Red Hat OpenStack Platform 13

기존 Red Hat Ceph 클러스터와 Overcloud 통합

Stand-Alone Red Hat Ceph Storage를 사용하도록 Overcloud 구성

Last Updated: 2023-05-25

Red Hat OpenStack Platform 13 기존 Red Hat Ceph 클러스터와 Overcloud 통합

Stand-Alone Red Hat Ceph Storage를 사용하도록 Overcloud 구성

OpenStack Team
rhos-docs@redhat.com

법적 공지

초록

이 가이드에서는 Red Hat OpenStack Platform director를 사용하여 Overcloud를 기존 독립 실행형 Red Hat Ceph 클러스터와 통합하는 방법을 설명합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 CTO Chris Wright의 메시지 를 참조하십시오.

1장. 소개

RHOSP(Red Hat OpenStack Platform) director는 *오버클라우드* 라는 클라우드 환경을 생성합니다. director는 오버클라우드에 추가 기능을 설정하는 기능을 제공합니다. 이러한 추가 기능 중 하나로 Red Hat Ceph Storage와의 통합이 포함됩니다. 여기에는 director 또는 기존 Ceph Storage 클러스터와 함께 생성된 Ceph Storage 클러스터가 모두 포함됩니다.

1.1. CEPH STORAGE 정의

Red Hat Ceph Storage는 우수한 성능, 안정성 및 확장성을 제공하도록 설계된 분산 데이터 오브젝트 저장소입니다. 분산 오브젝트 저장소는 구조화되지 않은 데이터를 수용하기 때문에 스토리지의 미래이며 클라이언트가 최신 개체 인터페이스와 기존 인터페이스를 동시에 사용할 수 있기 때문입니다. 모든 Ceph 배포의 핵심은 Ceph Storage 클러스터이며, 이는 두 가지 유형의 데몬으로 구성됩니다.

Ceph Object Storage Daemon

Ceph OSD(오브젝트 스토리지 데몬)는 Ceph 클라이언트를 대신하여 데이터를 저장합니다. 또한 Ceph OSD는 Ceph 노드의 CPU 및 메모리를 사용하여 데이터 복제, 리밸런싱, 복구, 모니터링 및 보고 기능을 수행합니다.

Ceph Monitor

Ceph 모니터(MON)는 스토리지 클러스터의 현재 상태와 함께 Ceph Storage 클러스터 맵의 마스터 사본을 유지 관리합니다.

Red Hat Ceph Storage에 대한 자세한 내용은 Red Hat Ceph Storage 아키텍처 가이드를 참조하십시오 .

중요

이 가이드에서는 블록 스토리지를 사용하여 Ceph를 구성하는 방법을 설명합니다.

Ceph Object Gateway(RGW): 이 기능은 이번 릴리스에서 *기술 프리뷰로 제공되므로 Red Hat에서 완전히* 지원되지 않습니다. 테스트 용도로만 사용해야 하며 프로덕션 환경에 배포해서는 안 됩니다. 기술 프리뷰 기능에 대한 자세한 내용은 적용 범위 상세 정보 를 참조하십시오.

Ceph 파일 시스템(CephFS)은 지원되지 않습니다.

1.2. SCENARIO 정의

이 가이드에서는 기존 Ceph Storage 클러스터를 오버클라우드와 통합하는 방법을 제공합니다. 즉, director에서 스토리지 요구 사항에 맞게 Ceph Storage 클러스터를 사용하도록 오버클라우드를 구성합니다. 오버클라우드 구성 외부에서 클러스터 자체를 관리하고 확장합니다.

2장. OVERCLOUD 노드 준비

이 장에서 설명하는 시나리오는 Overcloud의 6개의 노드로 구성되어 있습니다.

- 고가용성이 있는 컨트롤러 노드 3개
- 컴퓨팅 노드 세 개

director는 별도의 Ceph Storage 클러스터와 자체 노드를 Overcloud에 통합합니다. 오버클라우드와 독립적으로 이 클러스터를 관리합니다. 예를 들어 OpenStack Platform director가 아닌 Ceph 관리 툴을 사용하여 Ceph Storage 클러스터를 확장합니다. 자세한 내용은 Red Hat Ceph 설명서를 참조하십시오.

2.1. CEPH STORAGE에 대한 사전 배포 검증

오버클라우드 배포 실패를 방지하려면 서버에 필요한 패키지가 있는지 확인합니다.

