Red Hat OpenStack Platform 17.1

인스턴스 생성 및 관리

CLI를 사용하여 인스턴스 생성 및 관리

Last Updated: 2023-09-21

Red Hat OpenStack Platform 17.1 인스턴스 생성 및 관리

CLI를 사용하여 인스턴스 생성 및 관리

OpenStack Team
rhos-docs@redhat.com

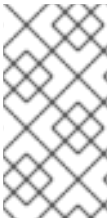
법적 공지

초록

이 가이드에서는 인스턴스 생성 및 관리를 위한 절차를 제공합니다.

차례

머리말



참고

인스턴스를 생성하는 동안 RBAC(역할 기반 액세스 제어) 공유 보안 그룹을 인스턴스에 직접 적용할 수 없습니다. 인스턴스에 RBAC-공유 보안 그룹을 적용하려면 먼저 포트를 만들고, 공유 보안 그룹을 해당 포트에 적용한 다음 해당 포트를 인스턴스에 할당해야 합니다. 포트에 보안 그룹 추가를 참조하십시오.

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.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 더 나은지 알려주십시오.

직접 문서 피드백(DDF) 기능 사용

피드백 추가 DDF 기능을 사용하여 특정 문장, 단락 또는 코드 블록에 대한 직접 의견을 제출할 수 있습니다.

1. *다중 페이지 HTML* 형식으로 문서를 봅니다.
2. 문서의 오른쪽 상단에 **피드백** 버튼이 표시되는지 확인합니다.
3. 주석 처리하려는 텍스트 부분을 강조 표시합니다.
4. **피드백 추가를** 클릭합니다.
5. 코멘트를 사용하여 **피드백 추가** 필드를 완료합니다.
6. 선택 사항: 문서 팀이 문제에 대한 자세한 설명을 위해 연락을 드릴 수 있도록 이메일 주소를 추가합니다.
7. **Submit** 을 클릭합니다.

1장. 인스턴스 정보

인스턴스는 클라우드 내부의 물리적 컴퓨팅 노드에서 실행되는 개별 가상 머신입니다. 인스턴스를 시작하려면 플레이버와 이미지 또는 부팅 가능한 볼륨이 필요합니다. 이미지를 사용하여 인스턴스를 시작하면 제공된 이미지가 부팅 가능한 운영 체제와 함께 설치된 가상 디스크가 포함된 기본 이미지가 됩니다. 각 인스턴스에는 루트 디스크가 필요하며, 이 디스크는 인스턴스 디스크로 참조합니다. Compute 서비스(nova)는 인스턴스에 지정한 플레이버의 사양과 일치하도록 인스턴스 디스크의 크기를 조정합니다.

이미지는 Image 서비스(glance)에서 관리합니다. 이미지 서비스 이미지 저장소에는 사전 정의된 여러 이미지가 포함되어 있습니다. 컴퓨팅 노드는 인스턴스에 사용 가능한 vCPU, 메모리 및 로컬 디스크 리소스를 제공합니다. Block Storage 서비스(cinder)는 사전 정의된 볼륨을 제공합니다. 인스턴스 디스크 데이터는 인스턴스를 삭제할 때 또는 블록 스토리지 서비스에서 제공하는 영구 볼륨에 삭제되는 임시 스토리지에 저장됩니다.

계산 서비스는 필요에 따라 인스턴스를 제공하는 중앙 구성 요소입니다. Compute 서비스는 인스턴스를 생성, 예약 및 관리하고 인증을 위해 ID 서비스, 인스턴스를 시작하는 데 사용되는 이미지의 이미지 서비스, 사용자 및 관리 인터페이스에 대한 대시보드 서비스(horizon)와 상호 작용합니다. 클라우드 사용자는 인스턴스를 생성하고 관리할 때 Compute 서비스와 상호 작용합니다. OpenStack CLI 또는 대시보드를 사용하여 인스턴스를 생성하고 관리할 수 있습니다.

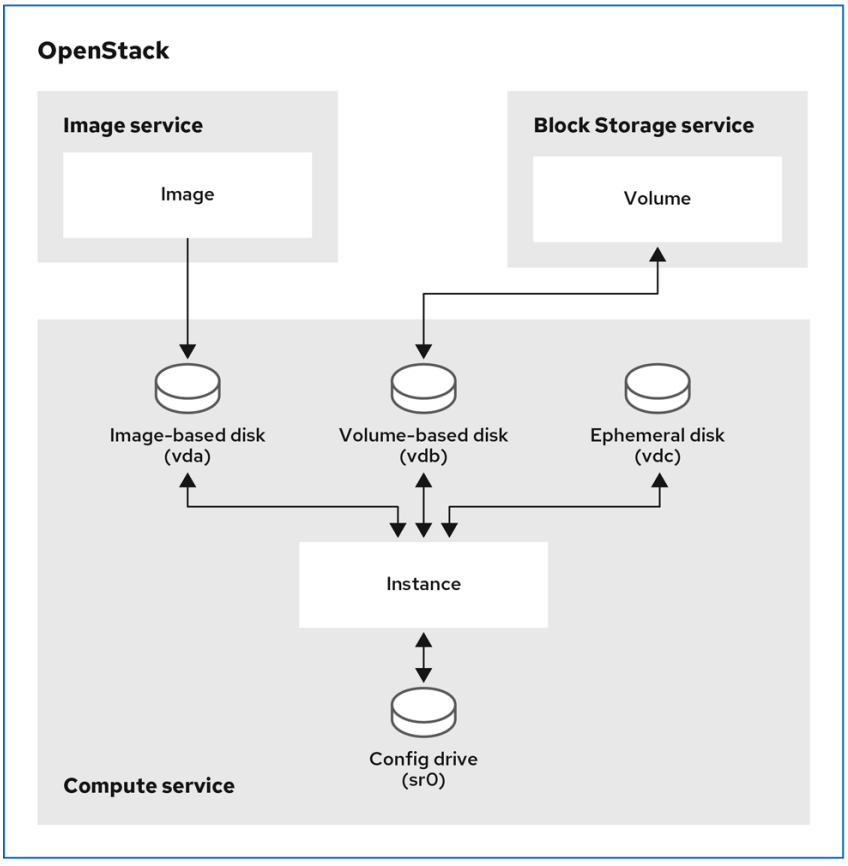
2장. 인스턴스 부팅 소스

인스턴스의 부팅 소스는 이미지 또는 부팅 가능한 볼륨이 될 수 있습니다. 이미지에서 부팅하는 인스턴스의 인스턴스 디스크는 Compute 서비스에서 제어하고 인스턴스가 삭제될 때 삭제됩니다. 볼륨에서 부팅하는 인스턴스의 인스턴스 디스크는 블록 스토리지 서비스에 의해 제어되며 원격으로 저장됩니다.

이미지에는 부팅 가능한 운영 체제가 포함되어 있습니다. Image 서비스(glance)는 이미지 스토리지 및 관리를 제어합니다. 동일한 기본 이미지에서 여러 인스턴스를 시작할 수 있습니다. 각 인스턴스는 기본 이미지의 사본에서 실행됩니다. 인스턴스에 대한 모든 변경 사항은 기본 이미지에 영향을 미치지 않습니다.

부팅 가능한 볼륨은 부팅 가능한 운영 체제가 포함된 이미지에서 생성된 블록 스토리지 볼륨입니다. 인스턴스는 인스턴스가 삭제될 때 부팅 가능한 볼륨을 사용하여 인스턴스 데이터를 유지할 수 있습니다. 인스턴스를 시작할 때 기존 영구 루트 볼륨을 사용할 수 있습니다. 인스턴스가 삭제될 때 인스턴스 데이터를 저장할 수 있도록 이미지에서 인스턴스를 시작할 때 영구 스토리지를 생성할 수도 있습니다. 볼륨 스냅샷에서 인스턴스를 생성할 때 새 영구 스토리지 볼륨이 자동으로 생성됩니다.

다음 다이어그램에서는 인스턴스를 시작할 때 생성할 수 있는 인스턴스 디스크 및 스토리지를 보여줍니다. 생성된 실제 인스턴스 디스크 및 스토리지는 사용되는 부팅 소스 및 플레이버에 따라 다릅니다.



145_OpenStack_0321

3장. 인스턴스 스토리지 유형

인스턴스에 사용할 수 있는 가상 스토리지는 인스턴스를 시작하는 데 사용되는 플레이버에서 정의합니다. 다음 가상 스토리지 리소스를 인스턴스와 연결할 수 있습니다.

- 인스턴스 디스크
- 임시 스토리지
- 스왑 스토리지
- 영구 블록 스토리지 볼륨
- 구성 드라이브

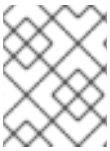
3.1. 인스턴스 디스크

인스턴스 데이터를 저장하기 위해 생성된 인스턴스 디스크는 인스턴스를 생성하는 데 사용하는 부팅 소스에 따라 다릅니다. 이미지에서 부팅하는 인스턴스의 인스턴스 디스크는 Compute 서비스에서 제어하고 인스턴스가 삭제될 때 삭제됩니다. 볼륨에서 부팅하는 인스턴스의 인스턴스 디스크는 블록 스토리지 서비스에서 제공하는 영구 볼륨입니다.

3.2. 인스턴스 임시 스토리지

임시 디스크를 구성하는 플레이버를 선택하여 인스턴스에 임시 디스크가 생성되도록 지정할 수 있습니다. 이 임시 스토리지는 인스턴스에서 사용할 수 있는 빈 추가 디스크입니다. 이 스토리지 값은 instance 플레이버에서 정의합니다. 기본값은 0이므로 보조 임시 스토리지가 생성되지 않습니다.

임시 디스크는 플러그인 하드 드라이브 또는 지문 드라이브와 동일한 방식으로 나타납니다. lsblk 명령을 사용하여 확인할 수 있는 블록 장치로 사용할 수 있습니다. 마운트하고 사용할 수 있지만 일반적으로 블록 장치를 사용합니다. 연결된 인스턴스 이외의 디스크를 보존하거나 참조할 수 없습니다.



참고

임시 스토리지 데이터는 인스턴스 스냅샷에 포함되지 않으며 보류되지 않은 인스턴스에서는 사용할 수 없습니다.

3.3. 인스턴스 스왑 스토리지

스왑 디스크를 구성하는 플레이버를 선택하여 인스턴스의 스왑 디스크가 생성되도록 지정할 수 있습니다. 이 스왑 스토리지는 실행 중인 운영 체제의 스왑 공간으로 사용하기 위해 인스턴스에서 사용할 수 있는 추가 디스크입니다.

3.4. 인스턴스 블록 스토리지

블록 스토리지 볼륨은 실행 중인 인스턴스의 상태와 관계없이 인스턴스에서 사용할 수 있는 영구 스토리지입니다. 여러 블록 장치를 인스턴스에 연결할 수 있으며 그 중 하나는 부팅 가능한 볼륨일 수 있습니다.

참고

인스턴스 디스크 데이터에 블록 스토리지 볼륨을 사용하면 새 볼륨이 생성되도록 요청하는 새 이미지로 인스턴스를 다시 빌드하는 경우에도 블록 스토리지 볼륨이 다시 빌드됩니다.

