MicroStack Up & Running

OpenInfra Community Days Korea 2021

임찬식

MicroStack ?

- https://microstack.run/
- 복잡한 OpenStack 배포를 간단하게 해결
- 명령어 두 개로 OpenStack 배포
- 포함하는 OpenStack 구성요소
  - Keystone, Nova, Neutron, Glance, Cinder
- 단일 노드 혹은 멀티 노드에 배포 가능
- 현재는 `beta` 상태

MicroStack for Developer

- 개발 과정에서 OpenStack 환경이 필요한 경우
- 기본적인 OpenStack 서비스만 있으면 충분
- 설정은 모두 기본 값으로 지정
- 개발 서버 안에서 작동하는 OpenStack
- 필요할 때만 띄워서 사용

MicroStack for Micro Clouds

- OpenStack 기반 Private Cloud 서비스 이용
- 적은 수의 노드를 이용해 시작 가능
- MicroStack 으로 쉽고 빠르게 배포 가능
- 단일 노드를 이용해 효율적인 Private Cloud 시작
- 필요하다면 멀티 노드로 확장

Setup MicroStack

Check MicroStack Version

- 최신 버전 확인
- 앞으로 설치 과정에서는 latest/edge 버전 사용

```
$ sudo snap info microstack

name:      microstack
summary:   OpenStack on your laptop
publisher: Canonical✓
store-url: https://snapcraft.io/microstack
license:   Apache-2.0
description: |
  MicroStack gives you an easy way to develop and test OpenStack
  workloads on your laptop or in a virtual machine.
channels:
  latest/beta:       ussuri 2021-10-27 (241) 436MB -
  latest/edge:       ussuri 2021-10-22 (241) 436MB -
  rocky/edge:        rocky  2019-12-06 (195) 398MB classic
```

Check Services and Ports

- MicroStack 에서 `nginx`, `mysql` 등의 서비스 실행
- 초기화 과정에서 포트가 겹칠 경우 초기화 실패
- 시스템에서 이미 사용 중인지 확인 필요
  - `nginx`: 80, 443
  - `mysql`: 3306
  - `rabbitmq`: 4369, 25672, 5671, 5672, 15672
- 초기화 과정을 마친 후 변경 가능

Install MicroStack

```
$ sudo snap install microstack --devmode --edge
microstack (edge) ussuri from Canonical✓ installed
```

- `--devmode`: 옵션을 지정해야 `beta`, `edge` 채널 선택 가능
- `--edge`: `beta` 와 `edge` 중에 여기서는 `edge` 채널 선택

Performance Considerations

- OpenStack 플랫폼은 많은 시스템 자원 요구
- 리눅스 시스템이 많은 자원을 할당할 수 있도록 설정 변경

```
echo fs.inotify.max_queued_events=1048576 | sudo tee -a /etc/sysctl.conf
echo fs.inotify.max_user_instances=1048576 | sudo tee -a /etc/sysctl.conf
echo fs.inotify.max_user_watches=1048576 | sudo tee -a /etc/sysctl.conf
echo vm.max_map_count=262144 | sudo tee -a /etc/sysctl.conf
echo vm.swappiness=1 | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
```

Initialize MicroStack

- `microstack init --auto --control` 실행
- `--auto`: 설정할 내용을 물어보지 않고 기본값으로 지정
- `--control`: 노드의 역할을 `control` 로 지정

```
$ sudo microstack init --auto --control
2021-10-31 01:23:52,405 - microstack_init - INFO - Configuring clustering ...
2021-10-31 01:23:52,622 - microstack_init - INFO - Setting up as a control node.
2021-10-31 01:23:55,194 - microstack_init - INFO - Generating TLS Certificate and Key
2021-10-31 01:23:56,366 - microstack_init - INFO - Configuring networking ...
2021-10-31 01:24:04,184 - microstack_init - INFO - Opening horizon dashboard up to *
2021-10-31 01:24:04,989 - microstack_init - INFO - Waiting for RabbitMQ to start ...
Waiting for 192.168.50.193:5672
2021-10-31 01:24:10,607 - microstack_init - INFO - RabbitMQ started!
2021-10-31 01:24:10,607 - microstack_init - INFO - Configuring RabbitMQ ...
2021-10-31 01:24:11,391 - microstack_init - INFO - RabbitMQ Configured!
2021-10-31 01:24:11,432 - microstack_init - INFO - Waiting for MySQL server to start ...
Waiting for 192.168.50.193:3306
2021-10-31 01:24:18,127 - microstack_init - INFO - Mysql server started! Creating databases ...
2021-10-31 01:24:19,857 - microstack_init - INFO - Configuring Keystone Fernet Keys ...
2021-10-31 01:24:31,771 - microstack_init - INFO - Bootstrapping Keystone ...
2021-10-31 01:24:38,734 - microstack_init - INFO - Creating service project ...
2021-10-31 01:24:42,944 - microstack_init - INFO - Keystone configured!
```

Initialize MicroStack (continued)

```
2021-10-31 01:24:42,992 - microstack_init - INFO - Configuring the Placement service...
2021-10-31 01:24:58,179 - microstack_init - INFO - Running Placement DB migrations...
2021-10-31 01:25:01,369 - microstack_init - INFO - Configuring nova control plane services ...
2021-10-31 01:25:09,739 - microstack_init - INFO - Running Nova API DB migrations (this may take a lot of time)...
2021-10-31 01:25:27,872 - microstack_init - INFO - Running Nova DB migrations (this may take a lot of time)...
Waiting for 192.168.50.193:8774
2021-10-31 01:26:14,258 - microstack_init - INFO - Creating default flavors...
2021-10-31 01:26:33,122 - microstack_init - INFO - Configuring nova compute hypervisor ...
2021-10-31 01:26:33,122 - microstack_init - INFO - Checking virtualization extensions presence on the host
2021-10-31 01:26:33,168 - microstack_init - INFO - Hardware virtualization is supported -
KVM will be used for Nova instances
2021-10-31 01:26:36,073 - microstack_init - INFO - Configuring the Spice HTML5 console service...
2021-10-31 01:26:36,763 - microstack_init - INFO - Configuring Neutron
Waiting for 192.168.50.193:9696
2021-10-31 01:27:41,786 - microstack_init - INFO - Configuring Glance ...
Waiting for 192.168.50.193:9292
2021-10-31 01:28:09,618 - microstack_init - INFO - Adding cirros image ...
2021-10-31 01:28:11,862 - microstack_init - INFO - Creating security group rules ...
2021-10-31 01:28:18,997 - microstack_init - INFO - Configuring the Cinder services...
2021-10-31 01:29:01,779 - microstack_init - INFO - Running Cinder DB migrations...
2021-10-31 01:29:12,013 - microstack_init - INFO - restarting libvirt and virtlogd ...
2021-10-31 01:29:30,216 - microstack_init - INFO - Complete. Marked microstack as initialized!
```

Check Running Services

- 초기화 과정에서 오류가 발생했다면, 겹치는 포트가 있는지 확인
- `snap services microstack` 명령으로 각 서비스 실행 여부 확인

```
$ sudo snap services microstack
Service                                 Startup   Current   Notes
microstack.cinder-backup                disabled  inactive  -
microstack.cinder-scheduler             enabled   active    -
microstack.cinder-uwsgi                 enabled   active    -
microstack.cinder-volume                disabled  inactive  -
microstack.cluster-uwsgi                enabled   active    -
microstack.external-bridge              enabled   inactive  -
microstack.filebeat                     disabled  inactive  -
microstack.glance-api                   enabled   active    -
microstack.horizon-uwsgi                enabled   active    -
microstack.iscsid                       disabled  inactive  -
microstack.keystone-uwsgi               enabled   active    -
microstack.libvirtd                     enabled   active    -
microstack.load-modules                 enabled   inactive  -
microstack.memcached                    enabled   active    -
```

Check Running Services (continued)

```
microstack.mysqld                       enabled   active    -
microstack.neutron-api                  enabled   active    -
microstack.neutron-ovn-metadata-agent   enabled   active    -
microstack.nginx                        enabled   active    -
microstack.nova-api                     enabled   active    -
microstack.nova-api-metadata            enabled   active    -
microstack.nova-compute                 enabled   active    -
microstack.nova-conductor               enabled   active    -
microstack.nova-scheduler               enabled   active    -
microstack.nova-spicehtml5proxy         enabled   active    -
microstack.nrpe                         disabled  inactive  -
microstack.ovn-controller               enabled   active    -
microstack.ovn-northd                   enabled   active    -
microstack.ovn-ovsdb-server-nb          enabled   active    -
microstack.ovn-ovsdb-server-sb          enabled   active    -
microstack.ovs-vswitchd                 enabled   active    -
microstack.ovsdb-server                 enabled   active    -
microstack.placement-uwsgi              enabled   active    -
microstack.rabbitmq-server              enabled   active    -
microstack.setup-lvm-loopdev            disabled  inactive  -
microstack.target                       disabled  inactive  -
microstack.telegraf                     disabled  inactive  -
microstack.virtlogd                     enabled   active    -
```

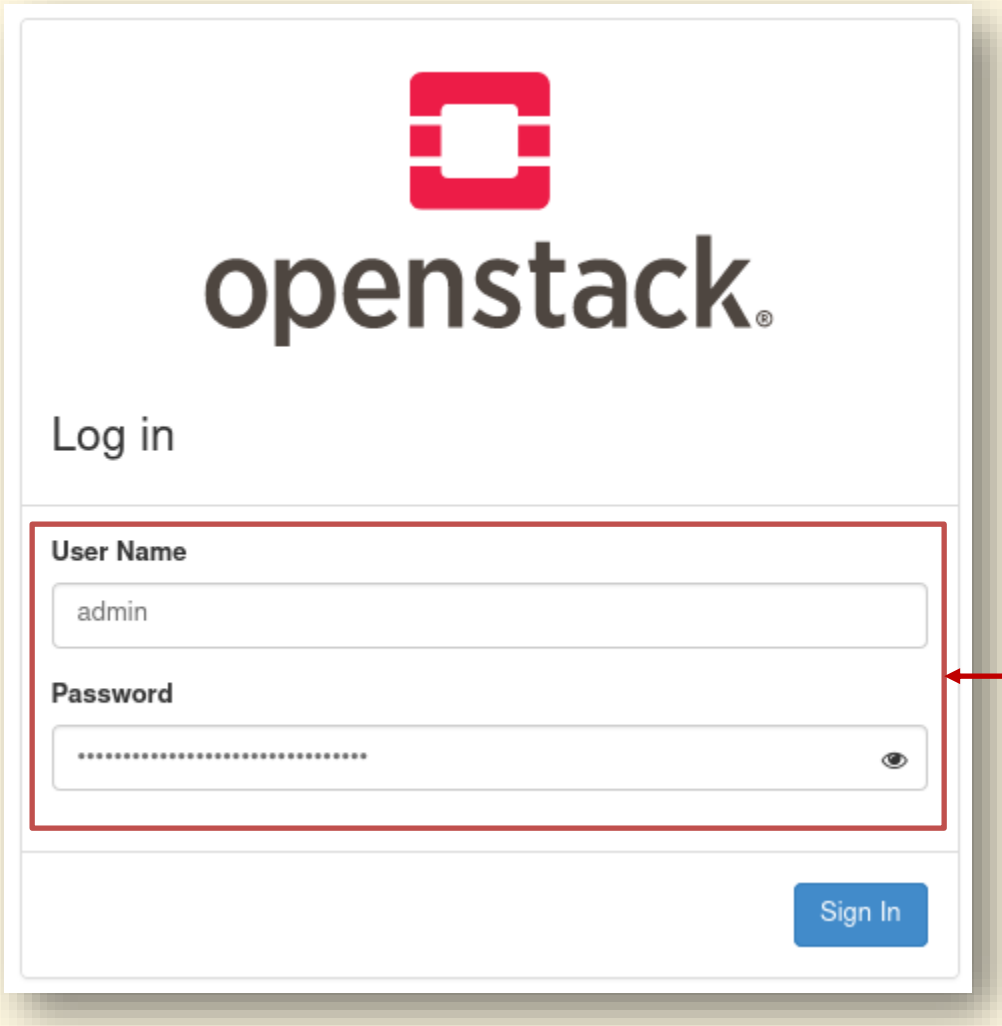
Access Web UI

- Web UI 주소: https://10.20.20.1
- 기본 계정 정보: `admin`
- 초기화 과정에서 생성한 비밀번호는 아래 명령으로 확인 가능

```
$ sudo snap get microstack config.credentials.keystone-password
qAlpQQqc..............JqzDhVRq
```

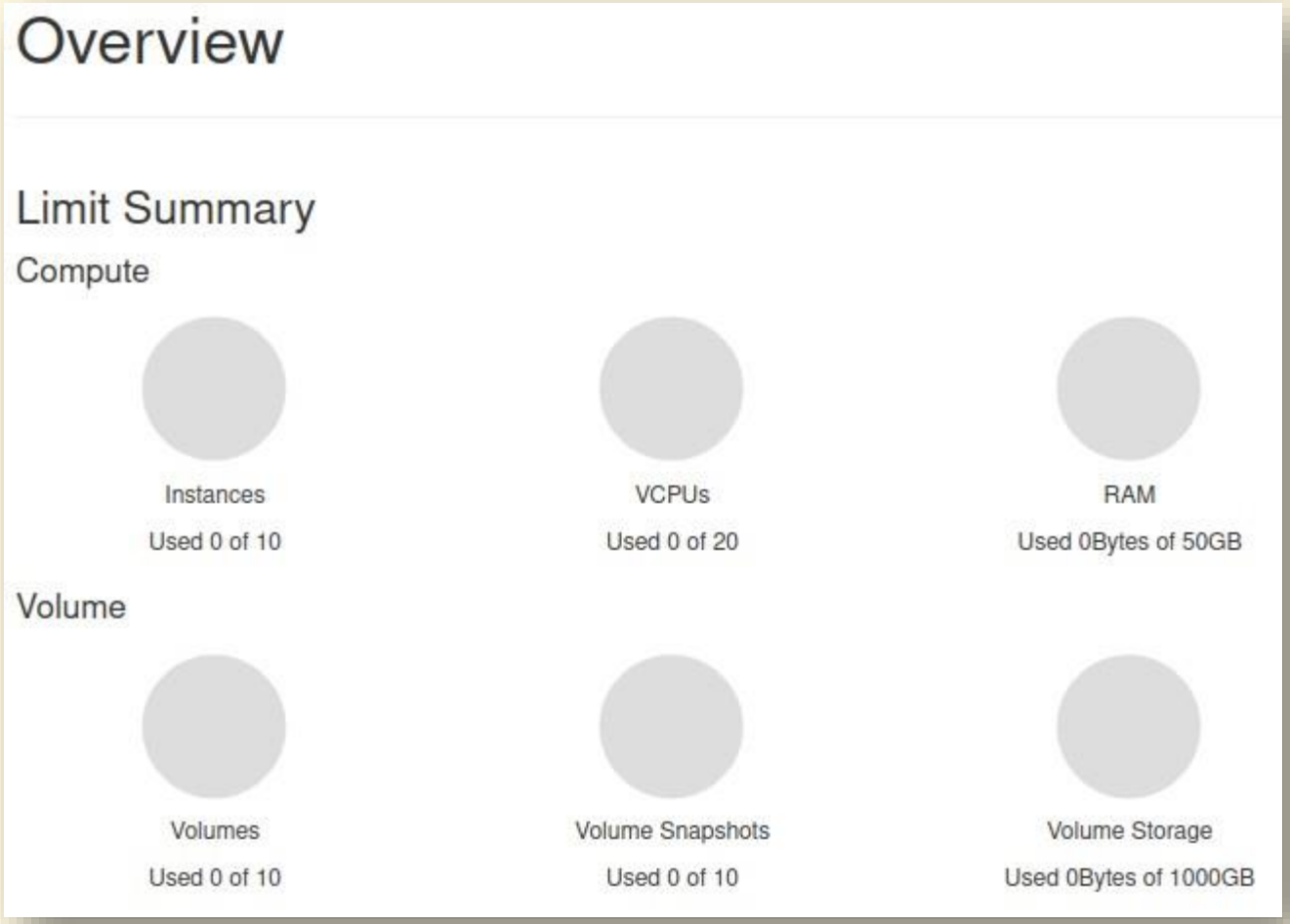
- `admin` 계정과 비밀번호를 이용해 Web UI 접속

Access Web UI (continued)



