OpenStack Assignment

Surbhi Sonkiya, Zsolt Dargo

December 11, 2018

1 Describe what are the floating IP addresses used in Openstack (if they use a NAT, what type,etc.). Are they visible from inside the virtual machines? Upload together with the results memory the two files with the traffic captures.

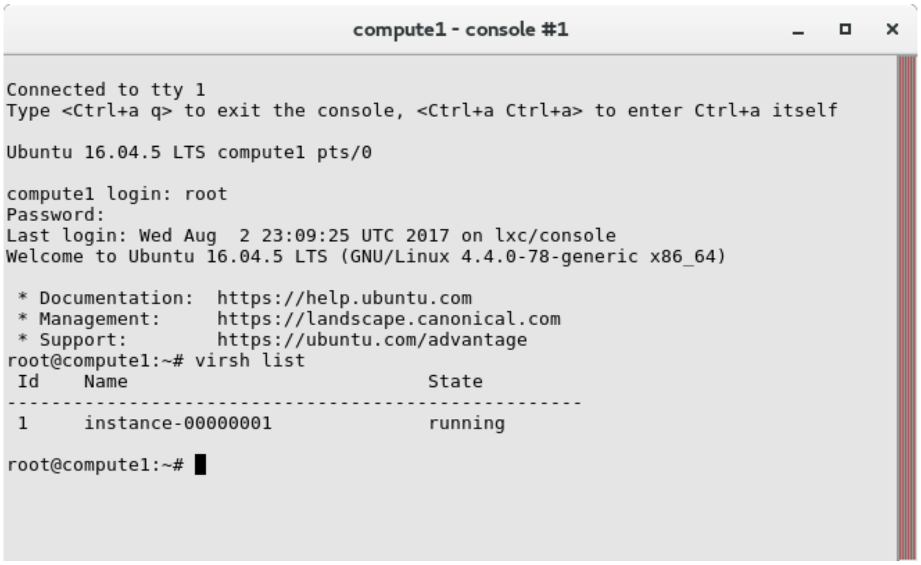
OpenStack has two IP types in use: private and floating IP addresses. While private addresses are used for internal communication, floating IP adresses provide the means of accessing OpenStack virtual machines from external networks.

The term “floating” comes from the fact that there is an IP address pool from which OpenStack admins can attach addresses to any of their instances and each instance can have a limited number of floating IP addresses assigned. OpenStack uses one-to-one NAT to implement floating IP addresses and they are also visible from the virtual machines.

The two capture files can be found attached with this document.

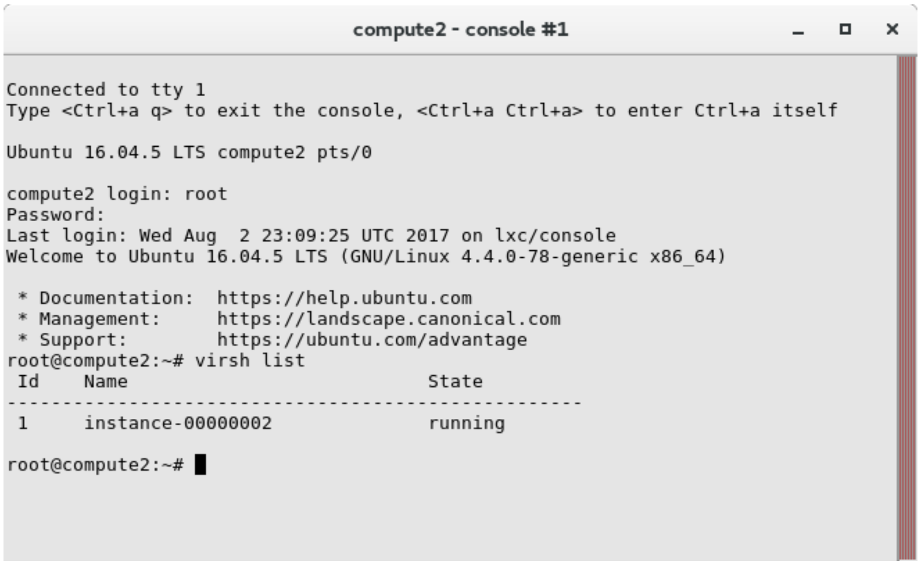
2 Describe in memory what is the purpose and utility of the flavors used in OpenStack.

Virtual machine instances have parameters like number of CPU cores, RAM size and disk capacity in OpenStack, also known as Flavors. They are used to facilitate easier creation of new instances by providing predefined characteristics that are later on applied on the instance in question. Flavors can be created, modified and deleted only through the “admin” account.



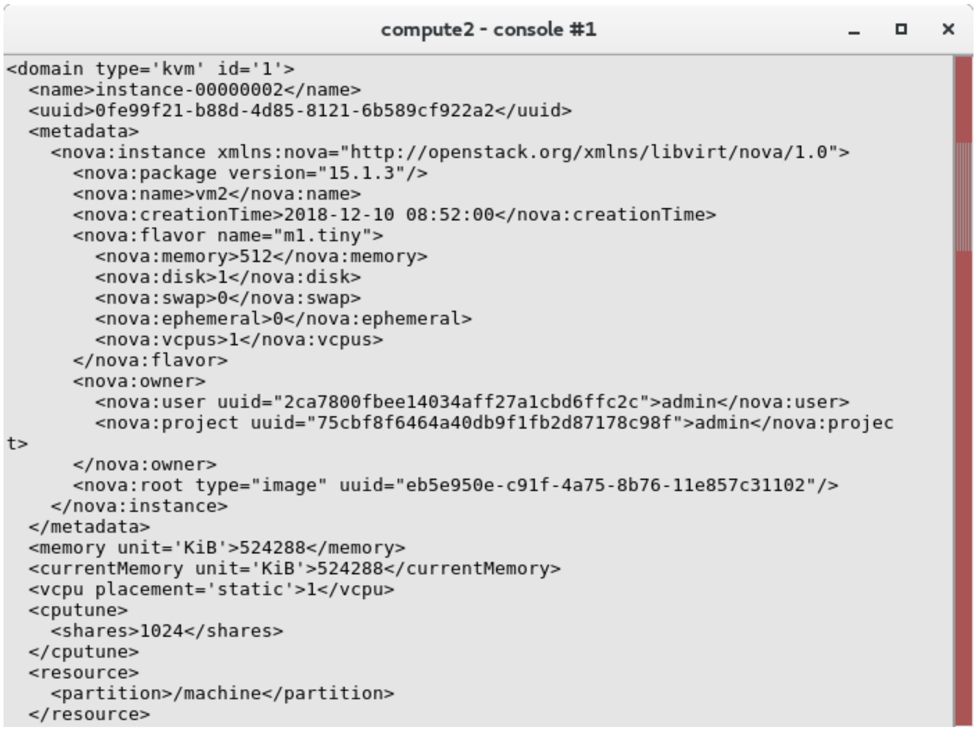
```
Connected to tty 1
Type <Ctrl+a q> to exit the console, <Ctrl+a Ctrl+a> to enter Ctrl+a itself

Ubuntu 16.04.5 LTS compute1 pts/0

compute1 login: root
Password:
Last login: Wed Aug  2 23:09:25 UTC 2017 on lxc/console
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
root@compute1:~# virsh list
 Id    Name                           State
----------------------------------------------------
 1     instance-00000001              running

root@compute1:~#
```

Figure 1: The list of instances on compute 1 node



```
Connected to tty 1
Type <Ctrl+a q> to exit the console, <Ctrl+a Ctrl+a> to enter Ctrl+a itself

Ubuntu 16.04.5 LTS compute2 pts/0

compute2 login: root
Password:
Last login: Wed Aug  2 23:09:25 UTC 2017 on lxc/console
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
root@compute2:~# virsh list
 Id    Name                           State
----------------------------------------------------
 1     instance-00000002              running

root@compute2:~#
```