3.5. 구성 드라이브

부팅 시 구성 드라이브를 인스턴스에 연결할 수 있습니다. 구성 드라이브가 인스턴스에 읽기 전용 드라이브로 표시됩니다. 인스턴스는 이 드라이브를 마운트하고 해당 드라이브에서 파일을 읽을 수 있습니다. 구성 드라이브를 **cloud-init** 정보의 소스로 사용할 수 있습니다. 구성 드라이브는 서버 부트스트랩을 위해 **cloud-init** 와 결합되고 대용량 파일을 인스턴스에 전달하려는 경우 유용합니다. 예를 들어 **cloud-init** 를 구성 드라이브를 자동으로 마운트하고 초기 인스턴스 부팅 중에 설정 스크립트를 실행하도록 구성할 수 있습니다. 구성 드라이브는 **config-2** 의 볼륨 레이블을 사용하여 생성되며 부팅 시 인스턴스에 연결됩니다. 구성 드라이브에 전달된 추가 파일의 콘텐츠는 구성 드라이브의 **openstack/{version}/** 디렉터리에 있는 **user_data** 파일에 추가됩니다. **cloud-init** 는 이 파일에서 사용자 데이터를 검색합니다.

4장. 인스턴스 플레이버

인스턴스 플레이버는 인스턴스의 가상 하드웨어 프로필을 지정하는 리소스 템플릿입니다. 인스턴스를 시작할 때 플레이버를 선택하여 인스턴스에 할당할 가상 리소스를 지정합니다. 플레이버는 가상 CPU 수, RAM 용량, 루트 디스크 크기, 보조 임시 스토리지 및 스왑 디스크를 포함하여 가상 스토리지의 크기를 정의합니다. 클라우드 내에서 프로젝트에 정의된 사용 가능한 플레이버 세트에서 플레이버를 선택합니다.

5장. 인스턴스 생성

인스턴스를 생성하려면 플레이버, 부팅 소스, 네트워크, 키 쌍 및 보안 그룹과 같은 기타 RHOSP(Red Hat OpenStack Platform) 구성 요소를 사용할 수 있어야 합니다. 이러한 구성 요소는 인스턴스 생성에 사용되며 기본적으로 사용할 수 없습니다.

인스턴스를 생성할 때 인스턴스에 필요한 부팅 가능한 운영 체제, 인스턴스에 필요한 하드웨어 프로필이 있는 플레이버, 인스턴스를 연결하려는 네트워크, 데이터 볼륨 및 임시 스토리지와 같은 추가 스토리지가 있는 부팅 소스를 선택합니다.

5.1. 사전 요구 사항

- 필요한 이미지 또는 볼륨은 부팅 소스로 사용할 수 있습니다.
  - 이미지를 생성하는 방법에 대한 자세한 내용은 이미지 *생성* 및 관리를 참조하십시오.
  - 볼륨을 생성하는 방법에 대한 자세한 내용은 *영구 스토리지 구성* *가이드에서 블록 스토리지 볼륨 생성* 을 참조하십시오.
  - 인스턴스의 부팅 소스에 사용할 수 있는 옵션에 대한 자세한 내용은 인스턴스 부팅 소스 를 참조하십시오.
- 필요한 CPU, 메모리 및 스토리지 용량을 지정하는 플레이버를 사용할 수 있습니다. 플레이버 설정은 선택한 이미지에서 지정한 디스크 및 메모리 크기에 대한 최소 요구 사항을 충족해야 합니다. 그렇지 않으면 인스턴스가 시작되지 않습니다.
- 필요한 네트워크를 사용할 수 있습니다. 네트워크를 생성하는 방법에 대한 자세한 내용은 *Red Hat OpenStack Platform 네트워킹 구성* 가이드에서 네트워크 생성 을 참조하십시오.

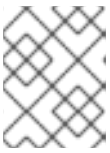
5.2. 이미지에서 인스턴스 생성

이미지를 부팅 소스로 사용하여 인스턴스를 생성할 수 있습니다.

절차

1. 인스턴스에 필요한 하드웨어 프로필이 있는 플레이버의 이름 또는 ID를 검색합니다.

```
$ openstack flavor list
```



참고

이미지가 성공적으로 부팅될 수 있는 충분한 크기의 플레이버를 선택합니다. 그렇지 않으면 인스턴스가 시작되지 않습니다.

2. 인스턴스에 필요한 소프트웨어 프로필이 있는 이미지의 이름 또는 ID를 검색합니다.

```
$ openstack image list
```

필요한 이미지를 사용할 수 없는 경우 새 이미지를 다운로드하거나 생성할 수 있습니다. 클라우드 이미지를 생성하거나 다운로드하는 방법에 대한 자세한 내용은 이미지 생성 을 참조하십시오.

참고

인스턴스에 26개 이상의 볼륨을 연결해야 하는 경우 인스턴스를 생성하는 데 사용하는 이미지에 다음과 같은 속성이 있어야 합니다.

- **hw_scsi_model=virtio-scsi**
- **hw_disk_bus=scsi**

3. 인스턴스를 연결할 네트워크의 이름 또는 ID를 검색합니다.

```
$ openstack network list
```

4. 인스턴스를 생성합니다.

```
$ openstack server create --flavor <flavor> \
  --image <image> --network <network> \
  --wait myInstanceFromImage
```

- & **lt;flavor** >를 1단계에서 검색한 플레이버의 이름 또는 ID로 바꿉니다.
- & **lt;image** >를 2단계에서 검색한 이미지의 이름 또는 ID로 바꿉니다.
- & **lt;network** >를 3단계에서 검색한 네트워크의 이름 또는 ID로 바꿉니다. 필요에 따라 **--network** 옵션을 두 번 이상 사용하여 인스턴스를 여러 네트워크에 연결할 수 있습니다.

5.3. 부팅 가능한 볼륨에서 인스턴스 생성

부팅 소스로 부팅 볼륨을 사용하여 인스턴스를 생성할 수 있습니다. 오류가 발생할 경우 인스턴스 데이터의 가용성을 개선해야 하는 경우 볼륨에서 인스턴스를 부팅합니다.

참고

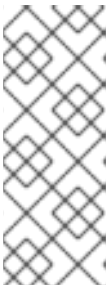
인스턴스 디스크 데이터에 블록 스토리지 볼륨을 사용하면 새 볼륨이 생성되도록 요청하는 새 이미지로 인스턴스를 다시 빌드하는 경우에도 블록 스토리지 볼륨이 다시 빌드됩니다.

절차

1. 인스턴스에 필요한 소프트웨어 프로필이 있는 이미지의 이름 또는 ID를 검색합니다.

```
$ openstack image list
```

필요한 이미지를 사용할 수 없는 경우 새 이미지를 다운로드하거나 생성할 수 있습니다. 클라우드 이미지를 생성하거나 다운로드하는 방법에 대한 자세한 내용은 이미지 생성 을 참조하십시오.



참고

인스턴스에 26개 이상의 볼륨을 연결해야 하는 경우 인스턴스를 생성하는 데 사용하는 이미지에 다음과 같은 속성이 있어야 합니다.

- **hw_scsi_model=virtio-scsi**
- **hw_disk_bus=scsi**

2. 이미지에서 부팅 가능한 볼륨을 생성합니다.

```
$ openstack volume create --image <image> \
--size <size_gb> --bootable myBootableVolume
```

- < **image** >를 볼륨에 쓸 이미지 이름 또는 ID로 1단계에서 검색했습니다.
- & **lt;size_gb&** gt;를 볼륨 크기(GB)로 바꿉니다.

3. 인스턴스에 필요한 하드웨어 프로필이 있는 플레이버의 이름 또는 ID를 검색합니다.

```
$ openstack flavor list
```

4. 인스턴스를 연결할 네트워크의 이름 또는 ID를 검색합니다.

```
$ openstack network list
```

5. 부팅 가능한 볼륨이 있는 인스턴스를 생성합니다.

```
$ openstack server create --flavor <flavor> \
 --volume myBootableVolume --network <network> \
 --wait myInstanceFromVolume
```

- & **lt;flavor** >를 3단계에서 검색한 플레이버의 이름 또는 ID로 바꿉니다.
- & **lt;network** >를 4단계에서 검색한 네트워크의 이름 또는 ID로 바꿉니다. 필요에 따라 **--network** 옵션을 두 번 이상 사용하여 인스턴스를 여러 네트워크에 연결할 수 있습니다.

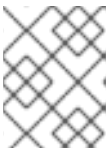
5.4. SR-IOV 네트워크 인터페이스를 사용하여 인스턴스 생성

SR-IOV(단일 루트 I/O 가상화) 네트워크 인터페이스로 인스턴스를 생성하려면 필요한 SR-IOV 포트를 생성해야 합니다.

절차

1. 인스턴스에 필요한 하드웨어 프로필이 있는 플레이버의 이름 또는 ID를 검색합니다.

```
$ openstack flavor list
```



참고

이미지가 성공적으로 부팅될 수 있는 충분한 크기의 플레이버를 선택합니다. 그렇지 않으면 인스턴스가 시작되지 않습니다.

작은 정보

필요한 정책이 있는 플레이버를 선택하여 PCI 패스스루 장치 및 SR-IOV 인터페이스에 대해 인스턴스에 적용되는 NUMA 선호도 정책을 지정할 수 있습니다. 사용 가능한 정책에 대한 자세한 내용은 *인스턴스 생성 가이드를 위한 Compute 서비스 구성 가이드의* *플레이버 메타데이터 의 Instance PCI NUMA 선호도 정책을* 참조하십시오. NUMA 선호도 정책을 사용하여 플레이버를 선택하는 경우 사용하는 이미지에 동일한 NUMA 선호도 정책 또는 NUMA 선호도 정책이 없어야 합니다.

2. 인스턴스에 필요한 소프트웨어 프로필이 있는 이미지의 이름 또는 ID를 검색합니다.

```
$ openstack image list
```

필요한 이미지를 사용할 수 없는 경우 새 이미지를 다운로드하거나 생성할 수 있습니다. 클라우드 이미지를 생성하거나 다운로드하는 방법에 대한 자세한 내용은 이미지 생성 을 참조하십시오.

작은 정보

필요한 정책이 있는 이미지를 선택하여 PCI 패스스루 장치 및 SR-IOV 인터페이스에 대해 인스턴스에 적용되는 NUMA 선호도 정책을 지정할 수 있습니다. 사용 가능한 정책에 대한 자세한 내용은 *인스턴스 생성 가이드를 위한 Compute 서비스 구성 가이드의 플레이버 메타데이터 의 Instance PCI NUMA 선호도 정책을* 참조하십시오. NUMA 선호도 정책을 사용하여 이미지를 선택하는 경우 사용하는 플레이버에 동일한 NUMA 선호도 정책 또는 NUMA 선호도 정책이 없어야 합니다.

