. 모든 모니터링 서비스에는 **yml** 파일에 구성된 네트워크와 포트가 있을 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.

절차

1. Ceph Manager 데몬에서 prometheus 모듈을 활성화합니다. 이렇게 하면 Prometheus에서 읽을 수 있도록 내부 Ceph 지표가 노출됩니다.

예제

```
[ceph: root@host01 /]# ceph mgr module enable prometheus
```



중요

Prometheus를 배포하기 전에 이 명령이 실행되는지 확인합니다. 배포 전에 명령이 실행되지 않은 경우 Prometheus를 재배포하여 구성을 업데이트해야 합니다.

```
ceph orch redeploy prometheus
```

2. 다음 디렉터리로 이동합니다.

구문

```
cd /var/lib/ceph/DAEMON_PATH/
```

예제

```
[ceph: root@host01 mds/]# cd /var/lib/ceph/monitoring/
```



참고

디렉터리 **모니터링**이 없으면 생성합니다.

3. **monitoring.yml** 파일을 생성합니다.

예제

```
[ceph: root@host01 monitoring]# touch monitoring.yml
```

4. 다음 예와 유사한 콘텐츠를 사용하여 사양 파일을 편집합니다.

예제

```
service_type: prometheus
service_name: prometheus
placement:
  hosts:
    - host01
```

```

networks:
- 192.169.142.0/24
port:
- 8080
---
service_type: node-exporter
---
service_type: alertmanager
service_name: alertmanager
placement:
  hosts:
  - host03
networks:
- 192.169.142.03/24
port:
- 9876
---
service_type: grafana
service_name: grafana
placement:
  hosts:
  - host02
networks:
- 192.169.142.02/24
port:
- 4100

```

5. 모니터링 서비스를 적용합니다.

예제

```
[ceph: root@host01 monitoring]# ceph orch apply -i monitoring.yml
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

```
ceph orch ps --service_name=SERVICE_NAME
```

예제

```
[ceph: root@host01 /]# ceph orch ps --service_name=prometheus
```



중요

Prometheus, Grafana 및 Ceph 대시보드는 모두 서로 통신하도록 자동 구성되므로 Ceph 대시보드에서 완전히 작동하는 Grafana 통합이 생성됩니다.

7.2. CEPHORCHESTRATOR를 사용하여 모니터링 스택 제거

ceph orch rm 명령을 사용하여 모니터링 스택을 제거할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. **ceph orch rm** 명령을 사용하여 모니터링 스택을 제거합니다.

구문

```
ceph orch rm SERVICE_NAME --force
```

예제

```
[ceph: root@host01 /]# ceph orch rm grafana
[ceph: root@host01 /]# ceph orch rm prometheus
[ceph: root@host01 /]# ceph orch rm node-exporter
[ceph: root@host01 /]# ceph orch rm alertmanager
[ceph: root@host01 /]# ceph mgr module disable prometheus
```

3. 프로세스의 상태를 확인합니다.

예제

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

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

```
ceph orch ps
```

예제

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

추가 리소스

- 자세한 내용은 Red Hat [Ceph Storage Operations 가이드](#)의 Ceph Orchestrator를 사용하여 모니터링 스택 배포 섹션을 참조하십시오.

8장. 기본 RED HAT CEPH STORAGE 클라이언트 설정

스토리지 관리자는 스토리지 클러스터와 상호 작용하기 위해 기본 구성으로 클라이언트 시스템을 설정해야 합니다. 대부분의 클라이언트 시스템에는 **ceph-common** 패키지와 해당 종속 항목만 설치해야 합니다. 기본 **ceph** 및 **rados** 명령과 **mount.ceph** 및 **rbd** 같은 기타 명령을 제공합니다.

8.1. 클라이언트 시스템에서 파일 설정 구성

클라이언트 시스템에는 일반적으로 완전한 스토리지 클러스터 멤버보다 작은 구성 파일이 필요합니다. Ceph 모니터에 도달하기 위해 클라이언트에 세부 정보를 제공할 수 있는 최소 구성 파일을 생성할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 액세스입니다.

절차

1. 파일을 설정하려는 노드에서 **/etc** 폴더에 디렉터리 **ceph** 를 생성합니다.

예제

```
[root@host01 ~]# mkdir /etc/ceph/
```

2. **/etc/ceph** 디렉터리로 이동합니다.

예제

```
[root@host01 ~]# cd /etc/ceph/
```

3. **ceph** 디렉터리에서 구성 파일을 생성합니다.

예제

```
[root@host01 ceph]# ceph config generate-minimal-conf

# minimal ceph.conf for 417b1d7a-a0e6-11eb-b940-001a4a000740
[global]
fsid = 417b1d7a-a0e6-11eb-b940-001a4a000740
mon_host = [v2:10.74.249.41:3300/0,v1:10.74.249.41:6789/0]
```

이 파일의 내용은 **/etc/ceph/ceph.conf** 경로에 설치되어야 합니다. 이 구성 파일을 사용하여 Ceph 모니터에 연결할 수 있습니다.

8.2. 클라이언트 시스템에서 인증 키 설정

대부분의 Ceph 클러스터는 인증이 활성화된 상태로 실행되며 클라이언트는 클러스터 시스템과 통신하기 위해 키가 필요합니다. Ceph 모니터에 도달하기 위해 클라이언트에 세부 정보를 제공할 수 있는 인증 키를 생성할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 액세스입니다.

절차

1. 인증 키를 설정하려는 노드에서 **/etc** 폴더에 디렉터리 **ceph** 를 생성합니다.

예제

```
[root@host01 ~]# mkdir /etc/ceph/
```

2. **ceph** 디렉터리의 **/etc/ceph** 디렉터리로 이동합니다.

예제

```
[root@host01 ~]# cd /etc/ceph/
```

3. 클라이언트의 인증 키를 생성합니다.

구문

```
ceph auth get-or-create client.CLIENT_NAME -o /etc/ceph/NAME_OF_THE_FILE
```

예제

```
[root@host01 ceph]# ceph auth get-or-create client.fs -o /etc/ceph/ceph.keyring
```

4. **ceph.keyring** 파일의 출력을 확인합니다.

예제

```
[root@host01 ceph]# cat ceph.keyring
```

```
[client.fs]
key = AQAvoH5gkUCsExAATz3xCBLd4n6B6jRv+Z7CVQ==
```

결과 출력은 인증 키 파일(예: **/etc/ceph/ceph.keyring**)에 배치해야 합니다.

9장. CEPHORCHESTRATOR를 사용한 MDS 서비스 관리

스토리지 관리자는 백엔드에서 Cephadm과 함께 Ceph Orchestrator를 사용하여 MDS 서비스를 배포할 수 있습니다. 기본적으로 Ceph 파일 시스템(CephFS)은 하나의 활성 MDS 데몬만 사용합니다. 그러나 많은 클라이언트가 있는 시스템에서는 여러 개의 활성 MDS 데몬을 활용할 수 있습니다.

이 섹션에서는 다음 관리 작업에 대해 설명합니다.

- 명령줄 인터페이스를 사용하여 MDS 서비스 배포.
- 서비스 사양을 사용하여 MDS 서비스 배포.
- Ceph Orchestrator를 사용하여 MDS 서비스 제거.

9.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

9.2. 명령줄 인터페이스를 사용하여 MDS 서비스 배포

Ceph Orchestrator를 사용하면 명령줄 인터페이스의 **배치** 사양을 사용하여 메타데이터 서버(MDS) 서비스를 배포할 수 있습니다. Ceph 파일 시스템(CephFS)에는 하나 이상의 MDS가 필요합니다.



참고

Ceph 파일 시스템(CephFS) 데이터용 데이터와 CephFS 메타데이터용 풀이 두 개 이상 있는지 확인합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

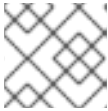
2. 배치 사양을 사용하여 MDS 데몬을 배포하는 방법은 다음 두 가지가 있습니다.

방법 1

- **ceph fs** 볼륨을 사용하여 MDS 데몬을 생성합니다. 그러면 CephFS와 연결된 CephFS 볼륨 및 풀이 생성되고 호스트에서 MDS 서비스도 시작합니다.

구문

```
ceph fs volume create FILESYSTEM_NAME --placement="NUMBER_OF_DAEMONS
HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```



참고

기본적으로 이 명령에 대해 복제된 풀이 생성됩니다.

예제

```
[ceph: root@host01 /]# ceph fs volume create test --placement="2 host01 host02"
```

방법 2

- 풀인 CephFS를 생성한 다음 배치 사양을 사용하여 MDS 서비스를 배포합니다.
 - a. CephFS의 풀을 생성합니다.

구문

```
ceph osd pool create DATA_POOL
ceph osd pool create METADATA_POOL
```

예제

```
[ceph: root@host01 /]# ceph osd pool create cephfs_data
[ceph: root@host01 /]# ceph osd pool create cephfs_metadata
```

- b. 데이터 풀 및 메타데이터 풀의 파일 시스템을 생성합니다.

구문

```
ceph fs new FILESYSTEM_NAME METADATA_POOL DATA_POOL
```

예제

```
[ceph: root@host01 /]# ceph fs new test cephfs_metadata cephfs_data
```

- c. **ceph orch apply** 명령을 사용하여 MDS 서비스를 배포합니다.

구문

```
ceph orch apply mds FILESYSTEM_NAME --placement="NUMBER_OF_DAEMONS
HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

예제

```
[ceph: root@host01 /]# ceph orch apply mds test --placement="2 host01 host02"
```

검증

- 서비스를 나열합니다.

예제

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

- CephFS 상태를 확인합니다.

예제

```
[ceph: root@host01 /]# ceph fs ls
[ceph: root@host01 /]# ceph fs status
```

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mds
```

추가 리소스

- [Ceph 파일 시스템\(CephFS\) 생성에 대한 자세한 내용은 Red Hat Ceph Storage File System 가이드를 참조하십시오.](#)

9.3. 서비스 사양을 사용하여 MDS 서비스 배포

Ceph Orchestrator를 사용하면 서비스 사양을 사용하여 MDS 서비스를 배포할 수 있습니다.



참고

Ceph File System(CephFS) 데이터용 풀과 CephFS 메타데이터용으로 각각 두 개 이상의 풀이 있는지 확인합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

절차

1. **mds.yaml** 파일을 생성합니다.

예제

```
[root@host01 ~]# touch mds.yaml
```

2. 다음 세부 사항을 포함하도록 **mds.yaml** 파일을 편집합니다.

구문

```
service_type: mds
service_id: FILESYSTEM_NAME
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
    - HOST_NAME_3
```

예제

```
service_type: mds
service_id: fs_name
placement:
  hosts:
    - host01
    - host02
```

3. 컨테이너의 디렉터리에 YAML 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount mds.yaml:/var/lib/ceph/mds/mds.yaml
```

4. 디렉터리로 이동합니다.

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/mds/
```

5. Cephadm 셸에 로그인합니다.

예제

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

6. 다음 디렉터리로 이동합니다.

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/mds/
```

7. 서비스 사양을 사용하여 MDS 서비스를 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yaml
```

예제

```
[ceph: root@host01 mds]# ceph orch apply -i mds.yaml
```

8. MDS 서비스가 배포되고 작동되면 CephFS를 생성합니다.

구문

```
ceph fs new CEPHFS_NAME METADATA_POOL DATA_POOL
```

예제

```
[ceph: root@host01 /]# ceph fs new test metadata_pool data_pool
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mds
```

추가 리소스

- [Ceph 파일 시스템\(CephFS\) 생성에 대한 자세한 내용은 Red Hat Ceph Storage File System 가이드를 참조하십시오.](#)

9.4. CEPHORCHESTRATOR를 사용하여 MDS 서비스 제거

ceph orch rm 명령을 사용하여 서비스를 제거할 수 있습니다. 또는 파일 시스템 및 관련 풀을 제거할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.

- 호스트는 클러스터에 추가됩니다.
- 호스트에 배포된 하나 이상의 MDS 데몬.

절차

- 클러스터에서 MDS 데몬을 제거하는 방법은 다음 두 가지가 있습니다.

방법 1

- CephFS 볼륨, 관련 풀 및 서비스를 제거합니다.
 - a. Cephadm 셸에 로그인합니다.

예제

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

- b. 설정 매개변수 **mon_allow_pool_delete** 를 **true** 로 설정합니다.

예제

```
[ceph: root@host01 /]# ceph config set mon mon_allow_pool_delete true
```

- c. 파일 시스템을 제거합니다.

구문

```
ceph fs volume rm FILESYSTEM_NAME --yes-i-really-mean-it
```

예제

```
[ceph: root@host01 /]# ceph fs volume rm cephfs-new --yes-i-really-mean-it
```

이 명령은 파일 시스템, 해당 데이터 및 메타데이터 풀을 제거합니다. 또한 활성화된 **ceph-mgr** Orchestrator 모듈을 사용하여 MDS를 제거합니다.

방법 2

- **ceph orch rm** 명령을 사용하여 전체 클러스터에서 MDS 서비스를 제거합니다.
 - a. 서비스를 나열합니다.

예제

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

- b. 서비스 제거

구문

```
ceph orch rm SERVICE_NAME
```

예제

```
[ceph: root@host01 /]# ceph orch rm mds.test
```

검증

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

```
ceph orch ps
```

예제

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

추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 의 명령행 인터페이스를 사용하여 MDS 서비스 배포를 참조하십시오.
- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 의 서비스 사양을 사용하여 MDS 서비스 배포를 참조하십시오.

10 장. CEPHORCHESTRATOR를 사용한 CEPH 개체 게이트웨이 관리

스토리지 관리자는 명령줄 인터페이스를 사용하거나 서비스 사양을 사용하여 Ceph 오브젝트 게이트웨이를 배포할 수 있습니다.

여러 사이트 개체 게이트웨이를 구성하고 Ceph Orchestrator를 사용하여 Ceph 개체 게이트웨이를 제거할 수도 있습니다.

Cephadm은 Ceph 오브젝트 게이트웨이를 다중 사이트 배포의 단일 클러스터 배포 또는 특정 영역 및 영역을 관리하는 데몬 컬렉션으로 배포합니다.



참고

Cephadm에서 오브젝트 게이트웨이 데몬은 **ceph.conf** 또는 명령줄 대신 모니터 구성 데이터베이스를 사용하여 구성됩니다. 해당 구성이 **client.rgw** 섹션에 없는 경우 오브젝트 게이트웨이 데몬은 기본 설정으로 시작하고 포트 **80**에 바인딩됩니다.



참고

.default.rgw.buckets.index 폴은 Ceph Object Gateway에서 버킷이 생성된 후 데이터를 버킷에 업로드한 후 **.default.rgw.buckets.data** 폴이 생성된 후에만 생성됩니다.

이 섹션에서는 다음 관리 작업에 대해 설명합니다.

- 명령줄 인터페이스를 사용하여 Ceph 오브젝트 게이트웨이 배포.
- 서비스 사양을 사용하여 Ceph 오브젝트 게이트웨이를 배포합니다.
- Ceph Orchestrator를 사용하여 다중 사이트 Ceph 개체 게이트웨이 배포.
- CephOrchestrator를 사용하여 Ceph 개체 게이트웨이 제거.

10.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터, OSD는 스토리지 클러스터에 배포됩니다.

10.2. 명령줄 인터페이스를 사용하여 CEPH OBJECT GATEWAY 배포

Ceph Orchestrator를 사용하여 명령줄 인터페이스에서 **ceph** 또는 **ch** 명령으로 Ceph Object Gateway를 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.

- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. Ceph 오브젝트 게이트웨이 데몬을 다음 세 가지 방법으로 배포할 수 있습니다.

방법 1

- realm, 영역 그룹, 영역을 생성한 다음 호스트 이름으로 배치 사양을 사용합니다.
 - a. 영역을 생성합니다.

구문

```
radosgw-admin realm create --rgw-realm=REALM_NAME --default
```

예제

```
[ceph: root@host01 /]# radosgw-admin realm create --rgw-realm=test_realm --default
```

- b. 영역 그룹을 생성합니다.