admin 계정과
비밀번호를 이용해 로그인

Compute: Overview



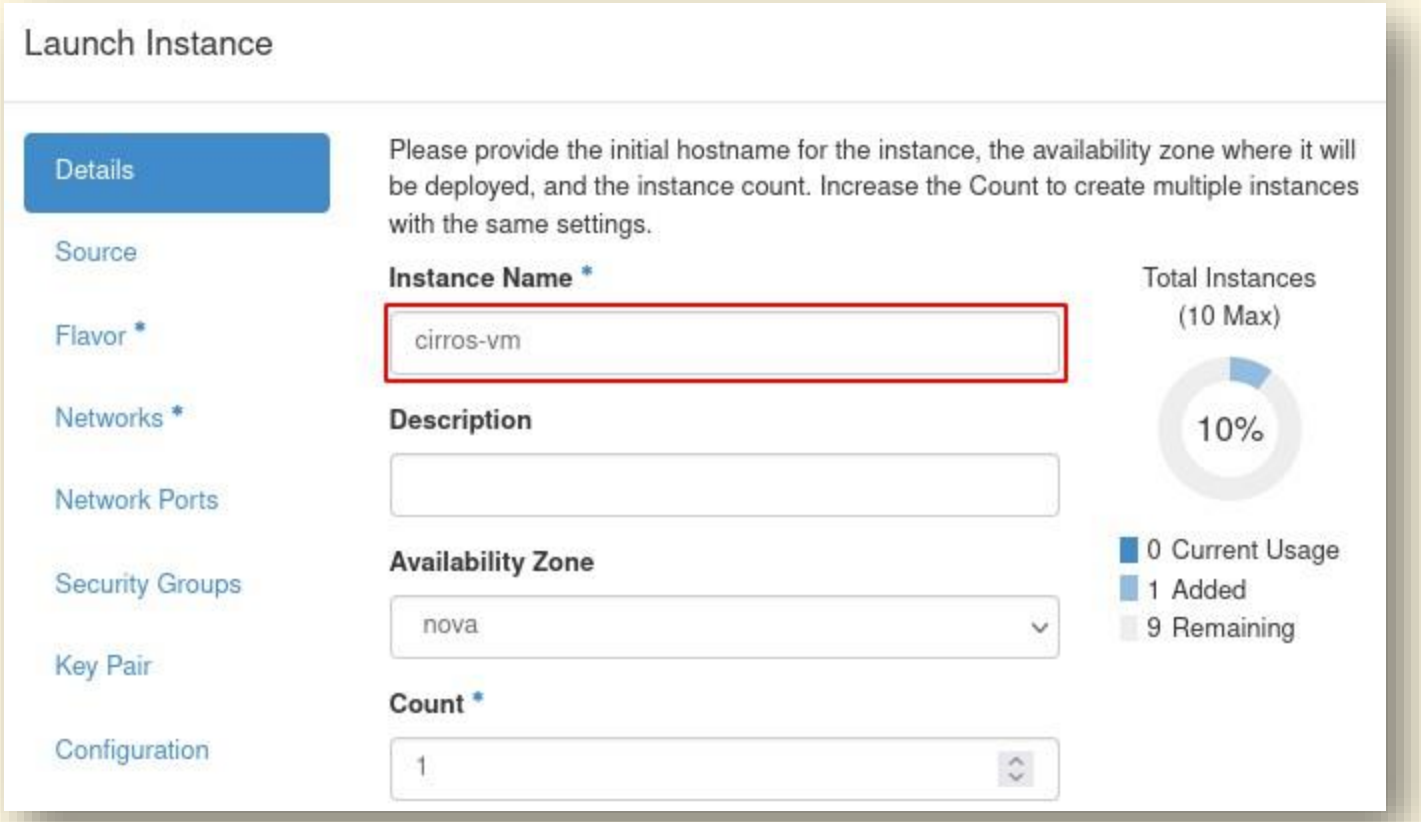
Launch Instance

- 터미널에서 `microstack.openstack` 명령 사용 가능
- 여기에서는 Web UI 를 이용해 `cirros` 이미지로 VM 생성

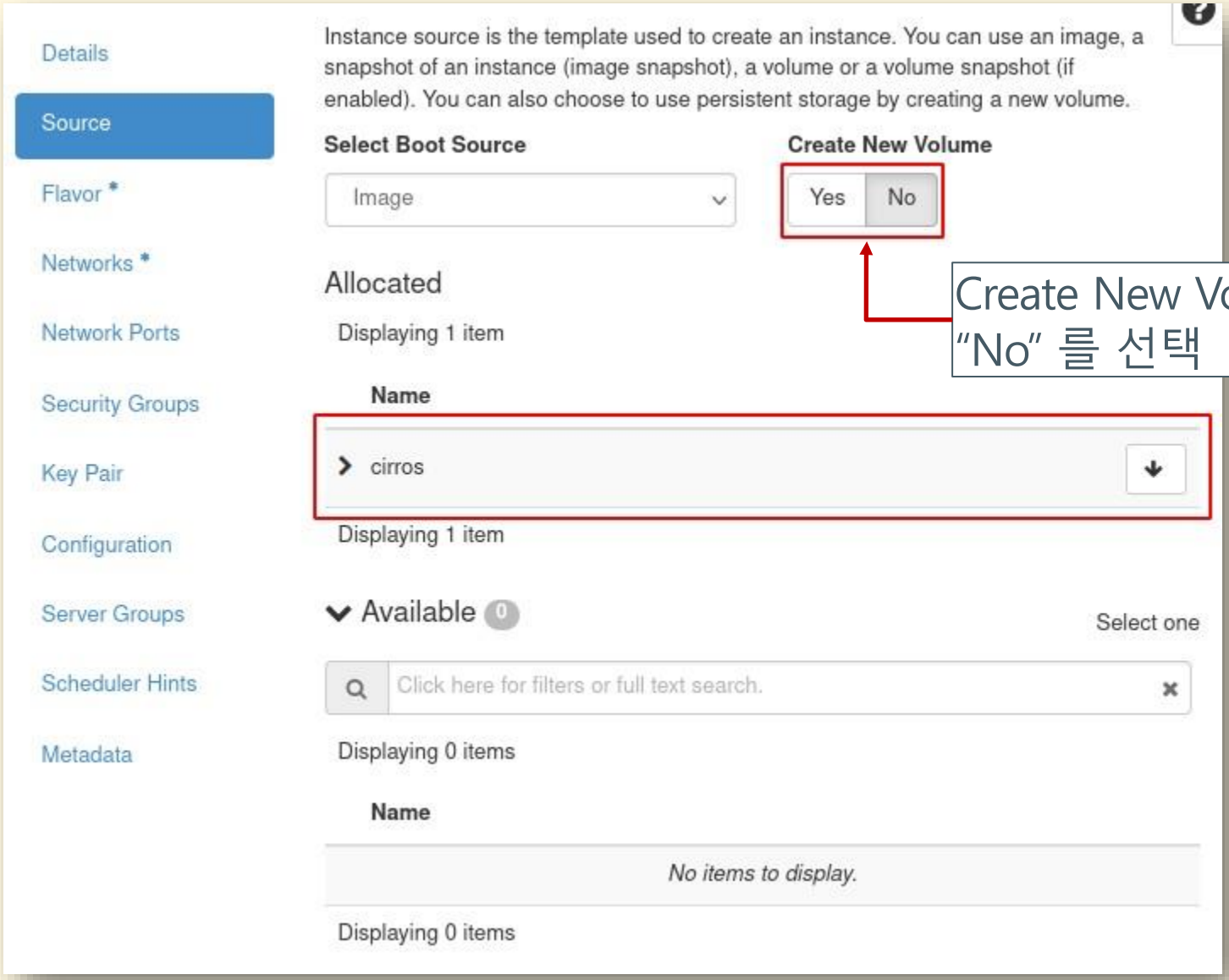
Launch Instance 를 클릭해 새로운 Instance 생성

Launch Instance: Detail

- `Instance Name` 항목에 `cirros-vm` 입력

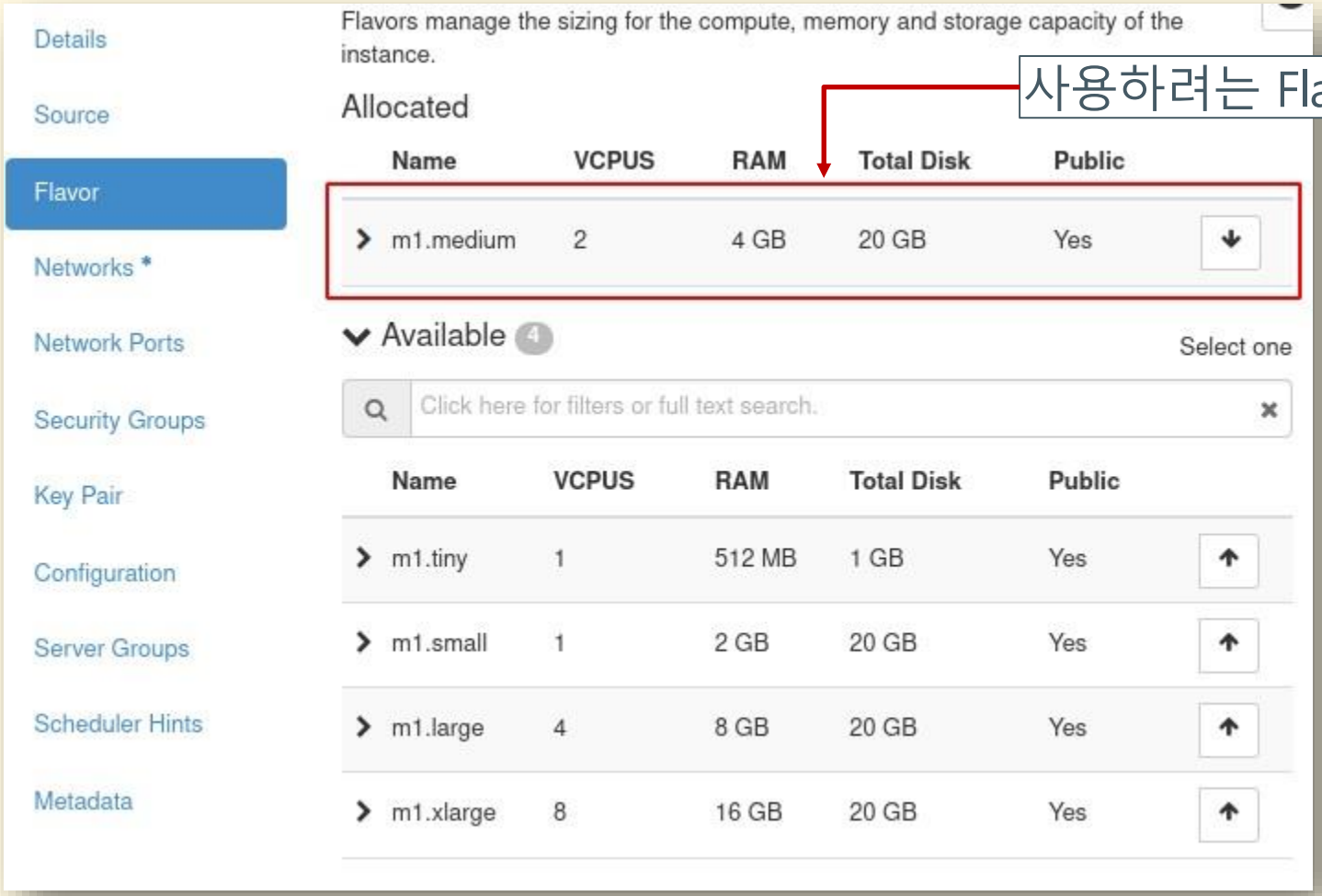


Launch Instance: Source



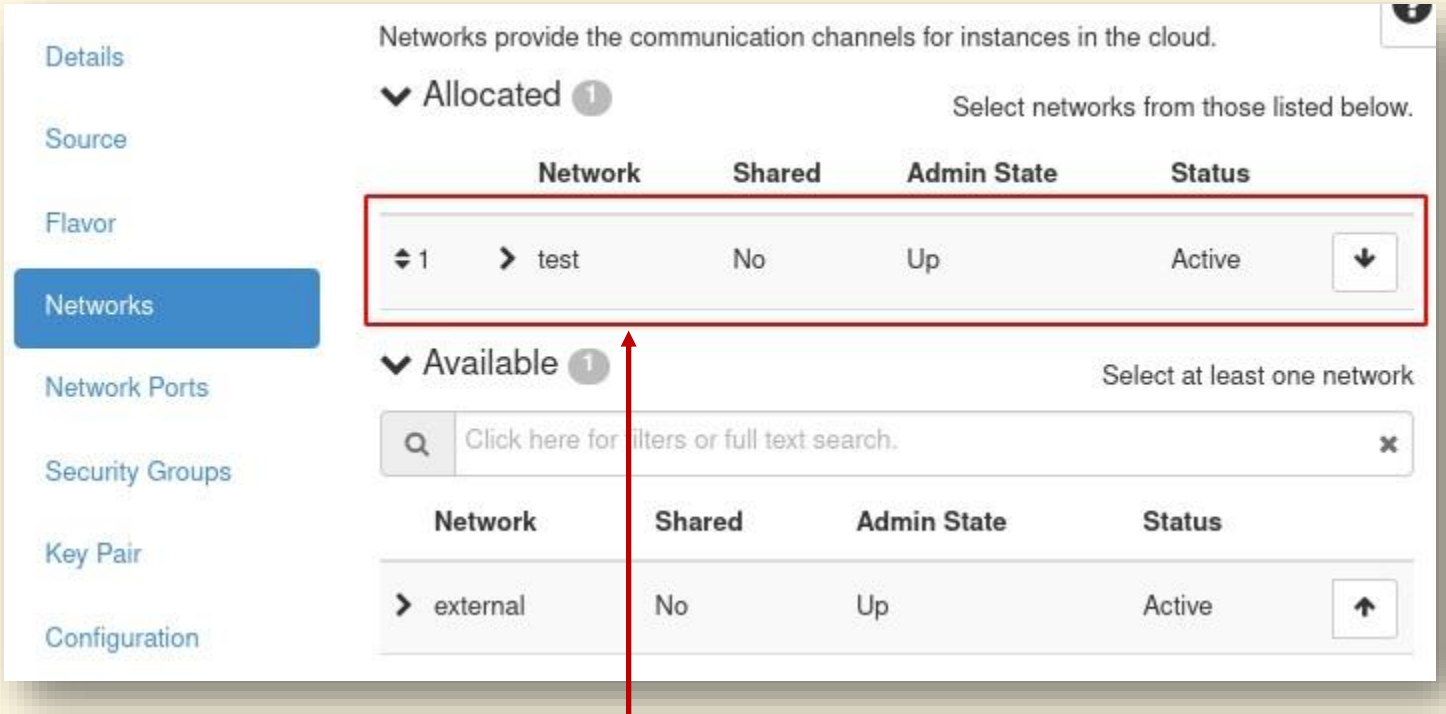