3. 인스턴스를 연결할 네트워크의 이름 또는 ID를 검색합니다.

```
$ openstack network list
```

4. SR-IOV 인터페이스에 필요한 포트 유형을 생성합니다.

```
$ openstack port create --network <network> \
 --vnic-type <vnic_type> mySriovPort
```

- & **lt;network** >를 3단계에서 검색한 네트워크의 이름 또는 ID로 바꿉니다.
- & **lt;vnic_type&** gt;을 다음 값 중 하나로 바꿉니다.
  - **direct**: 직접 모드 SR-IOV 가상 기능(VF) 포트를 생성합니다.
  - **direct-physical**: 직접 모드 SR-IOV 물리적 기능(PF) 포트를 생성합니다.
  - **MacVTap**: MacVTap을 사용하여 virtio 인터페이스를 인스턴스에 노출하는 간접 모드 SR-IOV VF 포트를 생성합니다.

5. 인스턴스를 생성합니다.

```
$ openstack server create --flavor <flavor> \
  --image <image> --port <port> \
  --wait mySriovInstance
```

- & **lt;flavor** >를 1단계에서 검색한 플레이버의 이름 또는 ID로 바꿉니다.
- & **lt;image** >를 2단계에서 검색한 이미지의 이름 또는 ID로 바꿉니다.
- & **lt;port** >를 4단계에서 생성한 포트의 이름 또는 ID로 바꿉니다.

5.5. 포트에서 NUMA 선호도를 사용하여 인스턴스 생성

포트에서 NUMA 선호도를 사용하여 인스턴스를 생성하려면 필요한 NUMA 선호도 정책을 사용하여 포트를 생성한 다음 인스턴스를 생성할 때 포트를 지정합니다.

참고

포트 NUMA 선호도 정책은 플레이버, 이미지 및 PCI NUMA 선호도 정책보다 우선 순위가 높습니다. 클라우드 Operator는 각 PCI 패스스루 장치에 대한 기본 NUMA 선호도 정책을 설정할 수 있습니다. 인스턴스 플레이버, 이미지 또는 포트를 사용하여 인스턴스에 적용된 기본 NUMA 선호도 정책을 덮어쓸 수 있습니다.

사전 요구 사항

- 클라우드 플랫폼에서 **port-numa-affinity-policy** 확장을 활성화해야 합니다.
- 서비스 플러그인은 Networking 서비스(neutron)에서 구성해야 합니다.

프로세스

1. 필요한 NUMA 선호도 정책을 사용하여 포트를 생성합니다.

```
$ openstack port create --network <network> \
 [--numa-policy-required | --numa-policy-preferred | --numa-policy-legacy] \
 myNUMAAffinityPort
```

   - & **lt;network** >를 인스턴스를 연결하려는 테넌트 네트워크의 이름 또는 ID로 바꿉니다.
   - 다음 옵션 중 하나를 사용하여 포트에 적용할 NUMA 선호도 정책을 지정합니다.
     - **--NUMA-policy-required** - 이 포트를 예약하는 데 필요한 NUMA 선호도 정책입니다.
     - **--NUMA-policy-preferred** - 이 포트를 예약하는 데 선호되는 NUMA 선호도 정책입니다.
     - **--NUMA-policy-legacy** - 이 포트를 예약하기 위해 레거시 모드를 사용하는 NUMA 선호도 정책입니다.

2. 인스턴스를 생성합니다.

```
$ openstack server create --flavor <flavor> \
 --image <image> --port <port> \
 --wait myNUMAAffinityInstance
```

   - & **lt;flavor** >를 인스턴스에 필요한 하드웨어 프로필이 있는 플레이버의 이름 또는 ID로 바꿉니다.
   - & **lt;image** >를 인스턴스에 필요한 소프트웨어 프로필이 있는 이미지의 이름 또는 ID로 바꿉니다.
   - & **lt;port** >를 1단계에서 생성한 포트의 이름 또는 ID로 바꿉니다.

5.6. 추가 리소스

- 사용자 지정 인스턴스 생성

6장. 보장된 최소 대역폭 QOS로 인스턴스 생성

QoS(Quality of Service) 정책을 사용하여 보장된 최소 대역폭을 요청하는 인스턴스를 생성할 수 있습니다.

최소 대역폭 규칙이 보장되는 QoS 정책은 특정 물리적 네트워크의 포트에 할당됩니다. 구성된 포트를 사용하는 인스턴스를 생성할 때 Compute 스케줄링 서비스는 이 요청을 충족하는 인스턴스에 대해 호스트를 선택합니다. Compute 스케줄링 서비스는 인스턴스를 배포할 호스트를 선택하기 전에 각 물리적 인터페이스에서 다른 인스턴스에서 예약한 대역폭의 양을 배치 서비스에 확인합니다.

제한 사항/제한

- 새 인스턴스를 생성할 때 보장된 최소 대역폭 QoS 정책만 할당할 수 있습니다. 컴퓨팅 서비스는 생성 또는 이동 작업 중 인스턴스의 리소스 사용량만 업데이트하므로, 보장된 최소 대역폭 QoS 정책을 인스턴스에 할당할 수 없습니다. 즉, 인스턴스에 사용 가능한 최소 대역폭을 보장할 수 없습니다.

사전 요구 사항

- 최소 대역폭 규칙이 있는 QoS 정책을 사용할 수 있습니다. 자세한 내용은 *Red Hat OpenStack Platform 네트워킹 구성 가이드에서 QoS(Quality of Service) 정책 구성*을 참조하십시오.

프로세스

1. 사용 가능한 QoS 정책을 나열합니다.

```
(overcloud)$ openstack network qos policy list
```

```
--------------------------------------------------------------+
| ID                                   | Name    | Shared | Default | Project                          |
--------------------------------------------------------------+
| 6d771447-3cf4-4ef1-b613-945e990fa59f | policy2 | True   | False   |
ba4de51bf7694228a350dd22b7a3dc24 |
| 78a24462-e3c1-4e66-a042-71131a7daed5 | policy1 | True   | False   |
ba4de51bf7694228a350dd22b7a3dc24 |
| b80acc64-4fc2-41f2-a346-520d7cfe0e2b | policy0 | True   | False   |
ba4de51bf7694228a350dd22b7a3dc24 |
--------------------------------------------------------------+
```

2. 사용 가능한 각 정책의 규칙을 확인하여 필요한 최소 대역폭이 있는지 확인합니다.

```
(overcloud)$ openstack network qos policy show policy0
```

```
-----------------------------------------------------------------------------------------------+
| Field       | Value                                                                            |
-----------------------------------------------------------------------------------------------+
| description |
|
| id          | b80acc64-4fc2-41f2-a346-520d7cfe0e2b
|
| is_default  | False
|
| location    | cloud=', project.domain_id=, project.domain_name='Default,
project.id=ba4de51bf7694228a350dd22b7a3dc24, project.name=admin,
```

```
region_name=regionOne, zone=
|
| name        | policy0
|
| project_id  | ba4de51bf7694228a350dd22b7a3dc24
|
| rules       | [{min_kbps: 100000, direction: egress, id: d46218fe-9218-4e96-952b-
9f45a5cb3b3c, qos_policy_id: b80acc64-4fc2-41f2-a346-520d7cfe0e2b, type:
minimum_bandwidth}, {min_kbps: 100000, direction: ingress, id: 1202c4e3-a03a-464c-80d5-
0bf90bb74c9d, qos_policy_id: b80acc64-4fc2-41f2-a346-520d7cfe0e2b, type:
minimum_bandwidth}] |
| shared      | True
|
| tags        | []
|
-----------------------------------------------------------------------------------------+
```

3. 적절한 정책에서 포트를 생성합니다.

```
(overcloud)$ openstack port create port-normal-qos --network net0 --qos-policy policy0
```

4. 사용할 NIC 포트를 지정하여 인스턴스를 생성합니다.

```
$ openstack server create --flavor cirros256 --image cirros-0.3.5-x86_64-disk --nic port-
id=port-normal-qos --wait qos_instance
```

출력의 "ACTIVE" 상태는 요청된 보장된 최소 대역폭을 제공할 수 있는 호스트에서 인스턴스를 성공적으로 생성했음을 나타냅니다.

6.1. 인스턴스에서 보장된 최소 대역폭 QOS 제거

인스턴스에서 보장된 최소 대역폭 QoS 정책 제한을 해제하려면 인터페이스를 분리할 수 있습니다.

절차

- 인터페이스를 분리하려면 다음 명령을 입력합니다.

```
$ openstack server remove port <vm_name|vm_id> <port_name|port_id>
```

7장. VDPA 인터페이스를 사용하여 인스턴스 생성

VDPA 유형의 vNIC 유형이 있는 인스턴스의 포트를 요청하여 VDPA 인터페이스를 사용하여 인스턴스를 생성할 수 있습니다.

제한

- VDPA 인터페이스가 있는 인스턴스를 일시 중지하거나 실시간 마이그레이션할 수 없습니다.
- 인스턴스에서 VDPA 인터페이스를 분리한 다음 인스턴스에 다시 연결할 수 없습니다.

프로세스

1. 물리적 네트워크에 매핑된 네트워크를 생성합니다.

```
$ openstack network create vdpa_network \
 --provider-physical-network tenant \
 --provider-network-type vlan \
 --provider-segment 1337
```

2. 네트워크의 서브넷을 생성합니다.

```
$ openstack subnet create vdpa_subnet \
 --network vdpa_net1 \
 --subnet-range 192.0.2.0/24 \
 --dhcp
```

3. VDPA 사용 NIC에서 포트를 생성합니다.

```
$ openstack port create vdpa_direct_port \
 --network vdpa_network \
 --vnic-type vdpa \
```

4. 사용할 NIC 포트를 지정하여 인스턴스를 생성합니다.

```
$ openstack server create vdpa_instance \
 --flavor cirros256 --image cirros-0.3.5-x86_64-disk \
 --nic port-id=vdpa_direct_port --wait
```

출력의 "ACTIVE" 상태는 요청된 VDPA 인터페이스를 제공할 수 있는 호스트에서 인스턴스를 성공적으로 생성했음을 나타냅니다.

8장. 인스턴스 업데이트

영구 볼륨 스토리지, 네트워크 인터페이스 또는 공용 IP 주소와 같은 실행 중인 인스턴스에서 추가 리소스를 추가하고 제거할 수 있습니다. 인스턴스 메타데이터와 인스턴스가 속하는 보안 그룹을 업데이트할 수도 있습니다.

8.1. 인스턴스에 네트워크 연결

실행 중인 인스턴스에 네트워크를 연결할 수 있습니다. 인스턴스에 네트워크를 연결하면 Compute 서비스에서 인스턴스의 네트워크에 포트를 생성합니다. 기본 보안 그룹을 사용하고 네트워크에는 서브넷이 하나뿐인 경우 네트워크를 사용하여 인스턴스에 네트워크 인터페이스를 연결합니다.