구문

```
radosgw-admin zonegroup create --rgw-zonegroup=ZONE_GROUP_NAME --master --default
```

예제

```
[ceph: root@host01 /]# radosgw-admin zonegroup create --rgw-zonegroup=default --master --default
```

- c. 영역을 생성합니다.

구문

```
radosgw-admin zone create --rgw-zonegroup=ZONE_GROUP_NAME --rgw-zone=ZONE_NAME --master --default
```

예제

```
[ceph: root@host01 /]# radosgw-admin zone create --rgw-zonegroup=default --rgw-zone=test_zone --master --default
```

- d. 변경 사항을 커밋합니다.

구문

```
radosgw-admin period update --rgw-realm=REALM_NAME --commit
```

예제

```
[ceph: root@host01 /]# radosgw-admin period update --rgw-realm=test_realm --commit
```

- e. **ceph orch apply** 명령을 실행합니다.

구문

```
ceph orch apply rgw NAME [--realm=REALM_NAME] [--zone=ZONE_NAME] --  
placement="NUMBER_OF_DAEMONS [HOST_NAME_1 HOST_NAME_2]"
```

예제

```
[ceph: root@host01 /]# ceph orch apply rgw test --realm=test_realm --zone=test_zone --  
placement="2 host01 host02"
```

방법 2

- 단일 클러스터 배포를 위한 두 개의 Ceph Object Gateway 데몬을 배포하려면 임의의 서비스 이름을 사용합니다.

구문

```
ceph orch apply rgw SERVICE_NAME
```

예제

```
[ceph: root@host01 /]# ceph orch apply rgw foo
```

방법 3

- 레이블이 지정된 호스트 집합에서 임의의 서비스 이름을 사용합니다.

구문

```
ceph orch host label add HOST_NAME_1 LABEL_NAME  
ceph orch host label add HOSTNAME_2 LABEL_NAME  
ceph orch apply rgw SERVICE_NAME '--placement=label:LABEL_NAME  
count_per_host:NUMBER_OF_DAEMONS' --port=8000
```



참고

NUMBER_OF_DAEMONS 는 각 호스트에 배포된 Ceph 개체 게이트웨이의 수를 제어합니다. 추가 비용을 유발하지 않고 최고 성능을 얻으려면 이 값을 2로 설정합니다.

예제

```
[ceph: root@host01 /]# ceph orch host label add host01 rgw # the 'rgw' label can be anything
[ceph: root@host01 /]# ceph orch host label add host02 rgw
[ceph: root@host01 /]# ceph orch apply rgw foo '--placement=label:rgw count_per_host:2' --
port=8000
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=rgw
```

10.3. 서비스 사양을 사용하여 CEPH OBJECT GATEWAY 배포

기본 또는 사용자 지정 영역, 영역 및 영역 그룹과 함께 서비스 사양을 사용하여 Ceph Object Gateway를 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 부트스트랩된 호스트에 대한 루트 수준 액세스입니다.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

절차

1. 루트 사용자로 사양 파일을 생성합니다.

예제

```
[root@host01 ~]# touch radosgw.yml
```

2. 기본 영역, 영역 및 영역 그룹에 대한 다음 세부 정보를 포함하도록 **radosgw.yml** 파일을 편집합니다.

구문

```

service_type: rgw
service_id: REALM_NAME.ZONE_NAME
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
  count_per_host: NUMBER_OF_DAEMONS
spec:
  rgw_realm: REALM_NAME
  rgw_zone: ZONE_NAME
  rgw_frontend_port: FRONT_END_PORT
networks:
  - NETWORK_CIDR # Ceph Object Gateway service binds to a specific network

```



참고

NUMBER_OF_DAEMONS 는 각 호스트에 배포된 Ceph Object Gateway 의 수를 제어합니다. 추가 비용을 유발하지 않고 최고 성능을 얻으려면 이 값을 2로 설정합니다.

예제

```

service_type: rgw
service_id: default
placement:
  hosts:
    - host01
    - host02
    - host03
  count_per_host: 2
spec:
  rgw_realm: default
  rgw_zone: default
  rgw_frontend_port: 1234
networks:
  - 192.169.142.0/24

```

3. 선택 사항: 사용자 지정 영역, 영역 및 영역 그룹의 경우 리소스를 생성한 다음 **radosgw.yml** 파일을 생성합니다.
 - a. 사용자 지정 영역, 영역 및 영역 그룹을 생성합니다.

예제

```

[root@host01 ~]# radosgw-admin realm create --rgw-realm=test_realm
[root@host01 ~]# radosgw-admin zonegroup create --rgw-zonegroup=test_zonegroup
[root@host01 ~]# radosgw-admin zone create --rgw-zonegroup=test_zonegroup --rgw-zone=test_zone
[root@host01 ~]# radosgw-admin period update --rgw-realm=test_realm --commit

```

- b. 다음 세부 정보를 사용하여 **radosgw.yml** 파일을 생성합니다.

예제

■

```

service_type: rgw
service_id: test_realm.test_zone
placement:
  hosts:
    - host01
    - host02
    - host03
  count_per_host: 2
spec:
  rgw_realm: test_realm
  rgw_zone: test_zone
  rgw_frontend_port: 1234
networks:
  - 192.169.142.0/24

```

4. 컨테이너의 디렉터리에 **radosgw.yml** 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount radosgw.yml:/var/lib/ceph/radosgw/radosgw.yml
```



참고

셸을 종료할 때마다 데몬을 배포하기 전에 컨테이너에 파일을 마운트해야 합니다.

5. 서비스 사양을 사용하여 Ceph Object Gateway를 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yml
```

예제

```
[ceph: root@host01 /]# ceph orch apply -i radosgw.yml
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=rgw
```

10.4. CEPH ORCHESTRATOR를 사용하여 다중 사이트 CEPH OBJECT GATEWAY 배포

Ceph Orchestrator는 Ceph Object Gateway에 대한 다중 사이트 구성 옵션을 지원합니다.

기본이 아닌 영역에 쓰기를 허용하는 활성-활성 영역 구성에서 작동하도록 각 오브젝트 게이트웨이를 구성할 수 있습니다. 다중 사이트 구성은 영역이라는 컨테이너 내에 저장됩니다.

영역은 영역 그룹, 영역, 및 기간을 저장합니다. **rgw** 데몬은 동기화를 처리하여 별도의 동기화 에이전트의 필요성을 제거하여 활성-활성 구성으로 작동합니다.

CLI(명령줄 인터페이스)를 사용하여 다중 사이트 영역을 배포할 수도 있습니다.



참고

다음 구성에서는 두 개 이상의 Red Hat Ceph Storage 클러스터가 지리적으로 분리된 위치에 있다고 가정합니다. 그러나 구성은 동일한 사이트에서도 작동합니다.

사전 요구 사항

- Red Hat Ceph Storage 클러스터를 두 개 이상 실행합니다.
- 각 Red Hat Ceph Storage 클러스터에 대해 각각 하나씩 두 개 이상의 Ceph Object Gateway 인스턴스.
- 모든 노드에 대한 루트 수준의 액세스.
- 노드 또는 컨테이너가 스토리지 클러스터에 추가됩니다.
- 모든 Ceph Manager, 모니터 및 OSD 데몬이 배포됩니다.

절차

1. **cephadm** 셸에서 기본 영역을 구성합니다.

- a. 영역을 생성합니다.

구문

```
radosgw-admin realm create --rgw-realm=REALM_NAME --default
```

예제

```
[ceph: root@host01 /]# radosgw-admin realm create --rgw-realm=test_realm --default
```

스토리지 클러스터에 단일 영역이 있는 경우 **--default** 플래그를 지정합니다.

- b. 기본 영역 그룹을 생성합니다.

구문

```
radosgw-admin zonegroup create --rgw-zonegroup=ZONE_GROUP_NAME --
endpoints=http://RGW_PRIMARY_HOSTNAME:RGW_PRIMARY_PORT_NUMBER_1 --
master --default
```

예제

```
[ceph: root@host01 /]# radosgw-admin zonegroup create --rgw-zonegroup=us --
endpoints=http://rgw1:80 --master --default
```

- c. 기본 영역을 생성합니다.

구문

```
radosgw-admin zone create --rgw-zonegroup=PRIMARY_ZONE_GROUP_NAME --rgw-
zone=PRIMARY_ZONE_NAME --
endpoints=http://RGW_PRIMARY_HOSTNAME:RGW_PRIMARY_PORT_NUMBER_1 --
access-key=SYSTEM_ACCESS_KEY --secret=SYSTEM_SECRET_KEY
```

예제

```
[ceph: root@host01 /]# radosgw-admin zone create --rgw-zonegroup=us --rgw-zone=us-
east-1 --endpoints=http://rgw1:80 --access-key=LIPEYZJLTWXRKXS9LPJC --secret-
key=lsAje0AVDNXNw48LjMAimpCpl7VaxJYSnfD0FFKQ
```

- d. 선택 사항: 기본 영역, 영역 그룹 및 관련 풀을 삭제합니다.



중요

데이터를 저장하기 위해 기본 영역 및 영역 그룹을 사용하는 경우 기본 영역과 해당 풀을 삭제하지 마십시오. 또한 기본 영역 그룹을 제거하면 시스템 사용자가 삭제됩니다.

기본 영역 및 영역 그룹의 이전 데이터에 액세스하려면 **radosgw-admin** 명령에서 **--rgw-zone default** 및 **--rgw-zonegroup default** 를 사용합니다.

예제

```
[ceph: root@host01 /]# radosgw-admin zonegroup delete --rgw-zonegroup=default
[ceph: root@host01 /]# ceph osd pool rm default.rgw.log default.rgw.log --yes-i-really-
really-mean-it
[ceph: root@host01 /]# ceph osd pool rm default.rgw.meta default.rgw.meta --yes-i-
really-really-mean-it
[ceph: root@host01 /]# ceph osd pool rm default.rgw.control default.rgw.control --yes-i-
really-really-mean-it
[ceph: root@host01 /]# ceph osd pool rm default.rgw.data.root default.rgw.data.root --
yes-i-really-really-mean-it
[ceph: root@host01 /]# ceph osd pool rm default.rgw.gc default.rgw.gc --yes-i-really-
really-mean-it
```

- e. 시스템 사용자를 생성합니다.

구문

```
radosgw-admin user create --uid=USER_NAME --display-name="USER_NAME" --
access-key=SYSTEM_ACCESS_KEY --secret=SYSTEM_SECRET_KEY --system
```

예제

```
[ceph: root@host01 /]# radosgw-admin user create --uid=zone.user --display-
name="Zone user" --system
```

access_key 및 **secret_key** 를 기록해 둡니다.

- f. 기본 영역에 액세스 키와 시스템 키를 추가합니다.

구문

```
radosgw-admin zone modify --rgw-zone=PRIMARY_ZONE_NAME --access-
key=ACCESS_KEY --secret=SECRET_KEY
```

예제

```
[ceph: root@host01 /]# radosgw-admin zone modify --rgw-zone=us-east-1 --access-
key=NE48APYCAODEPLKBCZVQ--
secret=u24GHQWRE3yxxNBnFBzjM4jn14mFlckQ4EKL6LoW
```

- g. 변경 사항을 커밋합니다.

구문

```
radosgw-admin period update --commit
```

예제

```
[ceph: root@host01 /]# radosgw-admin period update --commit
```

- h. **cephadm** 셸 외부에서 스토리지 클러스터 및 프로세스의 **FSID** 를 가져옵니다.

예제

```
[root@host01 ~]# systemctl list-units | grep ceph
```

- i. Ceph Object Gateway 데몬을 시작합니다.

구문

```
systemctl start ceph-FSID@DAEMON_NAME
systemctl enable ceph-FSID@DAEMON_NAME
```

예제

```
[root@host01 ~]# systemctl start ceph-62a081a6-88aa-11eb-a367-
001a4a000672@rgw.test_realm.us-east-1.host01.ahdtsw.service
[root@host01 ~]# systemctl enable ceph-62a081a6-88aa-11eb-a367-
```

```
001a4a000672@rgw.test_realm.us-east-1.host01.ahdtsw.service
```

2. Cephadm 셸에서 보조 영역을 구성합니다.

- a. 호스트에서 기본 영역 구성을 가져옵니다.

구문

```
radosgw-admin realm pull --url=URL_TO_PRIMARY_ZONE_GATEWAY --access-  
key=ACCESS_KEY --secret-key=SECRET_KEY
```

예제

```
[ceph: root@host04 /]# radosgw-admin realm pull --url=http://10.74.249.26:80 --access-  
key=LIPEYZJLTWXRKXS9LPJC --secret-  
key=lsAje0AVDNXNw48LjMAimpCpl7VaxJYSnfD0FFKQ
```

- b. 호스트에서 기본 기간 구성을 가져옵니다.

구문

```
radosgw-admin period pull --url=URL_TO_PRIMARY_ZONE_GATEWAY --access-  
key=ACCESS_KEY --secret-key=SECRET_KEY
```

예제

```
[ceph: root@host04 /]# radosgw-admin period pull --url=http://10.74.249.26:80 --access-  
key=LIPEYZJLTWXRKXS9LPJC --secret-  
key=lsAje0AVDNXNw48LjMAimpCpl7VaxJYSnfD0FFKQ
```

- c. 보조 영역을 구성합니다.

구문

```
radosgw-admin zone create --rgw-zonegroup=ZONE_GROUP_NAME \  
--rgw-zone=SECONDARY_ZONE_NAME --  
endpoints=http://RGW_SECONDARY_HOSTNAME:RGW_PRIMARY_PORT_NUMBER  
_1 \  
--access-key=SYSTEM_ACCESS_KEY --secret=SYSTEM_SECRET_KEY \  
--endpoints=http://FQDN:80 \  
[--read-only]
```

예제

```
[ceph: root@host04 /]# radosgw-admin zone create --rgw-zonegroup=us --rgw-zone=us-  
east-2 --endpoints=http://rgw2:80 --access-key=LIPEYZJLTWXRKXS9LPJC --secret-  
key=lsAje0AVDNXNw48LjMAimpCpl7VaxJYSnfD0FFKQ --  
endpoints=http://rgw.example.com:80
```

- d. 선택 사항: 기본 영역을 삭제합니다.



중요

데이터를 저장하기 위해 기본 영역 및 영역 그룹을 사용하는 경우 기본 영역과 해당 풀을 삭제하지 마십시오.

기본 영역 및 영역 그룹의 이전 데이터에 액세스하려면 **radosgw-admin** 명령에서 **--rgw-zone default** 및 **--rgw-zonegroup default** 를 사용합니다.

예제

```
[ceph: root@host04 /]# radosgw-admin zone rm --rgw-zone=default
[ceph: root@host04 /]# ceph osd pool rm default.rgw.log default.rgw.log --yes-i-really-mean-it
[ceph: root@host04 /]# ceph osd pool rm default.rgw.meta default.rgw.meta --yes-i-really-mean-it
[ceph: root@host04 /]# ceph osd pool rm default.rgw.control default.rgw.control --yes-i-really-mean-it
[ceph: root@host04 /]# ceph osd pool rm default.rgw.data.root default.rgw.data.root --yes-i-really-mean-it
[ceph: root@host04 /]# ceph osd pool rm default.rgw.gc default.rgw.gc --yes-i-really-mean-it
```

- e. Ceph 구성 데이터베이스를 업데이트합니다.

구문

```
ceph config set SERVICE_NAME rgw_zone SECONDARY_ZONE_NAME
```

예제

```
[ceph: root@host04 /]# ceph config set rgw rgw_zone us-east-2
```

- f. 변경 사항을 커밋합니다.

구문

```
radosgw-admin period update --commit
```

예제

```
[ceph: root@host04 /]# radosgw-admin period update --commit
```

- g. Cephadm 셸 외부에서 스토리지 클러스터 및 프로세스의 FSID를 가져옵니다.

