Lab #2 – Introduction to Private Infrastructure-as-a-Service

Introduction

Welcome to Lab #2. This lab assumes successful completion of Lab #1.

Background

In this lab we will explore the DevStack environment that is a developer version of the OpenStack Infrastructure as a Service (IaaS) cloud computing platform. DevStack is a fully functional cloud computing platform that provides a service analogous to the Amazon EC2/S3 Cloud.

OpenStack Basic Components (4 major ones)

1. Nova : VM management (including network configuration)
2. Glance : Storage for the VM images
3. Swift : Object Storage
4. Neutron: Networking

Additional components

1. Keystone : Identity Management (users and authentication)
2. Horizon : Web-based Management Console

Please read through the available online documentation before asking for help.

Note that the installation process can be problematic as the archives you are using are **constantly being updated** by OpenStack developers. This is out of our control and will be one of the hurdles you will need to overcome to successfully complete this lab.

IF THE INSTALLATION DOESN'T WORK THE FIRST TIME, TRY AGAIN LATER

Goals of Lab

- Configure DevStack
- Launch default instance in DevStack
- Introduce Management Console

Pre-requisites

- See Lab #1

Note: For clarity, the Virtual Machine (VM) instance we create with Vagrant will be referred to as a VVM. An instance that we create using DevStack will be referred to as a DevVM. Also, the OS of host we use is Ubuntu 15.04, but you can try the lab under Windows, the procedure is the same.

You are encouraged to read thru some pages at

- http://docs.openstack.org/developer/devstack/, in particular, the "walk-thru" for "All-In-One Single VM"
- http://docs.openstack.org/developer/devstack/configuration.html, section "Minimal Configuration"

Section 1 – Build DevStack VM and configure it automatically under Linux host

1. Make a folder in your host called "openstack/vag/".
2. Use vagrant and ubuntu/trustyXX (XX=32 or 64) to build a VM (check lab 1 on how to build a VM with vagrant). However, this VM is for devstack environment. So you must be able to write a script to automate the setup. The requirement of devstack's automation is as follow,
    a. Set the host IP address to 171.168.33.10. This is the IP address we use to log into devstack web console. ***Note: depending on the version of Vagrant and VirtualBox in use, the command "vagrant up" might fail while creating non-default private network. One fix is to use older VirtualBox version (e.g. 4.3.30 or older). Another fix is to run "vagrant up" command in the console with elevated priviliges.***
    b. Forward port 80 to 8080 (just like lab 1).
    c. Increase the VM memory to 4G or more (if you have problems with exercises below, try allocating 6.4G of RAM at least). Also, allocate 2 CPUs to VM. (Read vagrant documentation)
    d. Run a script to change the host name to "your_first_name"DevStack, e.g. johnDevStack. The following link - *http://docs.vagrantup.com/v2/vagrantfile/machine_settings.html* - can be useful but you might face a problem for setting the host name so you must find a work around via a shell script in Vagrantfile.
    e. Then run the main script.sh which configures the devstack VM as well and downloading devstack. Requirements for script.sh is as follow,
        - Update the system. `$sudo apt-get update`
        - Install git.
        - Clone devstack from '`git clone https://git.openstack.org/openstack-dev/devstack`'. Just a heads up, make sure you clone as vagrant user not root in your script.sh!
        - Now log into VVM (vagrant ssh). Change directory to devstack and then run stack.sh, if you do everything correctly then owner of stack.sh MUST be vagrant not root.

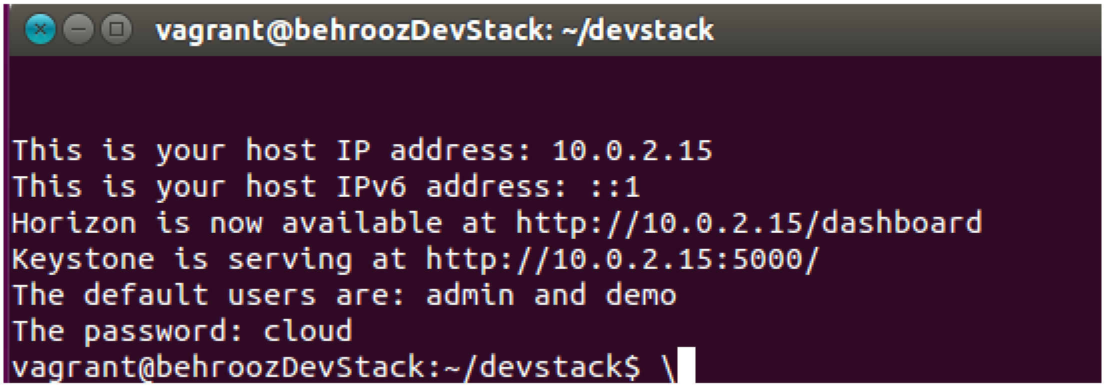
          In normal case when you run stack.sh it asks for password for its components, but we want the whole process to be automated. So, the trick here is that you can configure the devstack installation by a simple config file called "local.conf" as below,

```
[[local|localrc]]
DATABASE_PASSWORD=cloud
ADMIN_PASSWORD=cloud
MYSQL_PASSWORD=cloud
RABBIT_PASSWORD=cloud
SERVICE_PASSWORD=cloud
SWIFT_HASH=cloud
SERVICE_TOKEN=tokentoken
LOGDAYS=1
LOGFILE=/opt/stack/logs/stack.sh.log
enable_service s-proxy s-object s-container s-account
disable_service n-net
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta
enable_service q-metering
```

(1) In report you must explain the purpose of each line or groups of lines. After you have finished your research, you will notice we are using Neutron networking system instead of normal devstack networking system. At the time of this tutorial writing, Neuron was more reliable in action.

The "local.conf" file must be in devstack directory before you run stack.sh. So make sure your Vagrant file copy it to your VVM at /devstak directory before you run stack.sh.

Run stack.sh and the whole process takes up to 30 min. Then you must see this:



```
This is your host IP address: 10.0.2.15
This is your host IPv6 address: ::1
Horizon is now available at http://10.0.2.15/dashboard
Keystone is serving at http://10.0.2.15:5000/
The default users are: admin and demo
The password: cloud
vagrant@behroozDevStack:~/devstack$ \
```

As you can notice we have two users: **admin** and **demo** and the password is: **cloud**.