Create New Volume 은 "No" 를 선택

Launch Instance: Flavor



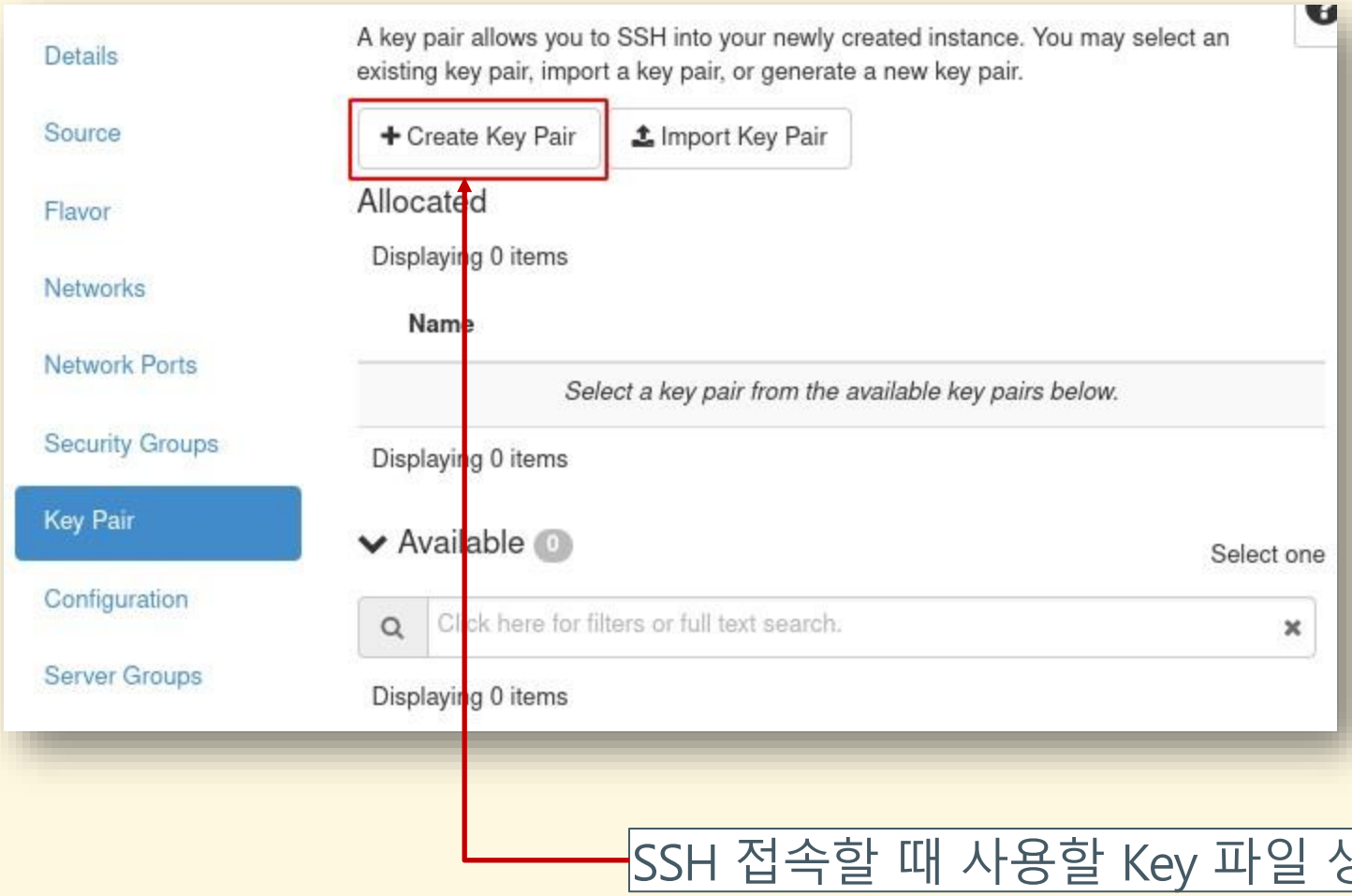
사용하려는 Flavor 선택

Launch Instance: Networks



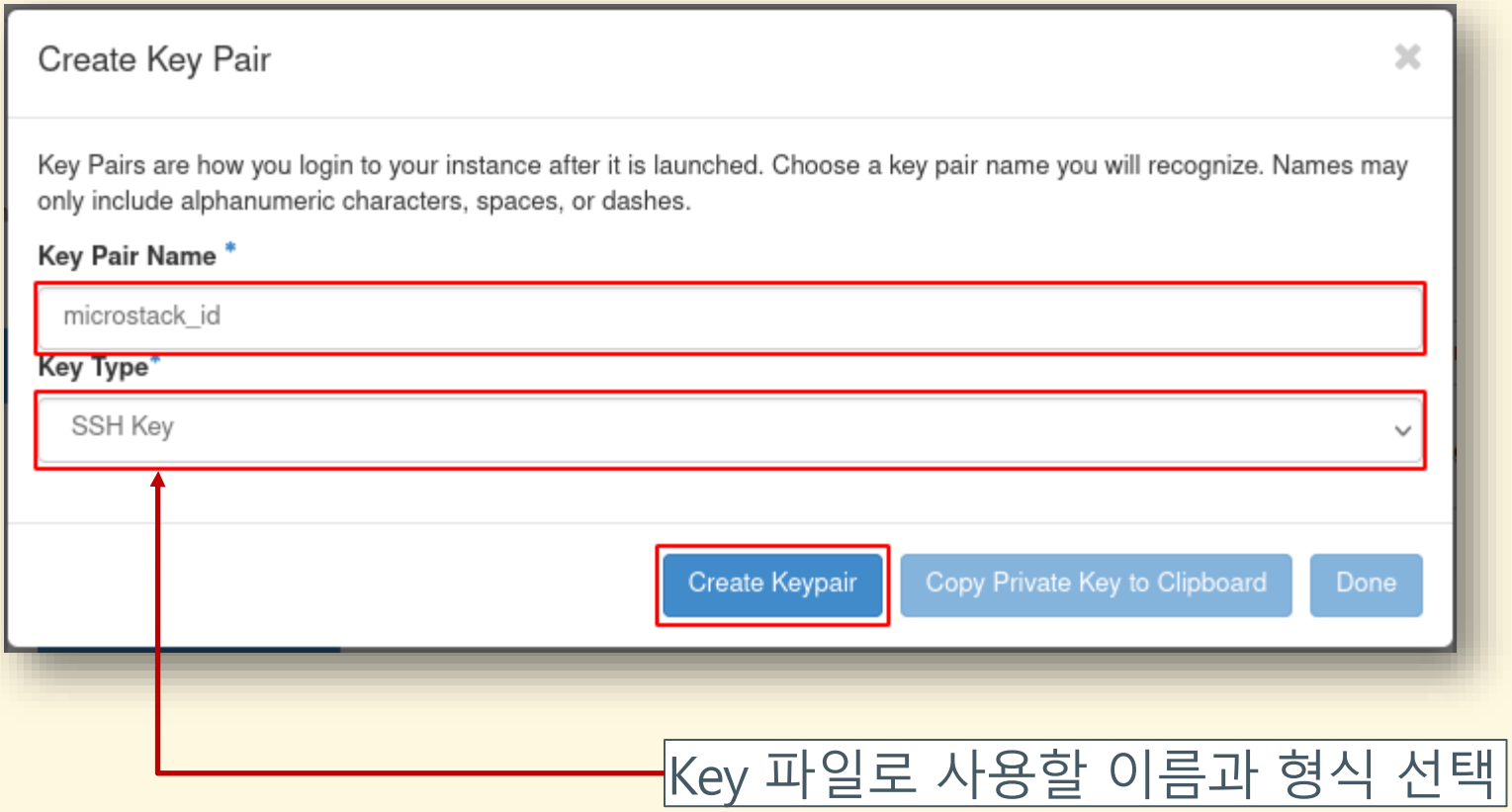
미리 생성되어 있는 test 네트워크 선택

Launch Instance: Key Pair

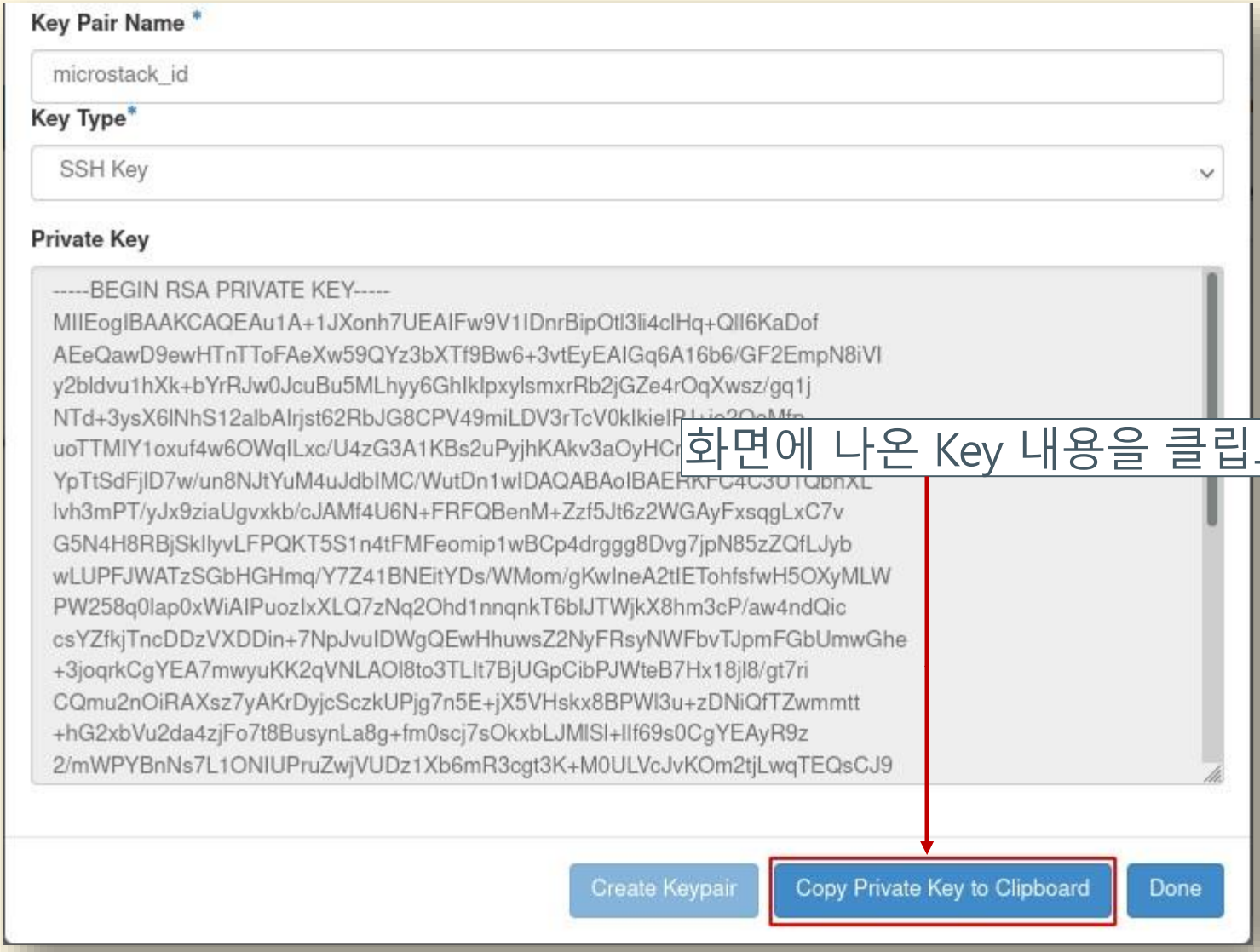


SSH 접속할 때 사용할 Key 파일 생성

Launch Instance: Key Pair (continued)



Launch Instance: Key Pair (continued)

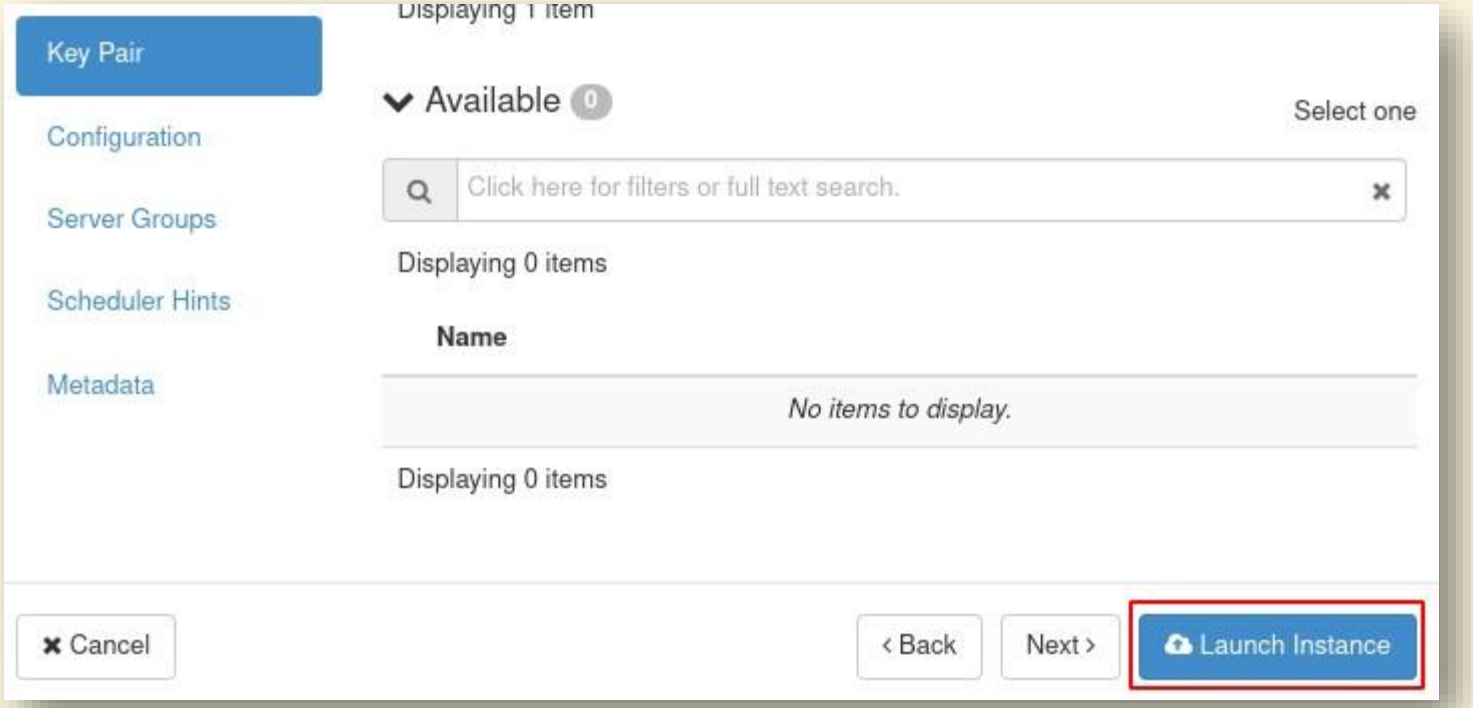


Launch Instance: Key Pair (continued)