예제

```
[root@host04 ~]# systemctl list-units | grep ceph
```

- h. Ceph Object Gateway 데몬을 시작합니다.

구문

```
systemctl start ceph-FSID@DAEMON_NAME
systemctl enable ceph-FSID@DAEMON_NAME
```

예제

```
[root@host04 ~]# systemctl start ceph-62a081a6-88aa-11eb-a367-
001a4a000672@rgw.test_realm.us-east-2.host04.ahdtsw.service
[root@host04 ~]# systemctl enable ceph-62a081a6-88aa-11eb-a367-
001a4a000672@rgw.test_realm.us-east-2.host04.ahdtsw.service
```

3. 선택 사항: 배치 사양을 사용하여 다중 사이트 Ceph 오브젝트 게이트웨이를 배포합니다.

구문

```
ceph orch apply rgw NAME --realm=REALM_NAME --zone=PRIMARY_ZONE_NAME --
placement="NUMBER_OF_DAEMONS HOST_NAME_1 HOST_NAME_2"
```

예제

```
[ceph: root@host04 /]# ceph orch apply rgw east --realm=test_realm --zone=us-east-1 --
placement="2 host01 host02"
```

검증

- 동기화 상태를 확인하여 배포를 확인합니다.

예제

```
[ceph: root@host04 /]# radosgw-admin sync status
```

10.5. CEPHORCHESTRATOR를 사용하여 CEPH OBJECT GATEWAY 제거

ceph orch rm 명령을 사용하여 Ceph 개체 게이트웨이 데몬을 제거할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.
- 호스트에 배포된 하나 이상의 Ceph 오브젝트 게이트웨이 데몬.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 서비스를 나열합니다.

예제

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

3. 서비스를 제거합니다.

구문

```
ceph orch rm SERVICE_NAME
```

예제

```
[ceph: root@host01 /]# ceph orch rm rgw.test_realm.test_zone_bb
```

검증

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

```
ceph orch ps
```

예제

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

추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 의 명령줄 인터페이스 섹션을 사용하여 [Ceph 개체 게이트웨이](#) 배포를 참조하십시오.
- 자세한 내용은 [Red Hat Ceph Storage Operations 가이드](#)의 서비스 사양을 사용하여 [Ceph 개체 게이트웨이](#) 배포를 참조하십시오.

11장. CEPH ORCHESTRATOR(LIMITED AVAILABILITY)를 사용한 NFS-GANESHA 게이트웨이 관리

스토리지 관리자는 백엔드에서 Cephadm과 함께 Orchestrator를 사용하여 NFS-Ganesha 게이트웨이를 배포할 수 있습니다. Cephadm은 사전 정의된 RADOS 풀 및 선택적 네임스페이스를 사용하여 NFS Ganesha를 배포합니다.



참고

이 기술은 제한적인 가용성입니다. 자세한 내용은 [사용 중단된 기능](#) 장을 참조하십시오.



참고

Red Hat은 NFS v4.0+ 프로토콜을 통해서만 내보내기를 지원합니다.

이 섹션에서는 다음 관리 작업에 대해 설명합니다.

- [Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터 생성.](#)
- [명령줄 인터페이스를 사용하여 NFS-Ganesha 게이트웨이 배포.](#)
- [서비스 사양을 사용하여 NFS-Ganesha 게이트웨이 배포.](#)
- [CephFS/NFS 서비스에 대한 HA 구현.](#)
- [Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터 업데이트](#)
- [Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터 정보 보기.](#)
- [Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터 로그 가져오기.](#)
- [Ceph Orchestrator를 사용하여 사용자 지정 NFS-Ganesha 구성을 설정합니다.](#)
- [Ceph Orchestrator를 사용하여 사용자 지정 NFS-Ganesha 구성 재설정.](#)
- [Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터를 삭제합니다.](#)
- [Ceph Orchestrator를 사용하여 NFS Ganesha 게이트웨이 제거.](#)

11.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

11.2. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 생성

Ceph Orchestrator의 **mgr/nfs** 모듈을 사용하여 NFS-Ganesha 클러스터를 생성할 수 있습니다. 이 모듈은 백엔드에서 Cephadm을 사용하여 NFS 클러스터를 배포합니다.

이렇게 하면 모든 NFS-Ganesha 데몬, **clusterid** 를 기반으로 하는 새 사용자, 공통 NFS-Ganesha 구성 RADOS 오브젝트에 대한 공통 복구 풀이 생성됩니다.

각 데몬마다 새 사용자와 공통 구성이 풀에 생성됩니다. 모든 클러스터에 클러스터 이름과 관련하여 다른 네임스페이스가 있지만 동일한 복구 풀을 사용합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. **mgr/nfs** 모듈을 활성화합니다.

예제

```
[ceph: root@host01 /]# ceph mgr module enable nfs
```

3. 클러스터를 생성합니다.

구문

```
ceph nfs cluster create CLUSTER_NAME ["HOST_NAME_1 HOST_NAME_2  
HOST_NAME_3"]
```

CLUSTER_NAME 은 임의의 문자열이며 HOST_NAME_1 은 NFS-Ganesha 데몬을 배포하기 위해 호스트를 나타내는 선택적 문자열입니다.

예제

```
[ceph: root@host01 /]# ceph nfs cluster create nfsganesha "host01 host02"  
NFS Cluster Created Successful
```

그러면 **host01** 및 **host02** 에서 하나의 데몬을 사용하여 NFS-Ganesha 클러스터 **nfsganesha** 가 생성됩니다.

검증

- 클러스터 세부 정보를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph nfs cluster ls
```

- NFS-Ganesha 클러스터 정보를 표시합니다.

구문

```
ceph nfs cluster info CLUSTER_NAME
```

예제

```
[ceph: root@host01 /]# ceph nfs cluster info nfsganesha
```

추가 리소스

- 자세한 내용은 Red Hat [Ceph Storage 파일 시스템 가이드의 NFS 프로토콜을 통해 Ceph 파일 시스템 네임스페이스 내보내기](#) 를 참조하십시오.
- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 의 서비스 사양 섹션을 사용하여 Ceph 데몬 배포를 참조하십시오.

11.3. 명령줄 인터페이스를 사용하여 NFS-GANESHA 게이트웨이 배포

백엔드의 Cephadm과 함께 CephOrchestrator를 사용하여 배치 사양을 사용하여 NFS-Ganesha 게이트웨이를 배포할 수 있습니다. 이 경우 게이트웨이를 배포하기 전에 RADOS 풀을 만들고 네임스페이스를 생성해야 합니다.



참고

Red Hat은 NFS v4.0+ 프로토콜을 통해서만 내보내기를 지원합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. RADOS 풀 네임스페이스를 생성하고 애플리케이션을 활성화합니다. RBD 풀의 경우 RBD를 활성화합니다.

구문

```
ceph osd pool create POOL_NAME
ceph osd pool application enable POOL_NAME freeform/rgw/rbd/cephfs/nfs
rbd pool init -p POOL_NAME
```

예제

```
[ceph: root@host01 /]# ceph osd pool create nfs-ganesha
[ceph: root@host01 /]# ceph osd pool application enable nfs-ganesha nfs
[ceph: root@host01 /]# rbd pool init -p nfs-ganesha
```

3. 명령줄 인터페이스의 배치 사양을 사용하여 NFS-Ganesha 게이트웨이를 배포합니다.

구문

```
ceph orch apply nfs SERVICE_ID --placement="NUMBER_OF_DAEMONS HOST_NAME_1
HOST_NAME_2 HOST_NAME_3"
```

예제

```
[ceph: root@host01 /]# ceph orch apply nfs foo --placement="2 host01 host02"
```

이렇게 하면 **host01** 및 **host02** 에서 하나의 데몬을 사용하여 NFS-Ganesha 클러스터 **nfsganesha** 가 배포됩니다.

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

추가 리소스

- 자세한 내용은 Red Hat [Ceph Storage Operations Guide](#) 의 명령행 인터페이스를 사용하여 Ceph 데몬 배포 섹션을 참조하십시오.
- 자세한 내용은 Red Hat Ceph Storage Block Device Guide 의 블록 장치 풀 생성 섹션을 참조하십시오.

11.4. 서비스 사양을 사용하여 NFS-GANESHA 게이트웨이 배포

백엔드의 Cephadm과 함께 CephOrchestrator를 사용하여 서비스 사양을 사용하여 NFS-Ganesha 게이트웨이를 배포할 수 있습니다. 이 경우 게이트웨이를 배포하기 전에 RADOS 풀을 만들고 네임스페이스를 생성해야 합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.

절차

1. **nfs.yaml** 파일을 생성합니다.

예제

```
[root@host01 ~]# touch nfs.yaml
```

2. 다음 세부 정보를 포함하도록 **nfs.yaml** 파일을 편집합니다.

구문

```
service_type: nfs
service_id: SERVICE_ID
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
```

예제

```
service_type: nfs
service_id: foo
placement:
  hosts:
    - host01
    - host02
```

3. 컨테이너의 디렉터리에 YAML 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount nfs.yaml:/var/lib/ceph/nfs.yaml
```

4. RADOS 풀, 네임스페이스를 생성하고 RBD를 활성화합니다.

구문

```
ceph osd pool create POOL_NAME
ceph osd pool application enable POOL_NAME rbd
rbd pool init -p POOL_NAME
```

예제

```
[ceph: root@host01 /]# ceph osd pool create nfs-ganesha
[ceph: root@host01 /]# ceph osd pool application enable nfs-ganesha rbd
[ceph: root@host01 /]# rbd pool init -p nfs-ganesha
```

5. 디렉터리로 이동합니다.

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/
```

6. 서비스 사양을 사용하여 NFS-Ganesha 게이트웨이를 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yaml
```

예제

```
[ceph: root@host01 ceph]# ceph orch apply -i nfs.yaml
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

추가 리소스

- 자세한 내용은 Red Hat Ceph Storage Block Device Guide 의 [블록 장치 풀 생성](#) 섹션을 참조하십시오.

11.5. CEPHFS/NFS 서비스용 HA 구현(기술 프리뷰)

--ingress 플래그를 사용하고 가상 IP 주소를 지정하여 HA(고가용성) 프런트 엔드, 가상 IP 및 로드 밸런서를 사용하여 NFS를 배포할 수 있습니다. 이는 **keepalived** 와 **haproxy** 의 조합을 배포하고 NFS 서비스에 대한 고가용성 NFS 프런트 엔드를 제공합니다.

--ingress 플래그를 사용하여 클러스터를 생성하면 NFS 서버에 대한 로드 밸런싱 및 고가용성을 제공하기 위해 Ingress 서비스가 추가로 배포됩니다. 가상 IP는 모든 NFS 클라이언트가 마운트에 사용할 수 있는 알려진 안정적인 NFS 엔드포인트를 제공하는 데 사용됩니다. Ceph는 가상 IP의 NFS 트래픽을 적절한 백엔드 NFS 서버로 리디렉션하는 세부 정보를 처리하고 실패할 때 NFS 서버를 다시 배포합니다.

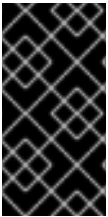
기존 서비스에 대한 Ingress 서비스를 배포하면 다음과 같은 이점이 있습니다.

- NFS 서버에 액세스하는 데 사용할 수 있는 안정적인 가상 IP입니다.
- 여러 NFS 게이트웨이에 대한 로드 배포입니다.
- 호스트 장애 발생 시 호스트 간 장애 조치.



중요

CephFS/NFS의 HA는 기술 프리뷰 기능 전용입니다. 기술 프리뷰 기능은 Red Hat 프로덕션 서비스 수준 계약(SLA)에서 지원되지 않으며 기능적으로 완전하지 않을 수 있으며 Red Hat은 프로덕션에 사용하지 않는 것이 좋습니다. 이러한 기능을 사용하면 향후 제품 기능을 조기에 이용할 수 있어 개발 과정에서 고객이 기능을 테스트하고 피드백을 제공할 수 있습니다. 자세한 내용은 [Red Hat 기술 프리뷰](#) 기능의 지원 범위를 참조하십시오.



중요

Ingress 서비스가 NFS 클러스터 앞에 배포되면 백엔드 NFS-ganesha 서버에서 클라이언트의 IP 주소가 아닌 haproxy의 IP 주소를 확인합니다. 결과적으로 IP 주소를 기반으로 클라이언트 액세스를 제한하는 경우 NFS 내보내기에 대한 액세스 제한이 예상대로 작동하지 않습니다.



참고

클라이언트를 제공하는 활성 NFS 서버가 다운되면 활성 NFS 서버를 교체하고 NFS 클러스터가 다시 활성 상태가 될 때까지 클라이언트의 I/O가 중단됩니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 manager, monitor, OSD 데몬이 배포됩니다.
- NFS 모듈이 활성화되었는지 확인합니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. **--ingress** 플래그를 사용하여 NFS 클러스터를 생성합니다.

구문

```
ceph nfs cluster create CLUSTER_ID [PLACEMENT] [--port PORT_NUMBER] [--ingress --virtual-ip IP_ADDRESS/CIDR_PREFIX]
```

- NFS Ganesha 클러스터 이름을 지정하려면 CLUSTER_ID 를 고유한 문자열로 바꿉니다.

- **PLACEMENT** 를 배포할 NFS 서버 수 및 NFS Ganesha 데몬 컨테이너를 배포하려는 호스트 또는 호스트 수로 교체합니다.
- **--port** **PORT_NUMBER** 플래그를 사용하여 2049의 기본 포트가 아닌 포트에 NFS를 배포합니다.
- **--virtual-ip** 플래그와 결합된 **--ingress** 플래그는 고가용성 프런트 엔드(가상 IP 및 로드 밸런서)를 사용하여 NFS를 배포합니다.
- **--virtual-ip** **IP_ADDRESS** 를 IP 주소로 교체하여 모든 클라이언트가 NFS 내보내기를 마운트하는 데 사용할 수 있는 알려진 안정적인 NFS 엔드포인트를 제공합니다. **--virtual-ip** 에는 CIDR 접두사 길이를 포함해야 합니다. 가상 IP는 일반적으로 동일한 서브넷에 있는 기존 IP가 있는 첫 번째 식별 네트워크 인터페이스에 구성됩니다.



참고

NFS 서비스에 할당된 호스트 수는 **deploy : count** 매개변수로 지정하는 활성 NFS 서버 수보다 커야 합니다. 아래 예제에서는 하나의 활성 NFS 서버가 요청되고 두 개의 호스트가 할당됩니다.

예제

```
[ceph: root@host01 /]# ceph nfs cluster create mycephnfs "1 host02,host03" --ingress --virtual-ip 10.10.128.75/22
```



참고

NFS 데몬과 수신 서비스는 비동기적으로 배포되고 서비스가 완전히 시작되기 전에 명령이 반환됩니다.

3. 서비스가 성공적으로 시작되었는지 확인합니다.

구문

```
ceph orch ls --service_name=nfs.CLUSTER_ID
ceph orch ls --service_name=ingress.nfs.CLUSTER_ID
```

예제

```
[ceph: root@host01 /]# ceph orch ls --service_name=nfs.mycephnfs
```

NAME	PORTS	RUNNING	REFRESHED	AGE	PLACEMENT
nfs.mycephnfs	?:12049	1/2	0s ago	20s	host02;host03

```
[ceph: root@host01 /]# ceph orch ls --service_name=ingress.nfs.mycephnfs
```

NAME	PORTS	RUNNING	REFRESHED	AGE	PLACEMENT
ingress.nfs.mycephnfs	10.10.128.75:2049,9049	4/4	46s ago	73s	count:2

검증

- **Ingress** 서비스의 IP 끝점, 개별 NFS 데몬의 IP, 가상 IP를 확인합니다.

구문

```
ceph nfs cluster info CLUSTER_ID
```

예제

```
[ceph: root@host01 /]# ceph nfs cluster info mycephnfs
```

```
{
  "mycephnfs": {
    "virtual_ip": "10.10.128.75",
    "backend": [
      {
        "hostname": "host02",
        "ip": "10.10.128.69",
        "port": 12049
      },
      {
        "hostname": "host03",
        "ip": "10.10.128.70",
        "port": 12049
      }
    ],
    "port": 2049,
    "monitor_port": 9049
  }
}
```

- 호스트 및 프로세스를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph orch ps | grep nfs
```

```
haproxy.nfs.cephnfs.host01.rftylv host01 *:2049,9000 running (11m) 10m ago 11m
23.2M - 2.2.19-7ea3822 5e6a41d77b38 f8cc61dc827e
haproxy.nfs.cephnfs.host02.zhtded host02 *:2049,9000 running (11m) 53s ago 11m
21.3M - 2.2.19-7ea3822 5e6a41d77b38 4cad324e0e23
keepalived.nfs.cephnfs.host01.zktmsk host01 running (11m) 10m ago 11m
2349k - 2.1.5 18fa163ab18f 66bf39784993
keepalived.nfs.cephnfs.host02.vyycvp host02 running (11m) 53s ago 11m
2349k - 2.1.5 18fa163ab18f 1ecc95a568b4
nfs.cephnfs.0.0.host02.fescmw host02 *:12049 running (14m) 3m ago 14m
76.9M - 3.5 cef6e7959b0a bb0e4ee9484e
nfs.cephnfs.1.0.host03.avaddf host03 *:12049 running (14m) 3m ago 14m
74.3M - 3.5 cef6e7959b0a ea02c0c50749
```

추가 리소스

- 클라이언트 호스트에서 NFS 내보내기를 마운트하는 방법에 대한 자세한 내용은 [Red Hat Ceph Storage 파일 시스템 가이드의 NFS 프로토콜을 통해 Ceph 파일 시스템 네임스페이스 내보내기 섹션](#)을 참조하십시오.