(2) Put the 'Vagrantfile' and 'script.sh' into report, also take screenshot of your VM terminal after 'stack.sh' has finished.

stack.sh is an opinionated OpenStack developer installation. It installs and configures various combinations of Ceilometer, Cinder, Glance, Heat, Horizon, Keystone, Nova, Neutron, Swift, and Trove. For details on what the script is doing see

- http://devstack.org/

Note: there could be more than one way to automate the whole process so use your imagination!

- Now, devstack is ready just source it by this command `$source ~/devstack/openrc`
- Now we need to create a RSA key for when we want to log into our DevVM.
  Run these command,

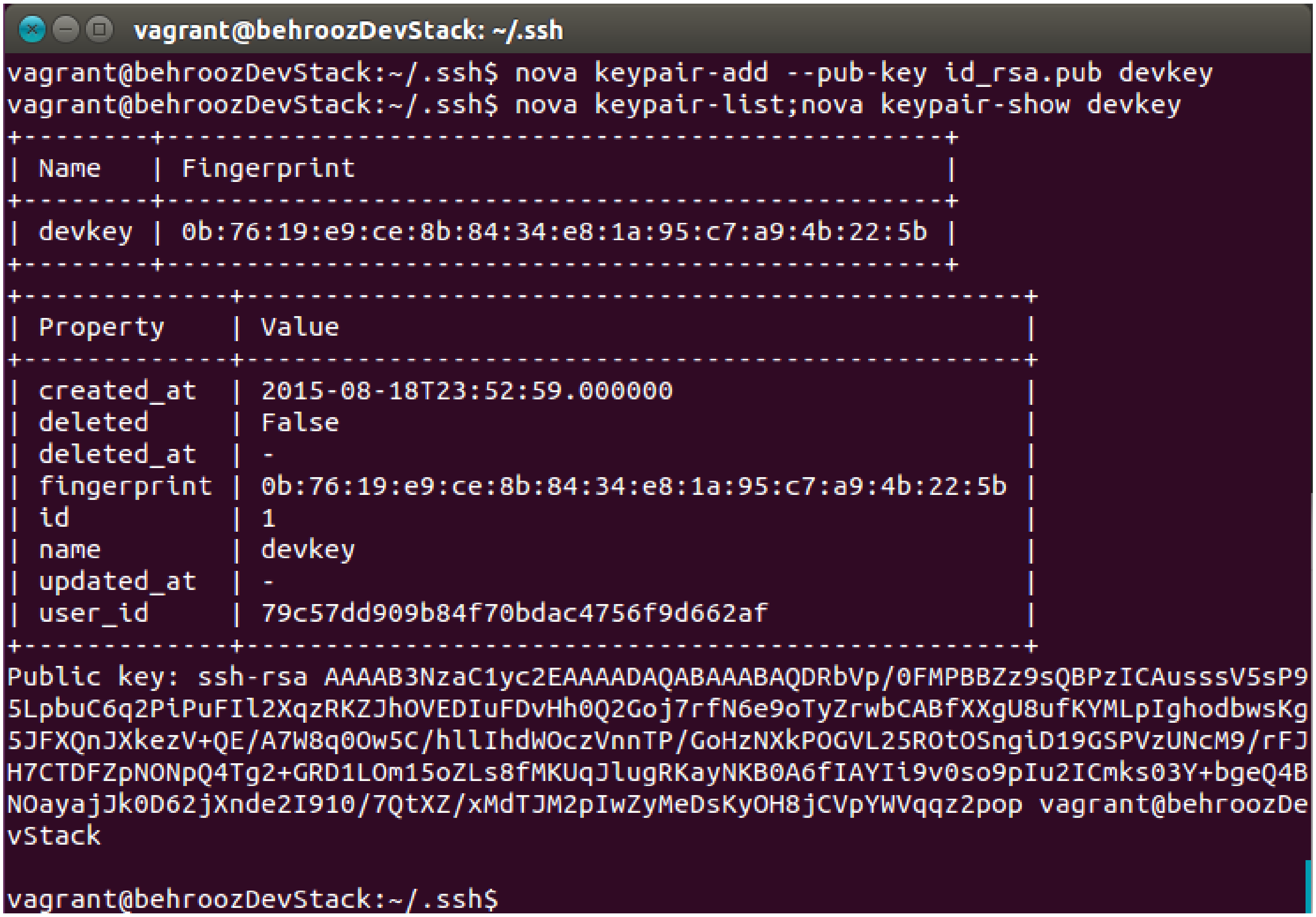
  ```
  $cd ~/.ssh
  $ssh-keygen -t rsa
  (just press enter to generate your public key)
  ```

- Now import your key via nova,

  ```
  $nova keypair-add --pub-key id_rsa.pub devkey
  ```

  Verify that the keypair has been imported

  ```
  $nova keypair-list;nova keypair-show devkey
  ```