Figure 2: The list of instances on compute 2 node



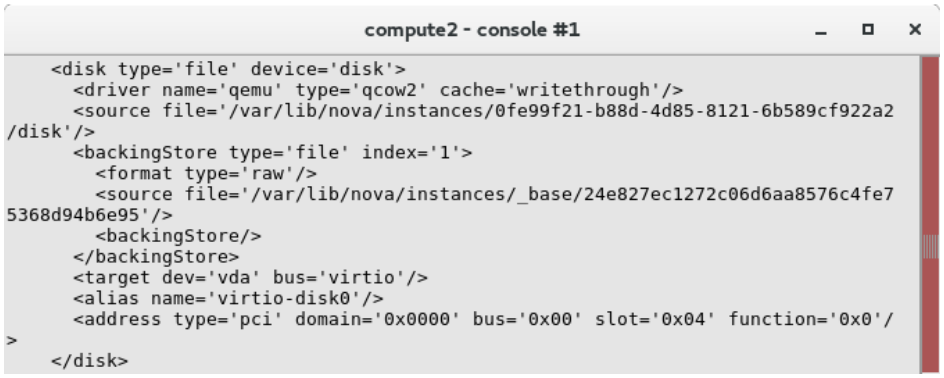
```
<domain type='kvm' id='1'>
  <name>instance-00000002</name>
  <uuid>0fe99f21-b88d-4d85-8121-6b589cf922a2</uuid>
  <metadata>
    <nova:instance xmlns:nova="http://openstack.org/xmlns/libvirt/nova/1.0">
      <nova:package version="15.1.3"/>
      <nova:name>vm2</nova:name>
      <nova:creationTime>2018-12-10 08:52:00</nova:creationTime>
      <nova:flavor name="m1.tiny">
        <nova:memory>512</nova:memory>
        <nova:disk>1</nova:disk>
        <nova:swap>0</nova:swap>
        <nova:ephemeral>0</nova:ephemeral>
        <nova:vcpus>1</nova:vcpus>
      </nova:flavor>
      <nova:owner>
        <nova:user uuid="2ca7800fbee14034aff27a1cbd6ffc2c">admin</nova:user>
        <nova:project uuid="75cbf8f6464a40db9f1fb2d87178c98f">admin</nova:projec
t>
      </nova:owner>
      <nova:root type="image" uuid="eb5e950e-c91f-4a75-8b76-11e857c31102"/>
    </nova:instance>
  </metadata>
  <memory unit='KiB'>524288</memory>
  <currentMemory unit='KiB'>524288</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <cputune>
    <shares>1024</shares>
  </cputune>
  <resource>
    <partition>/machine</partition>
  </resource>
```

Figure 3: Metadata of the virtual machine

3 Describe in the memory the main information of interest you find in the XML file. At least, mention the basic parameters such memory, virtual machine image and network interfaces. If from inside the compute node you would have to capture the traffic entering or leaving the virtual machine, what would be the command to execute?

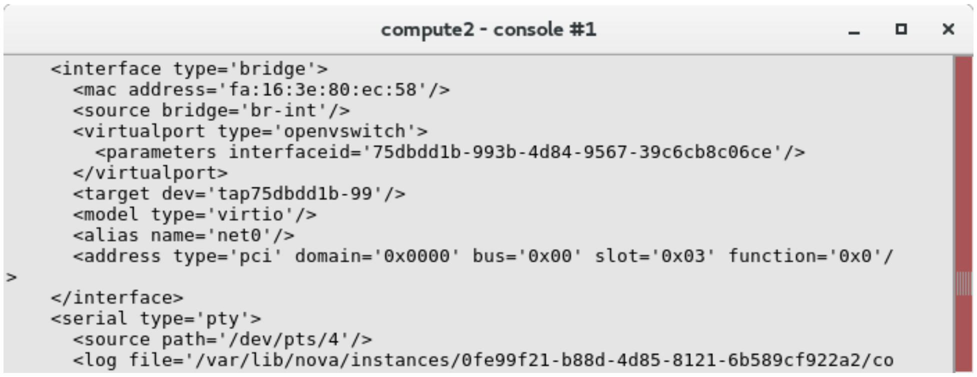
As we can see vm1 is running on the compute 1, while vm2 on the compute 2 server. The details below have been analyzed for vm2:

- **Memory:** 512 MB, which are in accordance with our expectations having chosen the “m1.tiny” flavor. The same can be seen in figure 3.
- **Virtual machine image:** The most relevant details (e.g driver name and type, alias) of the vm image can be seen in figure 4.
- **Network interfaces:** The network interface is described in the figure 5 as part of the output.



```
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' cache='writethrough'/>
      <source file='/var/lib/nova/instances/0fe99f21-b88d-4d85-8121-6b589cf922a2
/disk'/>
      <backingStore type='file' index='1'>
        <format type='raw'/>
        <source file='/var/lib/nova/instances/_base/24e827ec1272c06d6aa8576c4fe7
5368d94b6e95'/>
        <backingStore/>
      </backingStore>
      <target dev='vda' bus='virtio'/>
      <alias name='virtio-disk0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/
>
    </disk>
```

Figure 4: Virtual machine image data



```
    <interface type='bridge'>
      <mac address='fa:16:3e:80:ec:58'/>
      <source bridge='br-int'/>
      <virtualport type='openvswitch'>
        <parameters interfaceid='75dbdd1b-993b-4d84-9567-39c6cb8c06ce'/>
      </virtualport>
      <target dev='tap75dbdd1b-99'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/
>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/4'/>
      <log file='/var/lib/nova/instances/0fe99f21-b88d-4d85-8121-6b589cf922a2/co
```

Figure 5: Network interface data

The command for capturing network traffic is the ***tcpdump*** command in the corresponding compute console.

4 Describe in the results memory the utility of key pairs in the context of OpenStack. In what file inside the virtual machine is the public key that allows password-less access stored?

Key pairs are generated to protect the instances and provide the means of password-less access from remote networks. Public/private key pairs work by saving the private key on the local machine and public key on the server. Once the server verifies that the two keys match, a secure connection between the two machines is established.

The public key is stored in the root/.ssh/authorized_keys file. When using password-less authentication, we needed to pass the private key that we earlier downloaded. Therefore, the command that can be used to download has the following format:

- ***slogin cirros@10.0.10.X -i vm2.pem***

vm2.pem file is the private key of virtual machine. The above command will only work if we change the permissions of the PEM file using the following command:

- ***chmod 600 vm2.pem***

5 Describe in the memory if you have been able to access root account and, if it was not possible, describe how to allow the access. Hint: compare the SSH configuration files of cirrus account and the root account. What should be done to access the virtual machine using another user (vnx user, for example) account without specifying the password?

The authorized_keys file was present inside the **root/.ssh/** folder. However, it was missing from the **cirrus/.ssh/** folder. Therefore, it was not possible to access root account directly. Below command was used to copy the authorized_keys.