11.6. HA용 독립 실행형 CEPHFS/NFS 클러스터 업그레이드

스토리지 관리자는 기존 NFS 서비스에 수신 서비스를 배포하여 독립 실행형 스토리지 클러스터를 HA(고가용성) 클러스터로 업그레이드할 수 있습니다.

사전 요구 사항

- 기존 NFS 서비스를 사용하여 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 manager, monitor, OSD 데몬이 배포됩니다.
- NFS 모듈이 활성화되었는지 확인합니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 기존 NFS 클러스터를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph nfs cluster ls  
  
mynfs
```

참고

독립 실행형 NFS 클러스터가 하나의 노드에 생성되는 경우 HA의 두 개 이상의 노드로 늘려야 합니다. NFS 서비스를 늘리려면 **nfs.yaml** 파일을 편집하고 동일한 포트 번호로 배치를 늘립니다.

NFS 서비스에 할당된 호스트 수는 **deploy : count** 매개변수로 지정하는 활성 NFS 서버 수보다 커야 합니다.

구문

```
service_type: nfs
service_id: SERVICE_ID
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
  count: COUNT
spec:
  port: PORT_NUMBER
```

예제

```
service_type: nfs
service_id: mynfs
placement:
  hosts:
    - host02
    - host03
  count: 1
spec:
  port: 12345
```

이 예에서는 기존 NFS 서비스가 포트 **12345** 에서 실행되고 있으며 동일한 포트가 있는 NFS 클러스터에 추가 노드가 추가됩니다.

nfs.yaml 서비스 사양 변경 사항을 적용하여 두 개의 노드 NFS 서비스로 업그레이드합니다.

예제

```
[ceph: root@host01 ceph]# ceph orch apply -i nfs.yaml
```

3. 기존 NFS 클러스터 ID를 사용하여 **ingress.yaml** 사양 파일을 편집합니다.

구문

```
service_type: SERVICE_TYPE
service_id: SERVICE_ID
placement:
  count: PLACEMENT
spec:
  backend_service: SERVICE_ID_BACKEND 1
```

```
frontend_port: FRONTEND_PORT
monitor_port: MONITOR_PORT 2
virtual_ip: VIRTUAL_IP_WITH_CIDR
```

- 1 SERVICE_ID_BACKEND 에는 동일한 호스트에 배치될 수 있는 수신 서비스와 충돌하지 않도록 2049 가 아닌 PORT 속성이 포함되어야 합니다.
- 2 MONITOR_PORT 는 haproxy 부하 상태 페이지에 액세스하는 데 사용됩니다.

예제

```
service_type: ingress
service_id: nfs.mynfs
placement:
  count: 2
spec:
  backend_service: nfs.mynfs
  frontend_port: 2049
  monitor_port: 9000
  virtual_ip: 10.10.128.75/22
```

4. 수신 서비스를 배포합니다.

예제

```
[ceph: root@host01 /]# ceph orch apply -i ingress.yaml
```



참고

NFS 데몬과 수신 서비스는 비동기적으로 배포되고 서비스가 완전히 시작되기 전에 명령이 반환됩니다.

5. 수신 서비스가 성공적으로 시작되었는지 확인합니다.

구문

```
ceph orch ls --service_name=ingress.nfs.CLUSTER_ID
```

예제

```
[ceph: root@host01 /]# ceph orch ls --service_name=ingress.nfs.mynfs
```

NAME	PORTS	RUNNING	REFRESHED	AGE	PLACEMENT
ingress.nfs.mynfs	10.10.128.75:2049,9000	4/4	4m ago	22m	count:2

검증

- **Ingress** 서비스의 IP 끝점, 개별 **NFS** 데몬의 IP, 가상 IP를 확인합니다.

구문

```
ceph nfs cluster info CLUSTER_ID
```

예제

```
[ceph: root@host01 /]# ceph nfs cluster info mynfs
```

```
{
  "mynfs": {
    "virtual_ip": "10.10.128.75",
    "backend": [
      {
        "hostname": "host02",
        "ip": "10.10.128.69",
        "port": 12049
      },
      {
        "hostname": "host03",
        "ip": "10.10.128.70",
        "port": 12049
      }
    ],
    "port": 2049,
    "monitor_port": 9049
  }
}
```

- 호스트 및 프로세스를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph orch ps | grep nfs
```

```
haproxy.nfs.mynfs.host01.ruyyhq host01 *:2049,9000 running (27m) 6m ago 34m
9.85M - 2.2.19-7ea3822 5e6a41d77b38 328d27b3f706
haproxy.nfs.mynfs.host02.ctrhha host02 *:2049,9000 running (34m) 6m ago 34m
4944k - 2.2.19-7ea3822 5e6a41d77b38 4f4440dbfde9
keepalived.nfs.mynfs.host01.fqgjxd host01 running (27m) 6m ago 34m 31.2M
- 2.1.5 18fa163ab18f 0e22b2b101df
keepalived.nfs.mynfs.host02.fqzkbx host02 running (34m) 6m ago 34m 17.5M
- 2.1.5 18fa163ab18f c1e3cc074cf8
nfs.mynfs.0.0.host02.emoaut host02 *:12345 running (37m) 6m ago 37m 82.7M
- 3.5 91322de4f795 2d00faaa2ae5
nfs.mynfs.1.0.host03.nsxcfd host03 *:12345 running (37m) 6m ago 37m 81.1M
- 3.5 91322de4f795 d4bda4074f17
```

추가 리소스

- 클라이언트 호스트에서 NFS 내보내기를 마운트하는 방법에 대한 자세한 내용은 [Red Hat Ceph Storage 파일 시스템 가이드의 NFS 프로토콜을 통해 Ceph 파일 시스템 네임스페이스 내보내기 섹션](#)을 참조하십시오.

11.7. 사양 파일을 사용하여 CEPHFS/NFS에 대한 HA 배포

먼저 NFS 서비스를 배포한 다음 동일한 NFS 서비스에 Ingress를 배포하여 사양 파일을 사용하여 CephFS/NFS의 **HA**를 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 manager, monitor, OSD 데몬이 배포됩니다.
- NFS 모듈이 활성화되었는지 확인합니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. NFS 모듈이 활성화되었는지 확인합니다.

예제

```
[ceph: root@host01 /]# ceph mgr module ls | more
```

3. Cephadm 셸을 종료하고 **nfs.yaml** 파일을 생성합니다.

예제

```
[root@host01 ~]# touch nfs.yaml
```

4. 다음 세부 정보를 포함하도록 **nfs.yaml** 파일을 편집합니다.

구문

```
service_type: nfs
service_id: SERVICE_ID
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
  count: COUNT
spec:
  port: PORT_NUMBER
```



참고

NFS 서비스에 할당된 호스트 수는 **deploy : count** 매개변수로 지정하는 활성 NFS 서버 수보다 커야 합니다.

예제

```

service_type: nfs
service_id: cephfsnfs
placement:
  hosts:
    - host02
    - host03
  count: 1
spec:
  port: 12345

```

이 예에서 서버는 host02 및 host03에서 기본 포트가 **2049** 인 기본 포트가 아니라 **12345** 의 non-default 포트에서 실행됩니다.

- 컨테이너의 디렉터리에 YAML 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount nfs.yaml:/var/lib/ceph/nfs.yaml
```

- Cephadm 셸에 로그인하고 디렉터리로 이동합니다.

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/
```

- 서비스 사양을 사용하여 NFS 서비스를 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yaml
```

예제

```
[ceph: root@host01 ceph]# ceph orch apply -i nfs.yaml
```



참고

NFS 서비스의 배포는 비동기적으로 배포되고 서비스가 완전히 시작되기 전에 명령이 반환될 수 있습니다.

- NFS 서비스가 성공적으로 시작되었는지 확인합니다.

구문

```
ceph orch ls --service_name=nfs.CLUSTER_ID
```

예제

```
[ceph: root@host01 /]# ceph orch ls --service_name=nfs.cephfsnfs
```

NAME	PORTS	RUNNING	REFRESHED	AGE	PLACEMENT
------	-------	---------	-----------	-----	-----------

```
nfs.cephfsnfs ? :12345 2/2 3m ago 13m host02;host03
```

9. Cephadm 셸을 종료하고 **ingress.yaml** 파일을 생성합니다.

예제

```
[root@host01 ~]# touch ingress.yaml
```

10. 다음 세부 사항을 포함하도록 **ingress.yaml** 파일을 편집합니다.

구문

```
service_type: SERVICE_TYPE
service_id: SERVICE_ID
placement:
  count: PLACEMENT
spec:
  backend_service: SERVICE_ID_BACKEND
  frontend_port: FRONTEND_PORT
  monitor_port: MONITOR_PORT
  virtual_ip: VIRTUAL_IP_WITH_CIDR
```

예제

```
service_type: ingress
service_id: nfs.cephfsnfs
placement:
  count: 2
spec:
  backend_service: nfs.cephfsnfs
  frontend_port: 2049
  monitor_port: 9000
  virtual_ip: 10.10.128.75/22
```

참고

이 예에서는 **placement: count**입니다. 2는 **keepalived** 및 **haproxy** 서비스를 임의의 노드에 배포합니다. **keepalived** 및 **haproxy**를 배포할 노드를 지정하려면 **placement: hosts** 옵션을 사용합니다.

예제

```
service_type: ingress
service_id: nfs.cephfsnfs
placement:
  hosts:
    - host02
    - host03
```

11. 컨테이너의 디렉터리에 YAML 파일을 마운트합니다.

예제

■

```
[root@host01 ~]# cephadm shell --mount ingress.yaml:/var/lib/ceph/ingress.yaml
```

12. Cephadm 셸에 로그인하고 디렉터리로 이동합니다.

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/
```

13. 서비스 사양을 사용하여 수신 서비스를 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yaml
```

예제

```
[ceph: root@host01 ceph]# ceph orch apply -i ingress.yaml
```

14. 수신 서비스가 성공적으로 시작되었는지 확인합니다.

구문

```
ceph orch ls --service_name=ingress.nfs.CLUSTER_ID
```

예제

```
[ceph: root@host01 /]# ceph orch ls --service_name=ingress.nfs.cephfsnfs
```

NAME	PORTS	RUNNING	REFRESHED	AGE	PLACEMENT
ingress.nfs.cephfsnfs	10.10.128.75:2049,9000	4/4	4m ago	22m	count:2

검증

- **Ingress** 서비스의 IP 끝점, 개별 **NFS** 데몬의 IP, 가상 IP를 확인합니다.

구문

```
ceph nfs cluster info CLUSTER_ID
```

예제

```
[ceph: root@host01 /]# ceph nfs cluster info cephfsnfs
```

```
{
  "cephfsnfs": {
    "virtual_ip": "10.10.128.75",
    "backend": [
      {
        "hostname": "host02",
        "ip": "10.10.128.69",
        "port": 12345
      }
    ]
  }
}
```

```
{
  "hostname": "host03",
  "ip": "10.10.128.70",
  "port": 12345
},
"port": 2049,
"monitor_port": 9049
}
```

- 호스트 및 프로세스를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph orch ps | grep nfs
```

```
haproxy.nfs.cephfsnfs.host01.ruyyhq host01 *:2049,9000 running (27m) 6m ago 34m
9.85M - 2.2.19-7ea3822 5e6a41d77b38 328d27b3f706
haproxy.nfs.cephfsnfs.host02.ctrhha host02 *:2049,9000 running (34m) 6m ago 34m
4944k - 2.2.19-7ea3822 5e6a41d77b38 4f4440dbfde9
keepalived.nfs.cephfsnfs.host01.fqgjxd host01 running (27m) 6m ago 34m
31.2M - 2.1.5 18fa163ab18f 0e22b2b101df
keepalived.nfs.cephfsnfs.host02.fqzqxb host02 running (34m) 6m ago 34m
17.5M - 2.1.5 18fa163ab18f c1e3cc074cf8
nfs.cephfsnfs.0.0.host02.emoaut host02 *:12345 running (37m) 6m ago 37m
82.7M - 3.5 91322de4f795 2d00faaa2ae5
nfs.cephfsnfs.1.0.host03.nsxcd host03 *:12345 running (37m) 6m ago 37m
81.1M - 3.5 91322de4f795 d4bda4074f17
```

추가 리소스

- 클라이언트 호스트에서 NFS 내보내기를 마운트하는 방법에 대한 자세한 내용은 [Red Hat Ceph Storage 파일 시스템 가이드의 NFS 프로토콜을 통해 Ceph 파일 시스템 네임스페이스 내보내기 섹션](#)을 참조하십시오.

11.8. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 업데이트

백엔드에서 Cephadm과 함께 Cephadm을 사용하여 호스트에서 데몬 배치를 변경하여 NFS-Ganesha 클러스터를 업데이트할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.
- **mgr/nfs** 모듈을 사용하여 생성된 NFS-Ganesha 클러스터입니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 클러스터를 업데이트합니다.

구문

```
ceph orch apply nfs CLUSTER_NAME ["HOST_NAME_1,HOST_NAME_2,HOST_NAME_3"]
```

CLUSTER_NAME 은 임의의 문자열이며 HOST_NAME_1 은 배포된 NFS-Ganesha 데몬을 업데이트하기 위해 호스트를 나타내는 선택적 문자열입니다.

예제

```
[ceph: root@host01 /]# ceph orch apply nfs nfsganesha "host02"
```

그러면 **host02** 에서 **nfsganesha** 클러스터가 업데이트됩니다.