- 생성한 비밀키를 `$HOME/.ssh` 안에 저장하고 권한 지정
- 비밀키는 소유자만 읽고 쓸 수 있도록 권한 변경 필요

```
$ cd ~/.ssh
$ cat > microstack_id
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAu1A+1JXonh7UEAIFw9V1IDnrBipOtl3li4clHq+QlI6KaDof
AEeQawD9ewHTnTToFAeXw59QYz3bXTf9Bw6+3vtEyEAIGq6A16b6/GF2EmpN8iVI
...
y2bldvu1hXko/GDeyoqkX8Uzz/OGET0mNNN0aHMo7Cajp3P+biHEtfPL6HqNBrakmpP6iO+Lz3t
JZSiLhiiVQQ8brdrMnRJwbuiQViOeDfZFlfWSpTquAkofk5rzLM=
-----END RSA PRIVATE KEY-----
CTRL+D

$ chmod 600 microstack_id
```

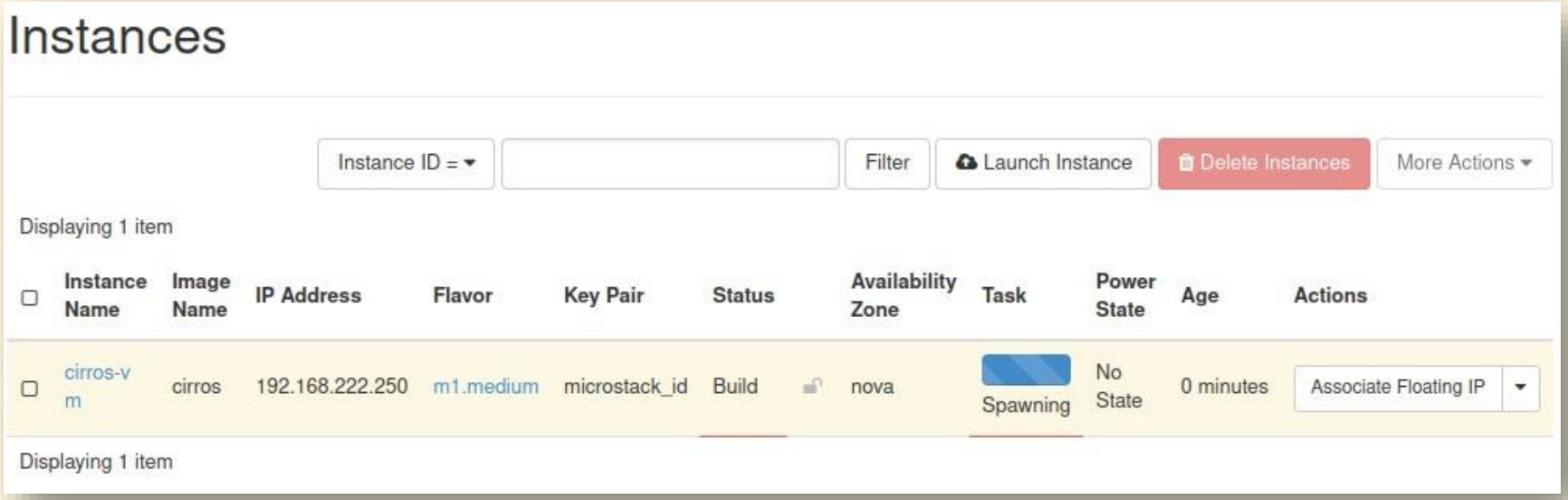
Launch Instance: Launch



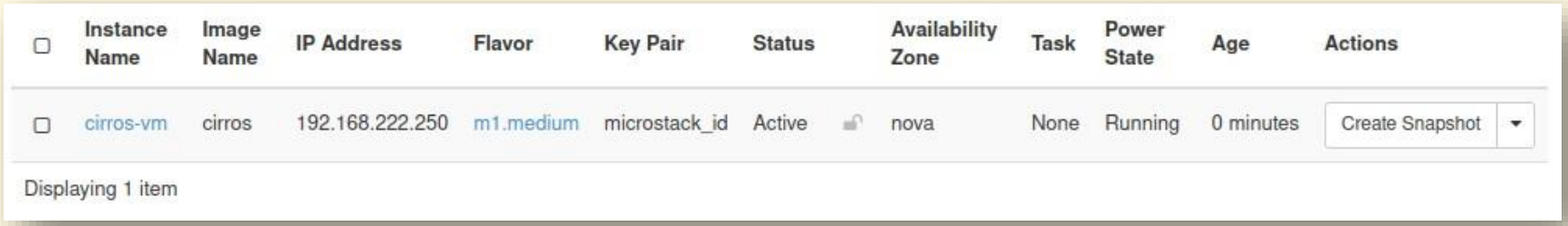
- `Launch Instance` 버튼을 눌러 VM 생성

Instances: Spawning

- 설정이 올바르다면 `cirros-vm` 생성 시작
- 만약에 오류가 발생한다면, `Source` 항목에서 `Create New Volume` 을 선택했는지 확인



Instances: Associate Floating IP



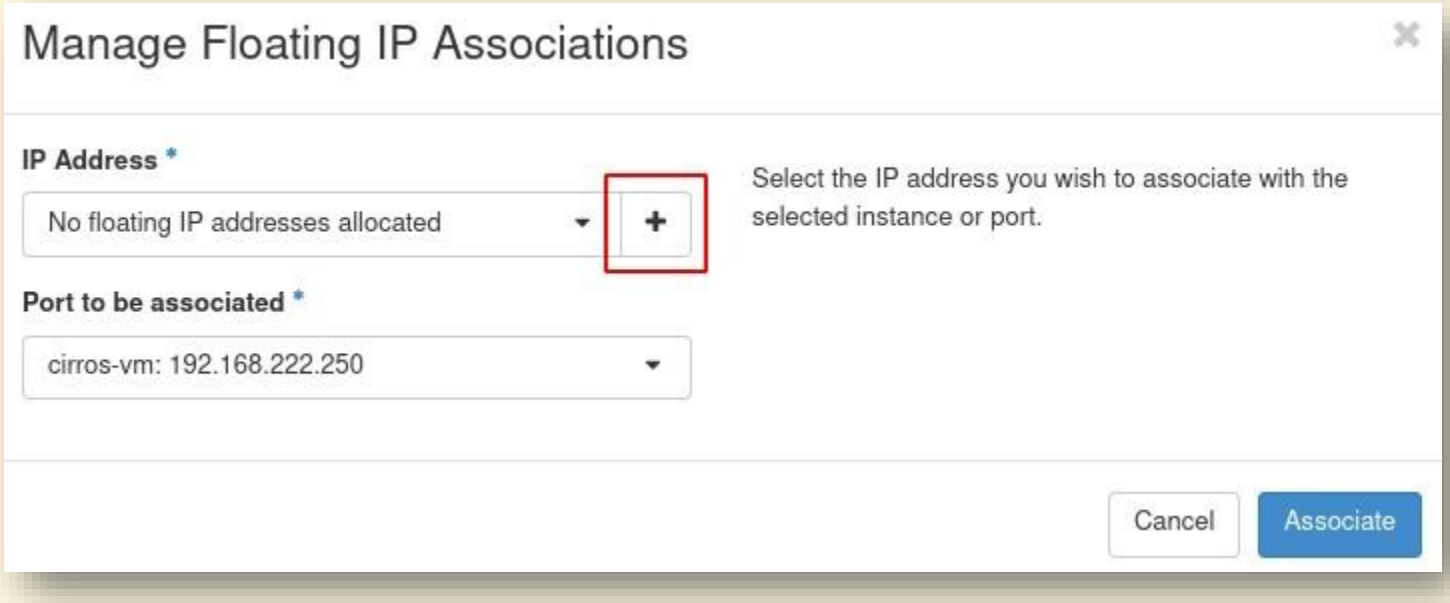
Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
cirros-vm	cirros	192.168.222.250	m1.medium	microstack_id	Active	nova	None	Running	0 minutes	Create Snapshot

Displaying 1 item

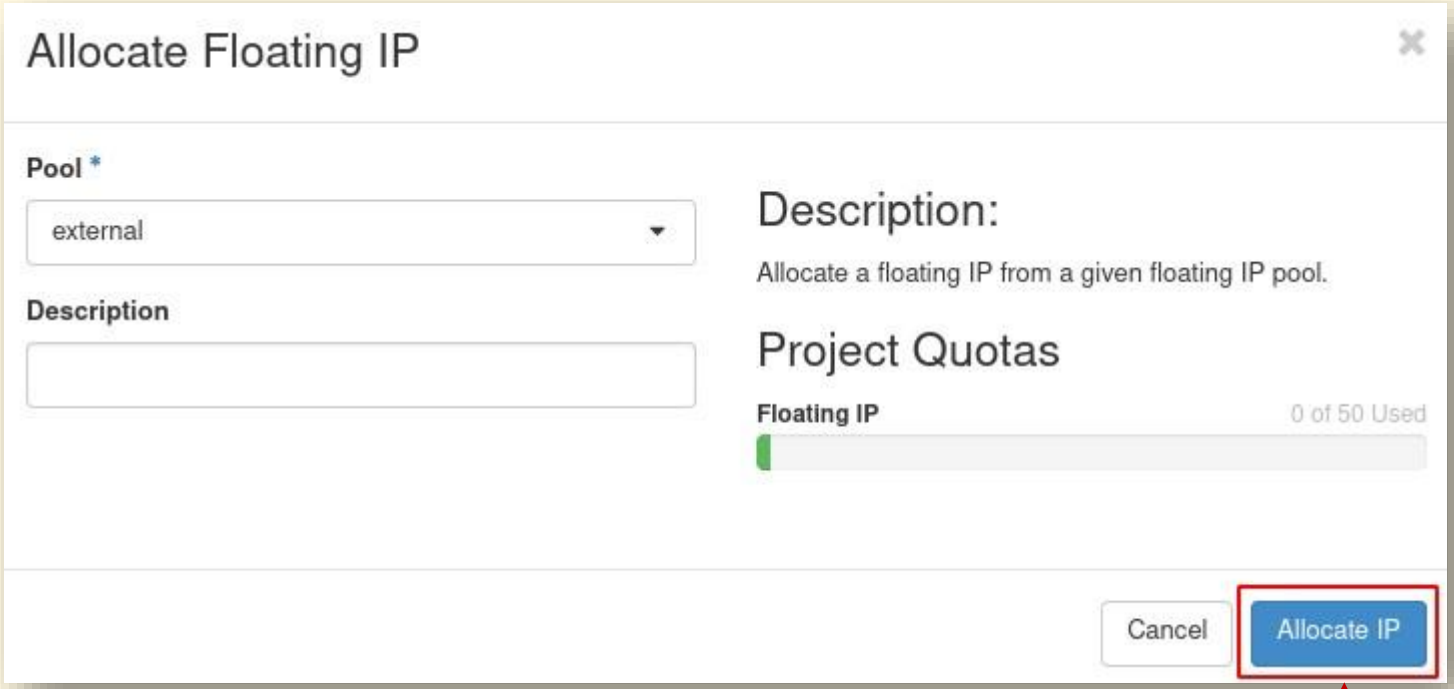
- 내부 네트워크에 생성한 VM 에 `Floating IP` 연결

Instances: Manage Floating IP

- Floating IP 추가

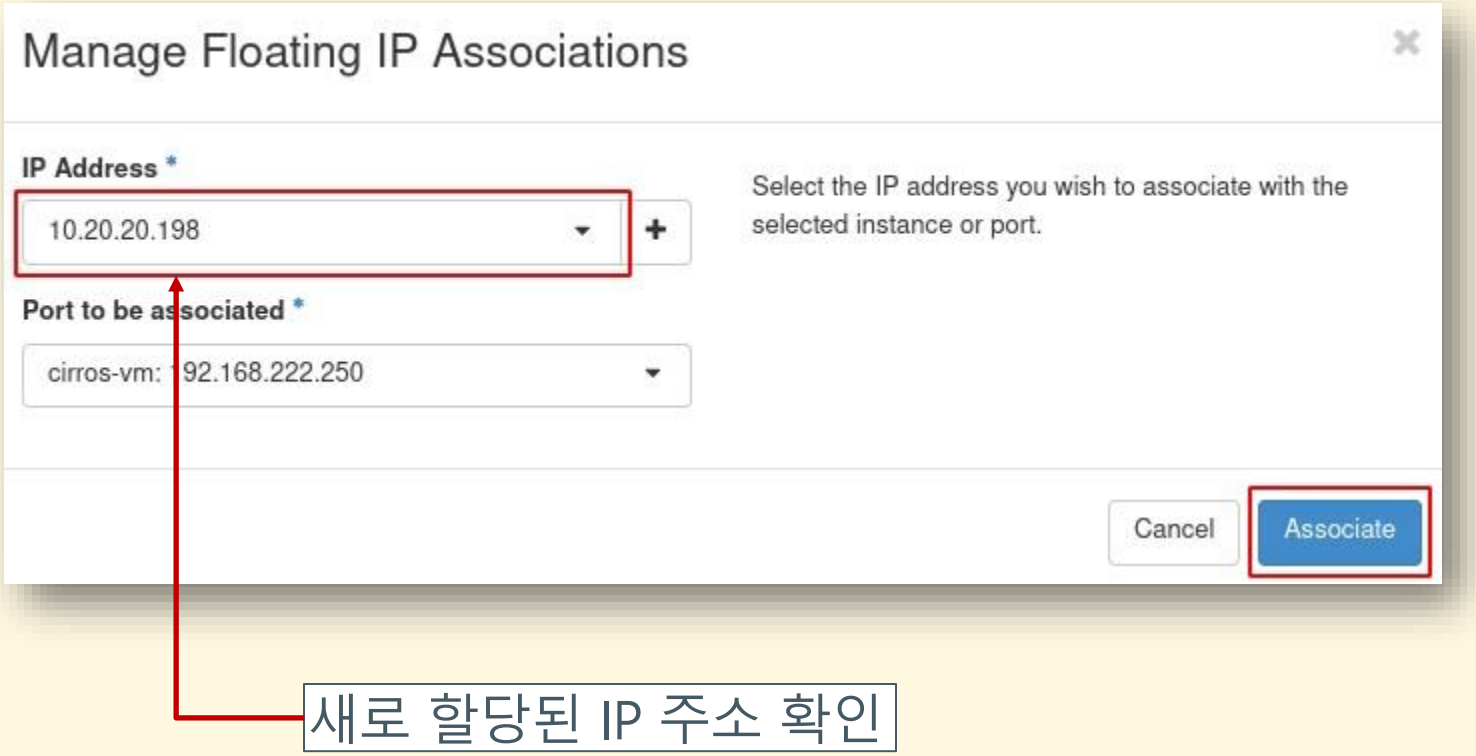


Instances: Allocate Floating IP



Allocate IP 버튼을 눌러 새로운 주소 할당

Instances: Assign Floating IP



Access VM

- `ssh` 명령을 이용해 Floating IP 로 접속

```
$ ssh -i ~/.ssh/microstack_id cirros@10.20.20.198
The authenticity of host '10.20.20.198 (10.20.20.198)' can't be established.
ECDSA key fingerprint is SHA256:CugRKPfWZTs1U9qRBlN6Os8cTzeQivzU+qfXjQp3q/4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.20.20.198' (ECDSA) to the list of known hosts.
$ hostname
cirros-vm
```