- ***slogin cirros@10.0.10.X -i vm2.pem***

As mentioned already, vm2.pem file is the private key of virtual machine. The above command will only work if we change the permissions of the PEM file using the following command:

- ***chmod 600 vm2.pem***

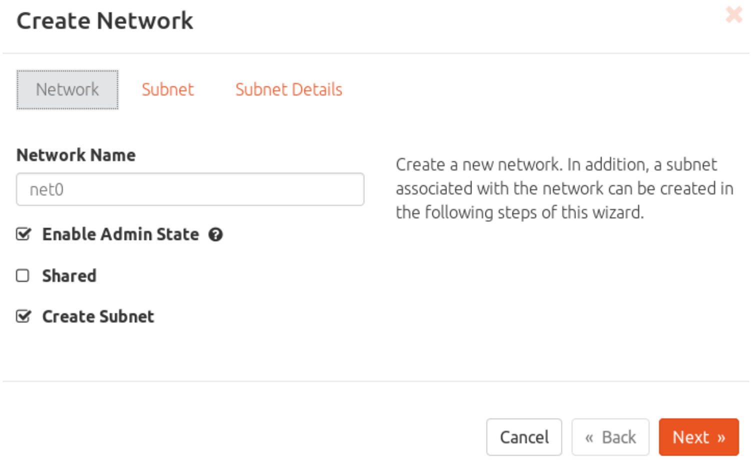
Alternatively, We can also use the following command to copy the private key of virtual machine.

- ***ssh-copy-id -i vm2.pem***

Now, there would an authorized_keys file also inside the **cirrus/.ssh/** folder. It will now allow password-less login access to virtual machine using another user.

6 Describe briefly in the results memory the procedure used to create the scenario.

1. Destroy the previously create virtual machines.
   - sudo vnx -f openstack_lab.xml –destroy
2. Create new virtual machines.
   - sudo vnx -f openstack_lab.xml –create
3. Start all the newly created virtual machines.
   - sudo vnx -f openstack_lab.xml -x start-all



Create Network

Network Name

net0

☑ Enable Admin State

☐ Shared

☑ Create Subnet

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Figure 6: Network tab

4. The image to be loaded for creating virtual machines in OpenStack scenario.

    - sudo vnx -f openstack_lab.xml -x load-img

5. After login, navigate to Project → Network → Networks. Click on *Create Network* button. Provide the details as shown in the figures 6, 7 and 8.

6. Login as admin on the below URL.

    - http://10.0.10.11/horizon

7. Navigate to Project → Compute → Key Pairs. Click on *Create Key Pair* button.

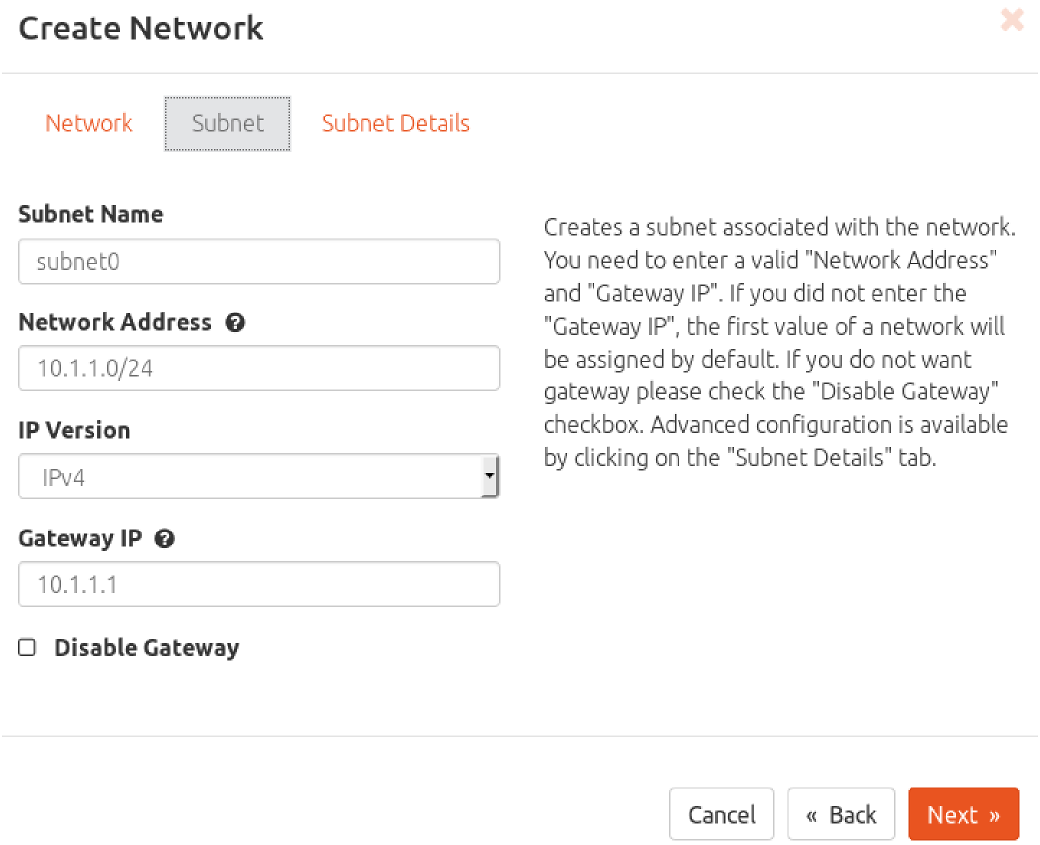
    - Create two key pairs named vm1 and vm2 (virtual machine vm1 is already created and vm2 would be created in further steps where this key pair would be used). Download the private keys and save it on the local machine.

8. Navigate to Project → Compute → Instances. Click on *Launch Instance* button.Provide the details as shown in the figures 9, 10, 11, 12 and 13.

9. Create a new instance named vm2. The steps to create vm2 is same as vm1 (detailed as part of point 8), but add the key pair which is named as vm2 instead of vm1.

10. Navigate to Project → Network → Networks. Click on *Create Network* button. Provide the details as shown in the figures 14, 15, and 16. Once the network is created, it has to be made *External* by navigating to Admin → System → Networks and clicking on *Edit Network*.

11. Navigate to Project → Network → Routers. Click on *Create Router* button. Provide the details as shown in the figure 17.



Create Network

Subnet Name
subnet0

Network Address
10.1.1.0/24

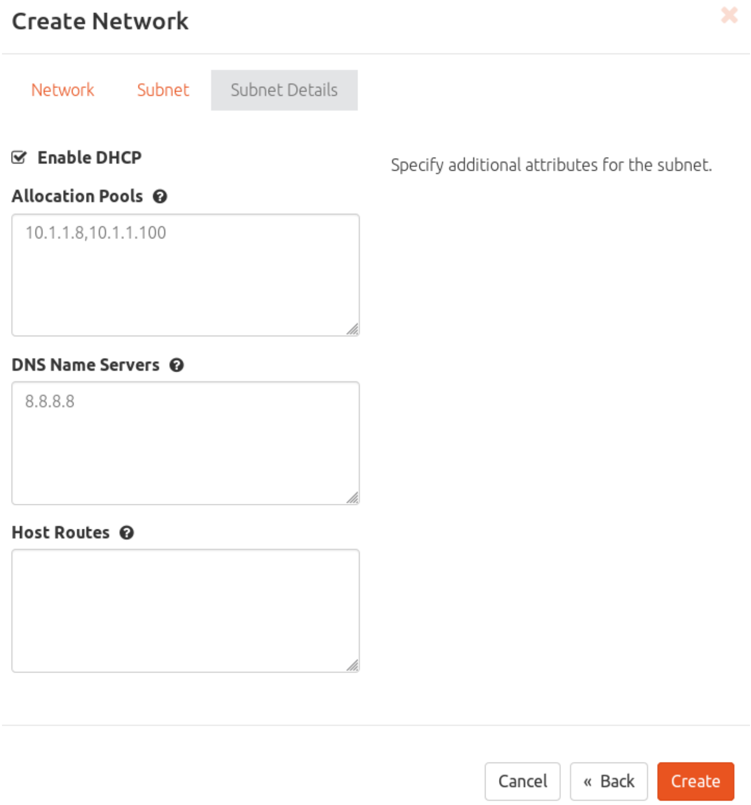
IP Version
IPv4

Gateway IP
10.1.1.1

□ Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Figure 7: Subnet tab



Specify additional attributes for the subnet.