절차

1. 사용 가능한 네트워크를 식별하고 인스턴스에 연결할 네트워크의 이름 또는 ID를 기록해 둡니다.

```
(overcloud)$ openstack network list
```

   필요한 네트워크를 사용할 수 없는 경우 새 네트워크를 생성합니다.

```
(overcloud)$ openstack network create <network>
```

2. 인스턴스에 네트워크를 연결합니다.

```
$ openstack server add network <instance> <network>
```

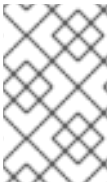
   - & **lt;instance** >를 네트워크를 연결할 인스턴스의 이름 또는 ID로 바꿉니다.
   - & **lt;network** >를 인스턴스에 연결할 네트워크의 이름 또는 ID로 바꿉니다.

추가 리소스

- *명령줄 인터페이스 참조* 의 OpenStack network create 명령 .
- *Red Hat OpenStack Platform 네트워킹 구성* 가이드에서 네트워크 생성.

8.2. 인스턴스에서 네트워크 분리

인스턴스에서 네트워크를 분리할 수 있습니다.



참고

네트워크를 분리하면 모든 네트워크 포트가 분리됩니다. 인스턴스에 네트워크에 여러 포트가 있고 해당 포트 중 하나만 분리하려면 인스턴스에서 포트 분리 절차에 따라 포트를 분리합니다.

절차

1. 인스턴스에 연결된 네트워크를 식별합니다.

```
(overcloud)$ openstack server show <instance>
```

2. 인스턴스에서 네트워크를 분리합니다.

```
$ openstack server remove network <instance> <network>
```

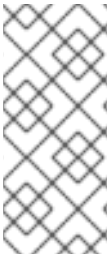
- & **lt;instance** >를 네트워크를 제거하려는 인스턴스의 이름 또는 ID로 바꿉니다.
- & **lt;network** >를 인스턴스에서 제거하려는 네트워크의 이름 또는 ID로 바꿉니다.

8.3. 인스턴스에 포트 연결

포트를 사용하여 실행 중인 인스턴스에 네트워크 인터페이스를 연결할 수 있습니다. 한 번에 하나의 인스턴스에만 포트를 연결할 수 있습니다. 사용자 지정 보안 그룹을 사용하려는 경우 또는 네트워크에 여러 서브넷이 있는 경우 포트를 사용하여 네트워크 인터페이스를 인스턴스에 연결합니다.

작은 정보

네트워크를 사용하여 네트워크 인터페이스를 연결하면 포트가 자동으로 생성됩니다. 자세한 내용은 인스턴스에 네트워크 연결을 참조하십시오.



참고

RHOSP(Red Hat OpenStack Platform)는 각 인스턴스에 대해 최대 24개의 인터페이스를 제공합니다. 기본적으로 인스턴스를 재부팅하기 전에 최대 16개의 PCIe 장치를 인스턴스에 추가할 수 있습니다. RHOSP 관리자는 인스턴스를 재부팅해야 장치를 추가하기 전에 **NovaLibvirtNumPciePorts** 매개변수를 사용하여 인스턴스에 추가할 수 있는 PCIe 장치 수를 구성할 수 있습니다.

사전 요구 사항

- SR-IOV vNIC를 사용하여 포트를 인스턴스에 연결하는 경우 적절한 물리적 네트워크의 호스트에 사용 가능한 SR-IOV 장치가 있어야 하며 인스턴스에 사용 가능한 PCIe 슬롯이 있어야 합니다.

프로세스

1. 인스턴스에 연결할 포트를 생성합니다.

```
$ openstack port create --network <network> [--vnic-type <vnic-type>] <port>
```

- & **lt;network** >를 포트를 생성할 네트워크의 이름 또는 ID로 바꿉니다.
- 선택 사항: SR-IOV 포트를 생성하려면 < **vnic-type** >을 다음 값 중 하나로 교체합니다.
  - **direct**: 직접 모드 SR-IOV 가상 기능(VF) 포트를 생성합니다.
  - **direct-physical**: 직접 모드 SR-IOV 물리적 기능(PF) 포트를 생성합니다.
  - **MacV** Tap: MacVTap 장치를 통해 인스턴스에 연결된 SR-IOV 포트를 생성합니다.
- & **lt;port** >를 인스턴스에 연결할 포트의 이름 또는 ID로 바꿉니다.

2. 인스턴스에 포트를 연결합니다.

```
$ openstack server add port <instance> <port>
```

- & **lt;instance** >를 포트를 연결할 인스턴스의 이름 또는 ID로 바꿉니다.
- & **lt;port** >를 인스턴스에 연결할 포트의 이름 또는 ID로 바꿉니다.

3. 포트가 인스턴스에 연결되어 있는지 확인합니다.

```
$ openstack port list --device-id <instance_UUID>
```

& **lt;instance_UUID** >를 포트를 연결한 인스턴스의 UUID로 바꿉니다.

추가 리소스

- *명령줄 인터페이스 참조* 의 OpenStack port create 명령 .

8.4. 인스턴스에서 포트 분리

인스턴스에서 포트를 분리할 수 있습니다.

절차

1. 인스턴스에 연결된 포트를 확인합니다.

```
(overcloud)$ openstack server show <instance>
```

2. 인스턴스에서 포트를 분리합니다.

```
$ openstack server remove port <instance> <port>
```

- & **lt;instance** >를 포트를 제거하려는 인스턴스의 이름 또는 ID로 바꿉니다.
- & **lt;port** >를 인스턴스에서 제거하려는 포트의 이름 또는 ID로 바꿉니다.

8.5. 인스턴스에 볼륨 연결

영구 스토리지를 위해 볼륨을 인스턴스에 연결할 수 있습니다. 볼륨이 다중 연결 볼륨으로 구성되지 않은 경우 한 번에 하나의 인스턴스에만 볼륨을 연결할 수 있습니다. 다중 연결 볼륨 생성에 대한 자세한 내용은 여러 인스턴스에 연결할 수 있는 볼륨 을 참조하십시오.

사전 요구 사항

- 다중 연결 볼륨을 연결하려면 환경 변수 **OS_COMPUTE_API_VERSION** 이 2.60 이상으로 설정됩니다.
- 인스턴스가 완전히 작동 중이거나 완전히 중지되었습니다. 인스턴스가 부팅 중이거나 종료되는 경우에는 볼륨을 인스턴스에 연결할 수 없습니다.
- 인스턴스에 26개 이상의 볼륨을 연결하려면 인스턴스를 생성하는 데 사용한 이미지에 다음과 같은 속성이 있어야 합니다.
  - **hw_scsi_model=virtio-scsi**
  - **hw_disk_bus=scsi**

프로세스

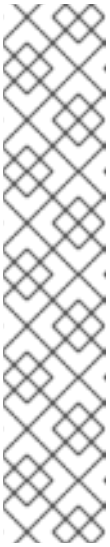
1. 사용 가능한 볼륨을 식별하고 인스턴스에 연결할 볼륨의 이름 또는 ID를 확인합니다.

```
(overcloud)$ openstack volume list
```

2. 볼륨을 인스턴스에 연결합니다.

```
$ openstack server add volume <instance> <volume>
```

- & **lt;instance** >를 볼륨을 연결할 인스턴스의 이름 또는 ID로 바꿉니다.
- & **lt;volume** >을 인스턴스에 연결할 볼륨의 이름 또는 ID로 바꿉니다.



참고

명령에서 다음 오류를 반환하는 경우 인스턴스에 연결하도록 선택한 볼륨이 다중 연결 볼륨이므로 Compute API 버전 2.60 이상을 사용해야 합니다.

```
Multiattach volumes are only supported starting with compute API version
2.60. (HTTP 400) (Request-ID: req-3a969c31-e360-4c79-a403-
75cc6053c9e5)
```

OS_COMPUTE_API_VERSION=2.72 환경 변수를 설정하거나 볼륨을 인스턴스에 추가할 때 **--os-compute-api-version** 인수를 포함할 수 있습니다.

```
$ openstack --os-compute-api-version 2.72 server add volume <instance>
<volume>
```

작은 정보

SHELVED 또는 **SHELVED_OFFLOADED** 인 인스턴스에 볼륨을 추가하려면 **--os-compute-api-version 2.20** 이상을 지정합니다.

3. 볼륨이 인스턴스 또는 인스턴스에 연결되어 있는지 확인합니다.

```
$ openstack volume show <volume>
```

& **lt;volume** >을 표시할 볼륨의 이름 또는 ID로 바꿉니다.

출력 예:

```
+----------------------------------------------------+---------------------+---------+-----+---------------
-------------------------------------------------------------------------+
| ID                                                 | Name                | Status | Size| Attached to
+----------------------------------------------------+---------------------+---------+-----+---------------
-------------------------------------------------------------------------+
| f3fb92f6-c77b-429f-871d-65b1e3afa750 | volMultiattach | in-use |   50 | Attached to
instance1 on /dev/vdb Attached to instance2 on /dev/vdb  |
+----------------------------------------------------+---------------------+---------+-----+---------------
-------------------------------------------------------------------------+
```

8.6. 인스턴스에 연결된 볼륨 보기

특정 인스턴스에 연결된 볼륨을 볼 수 있습니다.

사전 요구 사항

- **python-openstackclient 5.5.0** 을 사용하고 있습니다.

절차

- 인스턴스에 연결된 볼륨을 나열합니다.

```
$ openstack server volume list <instance>
+--------------------+---------+--------------------+----------------------+
| ID                 | Device  | Server ID          | Volume ID            |
+--------------------+---------+--------------------+----------------------+
| 1f9dcb02-9a20-4a4b- | /dev/vda | ab96b635-1e63-4487- | 1f9dcb02-9a20-4a4b-9f |
| 9f25-c7846a1ce9e8  |         | a85c-854197cd537b  | 25-c7846a1ce9e8      |
+--------------------+---------+--------------------+----------------------+
```

8.7. 인스턴스에서 볼륨 분리

인스턴스에서 볼륨을 분리할 수 있습니다.

참고

네트워크를 분리하면 모든 네트워크 포트가 분리됩니다. 인스턴스에 네트워크에 여러 포트가 있고 해당 포트 중 하나만 분리하려면 인스턴스에서 포트 분리 절차에 따라 포트를 분리합니다.

사전 요구 사항

- 인스턴스가 완전히 작동 중이거나 완전히 중지되었습니다. 인스턴스가 부팅 중이거나 종료되는 경우 인스턴스에서 볼륨을 분리할 수 없습니다.

프로세스

1. 인스턴스에 연결된 볼륨을 확인합니다.

```
(overcloud)$ openstack server show <instance>
```

2. 인스턴스에서 볼륨을 분리합니다.

```
$ openstack server remove volume <instance> <volume>
```

- & **lt;instance** >를 볼륨을 제거하려는 인스턴스의 이름 또는 ID로 바꿉니다.
- & **lt;volume** >을 인스턴스에서 제거하려는 볼륨의 이름 또는 ID로 바꿉니다.

참고

SHELVED 또는 **SHELVED_OFFLOADED** 인 인스턴스에서 볼륨을 제거하려면 **--os-compute-api-version 2.20** 이상을 지정합니다.