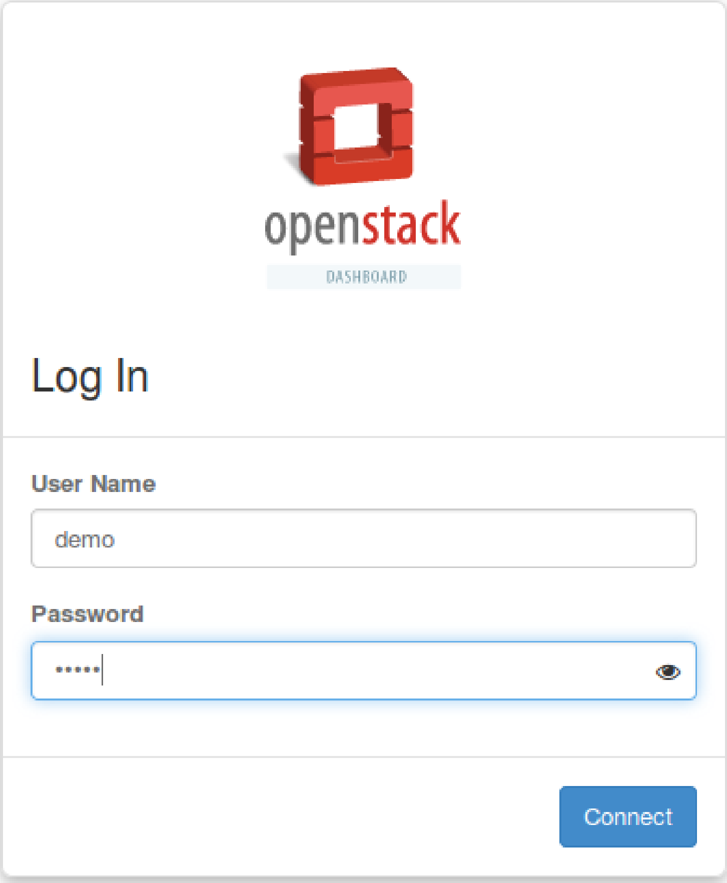
vagrant@behroozDevStack: ~/.ssh

```
vagrant@behroozDevStack:~/.ssh$ nova keypair-add --pub-key id_rsa.pub devkey
vagrant@behroozDevStack:~/.ssh$ nova keypair-list;nova keypair-show devkey
+--------+-------------------------------------------------+
| Name   | Fingerprint                                     |
+--------+-------------------------------------------------+
| devkey | 0b:76:19:e9:ce:8b:84:34:e8:1a:95:c7:a9:4b:22:5b |
+--------+-------------------------------------------------+
+-------------+-------------------------------------------------+
| Property    | Value                                           |
+-------------+-------------------------------------------------+
| created_at  | 2015-08-18T23:52:59.000000                      |
| deleted     | False                                           |
| deleted_at  | -                                               |
| fingerprint | 0b:76:19:e9:ce:8b:84:34:e8:1a:95:c7:a9:4b:22:5b |
| id          | 1                                               |
| name        | devkey                                          |
| updated_at  | -                                               |
| user_id     | 79c57dd909b84f70bdac4756f9d662af                |
+-------------+-------------------------------------------------+
Public key: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDRbVp/0FMPBBZz9sQBPzICAusssV5sP9
5LpbuC6q2PiPuFIl2XqzRKZJhOVEDIuFDvHh0Q2Goj7rfN6e9oTyZrwbCABfXXgU8ufKYMLpIghodbwsKg
5JFXQnJXkezV+QE/A7W8q0Ow5C/hllIhdWOczVnnTP/GoHzNXkPOGVL25ROtOSngiD19GSPVzUNcM9/rFJ
H7CTDFZpNONpQ4Tg2+GRD1LOm15oZLs8fMKUqJlugRKayNKB0A6fIAYIi9v0so9pIu2ICmks03Y+bgeQ4B
NOayajJk0D62jXnde2I910/7QtXZ/xMdTJM2pIwZyMeDsKyOH8jCVpYWVqqz2pop vagrant@behroozDe
vStack

vagrant@behroozDevStack:~/.ssh$
```

(3) Take a screenshot of your key and put it in report.

f. Now, we switch to horizon web console. Open your browser go to this address 171.168.33.10. You must see this,



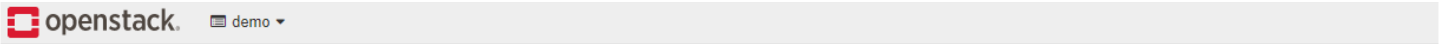
Log In

User Name: demo

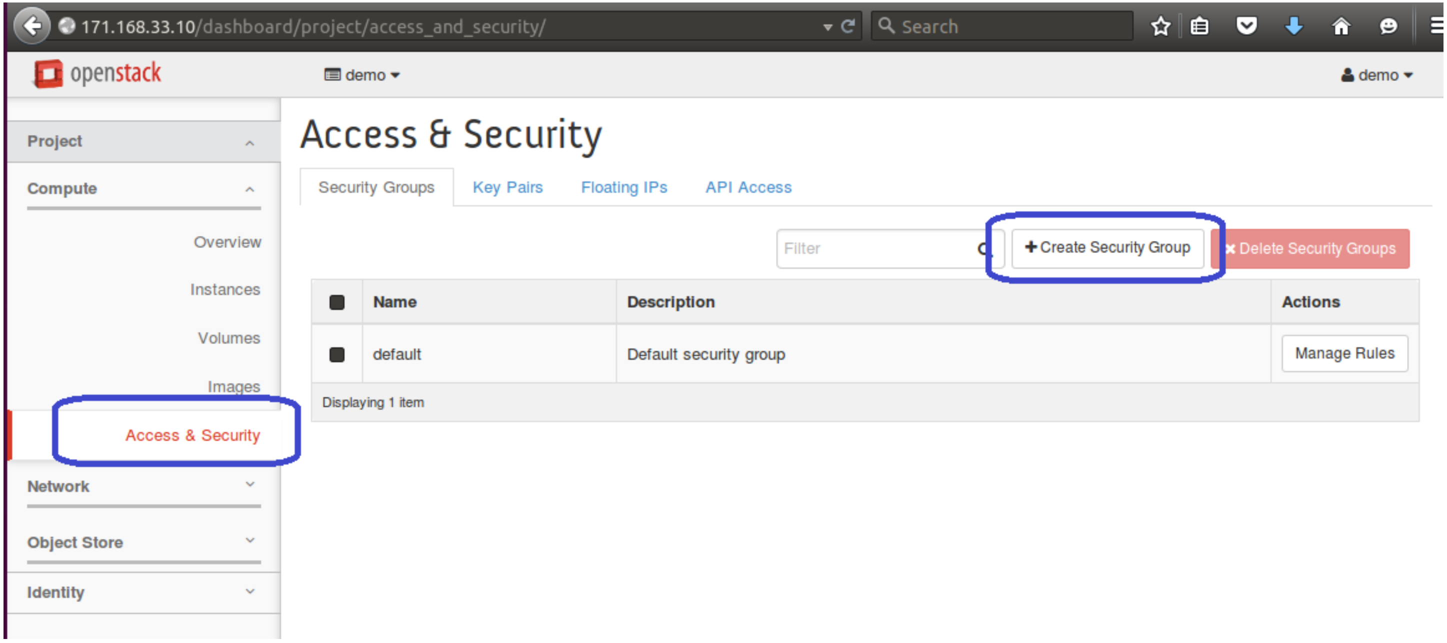
Password: •••••

Username: demo
Password: cloud

Make sure “demo” is selected on the upper left corner



- Go to Access & Security. In ‘Security Groups’ tab create a new Security Group.