검증

- 클러스터 세부 정보를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph nfs cluster ls
```

- NFS-Ganesha 클러스터 정보를 표시합니다.

구문

```
ceph nfs cluster info CLUSTER_NAME
```

예제

```
[ceph: root@host01 /]# ceph nfs cluster info nfsganesha
```

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

추가 리소스

- 자세한 내용은 Red Hat Ceph Storage 운영 가이드 [의 Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터 생성](#) 섹션을 참조하십시오.

11.9. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 정보 보기

Ceph intended orchestrator를 사용하여 NFS-Ganesha 클러스터의 정보를 볼 수 있습니다. 클러스터가 생성되는 호스트의 이름, 포트, IP 주소 및 특정 클러스터에서 모든 NFS Ganesha 클러스터 또는 특정 클러스터에 대한 정보를 가져올 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.
- **mgr/nfs** 모듈을 사용하여 생성된 NFS-Ganesha 클러스터입니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. NFS-Ganesha 클러스터 정보를 확인합니다.

구문

```
ceph nfs cluster info CLUSTER_NAME
```

예제

```
[ceph: root@host01 /]# ceph nfs cluster info nfsganesha
```

```
{
  "nfsganesha": [
    {
      "hostname": "host02",
      "ip": [
        "10.10.128.70"
      ],
      "port": 2049
    }
  ]
}
```

추가 리소스

- 자세한 내용은 Red Hat Ceph Storage 운영 가이드 [의 Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터 생성](#) 섹션을 참조하십시오.

11.10. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 로그 가져오기

Ceph Orchestrator를 사용하면 NFS-Ganesha 클러스터 로그를 가져올 수 있습니다. 서비스가 배포된 노드에 있어야 합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- NFS가 배포된 노드에 Cephadm이 설치되어 있습니다.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.
- mgr/nfs** 모듈을 사용하여 생성된 NFS-Ganesha 클러스터입니다.

절차

1. root 사용자로 스토리지 클러스터의 FSID 를 가져옵니다.

예제

```
[root@host03 ~]# cephadm ls
```

FSID 및 서비스 이름을 복사합니다.

2. 로그를 가져옵니다.

구문

```
cephadm logs --fsid FSID --name SERVICE_NAME
```

예제

```
[root@host03 ~]# cephadm logs --fsid 499829b4-832f-11eb-8d6d-001a4a000635 --name nfs.foo.host03
```

추가 리소스

- 자세한 내용은 Red Hat [Ceph Storage Operations Guide](#)의 [배치 사양을 사용하여 Ceph 데몬 배포](#)를 참조하십시오.

11.11. CEPH ORCHESTRATOR를 사용하여 사용자 지정 NFS-GANESHA 구성 설정

NFS-Ganesha 클러스터는 기본 구성 블록으로 정의됩니다. Ceph Orchestrator를 사용하면 구성을 사용자 지정할 수 있으며 기본 구성 블록보다 우선합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.
- **mgr/nfs** 모듈을 사용하여 생성된 NFS-Ganesha 클러스터입니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 다음은 NFS-Ganesha 클러스터의 기본 구성의 예입니다.

예제

```
# {{ cephadm_managed }}
NFS_CORE_PARAM {
    Enable_NLM = false;
    Enable_RQUOTA = false;
    Protocols = 4;
}

MDCACHE {
    Dir_Chunk = 0;
}

EXPORT_DEFAULTS {
    Attr_Expiration_Time = 0;
}

NFSv4 {
    Delegations = false;
    RecoveryBackend = 'rados_cluster';
    Minor_Versions = 1, 2;
}

RADOS_KV {
    UserId = "{{ user }}";
    nodeid = "{{ nodeid }}";
    pool = "{{ pool }}";
    namespace = "{{ namespace }}";
}

RADOS_URLS {
    UserId = "{{ user }}";
    watch_url = "{{ url }}";
}
```

```
RGW {
    cluster = "ceph";
    name = "client.{{ rgw_user }}";
}

%url {{ url }}
```

3. NFS-Ganesha 클러스터 구성을 사용자 지정합니다. 다음은 구성을 사용자 지정하는 두 가지 예입니다.

- 로그 수준을 변경합니다.

예제

```
LOG {
    COMPONENTS {
        ALL = FULL_DEBUG;
    }
}
```

- 사용자 정의 내보내기 블록 추가:

- a. 사용자를 만듭니다.



참고

FSAL 블록에 지정된 사용자는 NFS-Ganesha 데몬에서 Ceph 클러스터에 액세스할 수 있는 적절한 제한이 있어야 합니다.

구문

```
ceph auth get-or-create client.USER_ID mon 'allow r' osd 'allow rw pool=.nfs
namespace=NFS_CLUSTER_NAME, allow rw tag cephfs data=FS_NAME' mds
'allow rw path=EXPORT_PATH'
```

예제

```
[ceph: root@host01 /]# ceph auth get-or-create client.f64f341c-655d-11eb-8778-
fa163e914bcc mon 'allow r' osd 'allow rw pool=.nfs namespace=nfs_cluster_name,
allow rw tag cephfs data=fs_name' mds 'allow rw path=export_path'
```

- b. 다음 디렉터리로 이동합니다.

구문

```
cd /var/lib/ceph/DAEMON_PATH/
```

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/nfs/
```

nfs 디렉터리가 없는 경우 경로에 디렉터를 생성합니다.

- c. 새 구성 파일을 생성합니다.

구문

```
touch PATH_TO_CONFIG_FILE
```

예제

```
[ceph: root@host01 nfs]# touch nfs-ganesha.conf
```

- d. 사용자 지정 내보내기 블록을 추가하여 구성 파일을 편집합니다. 단일 내보내기를 생성하고 Ceph NFS 내보내기 인터페이스에서 관리합니다.

구문

```
EXPORT {
    Export_Id = NUMERICAL_ID;
    Transports = TCP;
    Path = PATH_WITHIN_CEPHFS;
    Pseudo = BINDING;
    Protocols = 4;
    Access_Type = PERMISSIONS;
    Attr_Expiration_Time = 0;
    Squash = None;
    FSAL {
        Name = CEPH;
        Filesystem = "FILE_SYSTEM_NAME";
        User_Id = "USER_NAME";
        Secret_Access_Key = "USER_SECRET_KEY";
    }
}
```

예제

```
EXPORT {
    Export_Id = 100;
    Transports = TCP;
    Path = /;
    Pseudo = /ceph/;
    Protocols = 4;
    Access_Type = RW;
    Attr_Expiration_Time = 0;
    Squash = None;
    FSAL {
        Name = CEPH;
        Filesystem = "filesystem name";
        User_Id = "user id";
        Secret_Access_Key = "secret key";
    }
}
```

4. 클러스터에 새 구성을 적용합니다.

구문

```
ceph nfs cluster config set CLUSTER_NAME -i PATH_TO_CONFIG_FILE
```

예제

```
[ceph: root@host01 nfs]# ceph nfs cluster config set nfs-ganesha -i /var/lib/ceph/nfs/nfs-ganesha.conf
```

또한 사용자 지정 구성을 위해 서비스가 다시 시작됩니다.

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

- 사용자 지정 구성을 확인합니다.

구문

```
/bin/rados -p POOL_NAME -N CLUSTER_NAME get userconf-nfs.CLUSTER_NAME -
```

예제

```
[ceph: root@host01 /]# /bin/rados -p nfs-ganesha -N nfsganesha get userconf-nfs.nfsganesha -
```

추가 리소스

- 자세한 내용은 Red Hat Ceph Storage Operations Guide [의 Ceph intended orchestrator](#)를 사용하여 사용자 지정 NFS-Ganesha 구성 재설정 섹션을 참조하십시오.

11.12. CEPHORCHESTRATOR를 사용하여 사용자 정의 NFS-GANESHA 구성 재설정

Ceph Orchestrator를 사용하여 사용자 정의 구성을 기본 구성으로 재설정할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.
- **mgr/nfs** 모듈을 사용하여 배포된 NFS-Ganesha.
- 사용자 지정 NFS 클러스터 구성이 설정됨

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. NFS-Ganesha 구성을 재설정합니다.

구문

```
ceph nfs cluster config reset CLUSTER_ID
```

예제

```
[ceph: root@host01 /]# ceph nfs cluster config reset f64f341c-655d-11eb-8778-fa163e914bcc
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

- 사용자 지정 구성이 삭제되었는지 확인합니다.

구문

```
/bin/rados -p POOL_NAME -N CLUSTER_NAME get userconf-nfs.CLUSTER_NAME -
```

예제

```
[ceph: root@host01 /]# /bin/rados -p nfs-ganesha -N nfsganesha get userconf-  
nfs.nfsganesha -
```

추가 리소스

- 자세한 내용은 Red Hat Ceph Storage 운영 가이드 [의 Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터 생성](#) 섹션을 참조하십시오.
- 자세한 내용은 Red Hat Ceph Storage 운영 가이드 [의 Ceph Orchestrator를 사용하여 사용자 지정 NFS-Ganesha 구성](#) 설정을 참조하십시오.

11.13. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 클러스터 삭제

백엔드의 Cephadm과 함께 Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터를 삭제할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.
- **mgr/nfs** 모듈을 사용하여 생성된 NFS-Ganesha 클러스터입니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 클러스터를 삭제합니다.

구문

```
ceph nfs cluster delete CLUSTER_NAME
```

CLUSTER_NAME 은 임의의 문자열입니다.

예제

```
[ceph: root@host01 /]# ceph nfs cluster delete nfsganesha  
NFS Cluster Deleted Successfully
```

검증

- 클러스터 세부 정보를 나열합니다.

예제

```
[ceph: root@host01 /]# ceph nfs cluster ls
```

추가 리소스

- 자세한 내용은 Red Hat [Ceph Storage File System 가이드](#)의 NFS 프로토콜을 통해 Ceph 파일 시스템 내보내기 섹션을 참조하십시오.
- 자세한 내용은 Red Hat Ceph Storage 운영 가이드 의 [Ceph Orchestrator를 사용하여 NFS-Ganesha 클러스터 생성](#) 섹션을 참조하십시오.

11.14. CEPH ORCHESTRATOR를 사용하여 NFS-GANESHA 게이트웨이 제거

ceph orch rm 명령을 사용하여 NFS-Ganesha 게이트웨이를 제거할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.
- 호스트에 배포된 하나 이상의 NFS-Ganesha 게이트웨이.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 서비스를 나열합니다.

예제

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

3. 서비스를 제거합니다.

구문

```
ceph orch rm SERVICE_NAME
```

예제

```
[ceph: root@host01 /]# ceph orch rm nfs.foo
```

검증

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

```
ceph orch ps
```

예제

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

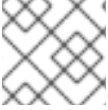
추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 의 서비스 사양 섹션을 사용하여 Ceph 데몬 배포를 참조하십시오.
- 자세한 내용은 [Red Hat Ceph Storage Operations 가이드](#) 의 서비스 사양을 사용하여 NFS-Ganesha 게이트웨이 배포를 참조하십시오.

12 장. CEPH ORCHESTRATOR (LIMITED AVAILABILITY) 를 사용한 ISCSI 게이트웨이 관리

스토리지 관리자는 Ceph Orchestrator를 사용하여 iSCSI 게이트웨이를 배포할 수 있습니다. iSCSI 게이트웨이는 RADOS 블록 장치(RBD) 이미지를 SCSI 디스크로 내보내는 고가용성(HA) iSCSI 대상을 제공합니다.

배치 사양 또는 YAML 파일과 같은 서비스 사양을 사용하여 iSCSI 게이트웨이를 배포할 수 있습니다.



참고

이 기술은 제한적인 가용성입니다. 자세한 내용은 [사용 중단된 기능](#) 장을 참조하십시오.

이 섹션에서는 다음 관리 작업에 대해 설명합니다.

- [배치 사양을 사용하여 iSCSI 게이트웨이 배포.](#)
- [서비스 사양을 사용하여 iSCSI 게이트웨이 배포.](#)
- [Ceph Orchestrator를 사용하여 iSCSI 게이트웨이 제거.](#)

12.1. 사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

12.2. 명령줄 인터페이스를 사용하여 ISCSI 게이트웨이 배포

Ceph Orchestrator를 사용하면 명령줄 인터페이스에서 **ceph orch** 명령을 사용하여 iSCSI 게이트웨이를 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 풀을 생성합니다.

구문

```
ceph osd pool create POOL_NAME
```

예제

```
[ceph: root@host01 /]# ceph osd pool create mypool
```

3. 명령줄 인터페이스를 사용하여 iSCSI 게이트웨이를 배포합니다.

구문

```
ceph orch apply iscsi POOLNAME admin admin --placement="NUMBER_OF_DAEMONS  
HOST_NAME_1 HOST_NAME_2"
```

예제

```
[ceph: root@host01 /]# ceph orch apply iscsi mypool admin admin --placement="1 host01"
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=iscsi
```

12.3. 서비스 사양을 사용하여 iSCSI 게이트웨이 배포

Ceph Orchestrator를 사용하면 서비스 사양을 사용하여 iSCSI 게이트웨이를 배포할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 호스트는 클러스터에 추가됩니다.
- 모든 관리자, 모니터 및 OSD 데몬이 배포됩니다.

절차

1. **iscsi.yml** 파일을 생성합니다.

예제

```
[root@host01 ~]# touch iscsi.yml
```

2. 다음 세부 사항을 포함하도록 **iscsi.yml** 파일을 편집합니다.

구문

```
service_type: iscsi
service_id: iscsi
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
spec:
  pool: POOL_NAME # RADOS pool where ceph-iscsi config data is stored.
  trusted_ip_list: "IP_ADDRESS_1,IP_ADDRESS_2" # optional
  api_port: ... # optional
  api_user: API_USERNAME # optional
  api_password: API_PASSWORD # optional
  api_secure: true/false # optional
  ssl_cert: | # optional
  ...
  ssl_key: | # optional
  ...
```

예제

```
service_type: iscsi
service_id: iscsi
placement:
  hosts:
    - host01
spec:
  pool: mypool
```

3. 컨테이너의 디렉터리에 YAML 파일을 마운트합니다.

예제

```
[root@host01 ~]# cephadm shell --mount iscsi.yaml:/var/lib/ceph/iscsi.yaml
```

4. 다음 디렉터리로 이동합니다.

구문

```
cd /var/lib/ceph/DAEMON_PATH/
```

예제

```
[ceph: root@host01 /]# cd /var/lib/ceph/
```

5. 서비스 사양을 사용하여 iSCSI 게이트웨이를 배포합니다.

구문

```
ceph orch apply -i FILE_NAME.yml
```

예제

```
[ceph: root@host01 iscsi]# ceph orch apply -i iscsi.yml
```

검증

- 서비스를 나열합니다.

예제

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

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=iscsi
```

12.4. CEPHORCHESTRATOR를 사용하여 iSCSI 게이트웨이 제거

ceph orch rm 명령을 사용하여 iSCSI 게이트웨이 데몬을 제거할 수 있습니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 모든 노드에 대한 루트 수준의 액세스.
- 호스트는 클러스터에 추가됩니다.
- 호스트에 배포된 하나 이상의 iSCSI 게이트웨이 데몬.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. 서비스를 나열합니다.

예제

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

- a. 서비스 제거

구문

```
ceph orch rm SERVICE_NAME
```

예제

```
[ceph: root@host01 /]# ceph orch rm iscsi.iscsi
```

검증

- 호스트, 데몬 및 프로세스를 나열합니다.

구문

```
ceph orch ps
```

예제

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

추가 리소스

- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 의 명령행 인터페이스를 사용하여 iSCSI 게이트웨이 배포를 참조하십시오.
- 자세한 내용은 [Red Hat Ceph Storage Operations Guide](#) 의 서비스 사양을 사용하여 iSCSI 게이트웨이 배포를 참조하십시오.

13장. SNMP 트랩 설정

스토리지 관리자는 Red Hat Ceph Storage 클러스터에 단순 네트워크 관리 프로토콜(SNMP) 게이트웨이를 배포 및 구성하여 Prometheus Alertmanager에서 경고를 수신하고 이를 SNMP 트랩으로 클러스터로 라우팅할 수 있습니다.

13.1. 간단한 네트워크 관리 프로토콜

SNMP(Simple Network Management Protocol)는 다양한 하드웨어 및 소프트웨어 플랫폼에서 분산 시스템 및 장치를 모니터링하기 위해 가장 널리 사용되는 오픈 프로토콜 중 하나입니다. Ceph의 SNMP 통합은 Prometheus Alertmanager 클러스터에서 게이트웨이 데몬으로 경고를 전달하는 데 중점을 둡니다. 게이트웨이 데몬은 경고를 SNMP 알람으로 변환하여 지정된 SNMP 관리 플랫폼으로 보냅니다. 게이트웨이 데몬은 **snmp_notifier_project**에서 가져온데, 이 프로젝트에서는 SNMP V2c 및 V3에서 인증 및 암호화를 지원합니다.