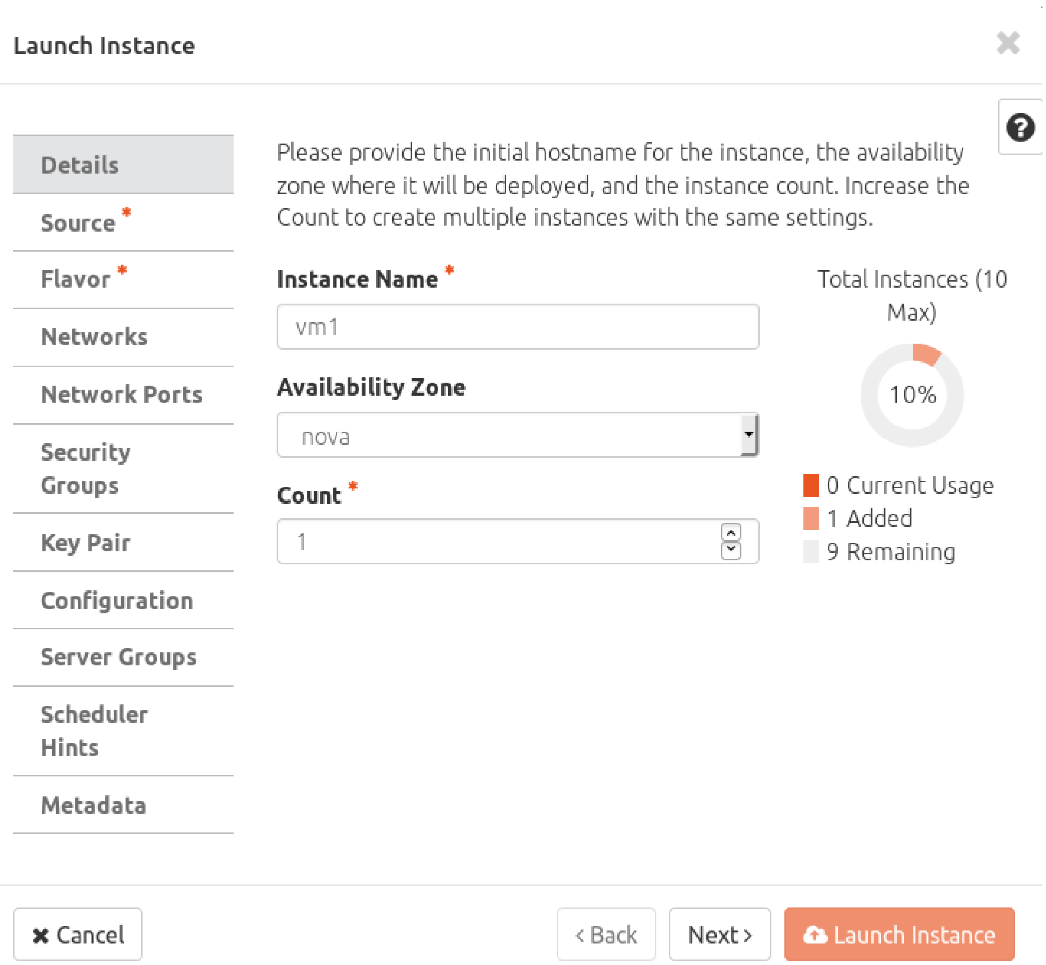
☑ Enable DHCP

Allocation Pools: 10.1.1.8,10.1.1.100

DNS Name Servers: 8.8.8.8

Host Routes:

Figure 8: Subnet details tab



Launch Instance

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name * vm1

Availability Zone nova

Count * 1

Total Instances (10 Max)

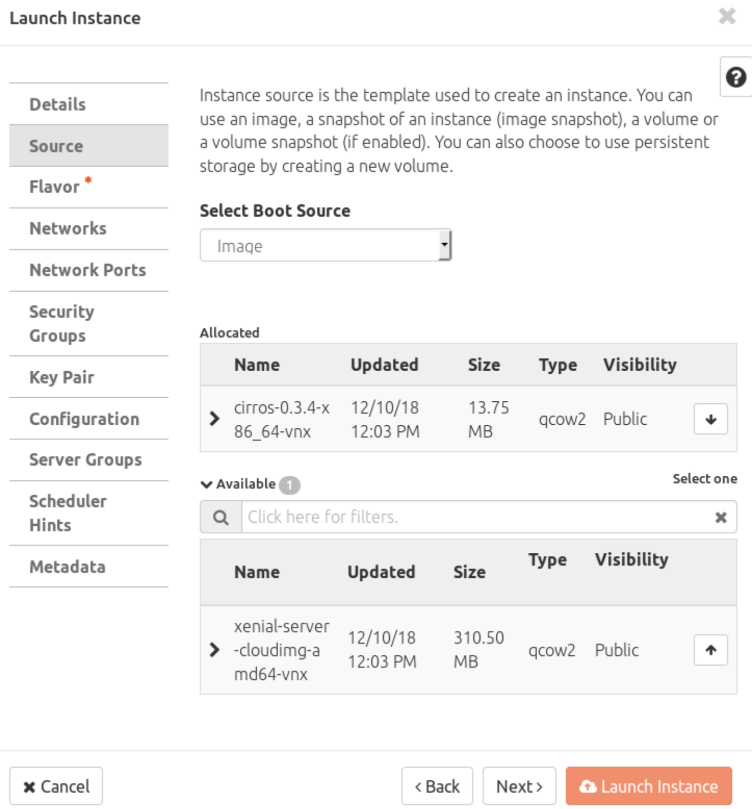
10%

🟥 0 Current Usage

🟧 1 Added

⬜ 9 Remaining

Figure 9: Details tab



Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Image

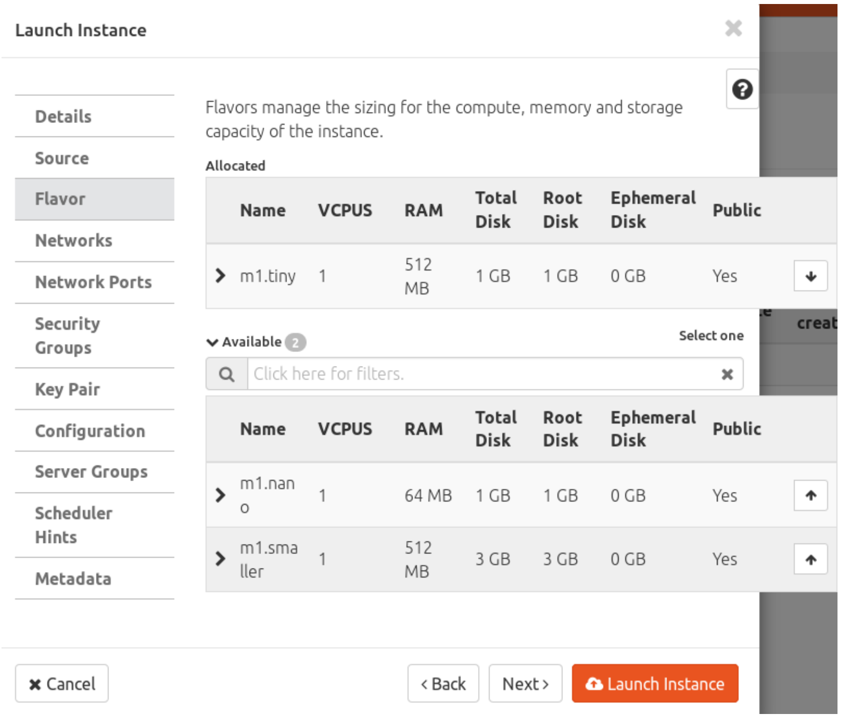
Allocated

Name	Updated	Size	Type	Visibility
cirros-0.3.4-x86_64-vnx	12/10/18 12:03 PM	13.75 MB	qcow2	Public

Available 1

Name	Updated	Size	Type	Visibility
xenial-server-cloudimg-amd64-vnx	12/10/18 12:03 PM	310.50 MB	qcow2	Public

Figure 10: Source tab



Flavors manage the sizing for the compute, memory and storage capacity of the instance.