2.1.1. ceph-ansible 패키지 버전 확인

언더클라우드에는 오버클라우드를 배포하기 전에 잠재적인 문제를 확인하기 위해 실행할 수 있는 Ansible 기반 검증이 포함되어 있습니다. 이러한 검증은 일반적인 문제가 발생하기 전에 오버클라우드 배포 실패를 방지하는 데 도움이 될 수 있습니다.

절차

ceph-ansible 패키지의 수정 버전이 설치되어 있는지 확인합니다.

```
$ ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/openstack-tripleo-validations/validations/ceph-ansible-installed.yaml
```

2.1.2. 사전 프로비저닝된 노드의 패키지 확인

오버클라우드 배포에서 사전 프로비저닝된 노드를 사용하는 경우 Ceph 서비스를 호스팅하는 오버클라우드 노드에 필요한 패키지가 서버에 있는지 확인할 수 있습니다.

사전 프로비저닝된 노드에 대한 자세 한 내용은 사전 프로비저닝된 노드를 사용하여 기본 오버클라우드 구성을 참조하십시오.

절차

서버에 필수 패키지가 포함되어 있는지 확인합니다.

```
ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/openstack-tripleo-validations/validations/ceph-dependencies-installed.yaml
```

2.2. 기존 CEPH STORAGE 클러스터 구성

1. 사용자 환경과 관련된 Ceph 클러스터에서 다음 풀을 생성합니다.
   - **볼륨**: OpenStack Block Storage(cinder)용 스토리지
   - **이미지**: OpenStack Image Storage(glance)용 스토리지
   - **VMs**: 인스턴스용 스토리지

- **백업**: OpenStack Block Storage Backup(cinder-backup)용 스토리지
- **metrics**: OpenStack Telemetry Metrics(gnocchi)용 스토리지
  다음 명령을 가이드로 사용하십시오.

```
[root@ceph ~]# ceph osd pool create volumes PGNUM
[root@ceph ~]# ceph osd pool create images PGNUM
[root@ceph ~]# ceph osd pool create vms PGNUM
[root@ceph ~]# ceph osd pool create backups PGNUM
[root@ceph ~]# ceph osd pool create metrics PGNUM
```

PGNUM 을 *배치 그룹* 수로 바꿉니다. OSD당 약 100 %를 권장합니다. 예를 들어, OSD의 총 수가 복제본 수로 100을 곱한 값입니다(**osd 풀 기본 크기**). 풀 계산기당 Ceph PG(배치 그룹)를 사용하여 적절한 값을 결정할 수도 있습니다.

2. 다음과 같은 기능을 사용하여 Ceph 클러스터에 **client.openstack** 사용자를 생성합니다.
   - **cap_mgr:** "허용 *"
   - **cap_mon:** Profile rbd
   - **cap_osd:** 프로필 rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd pool=backups, profile rbd pool=metrics
     다음 명령을 가이드로 사용하십시오.

```
[root@ceph ~]# ceph auth add client.openstack mgr 'allow *' mon 'profile rbd' osd 'profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd pool=backups, profile rbd pool=metrics'
```

3. **client.openstack** 사용자에 대해 생성된 Ceph 클라이언트 키를 확인합니다.

```
[root@ceph ~]# ceph auth list
...
[client.openstack]
 key = AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ItLdwUw==
 caps mgr = "allow *"
 caps mon = "profile rbd"
 caps osd = "profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd pool=backups, profile rbd pool=metrics"
...
```

이 예제의 **키** 값은 **AQC+vYNXgDAgAhAc8UoYt+OTz5uhV7ItLdwUwUw==**, Ceph 클라이언트 키입니다.

4. Ceph Storage 클러스터의 *파일 시스템 ID* 를 기록해 둡니다. 이 값은 클러스터의 구성 파일에서 **fsid** 설정을 사용하여 지정됩니다(**[global]** 섹션 참조):

```
[global]
fsid = 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
...
```

참고

Ceph Storage 클러스터 구성 파일에 대한 자세한 내용은 Configuration Reference (Red Hat Ceph Storage Configuration Guide)를 참조하십시오.

Ceph 클라이언트 키 및 파일 시스템 ID는 둘 다 *3장. 기존 Ceph 클러스터와 통합* 에서 나중에 사용됩니다.

2.3. 스택 사용자 초기화

stack 사용자로 director 호스트에 로그인하고 다음 명령을 실행하여 director 설정을 초기화합니다.

```
$ source ~/stackrc
```

이렇게 하면 인증 정보가 포함된 환경 변수가 director의 CLI 도구에 액세스할 수 있습니다.