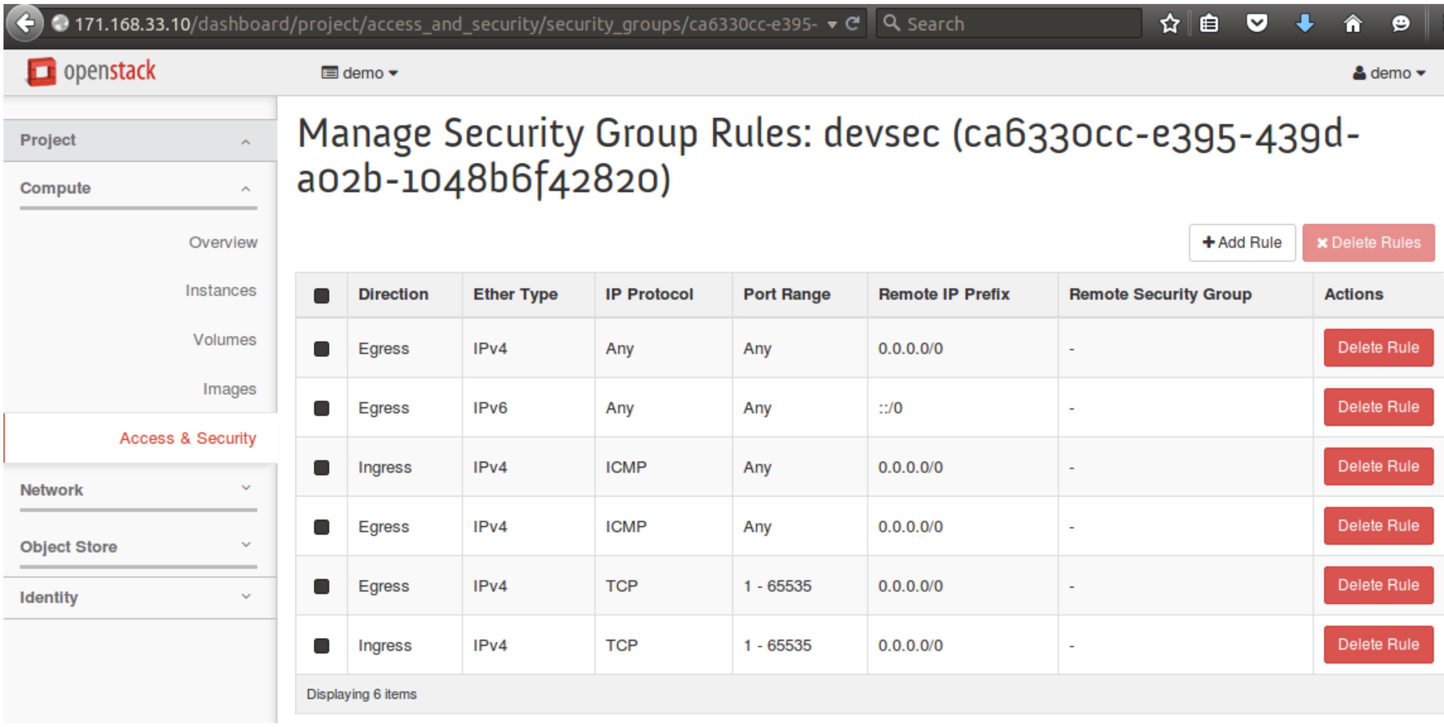
Access & Security

	Name	Description	Actions
■	default	Default security group	Manage Rules

Displaying 1 item

Name it “devsec” and for description enter “Security Group for Lab02 – DevStack”.

- Now select devsec and hit “Manage Rules”. Then in new window hit “Add Rule”. You need to add a rule for “All ICMP” Ingress and Engress (to ping DevVM) and TCP for port 22 so you can SSH into DevVM.



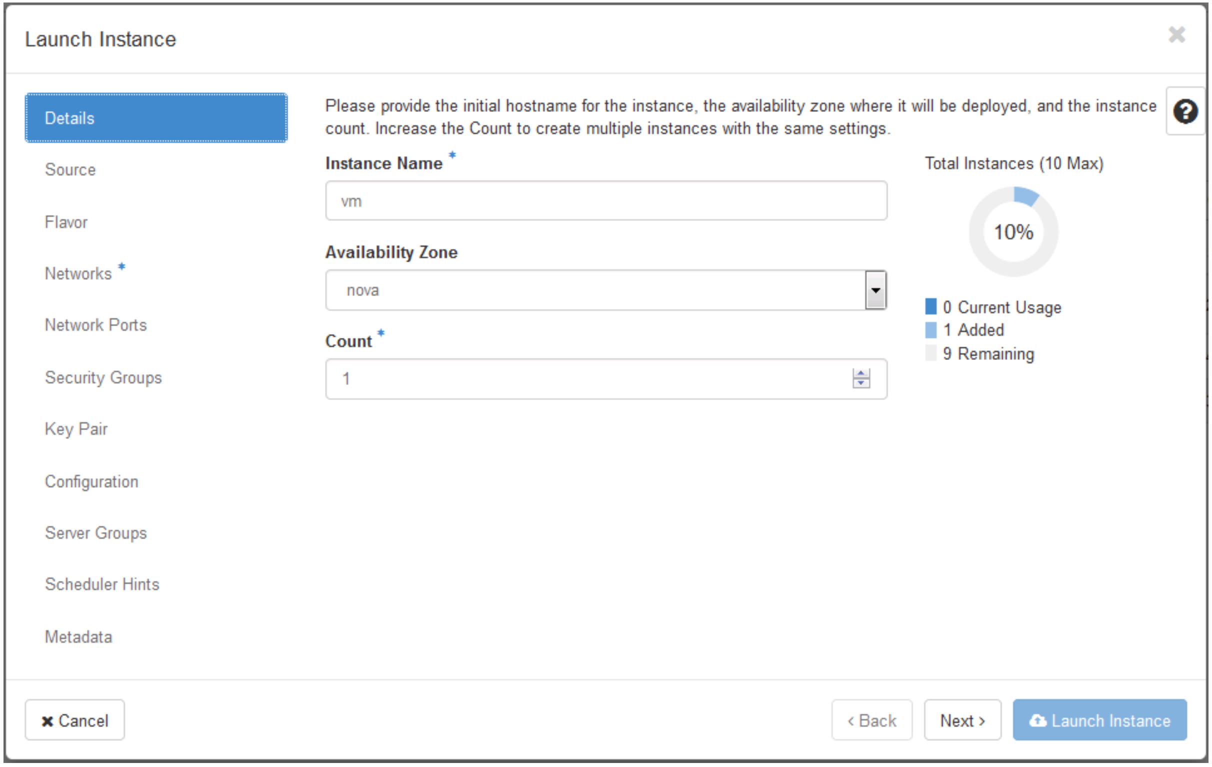
Manage Security Group Rules: devsec (ca6330cc-e395-439d-a02b-1048b6f42820)