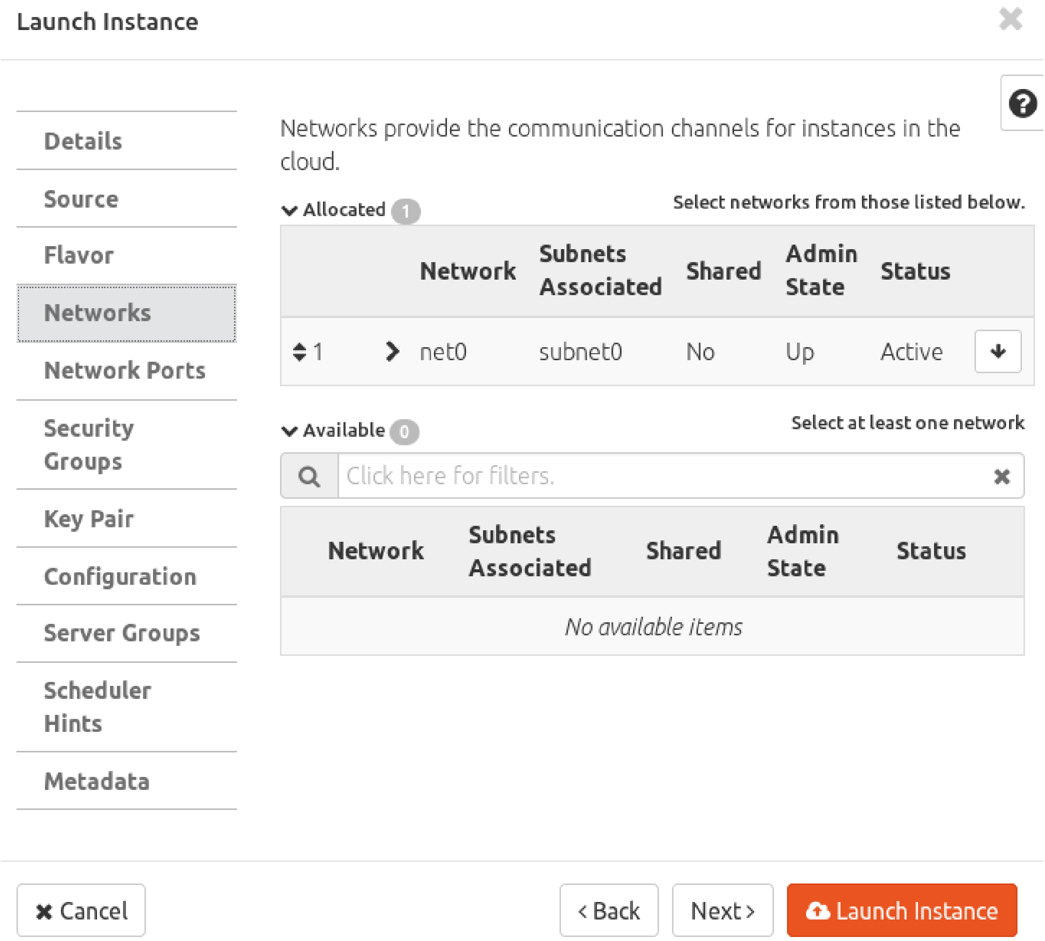
Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes

Available

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.nano	1	64 MB	1 GB	1 GB	0 GB	Yes
m1.smaller	1	512 MB	3 GB	3 GB	0 GB	Yes

Figure 11: Flavor tab



Networks provide the communication channels for instances in the cloud.

Allocated 1

Select networks from those listed below.

		Network	Subnets Associated	Shared	Admin State	Status	
⬍ 1	›	net0	subnet0	No	Up	Active	↓

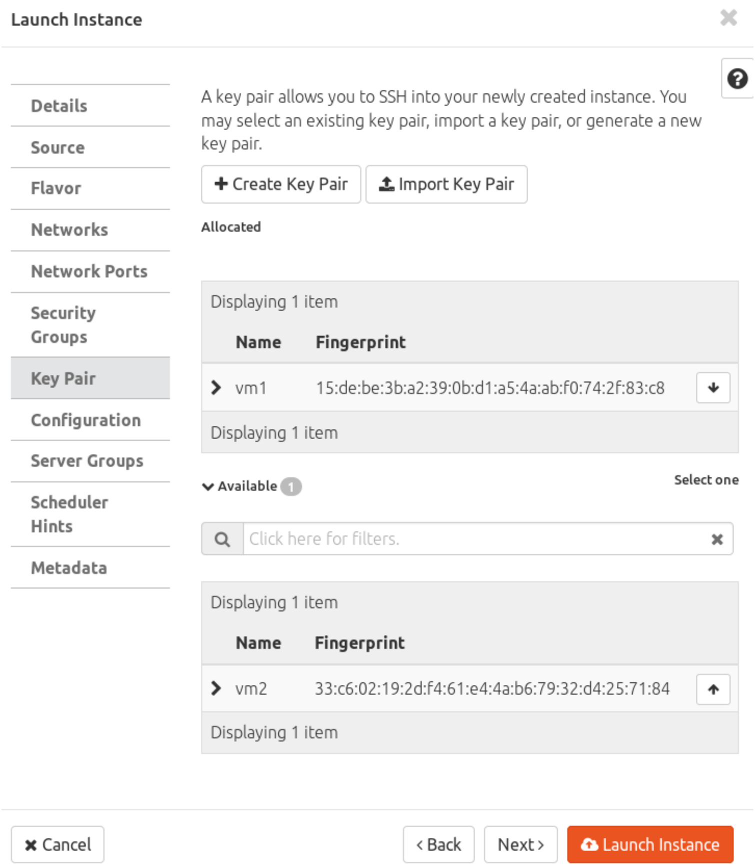
Available 0

Select at least one network

Click here for filters.

Network	Subnets Associated	Shared	Admin State	Status
No available items				

Figure 12: Networks tab



A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.