9장. 인스턴스에 대한 공용 액세스 제공

새 인스턴스는 인스턴스가 할당된 네트워크에서 고정 IP 주소가 있는 포트를 자동으로 수신합니다. 이 IP 주소는 비공개이며 인스턴스가 삭제될 때까지 인스턴스와 영구적으로 연결됩니다. 고정 IP 주소는 인스턴스 간 통신에 사용됩니다.

공용 IP 주소가 인스턴스에 직접 할당되는 공유 외부 네트워크에 공용 인스턴스를 직접 연결할 수 있습니다. 프라이빗 클라우드에서 작업하는 경우 유용합니다.

외부 공급자 네트워크에 대한 라우팅된 연결이 있는 프로젝트 네트워크를 통해 인스턴스에 대한 공용 액세스를 제공할 수도 있습니다. 퍼블릭 클라우드에서 작업하거나 공용 IP 주소가 제한된 경우 이 방법을 사용하는 것이 좋습니다. 프로젝트 네트워크를 통해 공용 액세스를 제공하려면 게이트웨이가 외부 네트워크로 설정된 라우터에 프로젝트 네트워크를 연결해야 합니다. 외부 트래픽이 인스턴스에 연결하려면 클라우드 사용자가 유동 IP 주소를 인스턴스와 연결해야 합니다.

공유 외부 네트워크 또는 라우팅된 공급자 네트워크에 연결되어 있는지 여부에 관계없이 인스턴스에서 액세스를 제공하려면 SSH, ICMP 또는 HTTP와 같은 필수 프로토콜이 있는 보안 그룹을 사용해야 합니다. 또한 원격으로 인스턴스에 액세스할 수 있도록 생성 중에 인스턴스에 키 쌍을 전달해야 합니다.

9.1. 사전 요구 사항

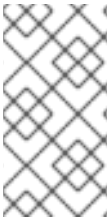
- 외부 네트워크에는 유동 IP 주소를 제공하는 서브넷이 있어야 합니다.
- 프로젝트 네트워크는 외부 네트워크가 게이트웨이로 구성된 라우터에 연결되어 있어야 합니다.
- 필요한 프로토콜이 있는 보안 그룹은 프로젝트에 사용할 수 있어야 합니다. 자세한 내용은 *Red Hat OpenStack Platform 네트워킹 구성에서 보안 그룹 구성을* 참조하십시오.

9.2. 보안 그룹 및 키 쌍을 사용하여 인스턴스 액세스 보안

보안 그룹은 인스턴스에 대한 네트워크 및 프로토콜 액세스를 제어하는 IP 필터 규칙 집합입니다(예: 인스턴스 ping 및 SSH를 통해 인스턴스에 연결할 수 있음).

모든 프로젝트에는 **default** 라는 기본 보안 그룹이 있으며, 인스턴스의 보안 그룹을 지정하지 않을 때 사용됩니다. 기본적으로 기본 보안 그룹은 나가는 모든 트래픽을 허용하고 동일한 보안 그룹의 인스턴스 이외의 소스에서 들어오는 모든 트래픽을 거부합니다. 인스턴스를 생성하는 동안 인스턴스에 하나 이상의 보안 그룹을 적용할 수 있습니다. 실행 중인 인스턴스에 보안 그룹을 적용하려면 인스턴스에 연결된 포트에 보안 그룹을 적용합니다.

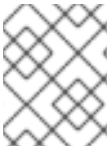
보안 그룹에 대한 자세한 내용은 *Red Hat OpenStack Platform 네트워킹 구성에서 보안 그룹 구성을* 참조하십시오.



참고

인스턴스를 생성하는 동안 RBAC(역할 기반 액세스 제어) 공유 보안 그룹을 인스턴스에 직접 적용할 수 없습니다. 인스턴스에 RBAC-공유 보안 그룹을 적용하려면 먼저 포트를 만들고, 공유 보안 그룹을 해당 포트에 적용한 다음 해당 포트를 인스턴스에 할당해야 합니다. 포트에 보안 그룹 추가를 참조하십시오.

키 쌍은 인스턴스에 대한 원격 액세스를 활성화하기 위해 인스턴스에 삽입되는 SSH 또는 x509 자격 증명입니다. RHOSP에서 새 키 쌍을 생성하거나 기존 키 쌍을 가져올 수 있습니다. 각 사용자에게는 하나 이상의 키 쌍이 있어야 합니다. 키 쌍은 여러 인스턴스에 사용할 수 있습니다.



참고

각 키 쌍이 프로젝트 대신 키 쌍을 만들거나 가져온 개별 사용자에 속하므로 프로젝트의 사용자 간에 키 쌍을 공유할 수 없습니다.

9.2.1. 포트에 보안 그룹 추가

기본 보안 그룹은 대체 보안 그룹을 지정하지 않는 인스턴스에 적용됩니다. 실행 중인 인스턴스의 포트에 대체 보안 그룹을 적용할 수 있습니다.

절차

1. 보안 그룹을 적용할 인스턴스의 포트를 확인합니다.

```
$ openstack port list --server myInstancewithSSH
```

2. 포트에 보안 그룹을 적용합니다.

```
$ openstack port set --security-group <sec_group> <port>
```

& **lt;sec_group** >을 실행 중인 인스턴스의 포트에 적용할 보안 그룹의 이름 또는 ID로 바꿉니다. 필요에 따라 **--security-group** 옵션을 두 번 이상 사용하여 여러 보안 그룹을 적용할 수 있습니다.

9.2.2. 포트에서 보안 그룹 제거

포트에서 보안 그룹을 제거하려면 먼저 모든 보안 그룹을 제거한 다음 포트에 할당된 보안 그룹을 다시 추가해야 합니다.

절차

1. 포트와 연결된 모든 보안 그룹을 나열하고 해당 포트와 연결된 상태로 유지하려는 보안 그룹의 ID를 기록합니다.

```
$ openstack port show <port>
```

2. 포트와 연결된 모든 보안 그룹을 제거합니다.

```
$ openstack port set --no-security-group <port>
```

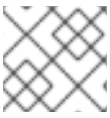
3. 보안 그룹을 포트에 다시 적용합니다.

```
$ openstack port set --security-group <sec_group> <port>
```

& **lt;sec_group** >을 실행 중인 인스턴스의 포트에 다시 적용할 보안 그룹의 ID로 바꿉니다. 필요에 따라 **--security-group** 옵션을 두 번 이상 사용하여 여러 보안 그룹을 적용할 수 있습니다.

9.2.3. 새 SSH 키 쌍 생성

프로젝트 내에서 사용할 새 SSH 키 쌍을 만들 수 있습니다.



참고

x509 인증서를 사용하여 Windows 인스턴스의 키 쌍을 만듭니다.

절차

1. 키 쌍을 생성하고 로컬 **.ssh** 디렉터리에 개인 키를 저장합니다.

```
$ openstack keypair create <keypair> > ~/.ssh/<keypair>.pem
```

& **lt;keypair&** gt;를 새 키 쌍의 이름으로 바꿉니다.

2. 개인 키 보호:

```
$ chmod 600 ~/.ssh/<keypair>.pem
```

9.2.4. 기존 SSH 키 쌍 가져오기

새 키 쌍을 생성할 때 공개 키 파일을 제공하여 RHOSP(Red Hat OpenStack Platform) 외부에서 생성한 프로젝트로 SSH 키를 가져올 수 있습니다.

절차

1. 기존 키 파일에서 키 쌍을 생성하고 로컬 **.ssh** 디렉터리에 개인 키를 저장합니다.
   - 기존 공개 키 파일에서 키 쌍을 가져오려면 다음 명령을 입력합니다.

```
$ openstack keypair create --public-key ~/.ssh/<public_key>.pub \
 <keypair> > ~/.ssh/<keypair>.pem
```

     - & **lt;public_key** >를 키 쌍을 생성하는 데 사용할 공개 키 파일의 이름으로 바꿉니다.
     - & **lt;keypair&** gt;를 새 키 쌍의 이름으로 바꿉니다.
   - 기존 개인 키 파일에서 키 쌍을 가져오려면 다음 명령을 입력합니다.

```
$ openstack keypair create --private-key ~/.ssh/<private_key> \
 <keypair> > ~/.ssh/<keypair>.pem
```

     - & **lt;private_key** >를 키 쌍을 생성하는 데 사용할 공개 키 파일의 이름으로 바꿉니다.
     - & **lt;keypair&** gt;를 새 키 쌍의 이름으로 바꿉니다.

2. 개인 키 보호:

```
$ chmod 600 ~/.ssh/<keypair>.pem
```

9.2.5. 추가 리소스

- *Red Hat OpenStack Platform 네트워킹 구성에서 보안 그룹 구성.*
- *OpenStack Identity 리소스 관리의 프로젝트 보안 관리.*

9.3. 인스턴스에 유동 IP 주소 할당

공용 유동 IP 주소를 인스턴스에 할당하여 인터넷을 포함하여 클라우드 외부 네트워크와의 통신을 활성화할 수 있습니다. 클라우드 관리자는 외부 네트워크에 사용 가능한 유동 IP 주소 풀을 구성합니다. 이 풀에서 유동 IP 주소를 프로젝트에 할당한 다음 유동 IP 주소를 인스턴스와 연결할 수 있습니다.

프로젝트에는 프로젝트의 인스턴스에서 사용할 수 있는 유동 IP 주소의 할당량이 제한되며, 기본적으로 50개입니다. 따라서 더 이상 필요하지 않을 때 재사용할 수 있도록 IP 주소를 릴리스합니다.

사전 요구 사항

- 인스턴스는 외부 네트워크에 있거나 외부 네트워크가 게이트웨이로 구성된 라우터에 연결된 프로젝트 네트워크에 있어야 합니다.
- 인스턴스에서 연결할 외부 네트워크에는 유동 IP 주소를 제공하는 서브넷이 있어야 합니다.

절차

1. 현재 프로젝트에 할당된 유동 IP 주소를 확인합니다.

```
$ openstack floating ip list
```

사용하려는 유동 IP 주소가 없는 경우 외부 네트워크 할당 풀에서 현재 프로젝트에 유동 IP 주소를 할당합니다.

```
$ openstack floating ip create <provider-network>
```

& **lt;provider-network** >를 외부 액세스를 제공하는 데 사용할 외부 네트워크의 이름 또는 ID로 바꿉니다.

작은 정보

기본적으로 유동 IP 주소는 외부 네트워크 풀에서 임의로 할당됩니다. 클라우드 관리자는 --floating-ip-address 옵션을 사용하여 외부 네트워크에서 특정 유동 IP 주소를 할당할 수 있습니다.

2. 인스턴스에 유동 IP 주소를 할당합니다.

```
$ openstack server add floating ip [--fixed-ip-address <ip_address>] \
 <instance> <floating_ip>
```

- & **lt;instance** >를 공용 액세스를 제공하려는 인스턴스의 이름 또는 ID로 바꿉니다.
- & **lt;floating_ip** >를 인스턴스에 할당하려는 유동 IP 주소로 바꿉니다.
- 선택 사항: **<ip_address** >를 유동 IP를 연결할 인터페이스의 IP 주소로 바꿉니다. 기본적으로 유동 IP 주소를 첫 번째 포트에 연결합니다.