	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
■	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
■	Egress	IPv6	Any	Any	::/0	-	Delete Rule
■	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
■	Egress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
■	Egress	IPv4	TCP	1 - 65535	0.0.0.0/0	-	Delete Rule
■	Ingress	IPv4	TCP	1 - 65535	0.0.0.0/0	-	Delete Rule

Displaying 6 items

If you pay attention to the screenshot above all TCP ports were opened. **(4) Is it a good idea to open all ports?**

- Before we continue, explore other sections of dashboard such as Network, Object Store, etc. Just explore.
- Now go to image section and launch “cirros-0.3.4-x86_64-uec” and do as follow until your machine is boot up.



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.

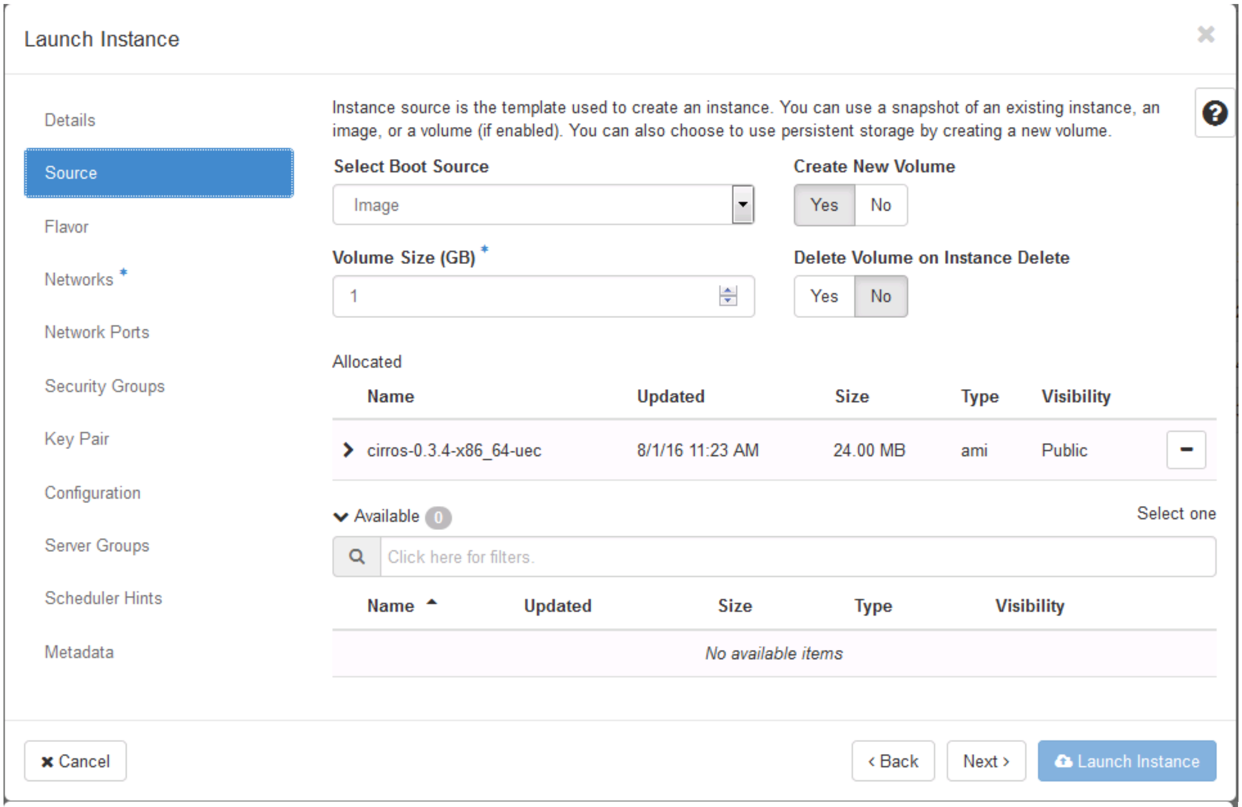
Instance Name *: vm

Availability Zone: nova

Count *: 1

Total Instances (10 Max): 10%

0 Current Usage
1 Added
9 Remaining



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

Select Boot Source: Image

Create New Volume: [✓ Yes] No

Volume Size (GB) *: 1

Delete Volume on Instance Delete: Yes [✓ No]