Allocated

Displaying 1 item

Name	Fingerprint
vm1	15:de:be:3b:a2:39:0b:d1:a5:4a:ab:f0:74:2f:83:c8

Displaying 1 item

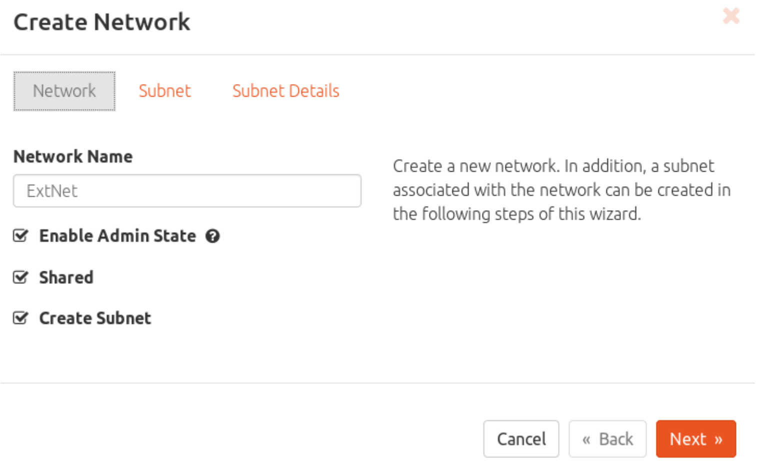
Available

Displaying 1 item

Name	Fingerprint
vm2	33:c6:02:19:2d:f4:61:e4:4a:b6:79:32:d4:25:71:84

Displaying 1 item

Figure 13: Key Pair tab



Network Name: ExtNet

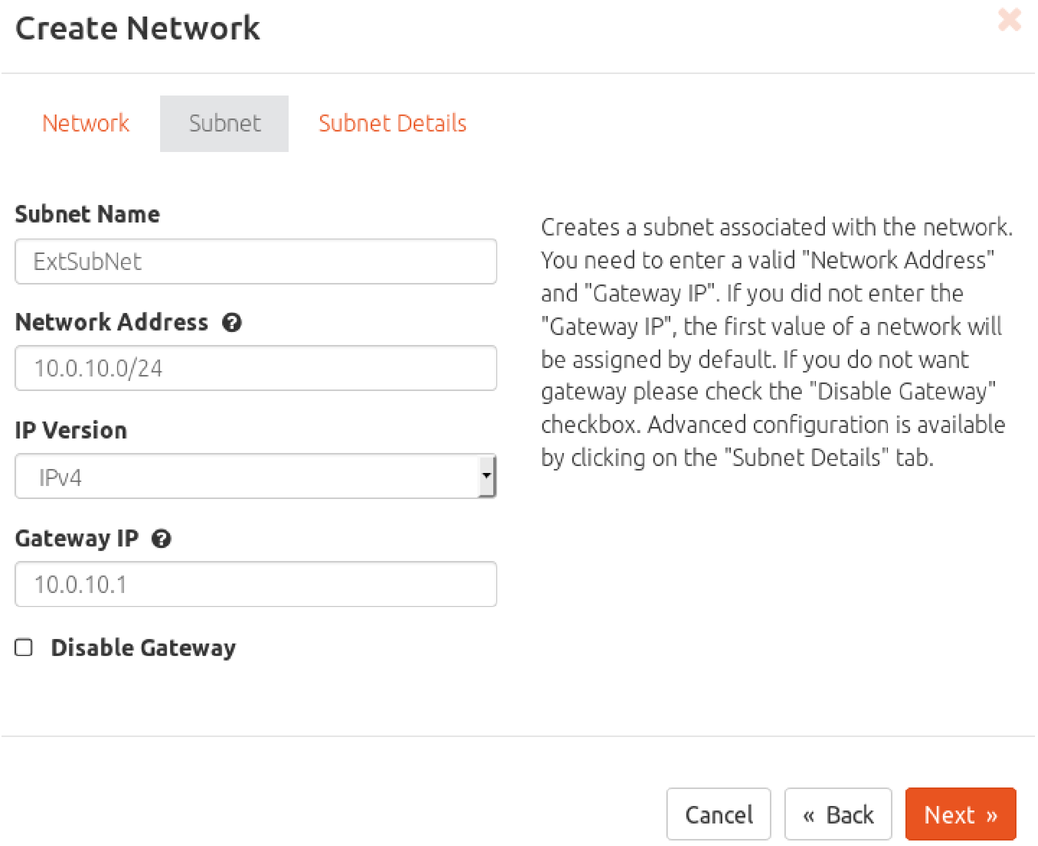
☑ Enable Admin State

☑ Shared

☑ Create Subnet

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Figure 14: Network tab



Subnet Name: ExtSubNet

Network Address: 10.0.10.0/24

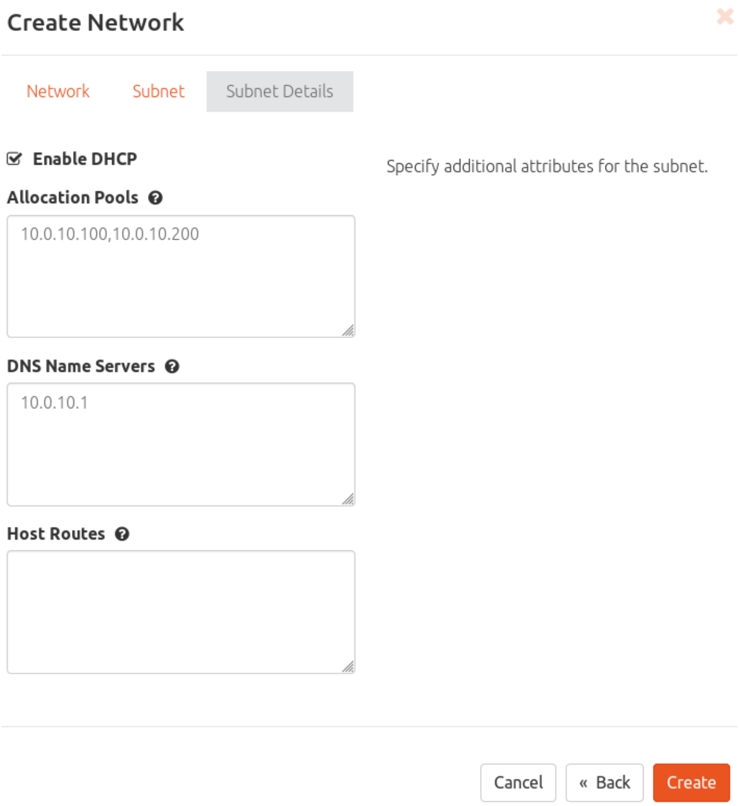
IP Version: IPv4

Gateway IP: 10.0.10.1

□ Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Figure 15: Subnet tab



☑ Enable DHCP

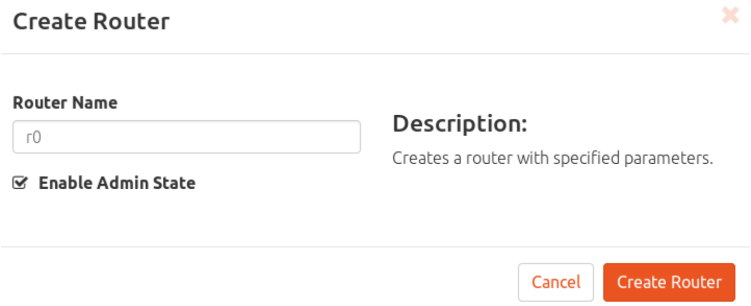
Allocation Pools: 10.0.10.100,10.0.10.200

DNS Name Servers: 10.0.10.1

Host Routes:

Specify additional attributes for the subnet.

Figure 16: Subnet details tab



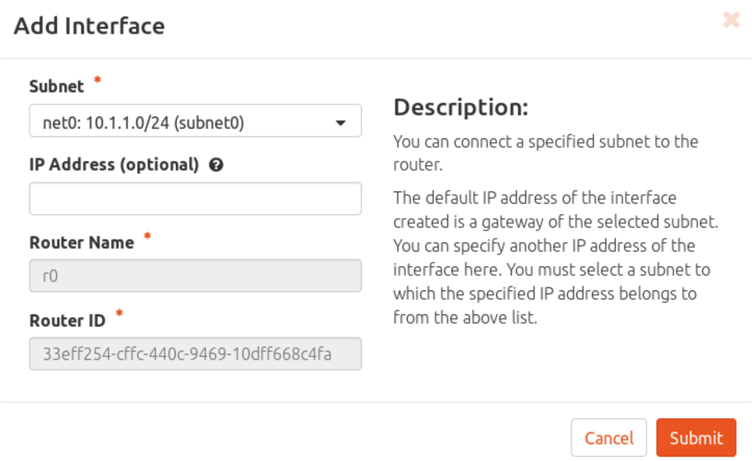
Router Name: r0

☑ Enable Admin State

Description:

Creates a router with specified parameters.