3. 유동 IP 주소가 인스턴스에 할당되었는지 확인합니다.

```
$ openstack server show <instance>
```

추가 리소스

- *Red Hat OpenStack Platform 네트워킹 구성* 가이드에서 유동 IP 풀 생성 .

9.4. 인스턴스에서 유동 IP 주소 연결 해제

인스턴스에 더 이상 공용 액세스가 필요하지 않으면 인스턴스에서 연결을 끊고 할당 풀로 반환합니다.

절차

1. 인스턴스에서 유동 IP 주소의 연결을 끊습니다.

```
$ openstack server remove floating ip <instance> <ip_address>
```

   - & **lt;instance** >를 공용 액세스를 제거하려는 인스턴스의 이름 또는 ID로 바꿉니다.
   - & **lt;floating_ip** >를 인스턴스에 할당된 유동 IP 주소로 바꿉니다.

2. 유동 IP 주소를 다시 할당 풀로 해제합니다.

```
$ openstack floating ip delete <ip_address>
```

3. 유동 IP 주소가 삭제되었으며 더 이상 할당에 사용할 수 없는지 확인합니다.

```
$ openstack floating ip list
```

9.5. SSH 액세스를 사용하여 인스턴스 생성

인스턴스를 생성할 때 키 쌍을 지정하여 인스턴스에 대한 SSH 액세스를 제공할 수 있습니다. 키 쌍은 시작 시 인스턴스에 삽입되는 SSH 또는 x509 자격 증명입니다. 각 프로젝트에는 하나 이상의 키 쌍이 있어야 합니다. 키 쌍은 프로젝트가 아닌 개별 사용자에게 속합니다.

참고

인스턴스를 생성한 후에는 키 쌍을 인스턴스와 연결할 수 없습니다.

인스턴스를 생성하는 동안 또는 실행 중인 인스턴스의 포트에 보안 그룹을 직접 적용할 수 있습니다.

참고

인스턴스를 생성하는 동안 RBAC(역할 기반 액세스 제어) 공유 보안 그룹을 인스턴스에 직접 적용할 수 없습니다. 인스턴스에 RBAC-공유 보안 그룹을 적용하려면 먼저 포트를 만들고, 공유 보안 그룹을 해당 포트에 적용한 다음 해당 포트를 인스턴스에 할당해야 합니다. 포트에 보안 그룹 추가를 참조하십시오.

사전 요구 사항

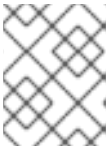
- 인스턴스에 SSH 연결을 수행하는 데 사용할 수 있는 키 쌍을 사용할 수 있습니다. 자세한 내용은 새 SSH 키 쌍 생성을 참조하십시오.
- 인스턴스를 생성하려는 네트워크는 외부 네트워크 또는 외부 네트워크가 게이트웨이로 구성된 라우터에 연결된 프로젝트 네트워크여야 합니다. 자세한 내용은 *Red Hat OpenStack Platform 네트워킹 구성* 가이드에서 라우터 추가 를 참조하십시오.
- 인스턴스에 연결하는 외부 네트워크에는 유동 IP 주소를 제공하는 서브넷이 있어야 합니다.
- 보안 그룹을 사용하면 인스턴스에 대한 SSH 액세스를 허용합니다. 자세한 내용은 보안 그룹 및 키 쌍을 사용한 인스턴스 액세스 보안 을 참조하십시오.
- 인스턴스가 기반으로 하는 이미지에는 SSH 공개 키를 인스턴스에 삽입할 **cloud-init** 패키지가 포함되어 있습니다.

- 인스턴스에 할당할 수 있는 유동 IP 주소를 사용할 수 있습니다. 자세한 내용은 인스턴스에 유동 IP 주소 할당을 참조하십시오.

절차

1. 인스턴스에 필요한 하드웨어 프로필이 있는 플레이버의 이름 또는 ID를 검색합니다.

```
$ openstack flavor list
```



참고

이미지가 성공적으로 부팅될 수 있는 충분한 크기의 플레이버를 선택합니다. 그렇지 않으면 인스턴스가 시작되지 않습니다.

2. 인스턴스에 필요한 소프트웨어 프로필이 있는 이미지의 이름 또는 ID를 검색합니다.

```
$ openstack image list
```

필요한 이미지를 사용할 수 없는 경우 새 이미지를 다운로드하거나 생성할 수 있습니다. 클라우드 이미지 생성 또는 다운로드에 대한 자세한 내용은 이미지 생성 을 참조하십시오.

3. 인스턴스를 연결할 네트워크의 이름 또는 ID를 검색합니다.

```
$ openstack network list
```

4. 인스턴스에 원격으로 액세스하는 데 사용할 키 쌍의 이름을 검색합니다.

```
$ openstack keypair list
```

5. SSH 액세스를 사용하여 인스턴스를 생성합니다.

```
$ openstack server create --flavor <flavor> \
  --image <image> --network <network> \
  [--security-group <secgroup>] \
  --key-name <keypair> --wait myInstancewithSSH
```

- & **lt;flavor** >를 1단계에서 검색한 플레이버의 이름 또는 ID로 바꿉니다.
- & **lt;image** >를 2단계에서 검색한 이미지의 이름 또는 ID로 바꿉니다.
- & **lt;network** >를 3단계에서 검색한 네트워크의 이름 또는 ID로 바꿉니다. 필요에 따라 **--network** 옵션을 두 번 이상 사용하여 인스턴스를 여러 네트워크에 연결할 수 있습니다.
- 선택 사항: **기본** 보안 그룹은 대체 보안 그룹을 지정하지 않는 인스턴스에 적용됩니다. 인스턴스를 생성하는 동안 또는 실행 중인 인스턴스의 포트에 대체 보안 그룹을 직접 적용할 수 있습니다. 인스턴스를 생성할 때 대체 보안 그룹을 지정하려면 **--security-group** 옵션을 사용합니다. 실행 중인 인스턴스의 포트에 보안 그룹을 추가하는 방법에 대한 자세한 내용은 포트에 보안 그룹 추가를 참조하십시오.
- & **lt;keypair** >를 4단계에서 검색한 키 쌍의 이름 또는 ID로 바꿉니다.

6. 인스턴스에 유동 IP 주소를 할당합니다.

```
$ openstack server add floating ip myInstancewithSSH <floating_ip>
```

& **lt;floating_ip** >를 인스턴스에 할당하려는 유동 IP 주소로 바꿉니다.

7. 자동으로 생성된 **cloud-user** 계정을 사용하여 SSH를 사용하여 인스턴스에 로그인할 수 있는지 확인합니다.

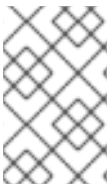
```
$ ssh -i ~/.ssh/<keypair>.pem cloud-user@<floatingIP>
[cloud-user@demo-server1 ~]$
```

9.6. 추가 리소스

- *Red Hat OpenStack Platform 네트워킹* 구성에서 네트워크 생성
- *Red Hat OpenStack Platform 네트워킹* 구성에 라우터 추가.
- *Red Hat OpenStack Platform 네트워킹 구성에서 보안 그룹 구성.*

10장. 인스턴스에 연결

인스턴스 보안 그룹 규칙에서 프로토콜을 허용하면 SSH 또는 WinRM과 같은 원격 쉘을 사용하여 클라우드에 외부 위치에서 인스턴스에 액세스할 수 있습니다. 네트워크 연결이 실패하더라도 디버그할 수 있도록 인스턴스의 콘솔에 직접 연결할 수도 있습니다.



참고

인스턴스에 키 쌍을 제공하지 않았거나 인스턴스에 보안 그룹을 할당하지 않은 경우 VNC를 사용하여 클라우드 내부에서만 인스턴스에 액세스할 수 있습니다. 인스턴스를 ping할 수 없습니다.

10.1. 인스턴스 콘솔에 액세스

브라우저에 VNC 콘솔 URL을 입력하여 인스턴스의 VNC 콘솔에 직접 연결할 수 있습니다.

절차

1. 인스턴스의 VNC 콘솔 URL을 표시하려면 다음 명령을 입력합니다.

```
$ openstack console url show <vm_name>
+-------+-----------------------------------------------------+
| Field | Value                                               |
+-------+-----------------------------------------------------+
| type  | novnc                                               |
| url | http://172.25.250.50:6080/vnc_auto.html?token=       |
| | 962dfd71-f047-43d3-89a5-13cb88261eb9                      |
+-------+-----------------------------------------------------+
```

2. VNC 콘솔에 직접 연결하려면 브라우저에 표시된 URL을 입력합니다.

10.2. 인스턴스에 로그인

공용 인스턴스에 원격으로 로그인할 수 있습니다.

사전 요구 사항

- 인스턴스의 키 쌍 인증서가 있습니다. 키 쌍이 생성되면 인증서가 다운로드됩니다. 키 쌍을 직접 생성하지 않은 경우 관리자에게 문의하십시오.
- 인스턴스는 공용 인스턴스로 구성됩니다. 공용 인스턴스의 요구 사항에 대한 자세한 내용은 인스턴스에 대한 공용 액세스 제공을 참조하십시오.
- 클라우드 사용자 계정이 있습니다.

절차

1. 로그인할 인스턴스의 유동 IP 주소를 검색합니다.

```
$ openstack server show <instance>
```

& **lt;instance** >를 연결하려는 인스턴스의 이름 또는 ID로 바꿉니다.

2. 자동으로 생성된 **cloud-user** 계정을 사용하여 인스턴스에 로그인합니다.

```
$ ssh -i ~/.ssh/<keypair>.pem cloud-user@<floatingIP>
[cloud-user@demo-server1 ~]$
```

- & **lt;keypair&** gt;를 키 쌍의 이름으로 바꿉니다.
- & **lt;floating_ip&** gt;를 인스턴스의 유동 IP 주소로 바꿉니다.

작은 정보

다음 명령을 사용하여 유동 IP 주소 없이 인스턴스에 로그인할 수 있습니다.

```
$ openstack server ssh --login cloud-user \
 --identity ~/.ssh/<keypair>.pem --private <instance>
```

- & **lt;keypair&** gt;를 키 쌍의 이름으로 바꿉니다.
- & **lt;instance** >를 연결하려는 인스턴스의 이름 또는 ID로 바꿉니다.

11장. 인스턴스 관리

인스턴스 크기 조정 또는 인스턴스 보류 등 인스턴스에서 관리 작업을 수행할 수 있습니다. 관리 작업의 전체 목록은 인스턴스 관리 작업을 참조하십시오.

11.1. 인스턴스 크기 조정

인스턴스의 메모리 또는 CPU 수를 늘리거나 줄여야 하는 경우 인스턴스의 크기를 조정할 수 있습니다. 인스턴스의 크기를 조정하려면 필요한 용량이 있는 인스턴스의 새 플레이버를 선택합니다. 인스턴스 크기 조정이 다시 빌드되고 인스턴스를 다시 시작합니다.