Red Hat Ceph Storage SNMP 게이트웨이 서비스는 기본적으로 하나의 게이트웨이 인스턴스를 배포합니다. 배치 정보를 제공하여 이를 늘릴 수 있습니다. 그러나 하나의 SNMP 게이트웨이 데몬을 활성화하는 경우, SNMP 관리 플랫폼은 동일한 이벤트에 대해 여러 개의 알람을 수신합니다.

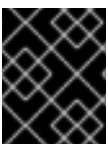
SNMP 트랩은 경고 메시지이며 Prometheus Alertmanager는 이러한 경고를 SNMP 알람에 보낸 다음 지정된 경고의 라벨에서 개체 식별자(OID)를 찾습니다. 각 SNMP 트랩에는 고유 ID가 있어 업데이트된 상태로 추가 트랩을 지정된 TPM 폴링자에게 보낼 수 있습니다. 모든 상태 경고가 특정 SNMP 트랩을 생성할 수 있도록 Ceph 상태 점검에 있는 SNMP 후크.

장치를 올바르게 작동하고 장치 상태에 대한 정보를 모니터링할 사용자로 전송하려면 SNMP는 여러 구성 요소를 사용합니다. SNMP를 구성하는 네 가지 주요 구성 요소가 있습니다.

- **SNMP Manager**- 관리 스테이션이라고도 하는 SNMP 관리자는 네트워크 모니터링 플랫폼을 실행하는 컴퓨터입니다. SNMP 지원 장치를 폴링하고 데이터를 검색하는 플랫폼이 있습니다. SNMP Manager 쿼리 에이전트는 에이전트에서 응답을 수신하고 에이전트에서 비동기 이벤트를 확인합니다.
- **SNMP 에이전트** - SNMP 에이전트는 관리할 시스템에서 실행되며 시스템의 MIB 데이터베이스를 포함하는 프로그램입니다. 이는 대역폭 및 디스크 공간과 같은 데이터를 수집하여 이를 집계하여 관리 정보 베이스(MIB)로 보냅니다.
- **MIB(관리 정보 기반)** - 이 구성 요소는 SNMP 에이전트에 포함된 구성 요소입니다. SNMP 관리자는 이를 데이터베이스로 사용하고 특정 정보에 대한 액세스를 에이전트에게 요청합니다. 이 정보는 네트워크 관리 시스템(NMS)에 필요합니다. NMS에서는 에이전트에서 이러한 파일의 정보를 가져온 다음 그래프로 변환하고 사용자가 볼 수 있는 디스플레이를 표시합니다. MIB에는 네트워크 장치에 의해 결정되는 통계 및 제어 값이 포함됩니다.
- **SNMP 장치**

다음 SNMP 버전은 호환 가능하며 게이트웨이 구현에 지원됩니다.

- **V2c** - 인증 없이 커뮤니티 문자열을 사용하며 외부 공격에 취약합니다.
- **V3 authNoPriv** - 암호화 없이 사용자 이름과 암호 인증을 사용합니다.
- **V3 authPriv** - encryption와 함께 SNMP 관리 플랫폼에 사용자 이름 및 암호 인증을 사용합니다.



중요

SNMP 트랩을 사용하는 경우 SNMP에 고유한 취약점을 최소화하고 무단 사용자로부터 네트워크를 보호하려면 버전 번호에 대한 올바른 보안 구성이 올바른지 확인하십시오.

13.2. SNMPTRAPD 구성

snmptrapd 데몬에는 **snmp-gateway** 서비스를 만들 때 지정해야 하는 인증 설정이 포함되어 있으므로 **snmp-gateway** 를 배포하기 전에 SNMP(Simple Network Management Protocol) 대상을 구성하는 것이 중요합니다.

SNMP 게이트웨이 기능은 Prometheus 스택에서 생성된 경고를 SNMP 관리 플랫폼에 노출하는 수단을 제공합니다. **snmptrapd** 툴을 기반으로 대상으로 SNMP 트랩을 구성할 수 있습니다. 이 도구를 사용하면 하나 이상의 SNMP 트랩 리스너를 설정할 수 있습니다.

다음 매개변수는 구성에 중요합니다.

- **engineid** 는 hex에서 장치의 고유 식별자이며, TPMV3 게이트웨이에 필요합니다. Red Hat은 이 매개변수에 '8000C53F_CLUSTER_FSID_WITHOUT_DASHES_'를 사용할 것을 권장합니다.
- SNMP_COMMUNITY_FOR_SNMPV2 매개 변수인 **snmp-community** (snmp-community)는 SNMPV2c 게이트웨이용으로 공개됩니다.
- AUTH_PROTOCOL 인 **auth-protocol** 은 SNMPV3 게이트웨이에는 필수입니다. 기본적으로 **SHA**입니다.
- 개인 정보 보호 프로토콜 (PRIVACY_PROTOCOL)은 TPMV3 게이트웨이를 필요로 합니다.
- PRIVACY_PASSWORD 는 암호화가 있는 SNMPV3 게이트웨이를 사용해야 합니다.
- SNMP_V3_AUTH_USER_NAME 은 사용자 이름이며, SNMPV3 게이트웨이에는 필수입니다.
- SNMP_V3_AUTH_PASSWORD 는 암호이며 SNMPV3 게이트웨이에는 필수입니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.
- Red Hat Enterprise Linux 시스템에 **firewalld** 를 설치합니다.

절차

1. SNMP 관리 호스트에서 SNMP 패키지를 설치합니다.

예제

```
[root@host01 ~]# dnf install -y net-snmp-utils net-snmp
```

2. TPM의 포트 162를 열어 경고를 받습니다.

예제

```
[root@host01 ~]# firewall-cmd --zone=public --add-port=162/udp
[root@host01 ~]# firewall-cmd --zone=public --add-port=162/udp --permanent
```

3. MIB(관리 정보 기반)를 구현하여 SNMP 알람을 감지하고 대상 호스트에서 SNMP 지원을 향상시킵니다. 기본 리포지토리에서 원시 파일을 복사합니다.

<https://github.com/ceph/ceph/blob/master/monitoring/snmp/CEPH-MIB.txt>

예제

```
[root@host01 ~]# curl -o CEPH_MIB.txt -L
https://raw.githubusercontent.com/ceph/ceph/master/monitoring/snmp/CEPH-MIB.txt
[root@host01 ~]# scp CEPH_MIB.txt root@host02:/usr/share/snmp/mibs
```

4. **snmptrapd** 디렉토리를 만듭니다.

예제

```
[root@host01 ~]# mkdir /root/snmptrapd/
```

5. SNMP 버전을 기반으로 각 프로토콜의 **snmptrapd** 디렉터리에 구성 파일을 생성합니다.

구문

```
format2 %V\n% Agent Address: %A \n Agent Hostname: %B \n Date: %H - %J - %K - %L -
%M - %Y \n Enterprise OID: %N \n Trap Type: %W \n Trap Sub-Type: %q \n
Community/Infosec Context: %P \n Uptime: %T \n Description: %W \n PDU Attribute/Value
Pair Array:\n%v \n ----- \n
createuser -e 0x_ENGINE_ID_ SNMPV3_AUTH_USER_NAME AUTH_PROTOCOL
SNMP_V3_AUTH_PASSWORD PRIVACY_PROTOCOL PRIVACY_PASSWORD
authuser log,execute SNMP_V3_AUTH_USER_NAME
authCommunity log,execute,net SNMP_COMMUNITY_FOR_SNMPV2
```

- SNMPV2c의 경우 다음과 같이 **snmptrapd_public.conf** 파일을 만듭니다.

예제

```
format2 %V\n% Agent Address: %A \n Agent Hostname: %B \n Date: %H - %J - %K -
%L - %M - %Y \n Enterprise OID: %N \n Trap Type: %W \n Trap Sub-Type: %q \n
Community/Infosec Context: %P \n Uptime: %T \n Description: %W \n PDU
Attribute/Value Pair Array:\n%v \n ----- \n

authCommunity log,execute,net public
```

여기에서 공개 설정은 **snmp-gateway** 서비스를 배포할 때 사용되는 **snmp_community** 설정과 일치해야 합니다.

- 인증만 사용하는 SNMPV3의 경우 다음과 같이 **snmptrapd_auth.conf** 파일을 만듭니다.

예제

```
format2 %V\n% Agent Address: %A \n Agent Hostname: %B \n Date: %H - %J - %K -
%L - %M - %Y \n Enterprise OID: %N \n Trap Type: %W \n Trap Sub-Type: %q \n
Community/Infosec Context: %P \n Uptime: %T \n Description: %W \n PDU
Attribute/Value Pair Array:\n%v \n ----- \n
createuser -e 0x8000C53F64f341c655d11eb8778fa163e914bcc myuser SHA
mypassword
authuser log,execute myuser
```

0x8000C53F64f341c655d11eb8778fa163e914bcc 문자열은 **engine_id** 이고 **myuser** 및 **mypassword** 는 자격 증명입니다. 암호 보안은 **SHA** 알고리즘에 의해 정의됩니다.

이는 **snmp-gateway** 데몬을 배포하기 위한 설정에 해당합니다.

예제

```
snmp_v3_auth_username: myuser
snmp_v3_auth_password: mypassword
```

- 인증 및 암호화를 사용하는 SNMPV3의 경우 다음과 같이 **snmptrapd_authpriv.conf** 파일을 만듭니다.

예제

```
format2 %V\n% Agent Address: %A \n Agent Hostname: %B \n Date: %H - %J - %K -
%L - %M - %Y \n Enterprise OID: %N \n Trap Type: %W \n Trap Sub-Type: %q \n
Community/Infosec Context: %P \n Uptime: %T \n Description: %W \n PDU
Attribute/Value Pair Array:\n%v \n ----- \n
createuser -e 0x8000C53F64f64f341c655d11eb8778fa163e914bcc myuser SHA
mypassword DES mysecret
authuser log,execute myuser
```

0x8000C53F64f64f341c655d11eb8778fa163e914bcc 문자열은 **engine_id** 이고 **myuser** 및 **mypassword** 는 자격 증명입니다. 암호 보안은 **SHA** 알고리즘에 의해 정의되며 **DES** 는 개인 정보 보호 암호화의 유형입니다.

이는 **snmp-gateway** 데몬을 배포하기 위한 설정에 해당합니다.

예제

```
snmp_v3_auth_username: myuser
snmp_v3_auth_password: mypassword
snmp_v3_priv_password: mysecret
```

- SNMP 관리 호스트에서 데몬을 실행합니다.

구문

```
/usr/sbin/snmptrapd -M /usr/share/snmp/mibs -m CEPH-MIB.txt -f -C -c
/root/snmptrapd/CONFIGURATION_FILE -Of -Lo :162
```

예제

```
[root@host01 ~]# /usr/sbin/snmptrapd -M /usr/share/snmp/mibs -m CEPH-MIB.txt -f -C -c
/root/snmptrapd/snmptrapd_auth.conf -Of -Lo :162
```

- 스토리지 클러스터에서 트리거된 경고가 있는 경우 SNMP 관리 호스트에서 출력을 모니터링할 수 있습니다. SNMP 트랩 및 MIB에 의해 디코딩된 트랩도 확인합니다.

예제

```
NET-SNMP version 5.8
Agent Address: 0.0.0.0
Agent Hostname: <UNKNOWN>
Date: 15 - 5 - 12 - 8 - 10 - 4461391
```

```

Enterprise OID: .
Trap Type: Cold Start
Trap Sub-Type: 0
Community/Infosec Context: TRAP2, SNMP v3, user myuser, context
Uptime: 0
Description: Cold Start
PDU Attribute/Value Pair Array:
.iso.org.dod.internet.mgmt.mib-2.1.3.0 = Timeticks: (292276100) 3 days, 19:52:41.00
.iso.org.dod.internet.snmpV2.snmpModules.1.1.4.1.0 = OID:
.iso.org.dod.internet.private.enterprises.ceph.cephCluster.cephNotifications.prometheus.promM
gr.promMgrPrometheusInactive
.iso.org.dod.internet.private.enterprises.ceph.cephCluster.cephNotifications.prometheus.promM
gr.promMgrPrometheusInactive.1 = STRING:
"1.3.6.1.4.1.50495.1.2.1.6.2[alertname=CephMgrPrometheusModuleInactive]"
.iso.org.dod.internet.private.enterprises.ceph.cephCluster.cephNotifications.prometheus.promM
gr.promMgrPrometheusInactive.2 = STRING: "critical"
.iso.org.dod.internet.private.enterprises.ceph.cephCluster.cephNotifications.prometheus.promM
gr.promMgrPrometheusInactive.3 = STRING: "Status: critical"
- Alert: CephMgrPrometheusModuleInactive
Summary: Ceph's mgr/prometheus module is not available
Description: The mgr/prometheus module at 10.70.39.243:9283 is unreachable. This could
mean that the module has been disabled or the mgr itself is down.
Without the mgr/prometheus module metrics and alerts will no longer function. Open a shell
to ceph and use 'ceph -s' to determine whether the mgr is active. If the mgr is not active,
restart it, otherwise you can check the mgr/prometheus module is loaded with 'ceph mgr
module ls' and if it's not listed as enabled, enable it with 'ceph mgr module enable
prometheus'"

```

위 예에서는 Prometheus 모듈이 비활성화된 후 경고가 생성됩니다.

추가 리소스

- Red Hat Ceph Storage Operations Guide 의 [Deploying the SNMP gateway](#) 섹션을 참조하십시오.

13.3. SNMP 게이트웨이 배포

SNMPV2c 또는 SNMPV3을 사용하여 SNMP(Simple Network Management Protocol) 게이트웨이를 배포할 수 있습니다. SNMP 게이트웨이를 배포하는 방법은 다음 두 가지가 있습니다.

1. 자격 증명 파일을 생성합니다.
2. 모든 세부 정보가 포함된 하나의 서비스 구성 yaml 파일을 생성합니다.

다음 매개변수를 사용하여 버전을 기반으로 SNMP 게이트웨이를 배포할 수 있습니다.

- **service_type** 은 **snmp-gateway** 입니다.
- **service_name** 은 사용자 정의 문자열입니다.
- **개수** 는 스토리지 클러스터에 배포할 SNMP 게이트웨이 수입니다.
- **snmp_destination** 매개변수는 hostname:port 형식이어야 합니다.
- **engineid** 는 hex에서 장치의 고유 식별자이며, TPMV3 게이트웨이에 필요합니다. Red Hat은 이 매개변수에 '8000C53F_CLUSTER_FSID_WITHOUT_DASHES_'를 사용할 것을 권장합니다.

- **snmp_community** 매개 변수는 SNMPV2c 게이트웨이의 **공용**입니다.
- **auth-protocol** 은 SNMPV3 게이트웨이에는 필수이며 기본적으로 **SHA** 입니다.
- **개인 정보 보호 프로토콜**은 인증 및 암호화를 사용하는 SNMPV3 게이트웨이에는 필수입니다.
- 이 포트는 기본적으로 **9464** 입니다.
- 오케스트레이터에 비밀 및 암호를 전달하려면 **-i FILENAME** 을 제공해야 합니다.

SNMP 게이트웨이 서비스가 배포되거나 업데이트되면 Prometheus Alertmanager 구성이 자동으로 업데이트되어 추가 처리를 위해 objectidentifier가 있는 모든 경고를 SNMP 게이트웨이 데몬으로 전달합니다.

사전 요구 사항

- 실행 중인 Red Hat Ceph Storage 클러스터.
- 노드에 대한 루트 수준 액세스입니다.
- SNMP 관리 호스트인 대상 호스트에서 **snmptrapd** 를 구성합니다.