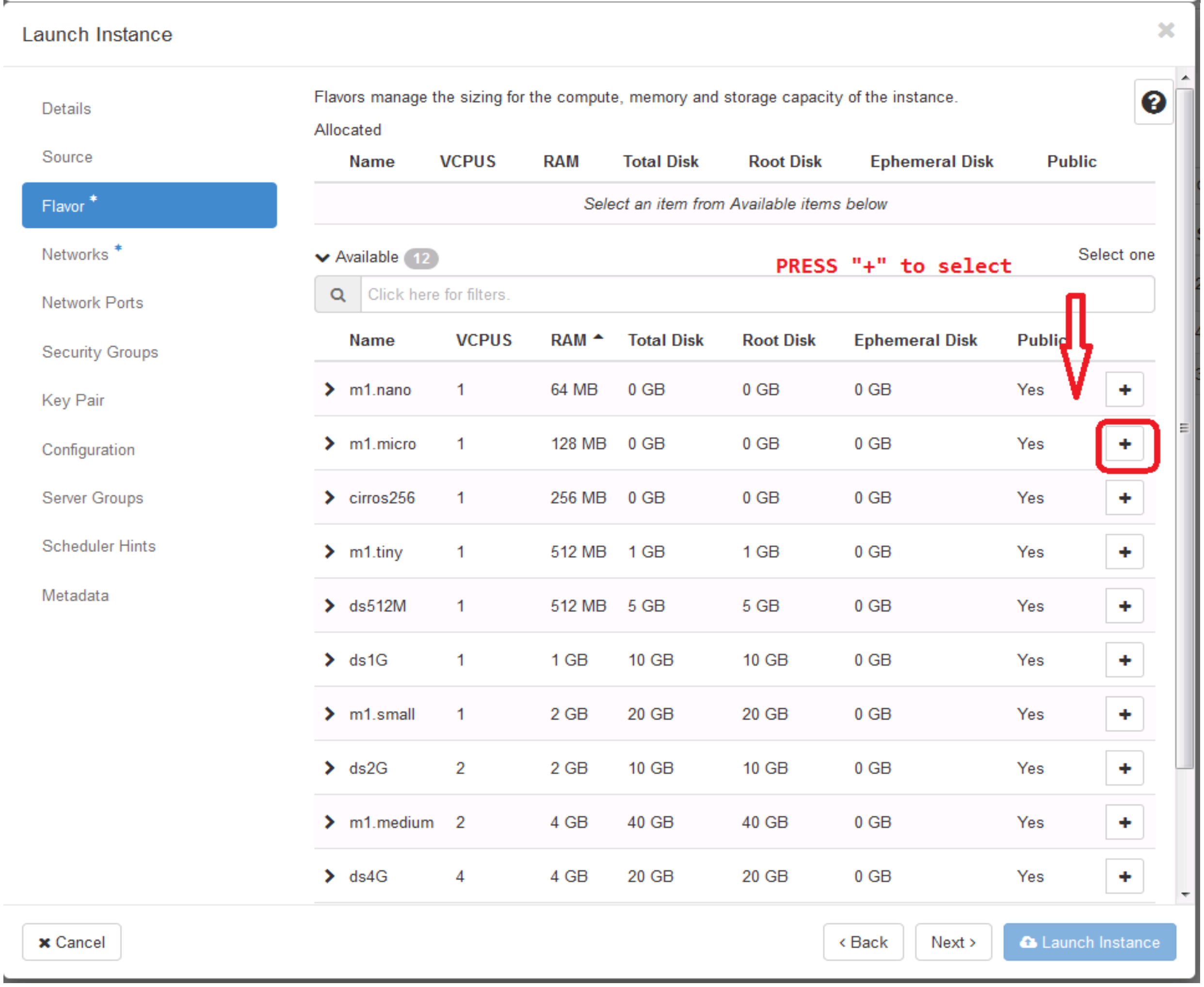
Allocated

Name	Updated	Size	Type	Visibility
cirros-0.3.4-x86_64-uec	8/1/16 11:23 AM	24.00 MB	ami	Public

Available 0 — Select one

Click here for filters.

Name	Updated	Size	Type	Visibility
No available items				



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

Allocated

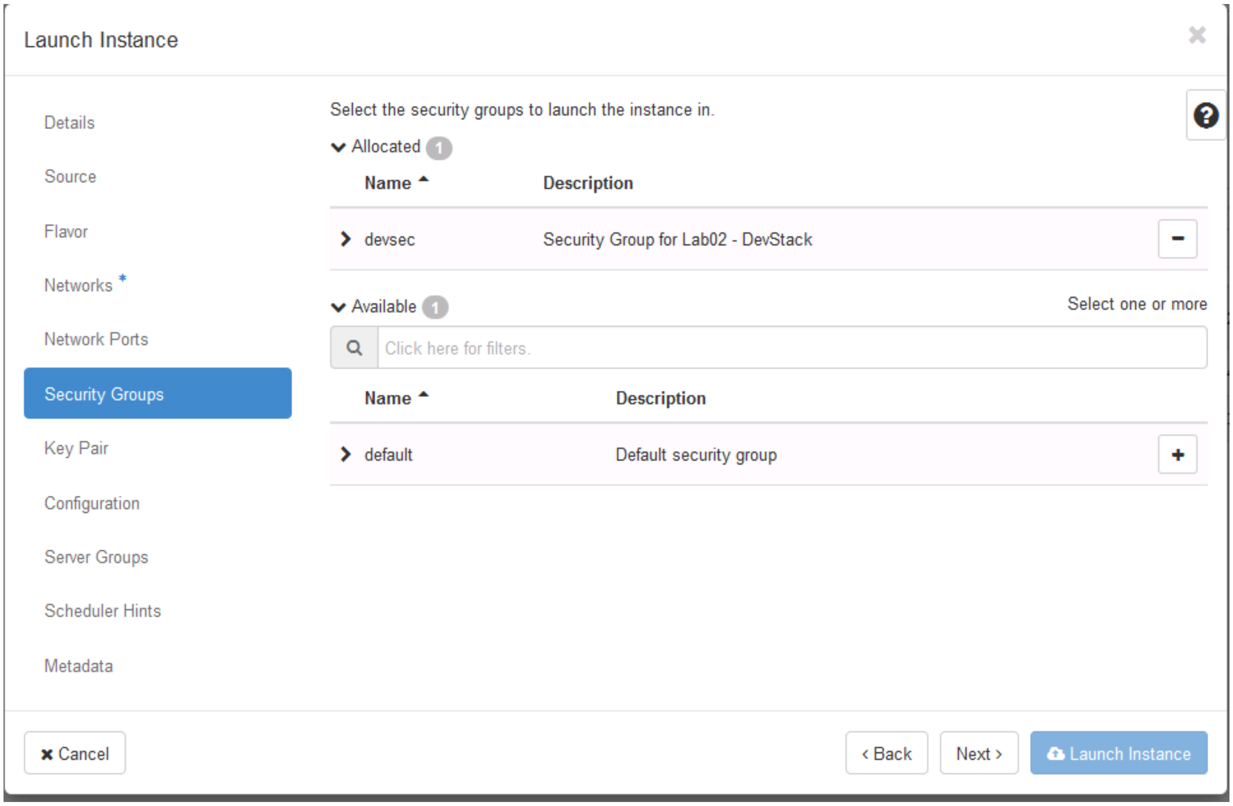
Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
Select an item from Available items below						

Available 12 — Select one

PRESS "+" to select

Click here for filters.