Figure 17: Create router window


Add Interface

Subnet *

net0: 10.1.1.0/24 (subnet0)

IP Address (optional)

Router Name *

r0

Router ID *

33eff254-cffc-440c-9469-10dff668c4fa

Description:

You can connect a specified subnet to the router.

The default IP address of the interface created is a gateway of the selected subnet. You can specify another IP address of the interface here. You must select a subnet to which the specified IP address belongs to from the above list.

Figure 18: Create router window

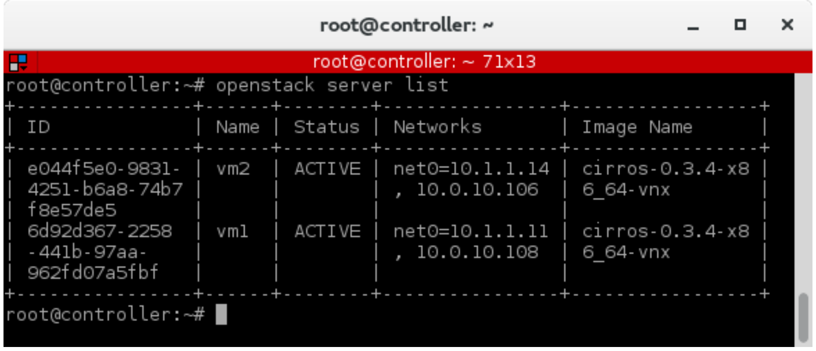
12. Navigate to Project → Network → Routers. Click on *Set Gateway* and choose the "ExtNet" network.

13. Navigate to Project → Network → Network Topology. Click on *Router* and click on the "Add Interface" button. Provide the details as shown in figure 18.

14. Navigate to Admin → System → Floating IPs. Click on *Allocate IP To Project*. Select the corresponding project and click on *Allocate floating IP* button.

15. Navigate to Project → Network → Floating IPs. Click on *Associate* button to assign floating IPs for vm1 and vm 2.

16. Navigate to Project → Network → Security Groups. Click on *Manage Rules* button. Click on *Add Rule* button to add the security rules to allow ICMP, SSH and WWW access.



```
root@controller:~# openstack server list
+----------------+------+--------+----------------+-----------------+
| ID             | Name | Status | Networks       | Image Name      |
+----------------+------+--------+----------------+-----------------+
| e044f5e0-9831- | vm2  | ACTIVE | net0=10.1.1.14 | cirros-0.3.4-x8 |
| 4251-b6a8-74b7 |      |        | , 10.0.10.106  | 6_64-vnx        |
| f8e57de5       |      |        |                |                 |
| 6d92d367-2258  | vm1  | ACTIVE | net0=10.1.1.11 | cirros-0.3.4-x8 |
| -441b-97aa-    |      |        | , 10.0.10.108  | 6_64-vnx        |
| 962fd07a5fbf   |      |        |                |                 |
+----------------+------+--------+----------------+-----------------+
root@controller:~# 
```

Figure 19: Server list command

7 Describe in the memory the commands necessary to obtain the aforementioned information.

Information about networks and virtual machines through command line are obtained by executing following commands.

1. **slogin root@controller**

2. **source bin/admin-openrc.sh**

3. **openstack server list** as shown in figure 19.

4. **openstack server show vm2** as shown in figure 20.

5. **openstack network list** as shown in figure 21.

6. **openstack network show net0** as shown in figure 22.

8 Include in memory commands needed to remove the demonstration scenario.

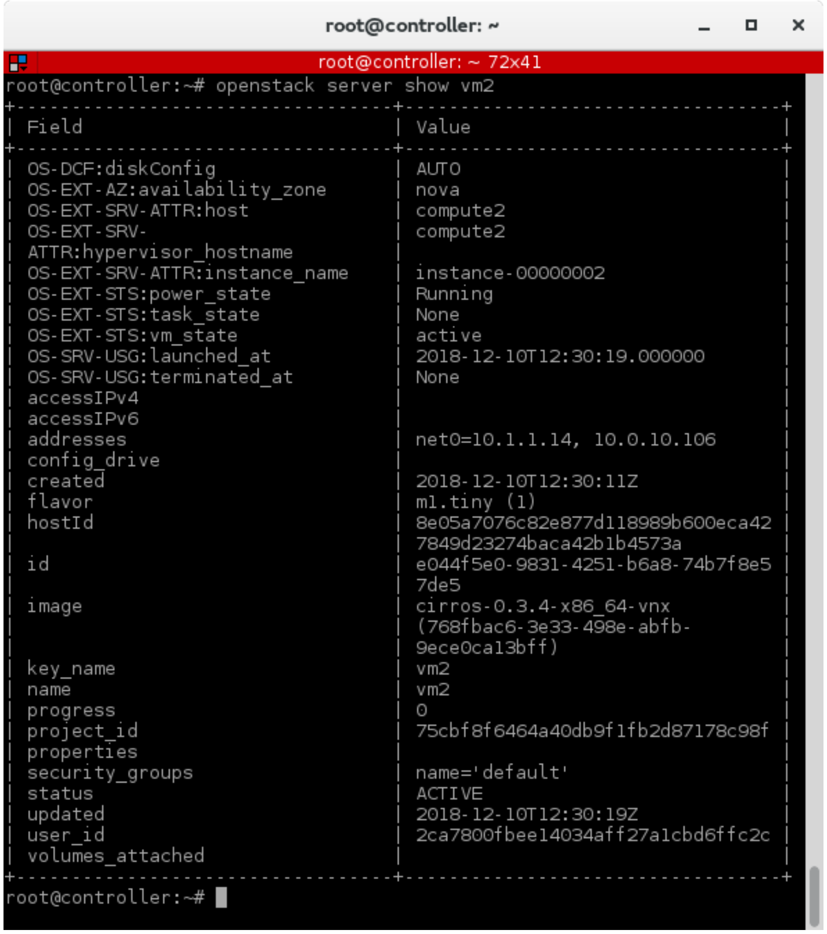
Use below commands to remove the demonstration scenario. The commands are also mentioned in the figure 23.

1. **slogin root@controller**

2. **source bin/admin-openrc.sh**

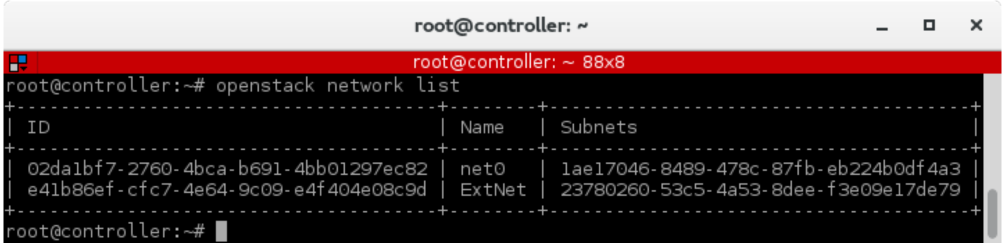
3. **openstack project list** to know the project-id.

4. **neutron purge *project-id***.