절차

1. 크기를 조정할 인스턴스의 이름 또는 ID를 검색합니다.

```
$ openstack server list
```

2. 인스턴스의 크기를 조정하는 데 사용할 플레이버의 이름 또는 ID를 검색합니다.

```
$ openstack flavor list
```

3. 인스턴스의 크기를 조정합니다.

```
$ openstack server resize --flavor <flavor> \
  --wait <instance>
```

- & **lt;flavor** >를 2단계에서 검색한 플레이버의 이름 또는 ID로 바꿉니다.
- & **lt;instance** >를 크기 조정 중인 인스턴스의 이름 또는 ID로 바꿉니다.

참고

크기 조정에는 시간이 걸릴 수 있습니다. 인스턴스의 운영 체제는 인스턴스의 전원이 꺼지고 인스턴스의 크기를 조정하기 전에 제어된 종료를 수행합니다. 이 시간 동안 인스턴스 상태는 **RESIZE** 입니다.

```
$ openstack server list
+----------------------+---------------+--------+----------------------------+
| ID                   | Name          | Status | Networks                   |
+----------------------+---------------+--------+----------------------------+
| 67bc9a9a-5928-47c... | myCirrosServer | RESIZE |
admin_internal_net=192.168.111.139    |
+----------------------+---------------+--------+----------------------------+
```

4. 크기 조정이 완료되면 인스턴스 상태가 **VERIFY_RESIZE** 로 변경됩니다. 이제 크기를 확인하거나 되돌리야 합니다.

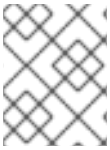
- 크기 조정을 확인하려면 다음 명령을 입력합니다.

```
$ openstack server resize confirm <instance>
```

- 크기 조정을 되돌리려면 다음 명령을 입력합니다.

```
$ openstack server resize revert <instance>
```

인스턴스가 원래 플레이버로 복원되고 상태가 **ACTIVE** 로 변경됩니다.



참고

클라우드는 구성된 시간 프레임 내에서 확인하지 않거나 되돌리지 않으면 인스턴스 크기 조정을 자동으로 확인하도록 구성할 수 있습니다.

11.2. 인스턴스 스냅샷 생성

스냅샷은 인스턴스의 실행 중인 디스크 상태를 캡처하는 이미지입니다. 인스턴스의 스냅샷을 작성하여 새 인스턴스를 생성하는 데 템플릿으로 사용할 수 있는 이미지를 생성할 수 있습니다. 스냅샷을 사용하면 다른 인스턴스에서 새 인스턴스를 생성하고 인스턴스 상태를 복원할 수 있습니다. 스냅샷이 기반으로 하는 인스턴스를 삭제하는 경우 스냅샷 이미지를 사용하여 스냅샷과 동일한 상태에 새 인스턴스를 만들 수 있습니다.

절차

1. 스냅샷을 만들 인스턴스의 이름 또는 ID를 검색합니다.

   ```
   $ openstack server list
   ```

2. 스냅샷을 생성합니다.

   ```
   $ openstack server image create --name <image_name> <instance>
   ```

   - & **lt;image_name&** gt;을 새 스냅샷 이미지의 이름으로 바꿉니다.
   - & **lt;instance** >를 스냅샷을 생성할 인스턴스의 이름 또는 ID로 바꿉니다.

3. 선택 사항: 새 인스턴스를 생성하기 위해 인스턴스 스냅샷을 템플릿으로 사용할 때 디스크 상태가 일관되게 유지되려면 QEMU 게스트 에이전트를 활성화하고 스냅샷 이미지에 다음 메타데이터를 추가하여 스냅샷 처리 중에 파일 시스템을 정지해야 합니다.

   ```
   $ openstack image set --property hw_qemu_guest_agent=yes \
    --property os_require_quiesce=yes <image_name>
   ```

   QEMU 게스트 에이전트는 애플리케이션을 관리하여 인스턴스 OS 수준 명령을 실행하는 데 도움이 되는 백그라운드 프로세스입니다. 이 에이전트를 활성화하면 PCI 슬롯을 사용하는 인스턴스에 다른 장치를 추가하고 인스턴스에 할당할 수 있는 다른 장치 수를 제한합니다. 또한 Windows 인스턴스가 알 수 없는 하드웨어 장치에 대한 경고 메시지를 표시하도록 합니다.

11.3. 인스턴스 복구

시스템 장애 또는 액세스 실패와 같은 긴급 상황에서 인스턴스를 복구 모드로 설정할 수 있습니다. 이렇게 하면 인스턴스가 종료되고 새 인스턴스 디스크로 재부팅되고 원래 인스턴스 디스크 및 구성 드라이브를 재부팅 인스턴스에 볼륨으로 마운트합니다. 재부팅된 인스턴스에 연결하여 원래 인스턴스 디스크를 보고 시스템을 복구하고 데이터를 복구할 수 있습니다.

프로세스

1. 인스턴스 복구를 수행합니다.

```
$ openstack server rescue [--image <image>] <instance>
```

- 선택 사항: 기본적으로 인스턴스는 클라우드 관리자가 제공한 복구 이미지 또는 원래 인스턴스 이미지의 새 사본에서 부팅됩니다. rescue 모드에서 인스턴스를 재부팅할 때 사용할 대체 이미지를 지정하려면 **--image** 옵션을 사용합니다.
- & **lt;instance** >를 복구하려는 인스턴스의 이름 또는 ID로 바꿉니다.

2. 복구된 인스턴스에 연결하여 문제를 해결합니다.
3. 일반 부팅 디스크에서 인스턴스를 다시 시작합니다.

```
$ openstack server unrescue <instance>
```

11.4. 인스턴스 보류

Shelving은 사용하지 않는 인스턴스가 있지만 삭제하려는 경우 유용합니다. 인스턴스를 보류하면 인스턴스 데이터 및 리소스 할당을 유지하지만 인스턴스 메모리를 지웁니다. 클라우드 구성에 따라 보류된 인스턴스는 즉시 또는 지연 후 **SHELVED_OFFLOADED** 상태로 이동합니다. **SHELVED_OFFLOADED** 에서는 인스턴스 데이터 및 리소스 할당이 삭제됩니다.

인스턴스를 보류하면 Compute 서비스에서 인스턴스 상태를 캡처하는 스냅샷 이미지를 생성하고 <instance> **-shelved** 형식의 이미지에 이름을 할당합니다. 이 스냅샷 이미지는 인스턴스가 보류 해제되거나 삭제될 때 삭제됩니다.

더 이상 보류된 인스턴스가 필요하지 않은 경우 삭제할 수 있습니다. 한 번에 두 개 이상의 인스턴스를 보류할 수 있습니다.

절차

1. 보류하려는 인스턴스 또는 인스턴스의 이름 또는 ID를 검색합니다.

```
$ openstack server list
```

2. 인스턴스 또는 인스턴스를 보류합니다.

```
$ openstack server shelve <instance> [<instance> ...]
```

& **lt;instance** >를 보류하려는 인스턴스의 이름 또는 ID로 바꿉니다. 필요에 따라 둘 이상의 인스턴스를 지정할 수 있습니다.

3. 인스턴스가 보류되었는지 확인합니다.

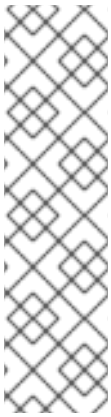
```
$ openstack server list
```

보류된 인스턴스의 상태는 **SHELVED_OFFLOADED** 입니다.

11.5. 인스턴스 관리 작업

인스턴스를 생성한 후 다음 관리 작업을 수행할 수 있습니다.

표 11.1. 관리 작업

작업	설명	명령
인스턴스 중지	인스턴스를 중지합니다.	**OpenStack server stop**
인스턴스 시작	중지된 인스턴스를 시작합니다.	**OpenStack 서버 시작**
실행 중인 인스턴스 일시 중지	실행 중인 인스턴스를 즉시 일시 중지합니다. 인스턴스 상태는 메모리(RAM)에 저장됩니다. 일시 중지된 인스턴스는 고정 상태로 계속 실행됩니다. 일시 중지 작업을 확인하라는 메시지가 표시되지 않습니다.	**OpenStack 서버 일시 중지**
일시 중지된 인스턴스 실행 재개	일시 중지된 인스턴스를 즉시 다시 시작합니다. 재개 작업을 확인하라는 메시지가 표시되지 않습니다.	**OpenStack 서버 일시 중지 해제**
실행 중인 인스턴스 일시 중단	실행 중인 인스턴스를 즉시 일시 중지합니다. 인스턴스 상태는 인스턴스 디스크에 저장됩니다. 일시 중단 작업을 확인하라는 메시지가 표시되지 않습니다.	**OpenStack 서버 일시 중단**
일시 중지된 인스턴스의 실행 재개	일시 중지된 인스턴스를 즉시 다시 시작합니다. 인스턴스 상태는 인스턴스 디스크에 저장됩니다. 재개 작업을 확인하라는 메시지가 표시되지 않습니다.	**OpenStack 서버 다시 시작**
인스턴스 삭제	인스턴스를 영구적으로 삭제합니다. 제거 작업을 확인하라는 메시지가 표시되지 않습니다. 클라우드가 소프트 삭제 활성화를 활성화하지 않으면 삭제된 인스턴스를 복구할 수 없습니다.  **참고** 인스턴스를 삭제해도 연결된 볼륨은 삭제되지 않습니다. 연결된 볼륨을 별도로 삭제해야 합니다. 자세한 내용은 *영구 스토리지 구성 가이드의 블록 스토리지 서비스 볼륨 삭제*를 참조하십시오.	**OpenStack server delete**

작업	설명	명령
인스턴스 메타데이터 편집	인스턴스 메타데이터를 사용하여 인스턴스의 속성을 지정할 수 있습니다. 자세한 내용은 사용자 지정 인스턴스 생성을 참조하십시오.	**openstack server set --property <key=value> [--property <key=value>] <instance>**
보안 그룹 추가	지정된 보안 그룹을 인스턴스에 추가합니다.	**OpenStack 서버에서 보안 그룹 추가**
보안 그룹 제거	인스턴스에서 지정된 보안 그룹을 제거합니다.	**OpenStack에서 보안 그룹 제거**
인스턴스 복구	시스템 장애 또는 액세스 실패와 같은 긴급 상황에서 인스턴스를 복구 모드로 설정할 수 있습니다. 인스턴스가 종료되고 root 디스크를 임시 서버에 마운트합니다. 임시 서버에 연결하여 시스템을 복구하고 데이터를 복구할 수 있습니다. 실행 중인 인스턴스를 복구 모드로 재부팅할 수도 있습니다. 예를 들어 인스턴스의 파일 시스템이 손상되면 이 작업이 필요할 수 있습니다.	**OpenStack 서버 복구**
복구된 인스턴스 복원	복구된 인스턴스를 재부팅합니다.	**OpenStack 서버 unrescue**
인스턴스 로그 보기	인스턴스 콘솔 로그의 최신 섹션을 확인합니다.	**OpenStack 콘솔 로그 표시**
인스턴스 보류	인스턴스를 보류하면 인스턴스 데이터 및 리소스 할당을 유지하지만 인스턴스 메모리를 지웁니다. 클라우드 구성에 따라 보류된 인스턴스는 즉시 또는 지연 후 **SHELVED_OFFLOADED** 상태로 이동합니다. 인스턴스가 **SHELVED_OFFLOADED** 상태에 있으면 인스턴스 데이터 및 리소스 할당이 삭제됩니다. 인스턴스 상태는 인스턴스 디스크에 저장됩니다. 인스턴스가 볼륨에서 부팅된 경우 즉시 **SHELVED_OFFLOADED** 로 이동합니다. shelve 작업을 확인하라는 메시지가 표시되지 않습니다.	**OpenStack 서버 shelve**
인스턴스 보류 해제	보류된 인스턴스의 디스크 이미지를 사용하여 인스턴스를 복원합니다.	**OpenStack 서버 unshelve**