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.nano	1	64 MB	0 GB	0 GB	0 GB	Yes
m1.micro	1	128 MB	0 GB	0 GB	0 GB	Yes
cirros256	1	256 MB	0 GB	0 GB	0 GB	Yes
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes
ds512M	1	512 MB	5 GB	5 GB	0 GB	Yes
ds1G	1	1 GB	10 GB	10 GB	0 GB	Yes
m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes
ds2G	2	2 GB	10 GB	10 GB	0 GB	Yes
m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes
ds4G	4	4 GB	20 GB	20 GB	0 GB	Yes



Select the security groups to launch the instance in.

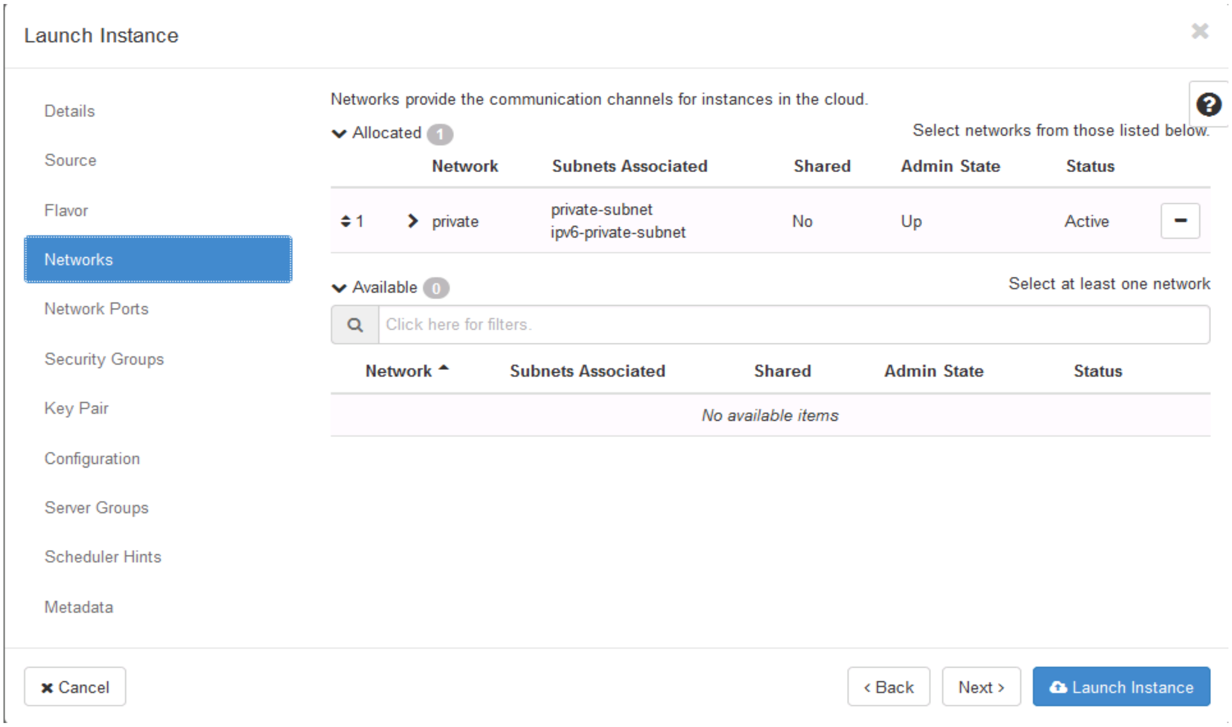
Allocated 1

Name	Description
devsec	Security Group for Lab02 - DevStack

Available 1 Select one or more

Click here for filters.

Name	Description
default	Default security group



Networks provide the communication channels for instances in the cloud.

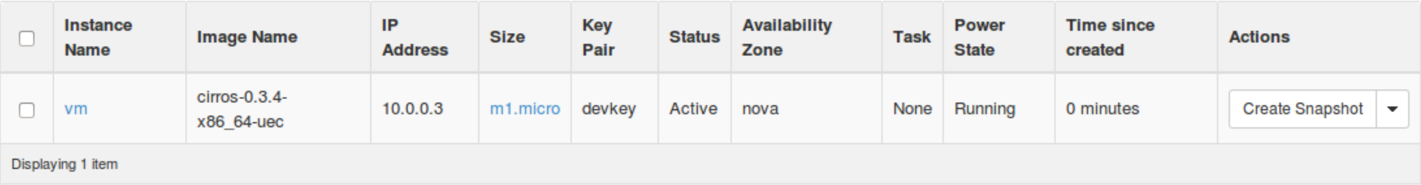
Allocated 1 Select networks from those listed below.

	Network	Subnets Associated	Shared	Admin State	Status
1	private	private-subnet ipv6-private-subnet	No	Up	Active

Available 0 Select at least one network

Click here for filters.

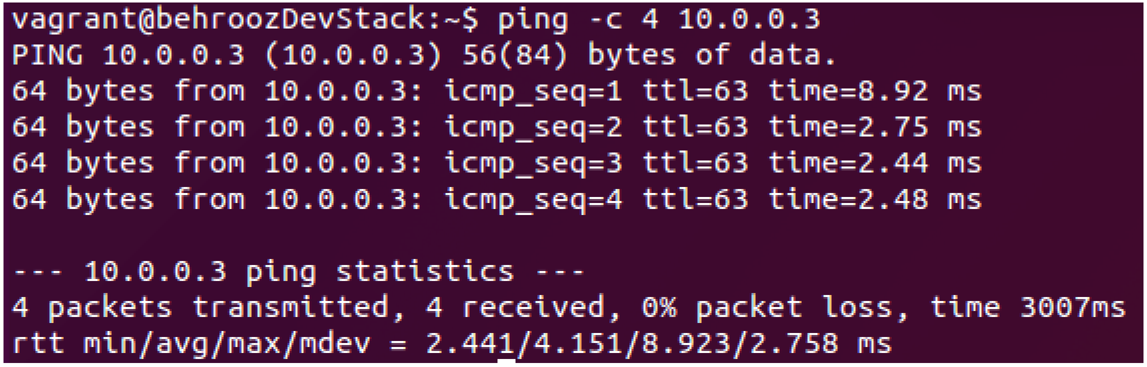
Network	Subnets Associated	Shared	Admin State	Status
No available items				