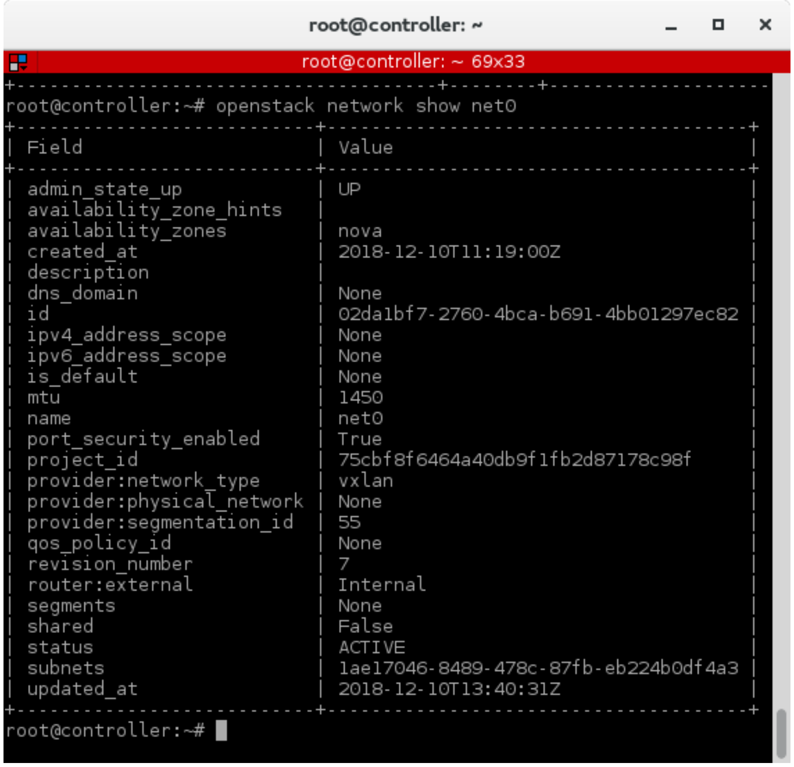
```
root@controller:~# openstack server show vm2
+-----------------------------------+----------------------------------+
| Field                             | Value                            |
+-----------------------------------+----------------------------------+
| OS-DCF:diskConfig                 | AUTO                             |
| OS-EXT-AZ:availability_zone       | nova                             |
| OS-EXT-SRV-ATTR:host              | compute2                         |
| OS-EXT-SRV-                       | compute2                         |
| ATTR:hypervisor_hostname          |                                  |
| OS-EXT-SRV-ATTR:instance_name     | instance-00000002                |
| OS-EXT-STS:power_state            | Running                          |
| OS-EXT-STS:task_state             | None                             |
| OS-EXT-STS:vm_state               | active                           |
| OS-SRV-USG:launched_at            | 2018-12-10T12:30:19.000000       |
| OS-SRV-USG:terminated_at          | None                             |
| accessIPv4                        |                                  |
| accessIPv6                        |                                  |
| addresses                         | net0=10.1.1.14, 10.0.10.106      |
| config_drive                      |                                  |
| created                           | 2018-12-10T12:30:11Z             |
| flavor                            | m1.tiny (1)                      |
| hostId                            | 8e05a7076c82e877d118989b600eca42 |
|                                   | 7849d23274baca42b1b4573a         |
| id                                | e044f5e0-9831-4251-b6a8-74b7f8e5 |
|                                   | 7de5                             |
| image                             | cirros-0.3.4-x86_64-vnx          |
|                                   | (768fbac6-3e33-498e-abfb-        |
|                                   | 9ece0ca13bff)                    |
| key_name                          | vm2                              |
| name                              | vm2                              |
| progress                          | 0                                |
| project_id                        | 75cbf8f6464a40db9f1fb2d87178c98f |
| properties                        |                                  |
| security_groups                   | name='default'                   |
| status                            | ACTIVE                           |
| updated                           | 2018-12-10T12:30:19Z             |
| user_id                           | 2ca7800fbee14034aff27a1cbd6ffc2c |
| volumes_attached                  |                                  |
+-----------------------------------+----------------------------------+
root@controller:~#
```

Figure 20: Server show command



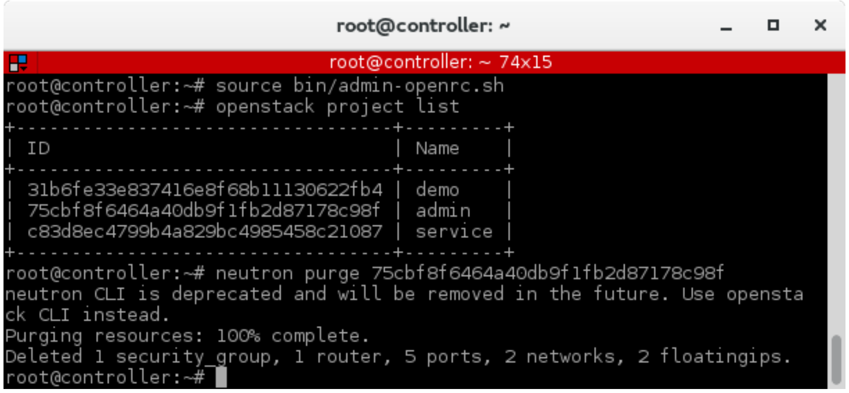
```
root@controller:~# openstack network list
+--------------------------------------+--------+--------------------------------------+
| ID                                   | Name   | Subnets                              |
+--------------------------------------+--------+--------------------------------------+
| 02da1bf7-2760-4bca-b691-4bb01297ec82 | net0   | 1ae17046-8489-478c-87fb-eb224b0df4a3 |
| e41b86ef-cfc7-4e64-9c09-e4f404e08c9d | ExtNet | 23780260-53c5-4a53-8dee-f3e09e17de79 |
+--------------------------------------+--------+--------------------------------------+
root@controller:~#
```

Figure 21: Network list command



```
+------------------------------------+--------+--------------------
root@controller:~# openstack network show net0
+---------------------------+-------------------------------------+
| Field                     | Value                               |
+---------------------------+-------------------------------------+
| admin_state_up            | UP                                  |
| availability_zone_hints   |                                     |
| availability_zones        | nova                                |
| created_at                | 2018-12-10T11:19:00Z                |
| description               |                                     |
| dns_domain                | None                                |
| id                        | 02da1bf7-2760-4bca-b691-4bb01297ec82 |
| ipv4_address_scope        | None                                |
| ipv6_address_scope        | None                                |
| is_default                | None                                |
| mtu                       | 1450                                |
| name                      | net0                                |
| port_security_enabled     | True                                |
| project_id                | 75cbf8f6464a40db9f1fb2d87178c98f    |
| provider:network_type     | vxlan                               |
| provider:physical_network | None                                |
| provider:segmentation_id  | 55                                  |
| qos_policy_id             | None                                |
| revision_number           | 7                                   |
| router:external           | Internal                            |
| segments                  | None                                |
| shared                    | False                               |
| status                    | ACTIVE                              |
| subnets                   | 1ae17046-8489-478c-87fb-eb224b0df4a3 |
| updated_at                | 2018-12-10T13:40:31Z                |
+---------------------------+-------------------------------------+
root@controller:~# 
```

Figure 22: Network show command



```
root@controller:~# source bin/admin-openrc.sh
root@controller:~# openstack project list
+----------------------------------+---------+
| ID                               | Name    |
+----------------------------------+---------+
| 31b6fe33e837416e8f68b11130622fb4 | demo    |
| 75cbf8f6464a40db9f1fb2d87178c98f | admin   |
| c83d8ec4799b4a829bc4985458c21087 | service |
+----------------------------------+---------+
root@controller:~# neutron purge 75cbf8f6464a40db9f1fb2d87178c98f
neutron CLI is deprecated and will be removed in the future. Use opensta
ck CLI instead.
Purging resources: 100% complete.
Deleted 1 security_group, 1 router, 5 ports, 2 networks, 2 floatingips.
root@controller:~# 
```

Figure 23: Purging scenario

9 References

1. https://docs.openstack.org/ocata/user-guide/cli-manage-ip-addresses.html

2. https://docs.openstack.org/horizon/latest/admin/manage-flavors.html

3. https://www.mirantis.com/blog/openstack-security-tip-create-a-keypair-for-accessing-vms/

4. https://www.ssh.com/ssh/copy-id

5. https://docs.openstack.org/python-openstackclient/pike/cli/command-list.html

6. https://docs.openstack.org/mitaka/networking-guide/ops-resource-purge.html