작업	설명	명령
인스턴스 잠금	관리자가 아닌 사용자가 인스턴스에서 작업을 실행하지 못하도록 인스턴스를 잠급니다.	**OpenStack 서버 잠금** **OpenStack 서버 잠금 해제**
인스턴스 소프트 재부팅	인스턴스를 정상적으로 중지하고 다시 시작합니다. 소프트 재부팅은 인스턴스를 다시 시작하기 전에 모든 프로세스를 정상적으로 종료합니다. 기본적으로 인스턴스를 재부팅하면 소프트 재부팅입니다.	**openstack server reboot --soft <server>**
하드 재부팅 인스턴스	인스턴스를 중지하고 다시 시작합니다. 하드 재부팅이 인스턴스에 대한 전원을 종료한 다음 다시 전환합니다.	**OpenStack 서버 reboot --hard <server>**
인스턴스 다시 빌드	새 이미지 및 디스크 파티션 옵션을 사용하여 인스턴스가 종료, 다시 이미지 종료, 재부팅되는 인스턴스를 다시 빌드합니다. 인스턴스를 종료하고 다시 시작하는 대신 운영 체제 문제가 발생하면 이 옵션을 사용합니다.	**OpenStack 서버 다시 빌드**

12장. 사용자 지정 인스턴스 생성

클라우드 사용자는 인스턴스를 시작할 때 사용할 추가 데이터(예: 인스턴스가 부팅 시 실행되는 쉘 스크립트)를 지정할 수 있습니다. 클라우드 사용자는 다음 방법을 사용하여 인스턴스에 데이터를 전달할 수 있습니다.

사용자 데이터

cloud-init 를 실행하기 위한 명령을 사용하여 인스턴스 시작 명령에 포함합니다.

인스턴스 메타데이터

인스턴스를 만들거나 업데이트할 때 지정할 수 있는 키-값 쌍 목록입니다.

구성 드라이브 또는 메타데이터 서비스를 사용하여 인스턴스에 전달된 추가 데이터에 액세스할 수 있습니다.

구성 드라이브

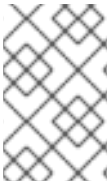
부팅 시 구성 드라이브를 인스턴스에 연결할 수 있습니다. 구성 드라이브가 인스턴스에 읽기 전용 드라이브로 표시됩니다. 인스턴스는 이 드라이브를 마운트하고 해당 드라이브에서 파일을 읽을 수 있습니다. 구성 드라이브를 **cloud-init** 정보의 소스로 사용할 수 있습니다. 구성 드라이브는 서버 부트스트랩을 위해 **cloud-init** 와 결합되고 대용량 파일을 인스턴스에 전달하려는 경우 유용합니다. 예를 들어 **cloud-init** 를 구성 드라이브를 자동으로 마운트하고 초기 인스턴스 부팅 중에 설정 스크립트를 실행하도록 구성할 수 있습니다. 구성 드라이브는 **config-2** 의 볼륨 레이블을 사용하여 생성되며 부팅 시 인스턴스에 연결됩니다. 구성 드라이브에 전달된 추가 파일의 콘텐츠는 구성 드라이브의 **openstack/{version}/** 디렉터리에 있는 **user_data** 파일에 추가됩니다. **cloud-init** 는 이 파일에서 사용자 데이터를 검색합니다.

메타데이터 서비스

인스턴스와 관련된 데이터를 검색하는 REST API를 제공합니다. 인스턴스는 **169.254.169.254** 또는 **fe80::a9fe:a9fe** 에서 이 서비스에 액세스합니다.

cloud-init 는 구성 드라이브와 메타데이터 서비스를 모두 사용하여 인스턴스를 사용자 정의하는 데 추가 데이터를 사용할 수 있습니다. **cloud-init** 패키지는 여러 데이터 입력 형식을 지원합니다. 쉘 스크립트 및 **cloud-config** 형식은 가장 일반적인 입력 형식입니다.

- 쉘 스크립트: 데이터 선언은 **#!** 또는 **Content-Type: text/x-shellscript** 로 시작합니다. 쉘 스크립트는 부팅 프로세스에서 마지막으로 호출됩니다.
- **cloud-config** 형식: 데이터 선언은 **#cloud-config** 또는 **Content-Type: text/cloud-config** 로 시작합니다. **cloud-config** 파일은 **cloud-init** 에서 구문 분석하고 실행하려면 유효한 YAML이어야 합니다.



참고

cloud-init 는 인스턴스에 전달되는 데이터에 대해 최대 사용자 데이터 크기가 16384바이트입니다. 크기 제한을 변경할 수 없으므로 크기 제한을 초과해야 하는 경우 gzip 압축을 사용합니다.

벤더별 데이터

RHOSP 관리자는 생성 시 데이터를 인스턴스에 전달할 수도 있습니다. 이 데이터는 예를 들어 Active Directory에 인스턴스를 등록하는 암호화 토큰과 같이 클라우드 사용자로 표시되지 않을 수 있습니다.

RHOSP 관리자는 vendordata 기능을 사용하여 인스턴스에 데이터를 전달합니다. Vendordata 구성은 읽기 전용이며 다음 파일 중 하나에 있습니다.

- **/openstack/{version}/vendor_data.json**

- **/openstack/{version}/vendor_data2.json**

메타데이터 서비스 또는 인스턴스의 구성 드라이브에서 이러한 파일을 볼 수 있습니다. 메타데이터 서비스를 사용하여 파일에 액세스하려면 **http://169.254.169.254/openstack/{version}/vendor_data.json** 또는 **http://169.254.169.254/openstack/{version}/vendor_data2.json** 에 GET 요청을 만듭니다.

12.1. 사용자 데이터를 사용하여 인스턴스 사용자 정의

사용자 데이터를 사용하여 instance launch 명령에 지침을 포함할 수 있습니다. **cloud-init** 는 이러한 명령을 실행하여 인스턴스를 부팅 프로세스의 마지막 단계로 사용자 지정합니다.

프로세스

1. **cloud-init** 에 대한 지침으로 파일을 생성합니다. 예를 들어 인스턴스에 웹 서버를 설치하고 활성화하는 bash 스크립트를 생성합니다.

```
$ vim /home/scripts/install_httpd
#!/bin/bash

yum -y install httpd python-psycopg2
systemctl enable httpd --now
```

2. **--user-data** 옵션으로 인스턴스를 시작하여 bash 스크립트를 전달합니다.

```
$ openstack server create \
--image rhel8 \
--flavor default \
--nic net-id=web-server-network \
--security-group default \
--key-name web-server-keypair \
--user-data /home/scripts/install_httpd \
--wait web-server-instance
```

3. 인스턴스 상태가 활성화되면 유동 IP 주소를 연결합니다.

```
$ openstack floating ip create web-server-network
$ openstack server add floating ip web-server-instance 172.25.250.123
```

4. SSH를 사용하여 인스턴스에 로그인합니다.

```
$ ssh -i ~/.ssh/web-server-keypair cloud-user@172.25.250.123
```

5. 사용자 지정이 성공적으로 수행되었는지 확인합니다. 예를 들어 웹 서버가 설치 및 활성화되어 있는지 확인하려면 다음 명령을 입력합니다.

```
$ curl http://localhost | grep Test
<title>Test Page for the Apache HTTP Server on Red Hat Enterprise Linux</title>
<h1>Red Hat Enterprise Linux <strong>Test Page</strong></h1>
```

6. **cloud-init** 가 실행되었는지 여부와 같은 관련 메시지의 **/var/log/cloud-init.log** 파일을 검토합니다.

```
$ sudo less /var/log/cloud-init.log
```

```
...output omitted...
...util.py[DEBUG]: Cloud-init v. 0.7.9 finished at Sat, 23 Jun 2018 02:26:02 +0000.
Datasource DataSourceOpenStack [net,ver=2].  Up 21.25 seconds
```

12.2. 메타데이터를 사용하여 인스턴스 사용자 정의

인스턴스 메타데이터를 사용하여 인스턴스 시작 명령에서 인스턴스의 속성을 지정할 수 있습니다.

절차

1. **--property <key=value&gt**; 옵션을 사용하여 인스턴스를 시작합니다. 예를 들어 인스턴스를 웹 서버로 표시하려면 다음 속성을 설정합니다.

```
$ openstack server create \
--image rhel8 \
--flavor default \
--property role=webservers \
--wait web-server-instance
```

2. 선택 사항: 인스턴스가 생성된 후 인스턴스에 속성을 추가합니다. 예를 들면 다음과 같습니다.

```
$ openstack server set \
--property region=emea \
--wait web-server-instance
```

12.3. 구성 드라이브를 사용하여 인스턴스 사용자 정의

인스턴스 부팅 프로세스 중에 연결된 인스턴스에 대한 구성 드라이브를 생성할 수 있습니다. 구성 드라이브가 인스턴스에서 사용할 수 있도록 하는 구성 드라이브에 콘텐츠를 전달할 수 있습니다.

절차

1. 구성 드라이브를 활성화하고 구성 드라이브에서 사용할 수 있도록 할 콘텐츠가 포함된 파일을 지정합니다. 예를 들어 다음 명령은 **config-drive-instance** 라는 새 인스턴스를 생성하고 **my-user-data.txt** 파일의 내용이 포함된 구성 드라이브를 연결합니다.

```
(overcloud)$ openstack server create --flavor m1.tiny \
  --config-drive true \
  --user-data ./my-user-data.txt \
  --image cirros config-drive-instance
```

   이 명령은 부팅 시 인스턴스에 연결된 **config-2** 의 볼륨 레이블을 사용하여 구성 드라이브를 생성하고 구성 드라이브의 **openstack/{version}/** 디렉터리의 **user_data** 파일에 **my-user-data.txt** 의 내용을 추가합니다.

2. 인스턴스에 로그인합니다.

3. 구성 드라이브를 마운트합니다.

   - 인스턴스 OS에서 **udev** 를 사용하는 경우 :

```
# mkdir -p /mnt/config
# mount /dev/disk/by-label/config-2 /mnt/config
```

- 인스턴스 OS에서 **udev** 를 사용하지 않는 경우 먼저 구성 드라이브에 해당하는 블록 장치를 식별해야 합니다.

```
# blkid -t LABEL="config-2" -odevice
/dev/vdb
# mkdir -p /mnt/config
# mount /dev/vdb /mnt/config
```