2.4. 노드 등록

노드 정의 템플릿(**instackenv.json**)은 JSON 형식 파일이며 노드 등록에 필요한 하드웨어 및 전원 관리 세부 정보를 포함합니다. 예를 들면 다음과 같습니다.

```
{
    "nodes":[
        {
            "mac":[
                "bb:bb:bb:bb:bb:bb"
            ],
            "cpu":"4",
            "memory":"6144",
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"pxe_ipmitool",
            "pm_user":"admin",
            "pm_password":"p@55w0rd!",
            "pm_addr":"192.0.2.205"
        },
        {
            "mac":[
                "cc:cc:cc:cc:cc:cc"
            ],
            "cpu":"4",
            "memory":"6144",
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"pxe_ipmitool",
            "pm_user":"admin",
            "pm_password":"p@55w0rd!",
            "pm_addr":"192.0.2.206"
        },
        {
            "mac":[
                "dd:dd:dd:dd:dd:dd"
            ],
            "cpu":"4",
            "memory":"6144",
```

```
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"pxe_ipmitool",
            "pm_user":"admin",
            "pm_password":"p@55w0rd!",
            "pm_addr":"192.0.2.207"
        },
        {
            "mac":[
                "ee:ee:ee:ee:ee:ee"
            ],
            "cpu":"4",
            "memory":"6144",
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"pxe_ipmitool",
            "pm_user":"admin",
            "pm_password":"p@55w0rd!",
            "pm_addr":"192.0.2.208"
        }
        {
            "mac":[
                "ff:ff:ff:ff:ff:ff"
            ],
            "cpu":"4",
            "memory":"6144",
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"pxe_ipmitool",
            "pm_user":"admin",
            "pm_password":"p@55w0rd!",
            "pm_addr":"192.0.2.209"
        }
        {
            "mac":[
                "gg:gg:gg:gg:gg:gg"
            ],
            "cpu":"4",
            "memory":"6144",
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"pxe_ipmitool",
            "pm_user":"admin",
            "pm_password":"p@55w0rd!",
            "pm_addr":"192.0.2.210"
        }
    ]
}
```

템플릿을 생성한 후 stack 사용자의 홈 디렉터리(**/home/stack/instackenv.json**)에 파일을 저장합니다. stack 사용자를 초기화한 다음 **instackenv.json** 을 director로 가져옵니다.

```
$ source ~/stackrc
$ openstack overcloud node import ~/instackenv.json
```

이렇게 하면 템플릿을 가져와서 템플릿의 각 노드를 director에 등록합니다.

각 노드에 커널 및 ramdisk 이미지를 할당합니다.

```
$ openstack overcloud node configure <node>
```

이제 노드가 director에 등록 및 설정됩니다.

2.5. 노드 수동 태그 지정

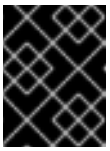
각 노드를 등록한 후 하드웨어를 검사하고 노드를 특정 프로필에 태그해야 합니다. 프로필 태그는 플레이버에 따라 노드에 일치하며, 그런 다음 플레이버가 배포 역할에 할당됩니다.

새 노드를 검사하고 태그를 지정하려면 다음 단계를 수행합니다.

1. 하드웨어 인트로스펙션을 트리거하여 각 노드의 하드웨어 속성을 검색합니다.

```
$ openstack overcloud node introspect --all-manageable --provide
```

   - **--all-manageable** 옵션은 관리 상태의 노드만 인트로스펙션합니다. 이 예제에서는 모든 것입니다.
   - **--provide** 옵션은 인트로스펙션 이후 모든 노드를 **활성** 상태로 재설정합니다.



중요

이 프로세스가 완료되었는지 확인합니다. 베어 메탈 노드의 경우 이 프로세스는 일반적으로 15분 정도 걸립니다.

2. 노드 목록을 검색하여 해당 UUID를 확인합니다.

```
$ openstack baremetal node list
```

3. 각 노드의 **properties/capabilities** 매개변수에 **profile** 옵션을 추가하여 노드를 특정 프로필에 수동으로 태그합니다.
   예를 들어 세 개의 노드를 태그하여 **compute** 프로필을 사용하도록 **control** 프로필과 다른 세 개의 노드를 태그하려면 다음을 실행합니다.

```
$ ironic node-update 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 6faba1a9-e2d8-4b7c-95a2-c7fbdc12129a add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 5e3b2f50-fcd9-4404-b0a2-59d79924b38e add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 484587b2-b3b3-40d5-925b-a26a2fa3036f add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d010460b-38f2-4800-9cc4-d69f0d067efe add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d930e613-3e14-44b9-8240-4f3559801ea6 add
properties/capabilities='profile:compute,boot_option:local'
```

profile 옵션을 추가하면 각 프로필에 노드를 태그합니다.