절차

1. Cephadm 셸에 로그인합니다.

예제

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

2. SNMP 게이트웨이를 배포해야 하는 호스트의 레이블을 생성합니다.

구문

```
ceph orch host label add HOSTNAME snmp-gateway
```

예제

```
[ceph: root@host01 /]# ceph orch host label add host02 snmp-gateway
```

3. SNMP 버전을 기반으로 자격 증명 파일 또는 서비스 구성 파일을 생성합니다.
 - SNMPV2c의 경우 다음과 같이 파일을 만듭니다.

예제

```
[ceph: root@host01 /]# cat snmp_creds.yml

snmp_community: public
```

또는

예제

```
[ceph: root@host01 /]# cat snmp-gateway.yml
```

```

service_type: snmp-gateway
service_name: snmp-gateway
placement:
  count: 1
spec:
  credentials:
    snmp_community: public
  port: 9464
  snmp_destination: 192.168.122.73:162
  snmp_version: V2c

```

- 인증만 사용하는 SNMPV3의 경우 다음과 같이 파일을 만듭니다.

예제

```

[ceph: root@host01 /]# cat snmp_creds.yml

snmp_v3_auth_username: myuser
snmp_v3_auth_password: mypassword

```

또는

예제

```

[ceph: root@host01 /]# cat snmp-gateway.yml

service_type: snmp-gateway
service_name: snmp-gateway
placement:
  count: 1
spec:
  credentials:
    snmp_v3_auth_password: mypassword
    snmp_v3_auth_username: myuser
  engine_id: 8000C53F64f341c655d11eb8778fa163e914bcc
  port: 9464
  snmp_destination: 192.168.122.1:162
  snmp_version: V3

```

- 인증 및 암호화를 사용하는 SNMPV3의 경우 다음과 같이 파일을 만듭니다.

예제

```

[ceph: root@host01 /]# cat snmp_creds.yml

snmp_v3_auth_username: myuser
snmp_v3_auth_password: mypassword
snmp_v3_priv_password: mysecret

```

또는

예제

```
[ceph: root@host01 /]# cat snmp-gateway.yml

service_type: snmp-gateway
service_name: snmp-gateway
placement:
  count: 1
spec:
  credentials:
    snmp_v3_auth_password: mypassword
    snmp_v3_auth_username: myuser
    snmp_v3_priv_password: mysecret
  engine_id: 8000C53F64f341c655d11eb8778fa163e914bcc
  port: 9464
  snmp_destination: 192.168.122.1:162
  snmp_version: V3
```

4. **ceph orch** 명령을 실행합니다.

구문

```
ceph orch apply snmp-gateway --snmp_version=V2c_OR_V3 --
destination=SNMP_DESTINATION [--port=PORT_NUMBER]\
[--engine-id=8000C53F_CLUSTER_FSID_WITHOUT_DASHES_] [--auth-
protocol=MDS_OR_SHA] [--privacy_protocol=DES_OR_AES] -i FILENAME
```

또는

구문

```
ceph orch apply -i FILENAME.yml
```

- **snmp_creds** 파일이 있는 SNMPV2c의 경우 **snmp-version** 을 **V2c** 로 사용하여 **ceph orch** 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph orch apply snmp-gateway --snmp-version=V2c --
destination=192.168.122.73:162 --port=9464 -i snmp_creds.yml
```

- 인증만 사용하는 SNMP **V3** 의 경우 **snmp_creds** 파일을 사용하여 v3 및 **engine-id** 로 **snmp-version** 을 사용하여 **ceph orch** 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph orch apply snmp-gateway --snmp-version=V3 --engine-
id=8000C53F64f341c655d11eb8778fa163e914bcc--destination=192.168.122.73:162 -i
snmp_creds.yml
```

- 인증 및 암호화가 있는 SNMPV3의 경우 **snmp_creds** 파일을 사용하여 **ceph orch** 명령을 **V3privacy-protocol, engine-id** 로 실행합니다.

예제

```
[ceph: root@host01 /]# ceph orch apply snmp-gateway --snmp-version=V3 --engine-id=8000C53F64f341c655d11eb8778fa163e914bcc--destination=192.168.122.73:162 --privacy-protocol=AES -i snmp_creds.yml
```

또는

- **snmp-gateway** 파일을 사용하는 모든 **SNMP** 버전의 경우 다음 명령을 실행합니다.

예제

```
[ceph: root@host01 /]# ceph orch apply -i snmp-gateway.yml
```

추가 리소스

- *Red Hat Ceph Storage Operations Guide* 의 [Configuring 'snmptrapd'](#) 섹션을 참조하십시오.

14장. 노드 오류 처리

스토리지 관리자는 스토리지 클러스터 내에서 전체 노드가 실패할 수 있으며 노드 오류를 처리하는 것은 디스크 오류를 처리하는 것과 유사합니다. 노드 장애가 발생하면 **Ceph** 복구 배치 그룹(**PG**) 대신 하나의 디스크에 대해서만 배치 그룹(**PG**)을 복구해야 합니다. **Ceph**에서 **OSD**가 모두 다운되었음을 감지하고 자동 복구라는 복구 프로세스를 자동으로 시작합니다.

노드 장애 시나리오는 세 가지입니다. 다음은 노드를 교체할 때 각 시나리오에 대한 상위 수준 워크플로입니다.

- 노드 교체는 하지만 오류가 발생한 노드에서 **root** 및 **Ceph OSD** 디스크를 사용합니다.
 1. 백필을 비활성화합니다.
 2. 노드를 교체하여 이전 노드에서 디스크를 가져와 새 노드에 추가합니다.
 3. 백필을 활성화합니다.
- 노드를 교체하고 운영 체제를 다시 설치하고 실패한 노드에서 **Ceph OSD** 디스크를 사용합니다.
 1. 백필을 비활성화합니다.
 2. **Ceph** 구성의 백업을 만듭니다.
 3. 노드를 교체하고 실패한 노드에서 **Ceph OSD** 디스크를 추가합니다.
 4. 디스크를 **JBOD**로 구성.
 5. 운영 체제를 설치합니다.

6. **Ceph** 구성을 복원합니다.
 7. **Ceph Orchestrator** 명령을 사용하여 스토리지 클러스터에 새 노드를 추가하고 **Ceph** 데몬이 해당 노드에 자동으로 배치됩니다.
 8. 백필을 활성화합니다.
- 노드를 교체하고 운영 체제를 다시 설치하고 모든 새 **Ceph OSD** 디스크를 사용합니다.
 1. 백필을 비활성화합니다.
 2. 스토리지 클러스터에서 장애가 발생한 노드의 모든 **OSD**를 제거합니다.
 3. **Ceph** 구성의 백업을 만듭니다.
 4. 노드를 교체하고 실패한 노드에서 **Ceph OSD** 디스크를 추가합니다.
 - a. 디스크를 **JBOD**로 구성.
 5. 운영 체제를 설치합니다.
 6. **Ceph Orchestrator** 명령을 사용하여 스토리지 클러스터에 새 노드를 추가하고 **Ceph** 데몬이 해당 노드에 자동으로 배치됩니다.
 7. 백필을 활성화합니다.

14.1. 사전 요구 사항

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

- 실패한 노드.

14.2. 노드를 추가하거나 제거하기 전에 고려해야 할 사항

Ceph의 뛰어난 기능 중 하나는 런타임에 **Ceph OSD** 노드를 추가하거나 제거하는 기능입니다. 즉, 스토리지 클러스터를 중단하지 않고도 스토리지 클러스터 용량의 크기를 조정하거나 하드웨어를 교체할 수 있습니다.

스토리지 클러스터가 성능이 저하된 상태에서 **Ceph** 클라이언트에 서비스를 제공하는 기능도 작동할 수 있습니다. 예를 들어, 시간 초과 또는 주말을 작업하는 대신 정규 업무 시간 동안 하드웨어를 추가하거나 제거할 수 있습니다. 그러나 **Ceph OSD** 노드를 추가하고 제거하면 성능에 큰 영향을 미칠 수 있습니다.

Ceph OSD 노드를 추가하거나 제거하기 전에 스토리지 클러스터 성능에 미치는 영향을 고려하십시오.

- 스토리지 클러스터 용량을 확장하거나 줄이는지 여부에 따라 **Ceph OSD** 노드를 추가하거나 제거하면 스토리지 클러스터 리밸런싱으로 백필이 발생합니다. 이 리밸런싱 기간 동안 **Ceph**는 추가 리소스를 사용하므로 스토리지 클러스터 성능에 영향을 줄 수 있습니다.
- 프로덕션 **Ceph** 스토리지 클러스터에서 **Ceph OSD** 노드에는 특정 유형의 스토리지 전략을 용이하게 하는 특정 하드웨어 구성이 있습니다.
- **Ceph OSD** 노드는 **CRUSH** 계층 구조의 일부이므로 노드를 추가하거나 제거하는 성능에는 일반적으로 **CRUSH** 규칙을 사용하는 풀의 성능에 영향을 미칩니다.

추가 리소스

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

14.3. 성능 고려 사항

다음 요소는 일반적으로 **Ceph OSD** 노드를 추가하거나 제거할 때 스토리지 클러스터의 성능에 영향을 미칩니다.

- **Ceph** 클라이언트는 I/O 인터페이스에 로드를 **Ceph**에 배치합니다. 즉, 클라이언트가 풀에 로드를 배치합니다. 풀은 **CRUSH** 규칙 세트에 매핑됩니다. 기본 **CRUSH** 계층 구조를 통해 **Ceph**는

장에 도메인에 데이터를 배치할 수 있습니다. 기본 **Ceph OSD** 노드에 클라이언트 부하가 높은 풀이 포함된 경우 클라이언트 로드가 복구 시간에 크게 영향을 미치고 성능을 저하시킬 수 있습니다. 쓰기 작업에 내구성을 위해 데이터 복제가 필요하므로 쓰기 집약적 클라이언트 로드가 특히 스토리지 클러스터를 복구하는 데 걸리는 시간이 증가할 수 있습니다.

•

일반적으로 추가하거나 제거하는 용량은 스토리지 클러스터의 복구 시간에 영향을 미칩니다. 또한 추가하거나 제거하는 노드의 스토리지 밀도가 복구 시간에 영향을 줄 수 있습니다. 예를 들어 **OSD가 36개인 노드**는 일반적으로 **OSD가 12개인 노드**보다 복구하는 데 시간이 오래 걸립니다.

•

노드를 제거할 때는 예비 용량이 충분하여 전체 비율 또는 거의 전체 비율에 도달하지 않도록 해야 합니다. 스토리지 클러스터가 전체 비율에 도달하면 **Ceph**에서 쓰기 작업을 일시 중단하여 데이터 손실을 방지합니다.

•

Ceph OSD 노드는 하나 이상의 **Ceph CRUSH** 계층 구조에 매핑되며 계층 구조는 하나 이상의 풀에 매핑됩니다. **CRUSH** 규칙 세트를 사용하는 각 풀에는 **Ceph OSD** 노드를 추가하거나 제거할 때 성능에 영향을 미칩니다.

•

복제 풀은 더 많은 네트워크 대역폭을 사용하여 데이터의 심층 복사본을 복제하는 경향이 있지만, 이레이저 코딩 풀은 더 많은 **CPU**를 사용하여 **k+m** 코딩 청크를 계산하는 경향이 있습니다. 데이터에 대한 복사본이 많을수록 스토리지 클러스터를 복구하는 데 시간이 오래 걸립니다. 예를 들어, 더 큰 풀 또는 **k+m** 청크 수가 더 많은 경우 동일한 데이터의 복사본이 적은 복제 풀보다 복구하는 데 시간이 더 오래 걸립니다.

•

드라이브, 컨트롤러 및 네트워크 인터페이스 카드는 모두 복구 시간에 영향을 줄 수 있는 처리량 특성을 갖습니다. 일반적으로 처리량이 높은 **10Gbps** 및 **SSD**와 같은 노드는 처리량이 낮은 노드(예: **1Gbps** 및 **SATA** 드라이브)보다 빠르게 복구됩니다.

14.4. 노드 추가 또는 제거 권장 사항

Red Hat은 노드 내에서 한 번에 하나의 **OSD**를 추가하거나 제거하고 다음 **OSD**로 진행하기 전에 스토리지 클러스터를 복구할 것을 권장합니다. 이렇게 하면 스토리지 클러스터 성능에 미치는 영향을 최소화하는 데 도움이 됩니다. 노드에 오류가 발생하면 한 번에 하나의 **OSD** 대신 전체 노드를 한 번에 변경해야 할 수 있습니다.

OSD를 제거하려면 다음을 수행합니다.

•

Ceph Orchestrator를 사용하여 **OSD** 데몬 제거.

OSD를 추가하려면 다음을 수행합니다.

- 사용 가능한 모든 장치에 **Ceph OSD** 배포 사용.
- 고급 서비스 사양을 사용하여 **Ceph OSD** 배포 사용.
- 특정 장치 및 호스트에 **Ceph OSD** 배포 사용.

Ceph OSD 노드를 추가하거나 제거할 때 다른 진행 중인 프로세스도 스토리지 클러스터 성능에 영향을 미칩니다. 클라이언트 I/O에 미치는 영향을 줄이기 위해 **Red Hat**은 다음을 권장합니다.

용량 계산

Ceph OSD 노드를 제거하기 전에 스토리지 클러스터가 전체 비율에 도달하지 않고 모든 **OSD**의 콘텐츠를 백업할 수 있는지 확인합니다. 전체 비율에 도달하면 스토리지 클러스터가 쓰기 작업을 거부합니다.

일시적으로 스크럽 비활성화

스크럽은 스토리지 클러스터 데이터의 내구성을 보장하는 데 필수적이지만 리소스 집약적입니다. **Ceph OSD** 노드를 추가하거나 제거하기 전에 스크럽 및 덤 스크럽을 비활성화하고 진행하기 전에 현재 스크럽 작업을 완료하십시오.

```
ceph osd set noscrub
ceph osd set nodeep-scrub
```

Ceph OSD 노드를 추가하거나 제거하고 스토리지 클러스터가 **active+clean** 상태로 반환되면 **noscrub** 및 **nodeep-scrub** 설정을 설정 해제합니다.

```
ceph osd unset noscrub
ceph osd unset nodeep-scrub
```

백업 및 복구 제한

적절한 데이터 내구성이 있는 경우 성능이 저하된 상태에서 작동하는 데 아무런 문제가 없습니다. 예를 들어 **osd_pool_default_size = 3** 및 **osd_pool_default_min_size = 2** 로 스토리지 클러스터를 작동할 수 있습니다. 가능한 가장 빠른 복구 시간을 위해 스토리지 클러스터를 조정할 수 있지만 **Ceph** 클라이언트 I/O 성능에 큰 영향을 미칩니다. 가장 높은 **Ceph** 클라이언트 I/O 성능을 유지하려면 백업 및 복구 작업을 제한하고 시간이 더 오래 걸릴 수 있습니다.

```
osd_max_backfills = 1
osd_recovery_max_active = 1
osd_recovery_op_priority = 1
```

osd_recovery_sleep 과 같은 **sleep** 및 **delay** 매개 변수를 설정하는 것도 고려할 수 있습니다.

배치 그룹 수 증가

마지막으로 스토리지 클러스터 크기를 확장하는 경우 배치 그룹의 수를 늘려야 할 수 있습니다. 배치 그룹 수를 확장해야 한다고 판단되면 **Red Hat**은 배치 그룹 수를 점진적으로 늘리는 것이 좋습니다. 배치 그룹의 수를 상당한 금액으로 늘리면 성능이 상당히 저하됩니다.

14.5. CEPH OSD 노드 추가

Red Hat Ceph Storage 클러스터의 용량을 확장하려면 **OSD** 노드를 추가할 수 있습니다.

사전 요구 사항

- 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 네트워크 연결이 있는 프로비저닝된 노드.

절차

1. 스토리지 클러스터의 다른 노드가 짧은 호스트 이름으로 새 노드에 연결할 수 있는지 확인합니다.
2. 일시적으로 스크럽을 비활성화합니다.

