Openstack | User guide

We'll show simple use of the CloudMIP platform within the **service** project.

We previously created and shared both a public and a private network.
These ones ought to be visible through the '**neutron net-list**' command

Granted roles and capabilities come from
`/etc/nova/policy.json`
`/etc/keystone/policy.json`
`/etc/<module>/policy.json`

Pre-requisite

Before going on, you must have an account → apply for an account to François ;)

[one-time operation] environment setup

The followings are **one-time only** operations:

```
osSetCredential.sh
    → will ask for your user account password
```

This will setup several files and environment variables that will get used by all of the Openstack's CLI.

[one-time operation] create SSH public key

```
ssh-keygen -t rsa
```

[one-time operation] add user's pubkey

```
nova keypair-add --pub-key ~/.ssh/id_rsa.pub mykey
```

```
nova keypair-list
```

Start instance

We'll now launch an instance of the previously installed FC24 image. Before that, there are some parameters required to launch an instance:

```
nova flavor-list
nova image-list
neutron net-list
nova secgroup-list
nova host-list
nova host-describe wn32.cloudmip.univ-tlse3.fr
nova hypervisor-list
nova hypervisor-show <ID>
```

Note: **only private network** is reachable from compute nodes so this is the one your instances ought to use.

```
nova boot --flavor m1.tiny --image cirros --nic net-id=<NETWORK_ID> \
--security-group default --key-name mykey name
```

```
nova boot --flavor m1.tiny --image cirros --nic
net-id=c1445469-4640-4c5a-ad86-9c0cb6650cca --security-group default
--key-name mykey private-instance
```

```
nova boot --flavor m1.small --image FC24 --nic net-id=<NETWORK_ID> \
--security-group default --key-name mykey name
```

```
nova boot --flavor m1.small --image FC24 --nic
net-id=c1445469-4640-4c5a-ad86-9c0cb6650cca --security-group default
--key-name mykey private-instance
```

VNC access to instance

Through a Web browser, you'll get access to the console of your VM :D

```
nova get-vnc-console <instance_ID|instance_name> novnc
```

```
nova get-vnc-console private-instance novnc
```

[root only] direct SSH access to an instance

This is a special case where you'll gain access to an instance without a public IP required

OpenStack command-line interface cheat sheet:
http://docs.openstack.org/user-guide/cli-cheat-sheet.html

```
neutron net-list
```

```
ip netns
ip netns exec NETNS_NAME ssh USER@SERVER
```

```
ip netns
sudo ip netns exec qdhcp-c1445469-4640-4c5a-ad86-9c0cb6650cca ssh -i ~/.ssh/id_rsa fedora@192.168.0.128
```

Note: qdhcp-xxxxx is part of neutron net-list private network

Securitygroup rules

We'll add some permissions (needed for the 'service' project)

```
nova secgroup-list
nova secgroup-list-rules default

If needed … add rules
nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
```

map public IP to instance

To enable instance to get reached from the internet, we grab a floating-ip
But before going-on with the floating public IP creation, is there any already created public IP available ?

```
nova floating-ip-list
```

Ok, if there are no Public IP available, let's create one

```
neutron floatingip-create public
Created a new floatingip:
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    |                                      |
| floating_ip_address | 195.220.53.4                         |
| floating_network_id | c254d472-6cfd-425a-9960-e9d38ea4c391 |
| id                  | b7015888-9dde-4273-a377-631fd4f235ac |
| port_id             |                                      |
| router_id           |                                      |
| status              | DOWN                                 |
| tenant_id           | 6ac3b0c5fd5641928a412ed2b0ad65e5     |
+---------------------+--------------------------------------+
```

```
nova floating-ip-associate private-instance 195.220.53.4
```

to allocate a specific public IP

```
neutron floatingip-create --floating-ip-address 195.220.53.14 public
```

Connect to your fedora VM

```
ssh fedora@195.220.53.4
[fedora@private-instance ~]$ sudo su -
[root@private-instance ~]#
```

... well done player one ;)

Start instance to specific host

This is an **admin** only feature:

```
francois@frontal[~] nova availability-zone-list
+-----------------------------------+----------------------------------------+
| Name                              | Status                                 |
+-----------------------------------+----------------------------------------+
| internal                          | available                              |
| |- frontal.cloudmip.univ-tlse3.fr |                                        |
| | |- nova-conductor               | enabled :-) 2016-09-08T11:16:12.000000 |
| | |- nova-consoleauth             | enabled :-) 2016-09-08T11:16:08.000000 |
| | |- nova-scheduler               | enabled :-) 2016-09-08T11:16:13.000000 |
| | |- nova-cert                    | enabled :-) 2016-09-08T11:16:14.000000 |
| nova                              | available                              |
| |- wn1.cloudmip.univ-tlse3.fr     |                                        |
| | |- nova-compute                 | enabled :-) 2016-09-08T11:16:13.000000 |
| |- wn10.cloudmip.univ-tlse3.fr    |                                        |
| | |- nova-compute                 | enabled :-) 2016-09-08T11:16:13.000000 |
.....
```

```
nova boot --flavor m1.small --image FC24 --nic
net-id=c1445469-4640-4c5a-ad86-9c0cb6650cca --security-group default
--key-name mykey --availability-zone nova:wn1.cloudmip.univ-tlse3.fr
private-instance-wn1
```