참고

수동 태그 지정 대신 AHC(Automated Health Check) 도구를 사용하여 벤치마킹 데이터를 기반으로 다수의 노드를 자동으로 태그합니다.

3장. 기존 CEPH 클러스터와 통합

director에서 제공하는 heat 템플릿 컬렉션에는 이미 오버클라우드를 배포하는 데 필요한 템플릿과 환경 파일이 포함되어 있습니다.

이 환경 파일은 기존 Ceph 클러스터를 배포 중인 오버클라우드에 통합하기 위해 배포(3.2절. "오버클라우드 배포") 중에 호출됩니다.

- **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml**

절차

1. director는 **ceph-ansible** 을 사용하여 기존 Ceph 클러스터와 통합하지만 **ceph-ansible** 은 기본적으로 언더클라우드에 설치되지 않습니다. 다음 명령을 입력하여 언더클라우드에 ceph-ansible 패키지를 설치합니다.

```
sudo yum install -y ceph-ansible
```

2. 통합을 구성하려면 Ceph 클러스터의 세부 정보를 director에 제공해야 합니다. 사용자 지정 환경 파일을 사용하여 **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible-external.yaml에서 사용하는 기본 설정을 재정의합니다.**

3. 다음 사용자 지정 환경 파일을 생성합니다.
/home/stack/templates/ceph-config.yaml

4. 이 파일에 **parameter_defaults:** 헤더를 추가합니다.

```
parameter_defaults:
```

5. 이 헤더에서 **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible-external.yaml** 에서 재정의할 모든 매개변수를 설정합니다. 최소한 다음 사항을 설정해야 합니다.

   - **CephClientKey**: Ceph Storage 클러스터의 Ceph 클라이언트 키입니다. 이는 2.2절. "기존 Ceph Storage 클러스터 구성" 이전에 검색한 **키** 의 값입니다. 예를 들어 **AQDLOh1VgE6FRAAFzT7Zw+Y9V6JJQAsRnRQ==**.

   - **CephClusterFSID**: Ceph Storage 클러스터의 파일 시스템 ID입니다. 이 값은 Ceph Storage 클러스터 구성 파일에서 이전에 2.2절. "기존 Ceph Storage 클러스터 구성" 에서 검색한 값 **fsid** 의 값입니다. 예를 들어 **4b5c8c0a-ff60-454b-a1b4-9747a737d19** 입니다.

   - **CephExternalMonHost**: Ceph Storage 클러스터에 있는 모든 MON 호스트의 IP의 쉼표로 구분된 목록입니다. 예: **172.16.1.7, 172.16.1.8**.

```
parameter_defaults:
  CephClientKey: AQDLOh1VgEp6FRAAFzT7Zw+Y9V6JJExQAsRnRQ==
  CephClusterFSID: 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
  CephExternalMonHost: 172.16.1.7, 172.16.1.8
```

6. 필요한 경우 다음 매개 변수와 값을 사용하여 OpenStack 풀의 이름과 클라이언트 사용자도 설정합니다.

   - **CephClientUserName: openstack**

   - **NovaRbdPoolName: vms**

- **CinderRbdPoolName: volumes**
- **GlanceRbdPoolName: 이미지**
- **CinderBackupRbdPoolName: backups**
- **GnocchiRbdPoolName: metrics**

7. 사용자 지정 환경 파일에 오버클라우드 매개변수를 추가할 수도 있습니다. 예를 들어 **vxlan** 을 **neutron** 네트워크 유형으로 설정하려면 **parameter_defaults** 에 다음을 추가합니다.

```
NeutronNetworkType: vxlan
```

3.1. 역할에 노드 및 플레이버 할당

오버클라우드 배포를 계획하려면 각 역할에 할당할 노드 수와 플레이버를 지정해야 합니다. 모든 Heat 템플릿 매개변수와 마찬가지로 이러한 역할 사양은 사용자 지정 환경 파일의 **parameter_defaults** 섹션에 선언됩니다(이 경우 **/home/stack/templates/ceph-config** 3장. *기존 Ceph 클러스터와 통합*).

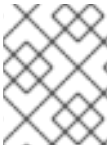
이를 위해 다음 매개변수를 사용합니다.