- 할당한 메모리 확인: 4GB

```
$ free -m
             total         used         free       shared      buffers
Mem:          3951           61         3890            0           14
-/+ buffers:                 46         3905
Swap:            0            0            0
```

Access VM (continued)

- 할당한 디스크 공간 확인: 20GB

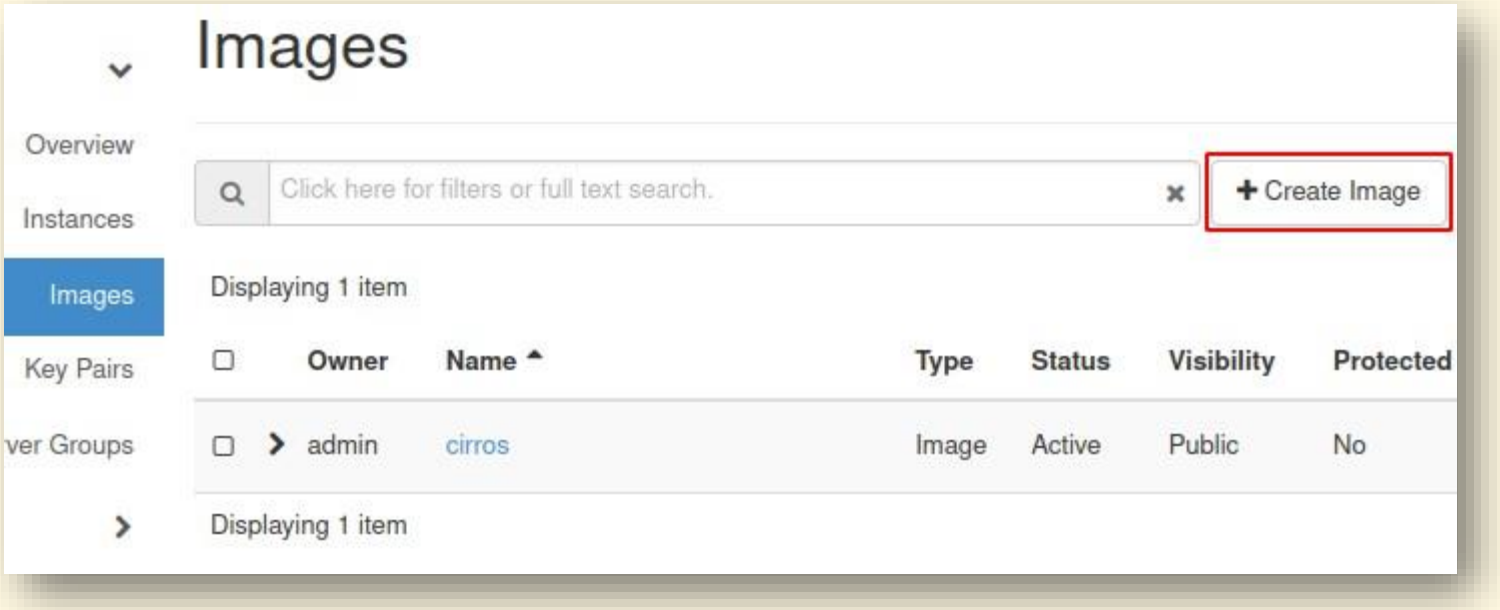
```
$ df -h
Filesyste                Size      Used Available Use% Mounted on
m
/dev                     1.9G         0      1.9G   0% /dev
/dev/vda1               19.4G     23.4M     18.7G   0% /
tmpfs                    1.9G         0      1.9G   0% /dev/shm
tmpfs                    1.9G     92.0K      1.9G   0% /run
```

- 원하는 사양으로 VM 이 생성되었음을 확인

Create Image

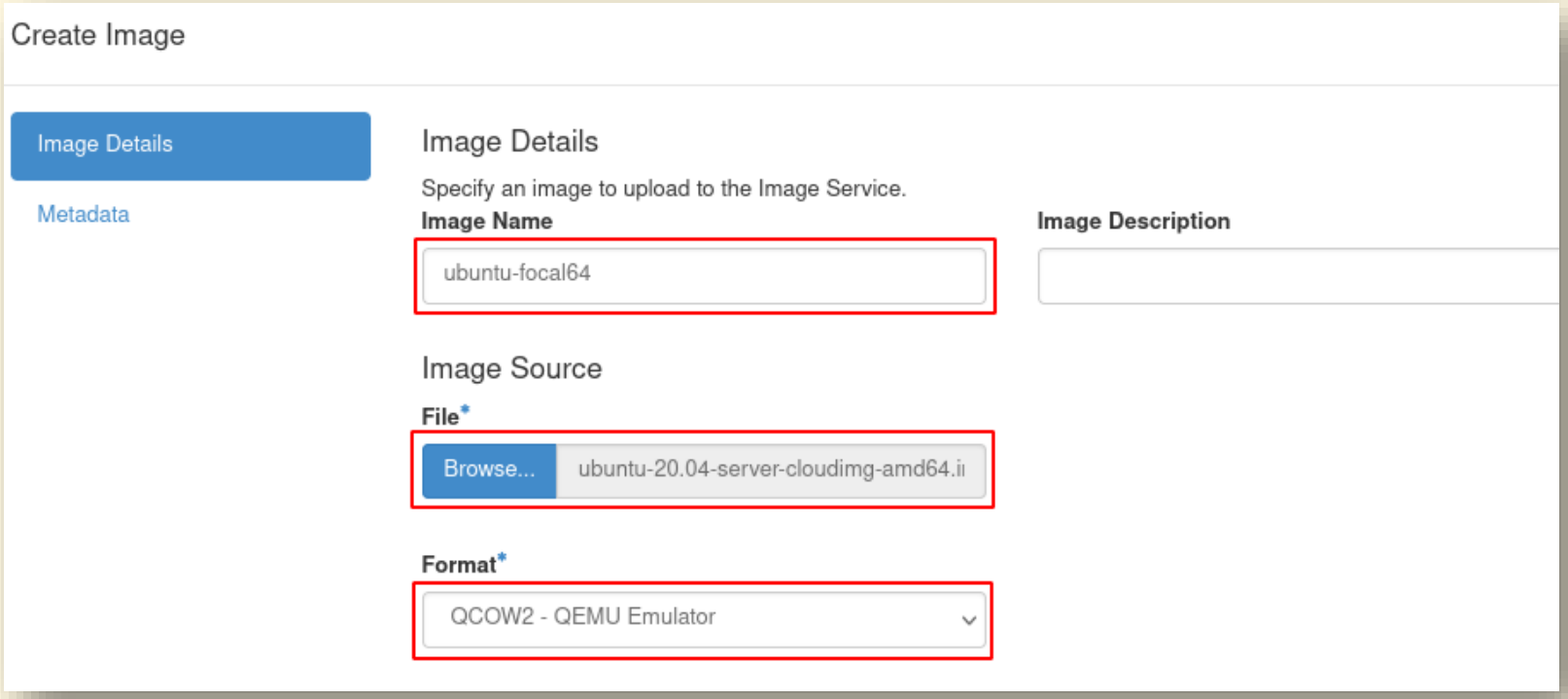
Download Ubuntu Image

- Ubuntu Cloud Image 사용
- 아래 링크에서 `ubuntu-20.04-server-cloudimg-arm64.img` 다운
  https://cloud-images.ubuntu.com/releases/focal/release/
- 받은 파일을 OpenStack 에 등록



Create Image

- `Image Name`: 사용할 이미지 이름 지정
- `Image Source`: 받은 파일 지정
- `Format`: `QCOW2` 선택



Create Image (continued)

- `Minimum Disk (GB)`: 필요한 최소 디스크 용량
- `Minimum Ram (MB)`: 필요한 최소 메모리 용량

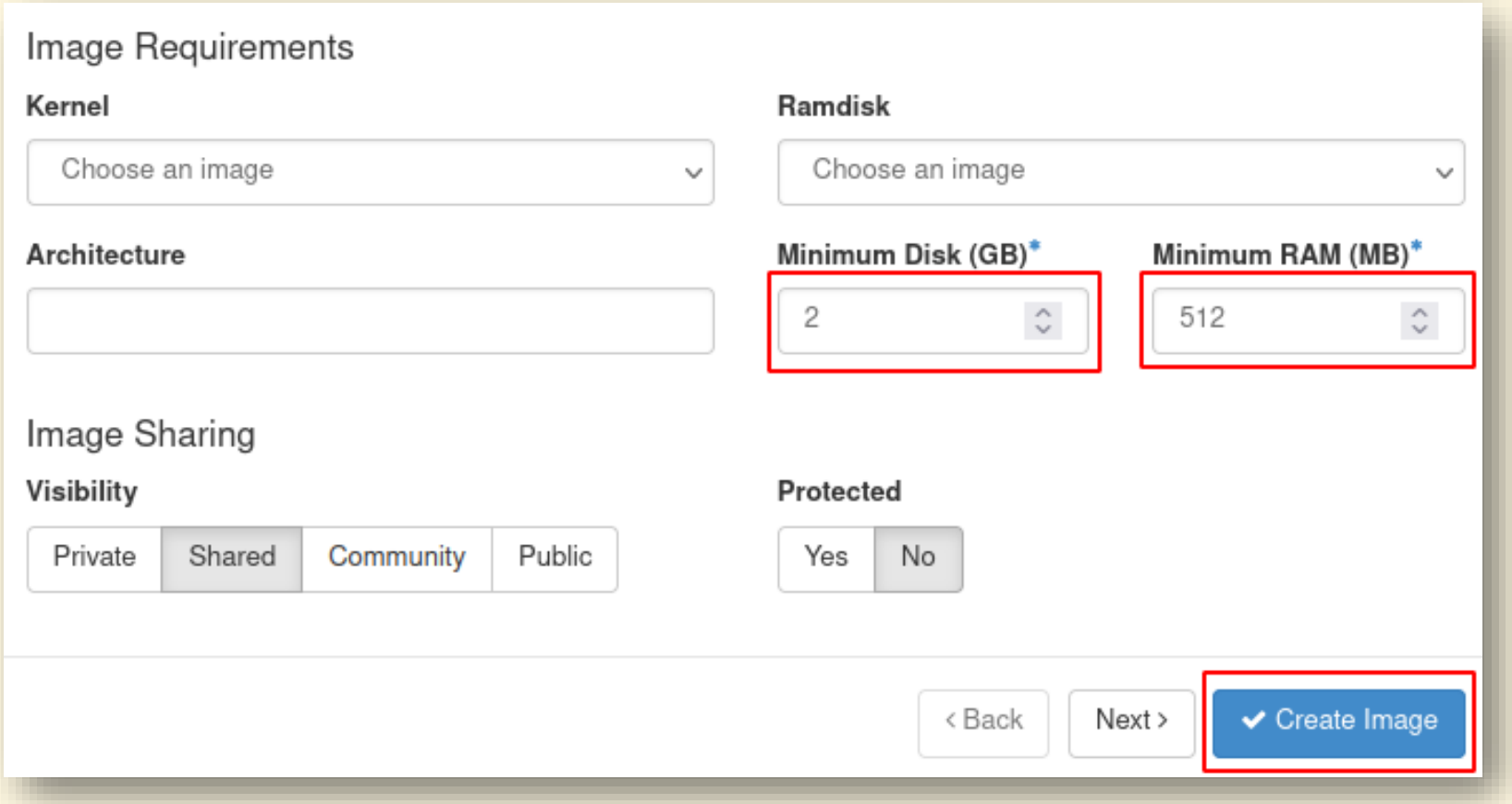
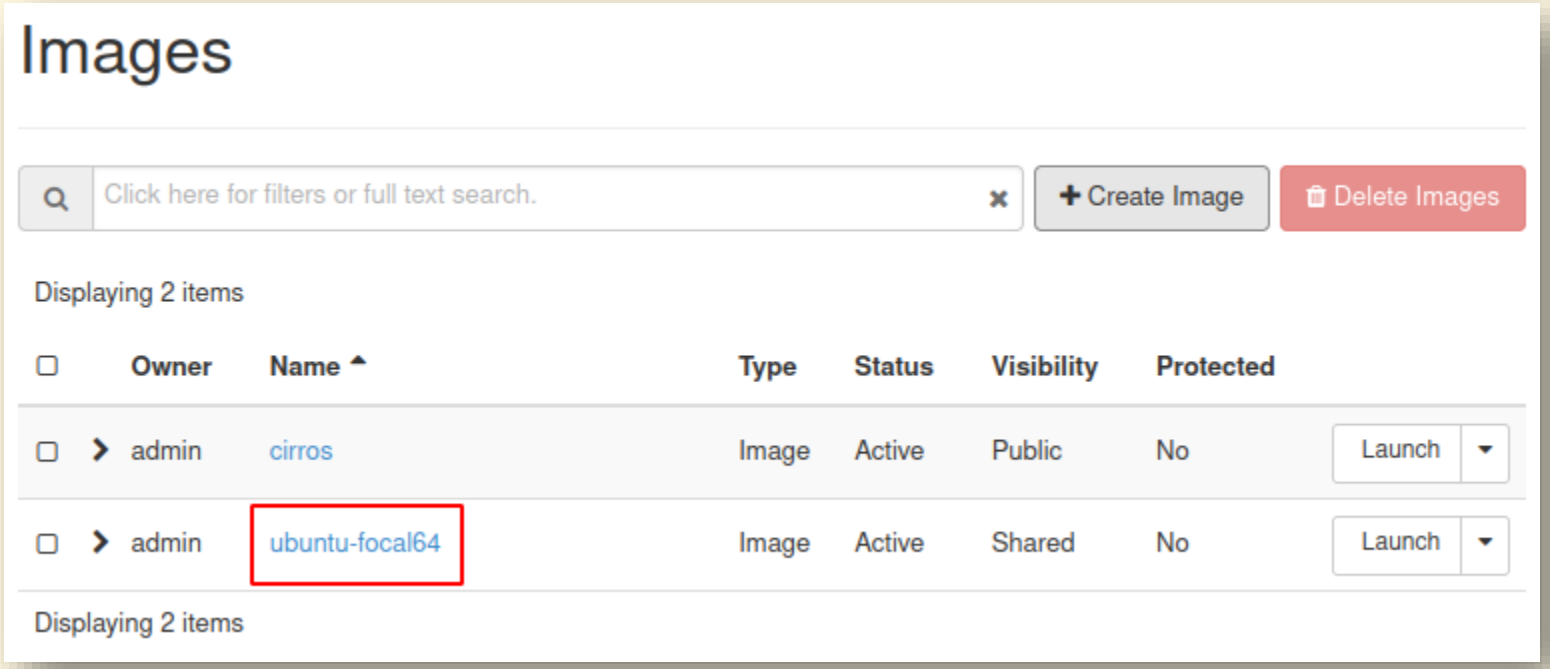