Note: hostname as described in 'availability-zone-list' (fqdn here)

migration

Users can't decide where a VM ought to be migrated ...

```
nova migrate --poll private-instance

once finished
nova resize-confirm private-instance
```

Live-migration

TODO!

create image from instance | nova snapshot

Creating a snapshot from a running instance

```
nova image-create --poll <instance_name> <snapshot_name>
```

```
nova image-create --poll private-instance FC24snap
```

… boot a new instance from this snapshot

```
nova boot --flavor same_flavour --image <snapshot_name> --nic
net-id=c1445469-4640-4c5a-ad86-9c0cb6650cca --security-group default
--key-name mykey <snapshot instance name>
```

```
nova boot --flavor m1.small --image FC24snap --nic
net-id=c1445469-4640-4c5a-ad86-9c0cb6650cca --security-group default
--key-name mykey myFC24snap
```

Persistent storage | cinder volumes

We'll add a persistent storage (cinder, i.e block) to an instance

```
cinder create --name FT_lv 4
```

Note: size in GB

```
nova list
cinder list or nova volume-list
nova volume-attach INSTANCE_NAME VOLUME_ID
```

```
nova volume-attach public-instance 1372f518-f06d-4ff5-9c3d-b31325ff3e51
+----------+--------------------------------------+
| Property | Value                                |
+----------+--------------------------------------+
| device   | /dev/vdb                             |
| id       | 1372f518-f06d-4ff5-9c3d-b31325ff3e51 |
| serverId | 49f31828-7ea8-444c-b518-6d8a957944ee |
| volumeId | 1372f518-f06d-4ff5-9c3d-b31325ff3e51 |
+----------+--------------------------------------+
```

log to VM [root only] direct SSH access to an instance and prepare newly attached storage

```
francois@frontal[~] sudo ip netns exec
qdhcp-c1445469-4640-4c5a-ad86-9c0cb6650cca ssh -i ~/.ssh/id_rsa
fedora@192.168.0.153
Warning: Permanently added '192.168.0.153' (ECDSA) to the list of known hosts.
Last login: Thu Sep  8 11:23:12 2016 from 192.168.0.1
[fedora@private-instance-wn1 ~]$ sudo su
[root@private-instance-wn1 fedora]# fdisk -l
Disk /dev/vda: 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
Disklabel type: dos
Disk identifier: 0x06d4f68c

Device     Boot Start      End  Sectors Size Id Type
/dev/vda1  *     2048 41943039 41940992  20G 83 Linux

Disk /dev/vdb: 1 TiB, 1099511627776 bytes, 2147483648 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
```

- *create a partition /dev/vdb1*
- *dnf -y install **xfsprogs***
- *mkfs.xfs /dev/vdb1*
- *mount newly created partition … et voilà :)*

Contextualisation | cloud-init

TODO!

Docker @ instance

Running docker in an instance requires some setup depending on the base image you are using. Additionally, due to the vxlan infrastructure under the hood, you'll have to change the MTU size of your containers (through a docker service configuration).

docker@Fedora

Once you are logged within your Fedora instance

```
dnf -y install docker
```

- `/etc/sysconfig/docker`

```
.............
OPTIONS='--log-driver=journald --signature-verification=false'
.............
```

- /etc/sysconfig/docker-network

```
# /etc/sysconfig/docker-network
DOCKER_NETWORK_OPTIONS="--mtu=1450 --bip=172.17.0.1/24
--fixed-cidr=172.17.0.0/24"
```

… then enable and start service

```
systemctl enable docker
systemctl start docker
```

docker@CentOS

Once you are logged within your CentOS instance

```
yum -y install docker
```

- /etc/sysconfig/docker

```
{
  "mtu": 1450,
  "bip": "172.17.0.1/24",
  "fixed-cidr": "172.17.0.0/24"
}
```

… then enable and start service

```
systemctl enable docker
systemctl start docker
```

Docker tests

```
docker pull fedora
docker run -it fedora /bin/bash
```

```
<@container> dnf -y update
```