표 3.1. Overcloud 노드의 역할 및 플레이버

Heat 템플릿 매개변수	설명
ControllerCount	확장할 컨트롤러 노드 수
OvercloudControlFlavor	컨트롤러 노드(**컨트롤**)에 사용할 플레이버
ComputeCount	확장할 컴퓨팅 노드 수
OvercloudComputeFlavor	컴퓨팅 노드(Compute)에 사용할 플레이버

예를 들어 각 역할(Controller 및 Compute)마다 3개의 노드를 배포하도록 오버클라우드를 구성하려면 **parameter_defaults** 에 다음을 추가합니다.

```
parameter_defaults:
  ControllerCount: 3
  ComputeCount: 3
  OvercloudControlFlavor: control
  OvercloudComputeFlavor: compute
```



참고

보다 완전한 Heat 템플릿 매개변수 목록은 Director 설치 및 사용 가이드에서 CLI 툴로 Overcloud 생성 을 참조하십시오.

3.2. 오버클라우드 배포

참고

언더클라우드 설치 중에 **undercloud.conf** 파일에 **generate_service_certificate=false** 를 설정합니다. 그러지 않으면 오버클라우드를 배포할 때 신뢰 앵커를 삽입해야 합니다. 신뢰 앵커를 삽입하는 방법에 대한 자세한 내용은 *Advanced Overcloud Customization* 가이드의 Overcloud Public Endpoints에서 SSL/TLS 를 참조하십시오.

오버클라우드를 생성하려면 **openstack overcloud deploy** 명령에 대한 추가 인수가 필요합니다. 예를 들면 다음과 같습니다.

```
$ openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-
external.yaml \
  -e /home/stack/templates/ceph-config.yaml \
  -e --ntp-server pool.ntp.org \
```

위의 명령은 다음 옵션을 사용합니다.

- **--templates** - 기본 Heat 템플릿 컬렉션(예: **/usr/share/openstack-tripleo-heat-templates/**)에서 Overcloud를 생성합니다.
- **-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible-ansible-external.yaml** - 기존 Ceph 클러스터를 오버클라우드에 통합하도록 director를 설정합니다.
- **-e /home/stack/templates/ceph-config.yaml** - - **e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible-external.yaml** 로 설정된 기본값을 덮어쓰도록 사용자 지정 환경 파일을 추가합니다. 이 경우 3장. *기존 Ceph 클러스터와 통합* 에서 생성한 사용자 지정 환경 파일입니다.
- **--NTP-server pool.ntp.org** - NTP 서버를 설정합니다.

작은 정보

응답 파일을 사용하여 모든 템플릿 및 환경 파일 을 호출할 수도 있습니다. 예를 들어 다음 명령을 사용하여 동일한 오버클라우드를 배포할 수 있습니다.

```
$ openstack overcloud deploy \
  --answers-file /home/stack/templates/answers.yaml \
  --ntp-server pool.ntp.org
```

이 경우 응답 파일 **/home/stack/templates/answers.yaml** 에는 다음이 포함됩니다.

```
templates: /usr/share/openstack-tripleo-heat-templates/
environments:
  - /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml
\
  - /home/stack/templates/ceph-config.yaml \
```

자세한 내용은 Overcloud Creation에서 환경 파일 포함 을 참조하십시오.

전체 옵션 목록은 다음을 실행합니다.

```
$ openstack help overcloud deploy
```

자세한 내용은 *Director 설치 및 사용 가이드* *의 CLI 도구를 사용하여 Overcloud 생성* 을 참조하십시오.

오버클라우드 생성 프로세스가 시작되고 director가 노드를 프로비저닝합니다. 이 프로세스를 완료하는 데 다소 시간이 걸립니다. 오버클라우드 생성 상태를 보려면 **stack** 사용자로 별도의 터미널을 열고 다음을 실행합니다.

```
$ source ~/stackrc
$ openstack stack list --nested
```

그러면 외부 Ceph Storage 클러스터를 사용하도록 Overcloud가 구성됩니다. 이 클러스터를 Overcloud와 독립적으로 관리합니다. 예를 들어 OpenStack Platform director가 아닌 Ceph 관리 툴을 사용하여 Ceph Storage 클러스터를 확장합니다.

4장. 오버클라우드 액세스

director는 director 호스트에서 오버클라우드와의 상호 작용을 구성하고 인증을 지원하는 스크립트를 생성합니다. director는 이 파일(**overcloudrc**)을 **stack** 사용자의 홈 디렉터리에 저장합니다. 이 파일을 사용하려면 다음 명령을 실행합니다.

```
$ source ~/overcloudrc
```

이렇게 하면 director 호스트의 CLI에서 오버클라우드와 상호 작용하는 데 필요한 환경 변수가 로드됩니다. director 호스트와의 상호 작용으로 돌아가려면 다음 명령을 실행합니다.

```
$ source ~/stackrc
```