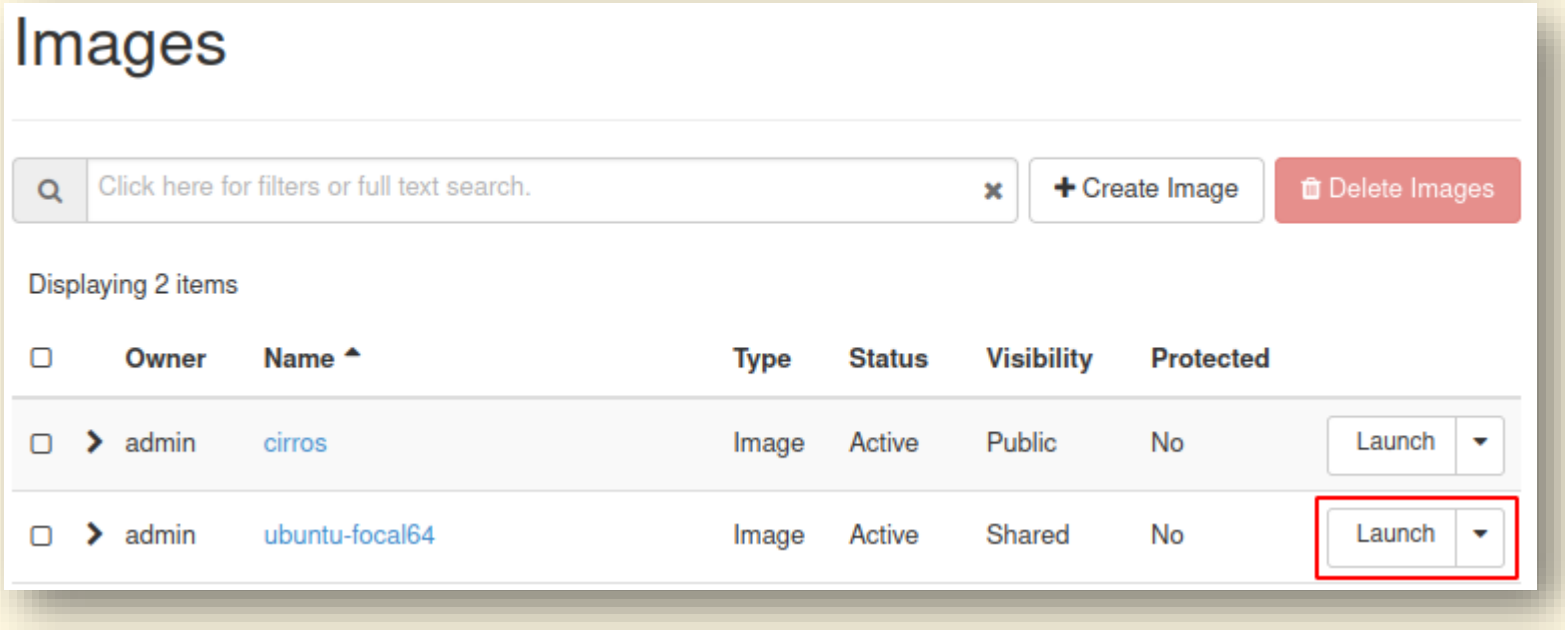
Image List

- 다운받은 파일을 이용해 이미지 생성



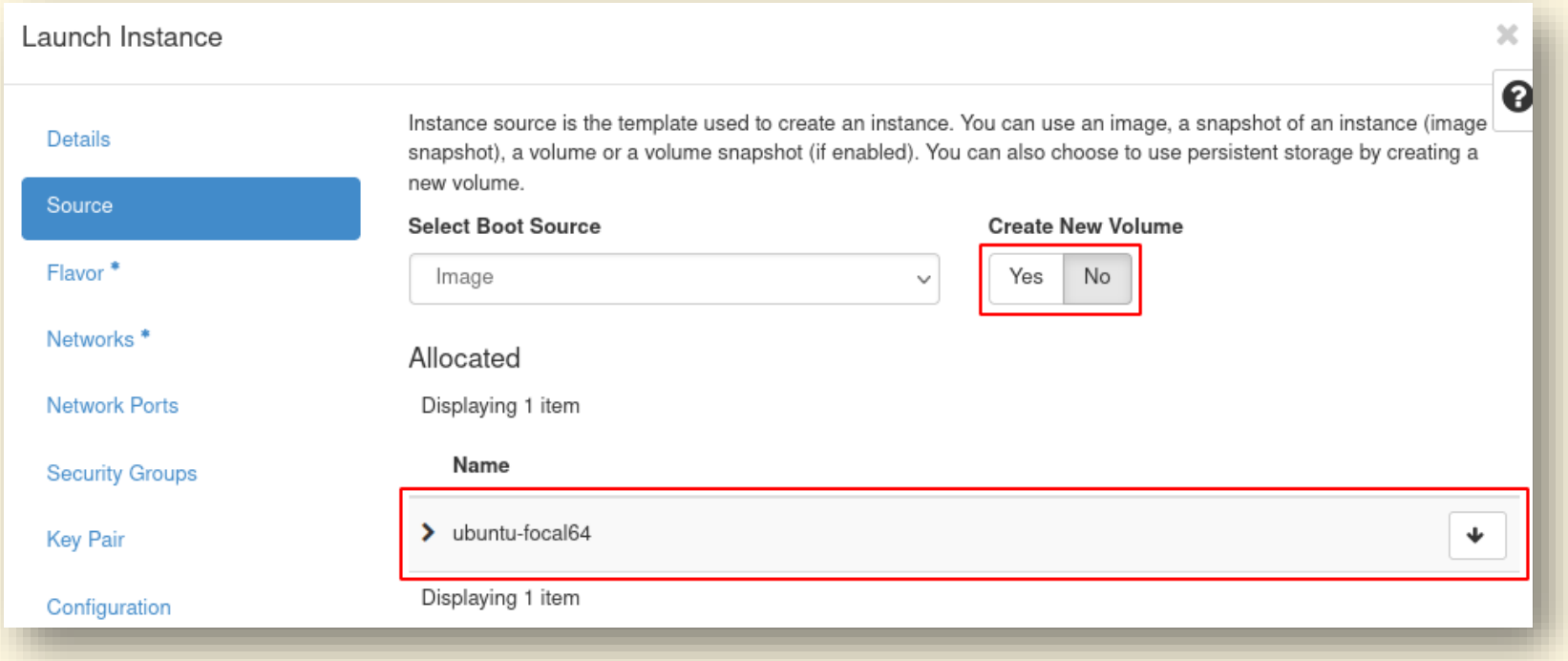
Create VM with Ubuntu Image

- 생성한 `ubuntu-focal64` 이미지를 이용해 가상 머신 생성



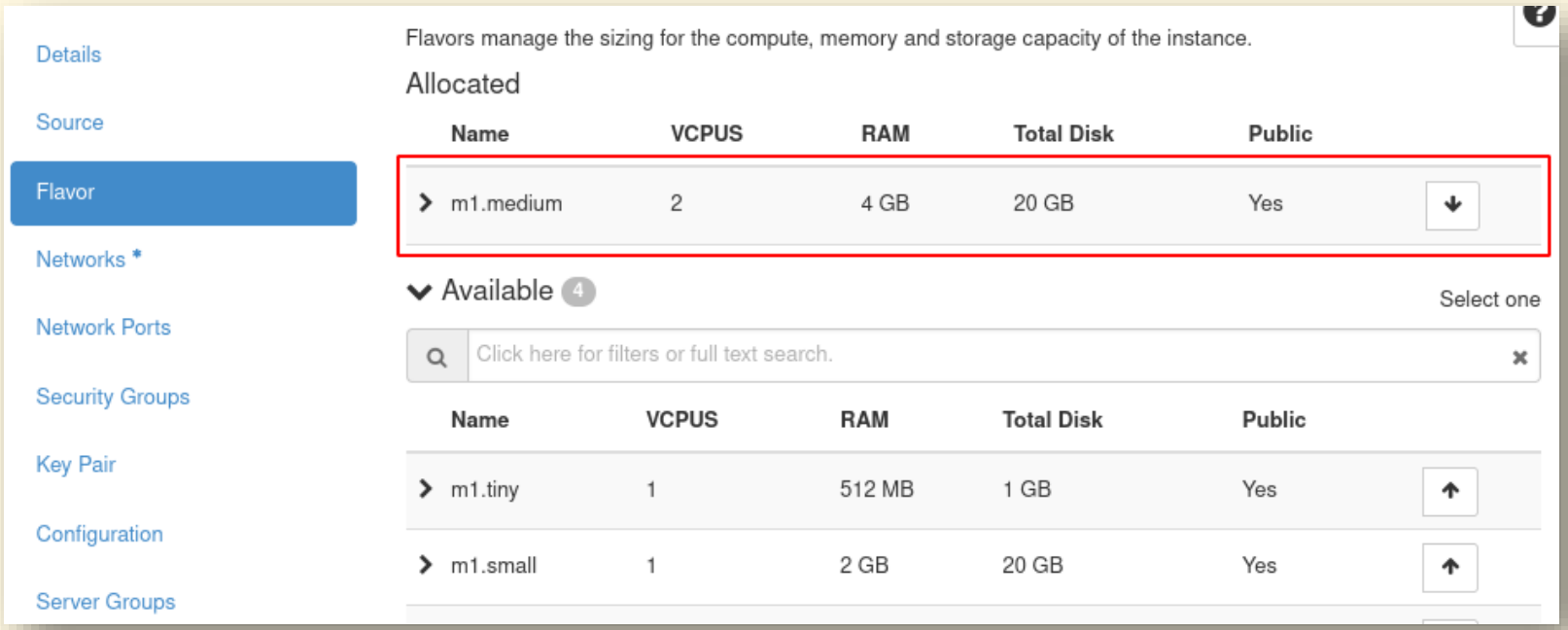
Select Source

- `ubuntu-focal64` 이미지 선택
- `Create New Volume` 은 `No` 선택



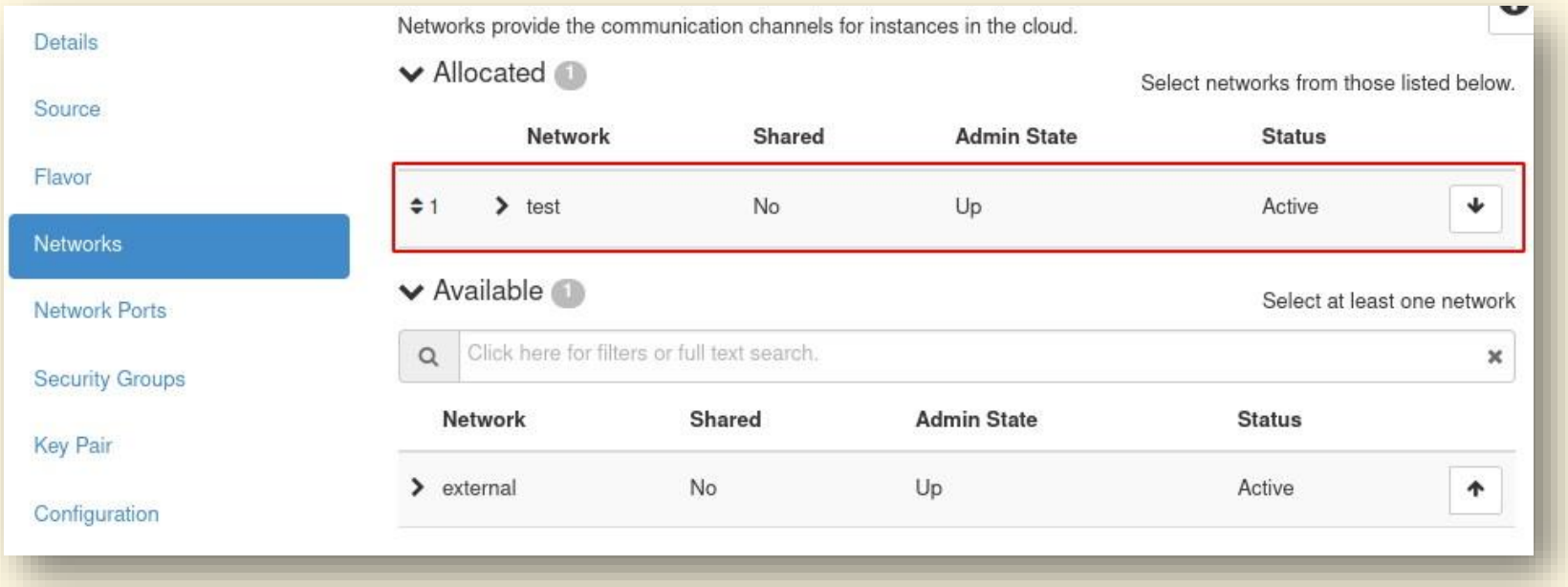
Select Flavor

- 원하는 `Flavor` 선택



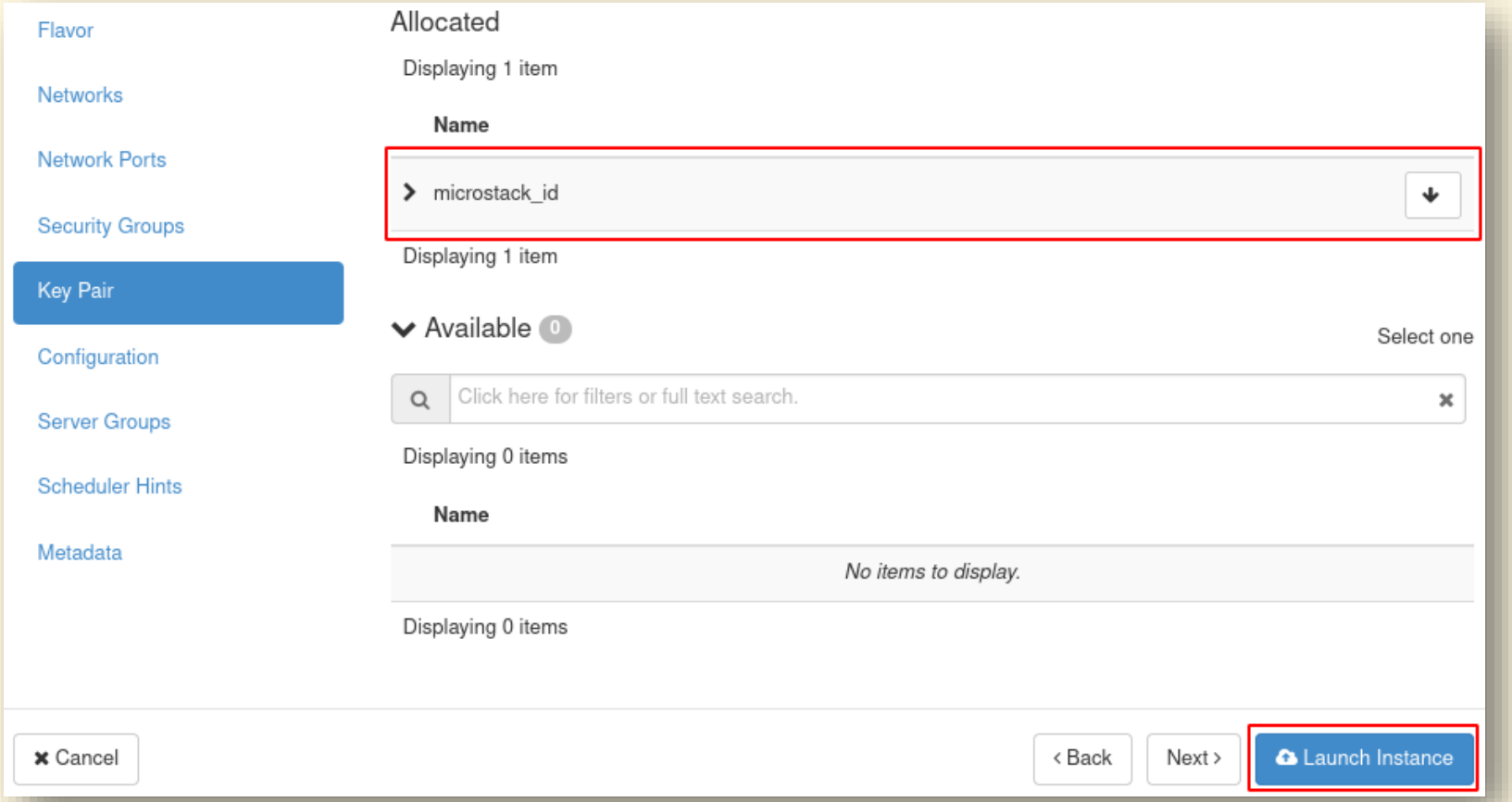
Select Networks

- 앞에서 생성한 VM 과 같은 `test` 네트워크 선택



Select Key Pair

- 앞에서 생성한 `microstack_id` 선택
- `Launch Instance` 를 선택해 VM 생성

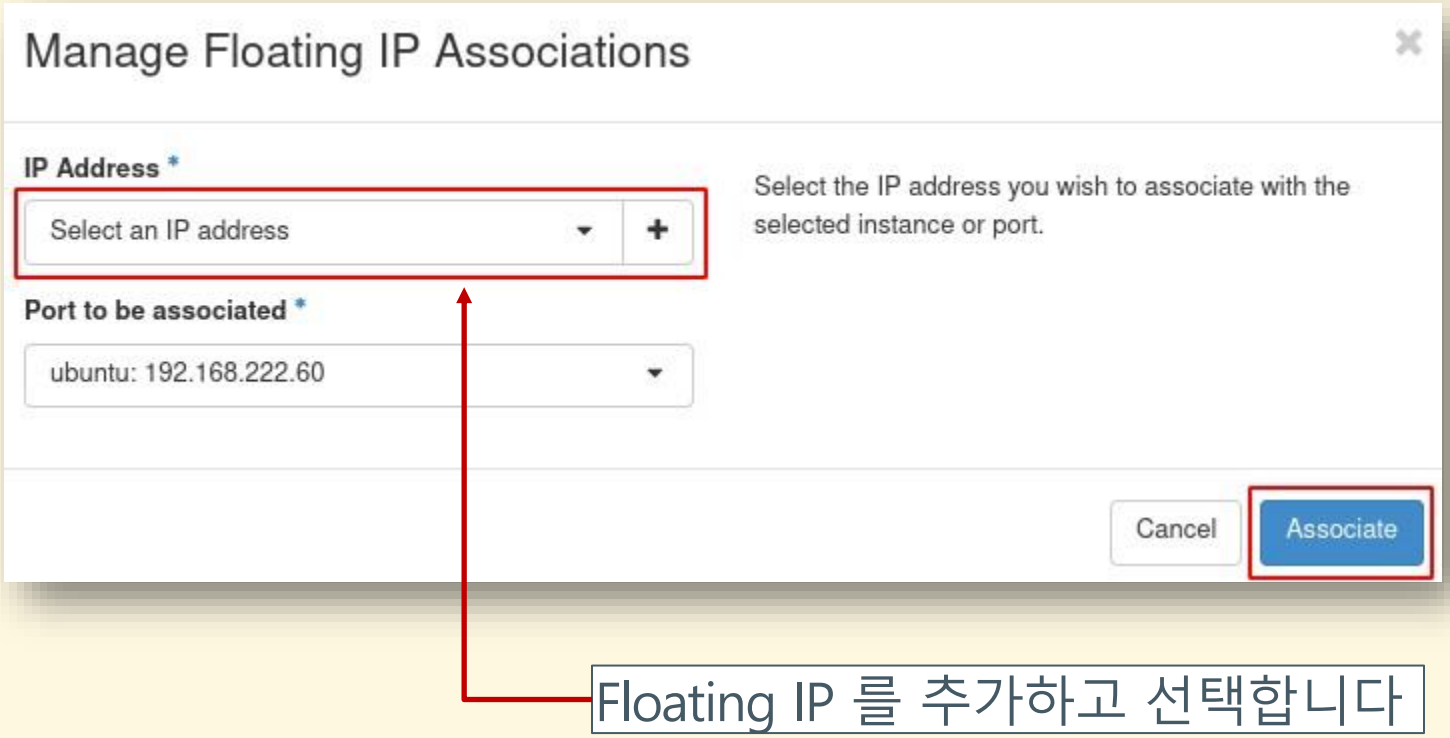


Instances

- Instance 목록에서 방금 생성한 `ubuntu` VM 확인

Attach Floating IP

- 외부에서 접속할 수 있도록 `Floating IP` 연결



Access VM

- `ubuntu` 에 할당한 `Floating IP` 주소 확인

	Instance Name	Image Name	IP Address	Flavor
☐	ubuntu	ubuntu-focal64	192.168.222.60, 10.20.20.85	m1.medium
☐	cirros-vm	cirros	192.168.222.250, 10.20.20.198	m1.medium

- 접근할 수 있는 IP 인지 확인

```
$ ping 10.20.20.85
PING 10.20.20.85 (10.20.20.85) 56(84) bytes of data.

64 bytes from 10.20.20.85: icmp_seq=1 ttl=63 time=1.71 ms
64 bytes from 10.20.20.85: icmp_seq=2 ttl=63 time=0.983 ms
64 bytes from 10.20.20.85: icmp_seq=3 ttl=63 time=0.158 ms
64 bytes from 10.20.20.85: icmp_seq=4 ttl=63 time=0.166 ms
```

Access VM (continued)

```
$ ssh -i ~/.ssh/microstack_id ubuntu@10.20.20.85
The authenticity of host '10.20.20.85 (10.20.20.85)' can't be established.
ECDSA key fingerprint is SHA256:r8pWGOfhta18u8U5JfoTQd6eVxqVhHo6lIoIuPWzBdY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.20.20.85' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com  https:
 * Management:     //landscape.canonical.com https
 * Support:        ://ubuntu.com/advantage

  System information as of Sun Oct 31 06:16:43 UTC 2021

  System load:  0.0               Processes:             117
  Usage of /:   6.6% of 19.21GB   Users logged in:       0
  Memory usage: 5%                IPv4 address for ens3: 192.168.222.60
  Swap usage:   0%

...

ubuntu@ubuntu:~$
```

Instance Networks

Internet Connectivity

- 생성한 VM 에서 외부 인터넷 연결이 제대로 되지 않는 문제

```
ubuntu@ubuntu:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1010ms

ubuntu@ubuntu:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1018ms
```

Internet Connectivity (continued)

- 인스턴스에서 외부 인터넷 연결은 막혀있는 상태
- 호스트 서버에서 인스턴스 접근은 가능

```
$ ping 10.20.20.85
PING 10.20.20.85 (10.20.20.85) 56(84) bytes of data.

64 bytes from 10.20.20.85: icmp_seq=1 ttl=63 time=1.80 ms
64 bytes from 10.20.20.85: icmp_seq=2 ttl=63 time=1.05 ms