예제

```
[ceph: root@host01 /]# ceph osd set noscrub
[ceph: root@host01 /]# ceph osd set nodeep-scrub
```

3. 백필 및 복구 기능을 제한합니다.

구문

```
ceph tell DAEMON_TYPE.* injectargs --OPTION_NAME VALUE [--OPTION_NAME VALUE]
```

예제

```
[ceph: root@host01 /]# ceph tell osd.* injectargs --osd-max-backfills 1 --osd-recovery-max-active 1 --osd-recovery-op-priority 1
```

4. 클러스터의 공용 **SSH** 키를 폴더에 추출합니다.

구문

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

예제

```
[ceph: root@host01 /]# ceph cephadm get-pub-key > ~/ceph.pub
```

5. **ceph** 클러스터의 공용 **SSH** 키를 새 호스트의 **root** 사용자 **authorized_keys** 파일에 복사합니다.

구문

```
ssh-copy-id -f -i ~/PATH root@HOST_NAME_2
```

예제

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

6.

CRUSH 맵에 새 노드를 추가합니다.

구문

```
ceph orch host add NODE_NAME IP_ADDRESS
```

예제

```
[ceph: root@host01 /]# ceph orch host add host02 10.10.128.70
```

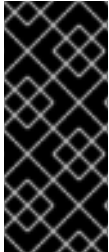
7.

노드의 각 디스크에 **OSD**를 스토리지 클러스터에 추가합니다.

-

사용 가능한 모든 장치에 **Ceph OSD** 배포 사용.

- 고급 서비스 사양을 사용하여 **Ceph OSD** 배포 사용.
- 특정 장치 및 호스트에 **Ceph OSD** 배포 사용.



중요

Red Hat Ceph Storage 클러스터에 **OSD** 노드를 추가할 때 **Red Hat**은 한 번에 하나의 **OSD** 데몬을 추가하고 다음 **OSD**로 진행하기 전에 **활성+clean** 상태로 클러스터를 복구할 것을 권장합니다.

추가 리소스

- 자세한 내용은 **Red Hat Ceph Storage Configuration Guide**의 **Runtime(런타임)**에서 **특정 구성 설정** 섹션을 참조하십시오.
- **CRUSH** 계층 구조의 적절한 위치에 노드를 배치하는 방법에 대한 자세한 내용은 **Red Hat Ceph Storage** 전략 가이드의 **버킷 및 버킷 전환** 섹션을 참조하십시오.

14.6. CEPH OSD 노드 제거

스토리지 클러스터의 용량을 줄이려면 **OSD** 노드를 제거합니다.



주의

Ceph OSD 노드를 제거하기 전에 스토리지 클러스터가 전체 비율에 도달하지 않고 모든 **OSD**의 콘텐츠를 백업할 수 있는지 확인합니다. 전체 비율에 도달하면 스토리지 클러스터가 쓰기 작업을 거부합니다.

사전 요구 사항

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

- 스토리지 클러스터의 모든 노드에 대한 루트 수준의 액세스.

절차

1. 스토리지 클러스터의 용량을 확인합니다.

구문

```
ceph df
rados df
ceph osd df
```

2. 일시적으로 스크럽을 비활성화합니다.

구문

```
ceph osd set noscrub
ceph osd set nodeep-scrub
```

3. 백필 및 복구 기능을 제한합니다.

구문

```
ceph tell DAEMON_TYPE.* injectargs --OPTION_NAME VALUE [--OPTION_NAME VALUE]
```

예제

```
[ceph: root@host01 /]# ceph tell osd.* injectargs --osd-max-backfills 1 --osd-recovery-max-active 1 --osd-recovery-op-priority 1
```

4.

스토리지 클러스터에서 노드의 각 **OSD**를 제거합니다.

•

Ceph Orchestrator를 사용하여 **OSD** 데몬 제거.



중요

스토리지 클러스터에서 **OSD** 노드를 제거할 때 노드 내에서 하나의 **OSD**를 제거하고 다음 **OSD**를 제거하기 전에 클러스터가 **active+clean** 상태로 복구되도록 하는 것이 좋습니다.

a.

OSD를 제거한 후 스토리지 클러스터가 거의 전체 비율에 도달하지 않는지 확인합니다.

구문

```
ceph -s  
ceph df
```

b.

노드의 모든 **OSD**가 스토리지 클러스터에서 제거될 때까지 이 단계를 반복합니다.

5.

모든 **OSD**가 제거되면 호스트를 제거합니다.

•

Ceph Orchestrator를 사용하여 호스트 제거.

하기 내용

- 자세한 내용은 **Red Hat Ceph Storage 구성 가이드의 런타임 설정** 섹션을 참조하십시오.

14.7. 노드 오류 시뮬레이션

하드 노드 오류를 시뮬레이션하려면 노드의 전원을 끄고 운영 체제를 다시 설치합니다.

사전 요구 사항

- 정상 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준의 액세스.

절차

1. 스토리지 클러스터의 용량을 확인하여 노드 제거의 영향을 확인합니다.

예제

```
[ceph: root@host01 /]# ceph df
[ceph: root@host01 /]# rados df
[ceph: root@host01 /]# ceph osd df
```

2. 선택적으로 복구 및 백업을 비활성화합니다.

예제

```
[ceph: root@host01 /]# ceph osd set noout
[ceph: root@host01 /]# ceph osd set noscrub
[ceph: root@host01 /]# ceph osd set nodeep-scrub
```

3. 노드를 종료합니다.
4. 호스트 이름을 변경하는 경우 **CRUSH** 맵에서 노드를 제거합니다.

예제

```
[ceph: root@host01 /]# ceph osd crush rm host03
```

5. 스토리지 클러스터의 상태를 확인합니다.

예제

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

6. 노드에 운영 체제를 다시 설치합니다.

7. 새 노드를 추가합니다.

- **Ceph Orchestrator**를 사용하여 호스트 추가 사용.

8. 선택적으로 복구 및 백필을 활성화합니다.

예제

```
[ceph: root@host01 /]# ceph osd unset noout  
[ceph: root@host01 /]# ceph osd unset noscrub  
[ceph: root@host01 /]# ceph osd unset nodeep-scrub
```

9.

Ceph의 상태를 확인합니다.

예제

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

추가 리소스

-

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

15장. 데이터 센터 오류 처리

스토리지 관리자는 데이터 센터 장애 발생을 방지하기 위해 예방 조치를 취할 수 있습니다. 이러한 예방 조치에는 다음이 포함됩니다.

- 데이터 센터 인프라 구성.
- **CRUSH** 맵 계층 구조 내에서 장애 도메인 설정.
- 도메인 내에서 오류 노드 지정.

15.1. 사전 요구 사항

- 정상 실행 중인 **Red Hat Ceph Storage** 클러스터.
- 스토리지 클러스터의 모든 노드에 대한 루트 수준의 액세스.

15.2. 데이터 센터 오류 방지

데이터 센터 인프라 구성

확장 클러스터 내의 각 데이터 센터는 로컬 기능 및 종속 항목을 반영하기 위해 다른 스토리지 클러스터 구성을 가질 수 있습니다. 데이터 센터 간 복제를 설정하여 데이터를 보존합니다. 한 데이터 센터에 오류가 발생하면 스토리지 클러스터의 다른 데이터 센터에 데이터 복사본이 포함됩니다.

CRUSH 맵 계층 구조 내에서 장애 도메인 설정

실패 또는 페일오버 도메인은 스토리지 클러스터 내 도메인의 중복 사본입니다. 활성 도메인이 실패하면 장애 도메인이 활성 도메인이 됩니다.

기본적으로 **CRUSH** 맵은 플랫 계층 구조 내의 스토리지 클러스터의 모든 노드를 나열합니다. 그러나 최상의 결과를 위해 **CRUSH** 맵 내에 논리적 계층 구조를 생성합니다. 계층 구조는 각 노드가 속한 도메인과 장애 도메인을 포함하여 스토리지 클러스터 내 해당 도메인 간의 관계를 지정합니다. 계층 구조 내에서 각 도메인의 장애 도메인을 정의하면 스토리지 클러스터의 안정성이 향상됩니다.

여러 데이터 센터가 포함된 스토리지 클러스터를 계획할 때 **CRUSH** 맵 계층 구조에 노드를 배치하여 하나의 데이터 센터가 다운되면 나머지 스토리지 클러스터가 가동되어 실행됩니다.

도메인 내에서 오류 노드 지정

스토리지 클러스터 내 데이터에 3방향 복제를 사용하려는 경우 장애 도메인 내의 노드 위치를 고려합니다. 데이터 센터 내에서 중단이 발생하는 경우 일부 데이터가 하나의 복사본에만 있을 수 있습니다. 이 시나리오가 발생하면 다음 두 가지 옵션이 있습니다.

- 표준 설정을 사용하여 데이터를 읽기 전용 상태로 둡니다.
- 정전 기간 동안 단 하나의 사본으로만 라이브.

표준 설정을 사용하고 노드에서 데이터 배치의 임의성 때문에 모든 데이터가 영향을 받는 것은 아니지만 일부 데이터는 하나의 복사본만 가질 수 있으며 스토리지 클러스터는 읽기 전용 모드로 되돌아갑니다. 그러나 일부 데이터가 하나의 복사본에만 있는 경우 스토리지 클러스터는 읽기 전용 모드로 돌아갑니다.

15.3. 데이터 센터 오류 처리

Red Hat Ceph Storage는 확장 클러스터에서 데이터 센터 중 하나를 분실하는 등 인프라 장애에 대처할 수 있습니다. 표준 오브젝트 저장소 사용 사례의 경우 세 개의 데이터 센터를 둘 간에 설정한 복제와 독립적으로 구성할 수 있습니다. 이 시나리오에서는 로컬 기능 및 종속성을 반영하여 각 데이터 센터의 스토리지 클러스터 구성이 다를 수 있습니다.

배치 계층 구조의 논리적 구조를 고려해야 합니다. 인프라 내에서 장애 도메인의 계층 구조를 반영하여 적절한 **CRUSH** 맵을 사용할 수 있습니다. 논리적 계층 구조 정의를 사용하면 표준 계층 구조 정의를 사용하는 대신 스토리지 클러스터의 안정성이 향상됩니다. 실패 도메인은 **CRUSH** 맵에 정의되어 있습니다. 기본 **CRUSH** 맵에는 플랫 계층 구조의 모든 노드가 포함됩니다. 확장 클러스터와 같은 3개의 데이터 센터 환경에서 하나의 데이터 센터가 다운될 수 있는 방식으로 노드 배치를 관리해야 하지만 스토리지 클러스터는 계속 가동되고 실행됩니다. 데이터에 3방향 복제를 사용할 때 노드가 상주하는 장애 도메인을 고려하십시오.

아래 예제에서는 6개의 **OSD** 노드가 있는 스토리지 클러스터의 초기 설정에서 결과 맵을 가져옵니다. 이 예제에서 모든 노드에는 하나의 디스크만 있으므로 **OSD**가 한 개만 있습니다. 모든 노드는 기본 루트, 즉 계층 구조 트리의 표준 루트에 따라 정렬됩니다. 2개의 **OSD**에 할당된 가중치가 있으므로 이러한 **OSD**에는 다른 **OSD**보다 더 적은 데이터 체크가 부여됩니다. 이러한 노드는 초기 **OSD** 디스크보다 큰 디스크가 나중에 도입되었습니다. 이는 노드 그룹의 실패를 견디도록 데이터 배치에는 영향을 미치지 않습니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
```

ID	WEIGHT	TYPE	NAME	UP/DOWN	REWEIGHT	PRIMARY-AFFINITY
-1	0.33554	root	default			
-2	0.04779	host	host03			
0	0.04779	osd	osd.0	up	1.00000	1.00000
-3	0.04779	host	host02			
1	0.04779	osd	osd.1	up	1.00000	1.00000
-4	0.04779	host	host01			
2	0.04779	osd	osd.2	up	1.00000	1.00000
-5	0.04779	host	host04			
3	0.04779	osd	osd.3	up	1.00000	1.00000
-6	0.07219	host	host06			
4	0.07219	osd	osd.4	up	0.79999	1.00000
-7	0.07219	host	host05			
5	0.07219	osd	osd.5	up	0.79999	1.00000

논리적 계층적 정의를 사용하여 노드를 동일한 데이터 센터로 그룹화하면 데이터 배치 완성도를 달성할 수 있습니다. 루트, 데이터 센터, 랙, 행 및 호스트의 가능한 정의 유형을 사용하면 세 개의 데이터 센터에 대한 장애 도메인을 반영할 수 있습니다.

- **host01** 및 **host02** 노드가 데이터 센터 1 (**DC1**)에 있습니다.
- 노드 **host03** 및 **host05**는 데이터 센터 2 (**DC2**)에 있습니다.
- 노드 **host04** 및 **host06**은 데이터 센터 3 (**DC3**)에 있습니다.
- 모든 데이터 센터는 동일한 구조(모든 **DC**)에 속합니다.

호스트의 모든 **OSD**가 호스트 정의에 속해 있으므로 변경할 필요가 없습니다. 스토리지 클러스터 런타임 중에 다음을 통해 다른 모든 할당을 조정할 수 있습니다.

- 다음 명령을 사용하여 버킷 구조를 정의합니다.

```
ceph osd crush add-bucket allDC root
ceph osd crush add-bucket DC1 datacenter
```

```
ceph osd crush add-bucket DC2 datacenter
ceph osd crush add-bucket DC3 datacenter
```

•

CRUSH 맵을 수정하여 이 구조 내에서 노드를 적절한 위치로 이동합니다.

```
ceph osd crush move DC1 root=allDC
ceph osd crush move DC2 root=allDC
ceph osd crush move DC3 root=allDC
ceph osd crush move host01 datacenter=DC1
ceph osd crush move host02 datacenter=DC1
ceph osd crush move host03 datacenter=DC2
ceph osd crush move host05 datacenter=DC2
ceph osd crush move host04 datacenter=DC3
ceph osd crush move host06 datacenter=DC3
```

이 구조 내에서 새 호스트도 추가할 수 있으며 새 디스크도 추가할 수 있습니다. 계층 구조에서 **OSD**를 올바른 위치에 배치하면 **CRUSH** 알고리즘은 구조 내의 다른 장애 도메인에 중복 조각을 배치하도록 변경됩니다.

위 예제는 다음과 같습니다.

예제

```
[ceph: root@host01 /]# ceph osd tree
ID WEIGHT TYPE NAME UP/DOWN REWEIGHT PRIMARY-AFFINITY
-8 6.00000 root allDC
-9 2.00000 datacenter DC1
-4 1.00000 host host01
2 1.00000 osd.2 up 1.00000 1.00000
-3 1.00000 host host02
1 1.00000 osd.1 up 1.00000 1.00000
-10 2.00000 datacenter DC2
-2 1.00000 host host03
0 1.00000 osd.0 up 1.00000 1.00000
-7 1.00000 host host05
5 1.00000 osd.5 up 0.79999 1.00000
-11 2.00000 datacenter DC3
-6 1.00000 host host06
4 1.00000 osd.4 up 0.79999 1.00000
-5 1.00000 host host04
3 1.00000 osd.3 up 1.00000 1.00000
-1 0 root default
```

위의 목록은 **osd** 트리를 표시하여 결과 **CRUSH** 맵을 보여줍니다. 이제 호스트가 데이터 센터에 속한 방법 및 모든 데이터 센터가 동일한 최상위 수준에 속하지만 위치를 명확하게 구분할 수 있습니다.



참고

맵에 따라 적절한 위치에 데이터를 배치하면 정상 클러스터 내에서만 올바르게 작동합니다. 일부 **OSD**를 사용할 수 없는 경우에는 **Misplacement**가 발생할 수 있습니다. 이러한 불일치는 가능한 한 자동으로 수정됩니다.

추가 리소스



자세한 내용은 **Red Hat Ceph Storage** 전략 가이드의 **CRUSH 관리** 장을 참조하십시오.