64 bytes from 10.20.20.85: icmp_seq=3 ttl=63 time=0.177 ms
64 bytes from 10.20.20.85: icmp_seq=4 ttl=63 time=0.193 ms
```

IPTable & NAT

- 인스턴스에서 외부 인터넷과 연결하기 위해 `iptables` 사용
- `10.20.20.0/24` 대역 패킷을 `Gateway` 인터페이스로 전달
- `Gateway` 인터페이스에는 `MASQUERADE` 기능 활성화
- `10.20.20.0/24` 대역 패킷을 외부 인터넷으로 전송하도록 설정

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -A FORWARD -s 10.20.20.0/24 -i br-ex -o enp7s0 \
     -m conntrack --ctstate NEW -j ACCEPT
sudo iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED \
     -j ACCEPT
sudo iptables -t nat -A POSTROUTING -o enp7s0 -j MASQUERADE
```

Check Instance Networks

- `iptables` 명령으로 `10.20.20.0/24` 대역 패킷이 외부망으로 제대로 전송되는지 확인

```
ubuntu@ubuntu:~$ ping 1.1.1.1

PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=50 time=3.90 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=50 time=2.89 ms
^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 2.894/3.398/3.903/0.504 ms
```

Check Instance Networks (continued)

```
ubuntu@ubuntu:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
...
Fetched 20.4 MB in 6s (3293 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
10 packages can be upgraded. Run 'apt list --upgradable' to see them.

ubuntu@ubuntu:~$ sudo apt upgrade -y
Reading package lists... Done  Build
ing dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apport bind9-dnsutils bind9-host bind9-libs libnetplan0
  netplan.io open-vm-tools python3-apport
  python3-problem-report tzdata
...
```

Volumes

LVM Volumes

- LVM 으로 `Logical Volume` 을 생성해 추가 가능
- 디스크 전체를 사용하거나 디스크 이미지 파일을 사용할 수 있음
- LVM 을 사용하기 위해 `lvm2` 패키지 설치 필요

```
$ sudo apt install lvm2
```

- 디스크 이미지 파일을 생성해 볼륨으로 사용할 수 있도록 설정

Create LVM Disk Image

- `/dev/loop*` 을 이용해 이미지 파일을 블록 디바이스로 사용

```
sudo fallocate -l 400G /mnt/ext-disk/cinder-volumes.img
sudo losetup /dev/loop99 /mnt/ext-disk/cinder-volumes.img
sudo pvcreate /dev/loop99
sudo vgcreate cinder-volumes /dev/loop99
```

- 블록 스토리지와 관련된 서비스 실행

```
sudo snap start --enable microstack.cinder-volume
sudo snap start --enable microstack.cinder-backup
sudo snap start --enable microstack.iscsid
```

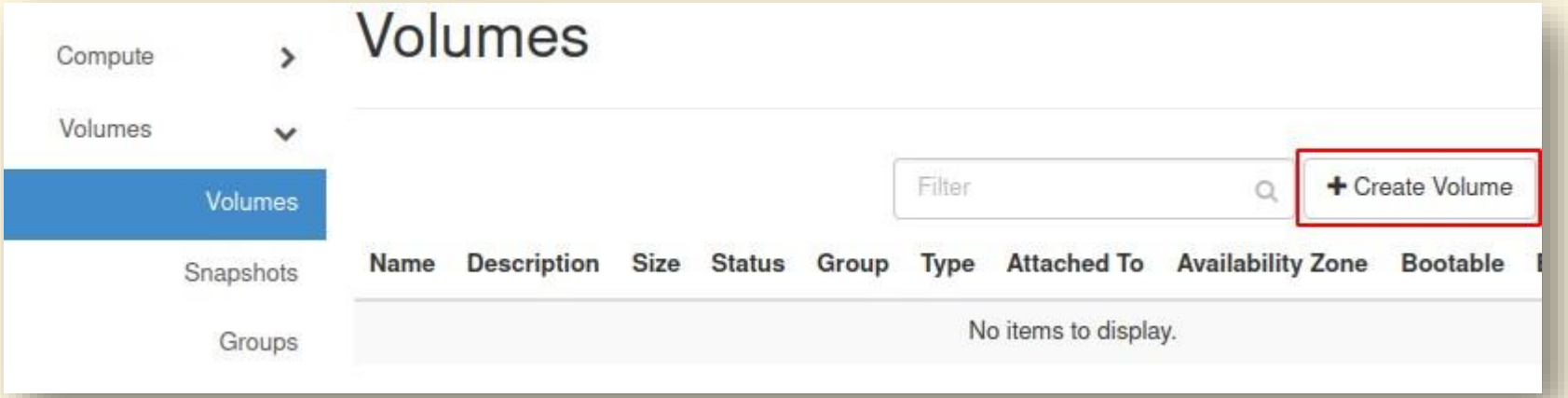
Create LVM Disk Image (continued)

- 생성한 LVM Volume 확인
  - Physical Volume
  - Volume Group
  - Logical Volume

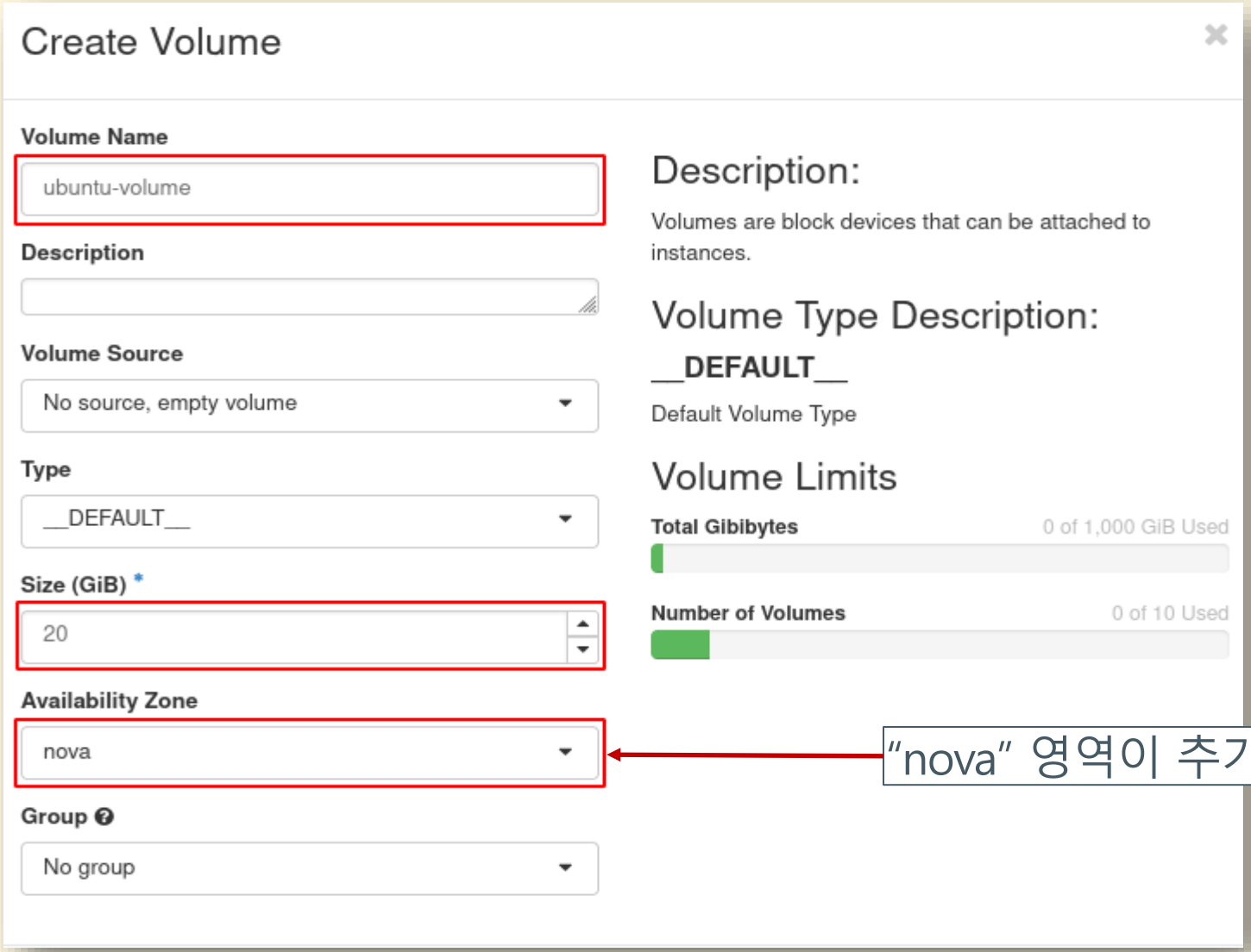
```
$ sudo pvscan
  PV /dev/loop99   VG cinder-volumes   lvm2 [<400.00 GiB / <19.81 GiB free]
  Total: 1 [<400.00 GiB] / in use: 1 [<400.00 GiB] / in no VG: 0 [0   ]

$ sudo vgscan
  Found volume group "cinder-volumes" using metadata type lvm2

$ sudo lvscan
  ACTIVE            '/dev/cinder-volumes/cinder-volumes-pool' [380.00 GiB] inherit
```

Create Volume

- `Create Volume` 을 선택해 볼륨 생성

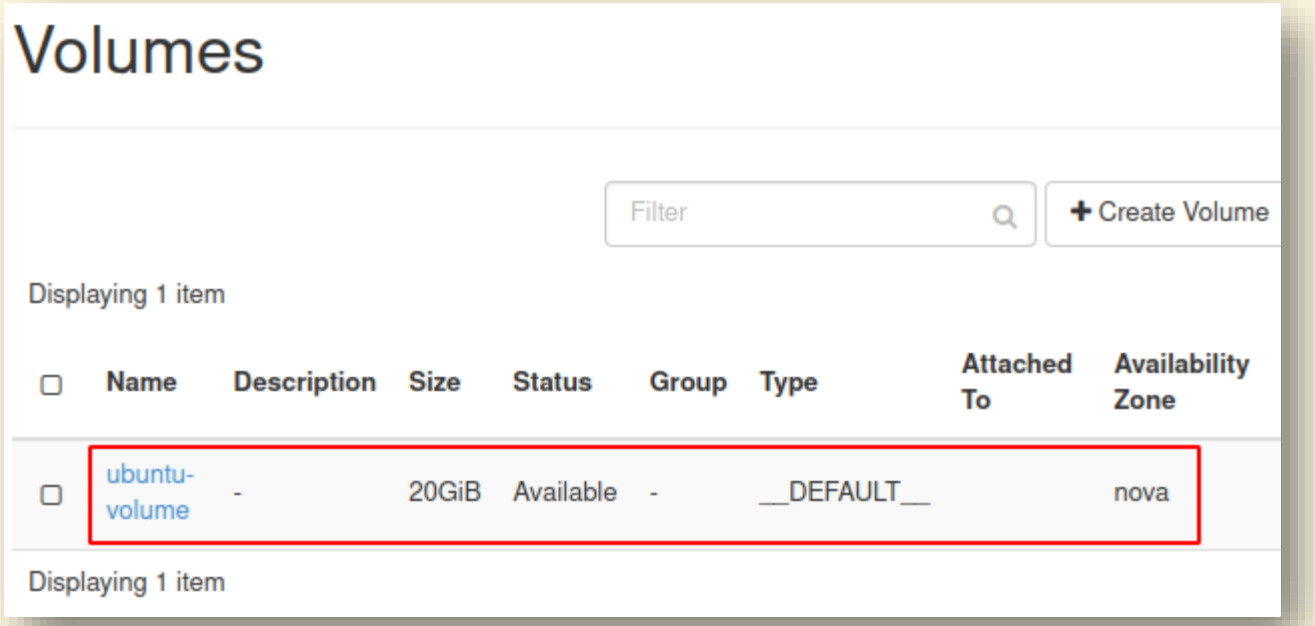


Create Volume (continued)



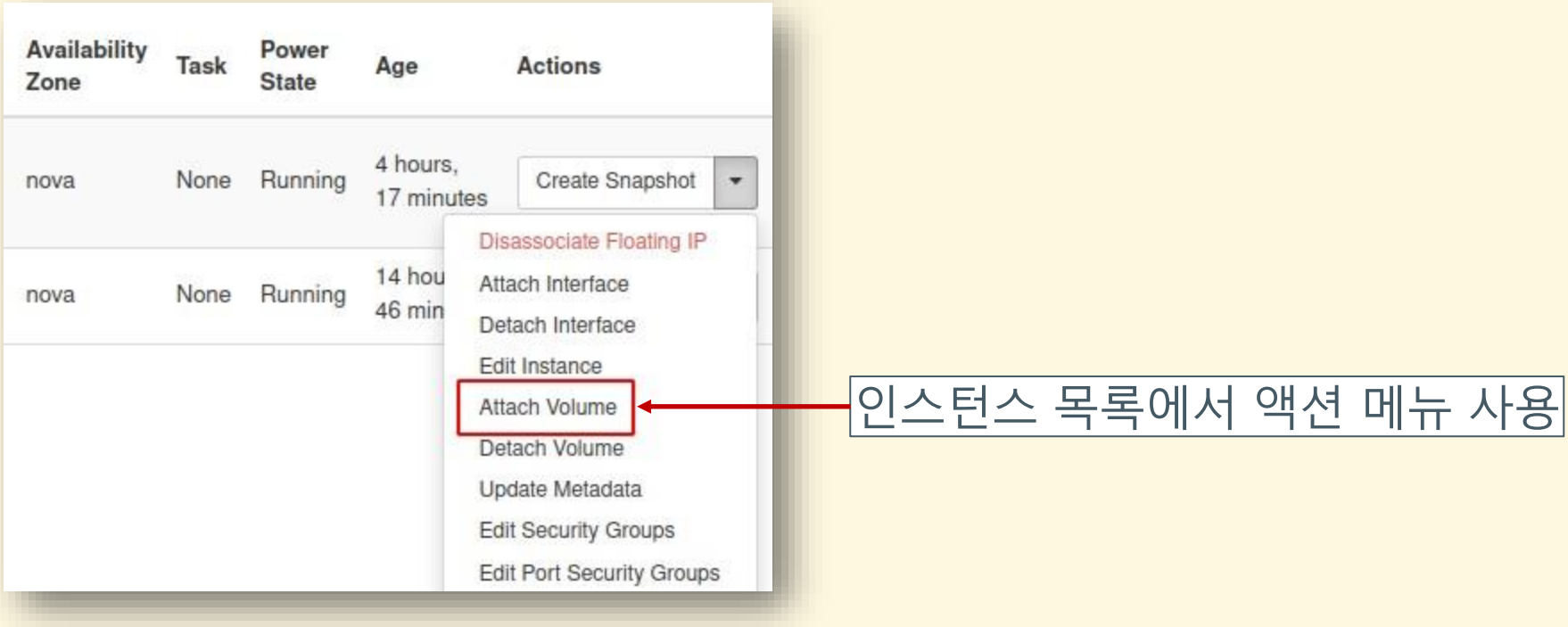
Create Volume (continued)

- `Availability Zone` 에 `nova` 영역 표시
- `ubuntu-volume` 으로 `20GiB` 크기로 볼륨 생성



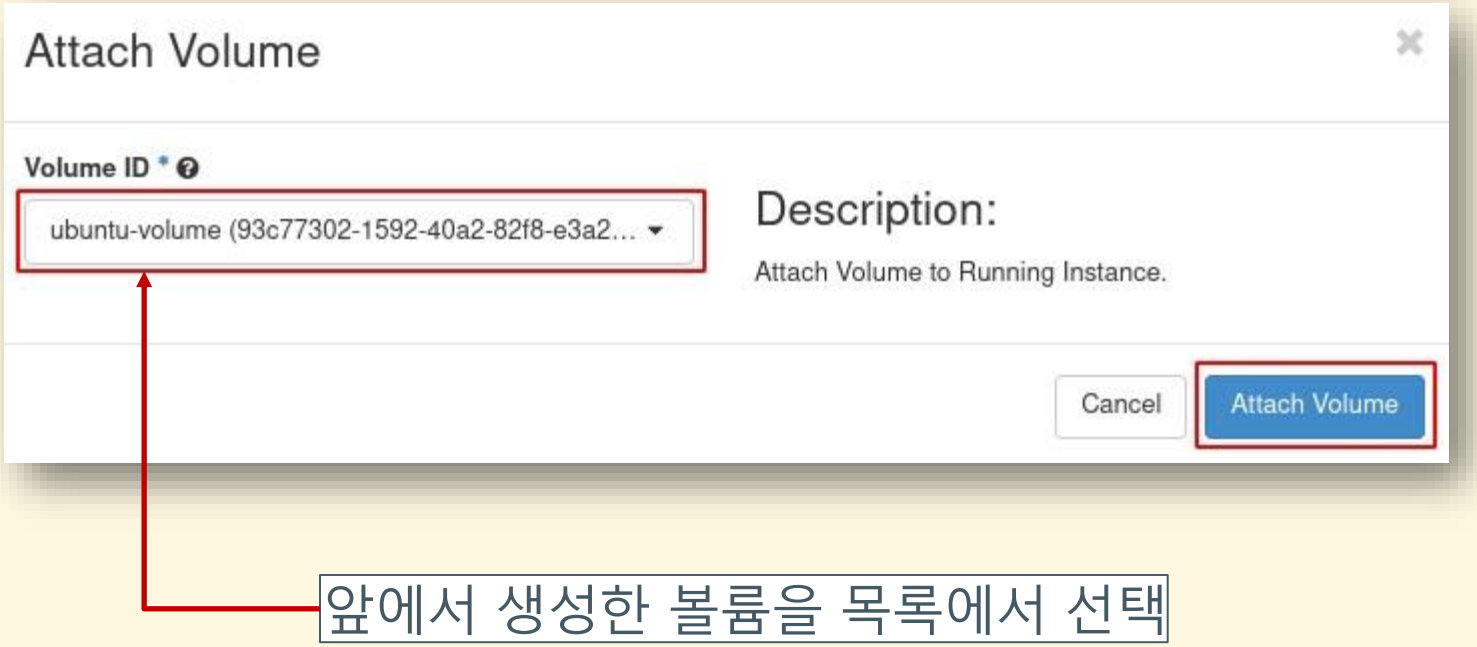
Attach Volume

- 볼륨을 추가할 인스턴스에서 `Attach Volume` 선택



Attach Volume (continued)

- 미리 생성해둔 볼륨 선택



Check Attached Volume

- `ubuntu` VM 안에서 추가된 볼륨 확인

```
ubuntu@ubuntu:~$ ls -al /dev/vd*
brw-rw---- 1 root disk 252,  0 Oct 31 08:34 /dev/vda
brw-rw---- 1 root disk 252,  1 Oct 31 08:34 /dev/vda1
brw-rw---- 1 root disk 252, 14 Oct 31 08:34 /dev/vda14
brw-rw---- 1 root disk 252, 15 Oct 31 08:34 /dev/vda15
brw-rw---- 1 root disk 252, 16 Oct 31 10:31 /dev/vdb

ubuntu@ubuntu:/mnt$ sudo fdisk -l /dev/vdb
Disk /dev/vdb: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

- `/dev/vdb` 볼륨이 추가된 것 확인

Mount Attached Volume

- `ext4` 파일 시스템으로 포맷 후에 시스템에 마운트

```
ubuntu@ubuntu:~$ sudo mkfs.ext4 /dev/vdb
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 5242880 4k blocks and 1310720 inodes
Filesystem UUID: 2193249c-b2bb-4997-9c95-6b2239e76c46
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Mount Attached Volume (continued)

```
ubuntu@ubuntu:~$ sudo mkdir /mnt/new-volume
ubuntu@ubuntu:~$ sudo mount /dev/vdb /mnt/new-volume
ubuntu@ubuntu:~$ cd /mnt/new-volume/
ubuntu@ubuntu:/mnt/new-volume$ ls -l
total 16
drwx------ 2 root root 16384 Oct 31 10:53 lost+found

ubuntu@ubuntu:/mnt/new-volume$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G     0  2.0G   0% /dev
tmpfs           394M  968K  393M   1% /run
/dev/vda1        20G  1.5G   18G   8% /
...
/dev/loop1       68M   68M     0 100% /snap/lxd/21545
/dev/loop0       62M   62M     0 100% /snap/core20/1169
/dev/vda15      105M  5.2M  100M   5% /boot/efi
/dev/loop2       33M   33M     0 100% /snap/snapd/13640
tmpfs           394M     0  394M   0% /run/user/1000
/dev/vdb         20G   45M   19G   1% /mnt/new-volume
```

Summary

MicroStack Up & Running

- 장점
  - 명령어 몇 개로 쉽고 빠르게 OpenStack 환경 구축
  - MicroStack 을 띄우는 서버에 영향을 주지 않음
  - LVM 볼륨을 생성해 인스턴스에 붙일 디스크 공간 제공 가능
- 단점
  - 초기화 단계에서 내부 서비스 포트 변경 불가
  - 인스턴스에서 외부 네트워크 연결을 위해 `iptables` 필요
  - 아직은 `beta` 단계의 프로젝트 성숙도

MicroStack Up & Running

감사합니다

OpenInfra Community Days Korea 2021

임찬식
chanshik@gmail.com