□	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	vm	cirros-0.3.4-x86_64-uec	10.0.0.3	m1.micro	devkey	Active	nova	None	Running	0 minutes	Create Snapshot ▾

Displaying 1 item

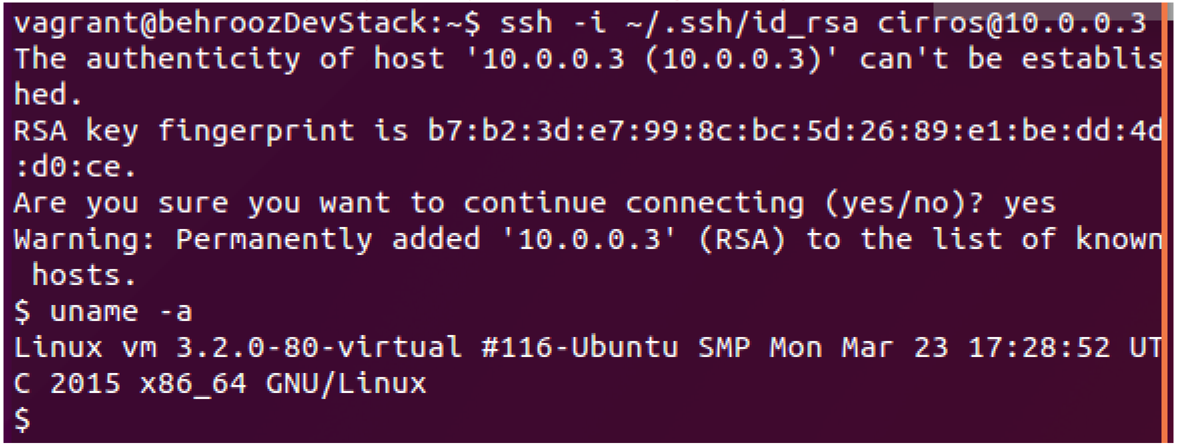
- As you can see the ip of our DevVM is 10.0.0.3. **(5) Go to your VVM and ping your DevVM as `$ping -c 4 10.0.0.3`, take a screenshot and put in report. (May not work now in 2017. Try to use floating IP described below to ping instead)**



```
vagrant@behroozDevStack:~$ ping -c 4 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=63 time=8.92 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=63 time=2.75 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=63 time=2.44 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=63 time=2.48 ms

--- 10.0.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 2.441/4.151/8.923/2.758 ms
```

- Now we ssh into our DevVM,
  `$ssh -i ~/.ssh/id_rsa cirros@10.0.0.3`
  (type yes afterwards)
  `$uname -a`
  (6) Then take screenshot and put in report. (use floating IP to login)



```
vagrant@behroozDevStack:~$ ssh -i ~/.ssh/id_rsa cirros@10.0.0.3
The authenticity of host '10.0.0.3 (10.0.0.3)' can't be establis
hed.
RSA key fingerprint is b7:b2:3d:e7:99:8c:bc:5d:26:89:e1:be:dd:4d
:d0:ce.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.3' (RSA) to the list of known
 hosts.
$ uname -a
Linux vm 3.2.0-80-virtual #116-Ubuntu SMP Mon Mar 23 17:28:52 UT
C 2015 x86_64 GNU/Linux
$
```

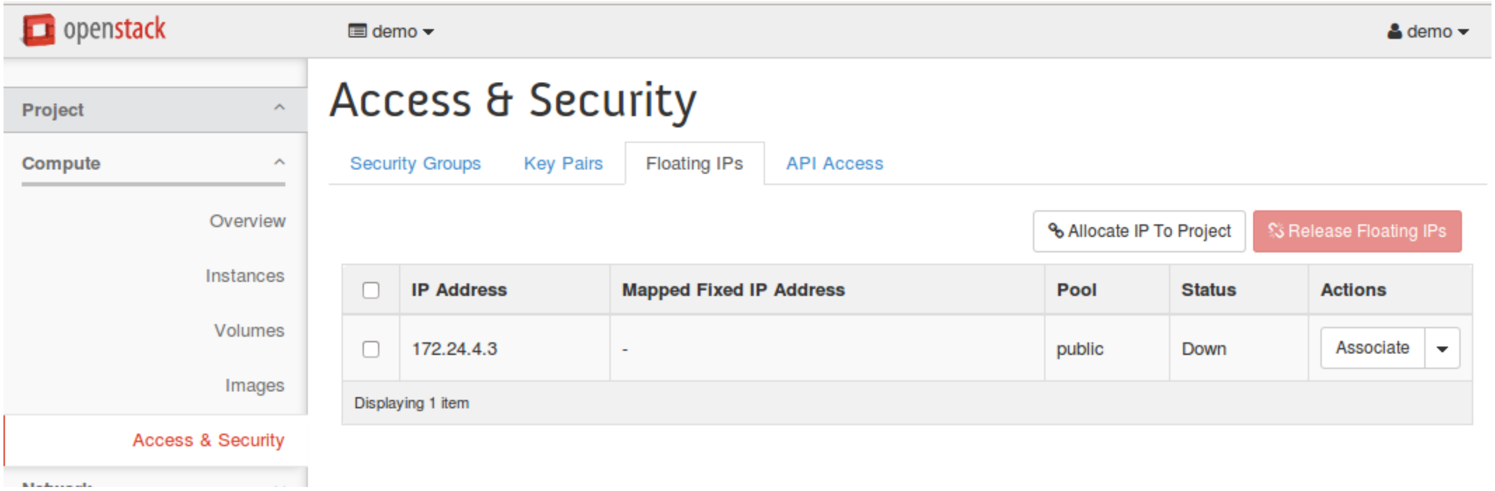
Now ping 10.0.0.1 the router and ping must be ok. But now ping 8.8.8.8 (Google Public DNS), ping will fail. Now in VVM run

`$sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE`

and now ping 8.8.8.8 in DevVM, it works, and it means our DevVM is connected to internet!

(7) Take a screenshot of your success.

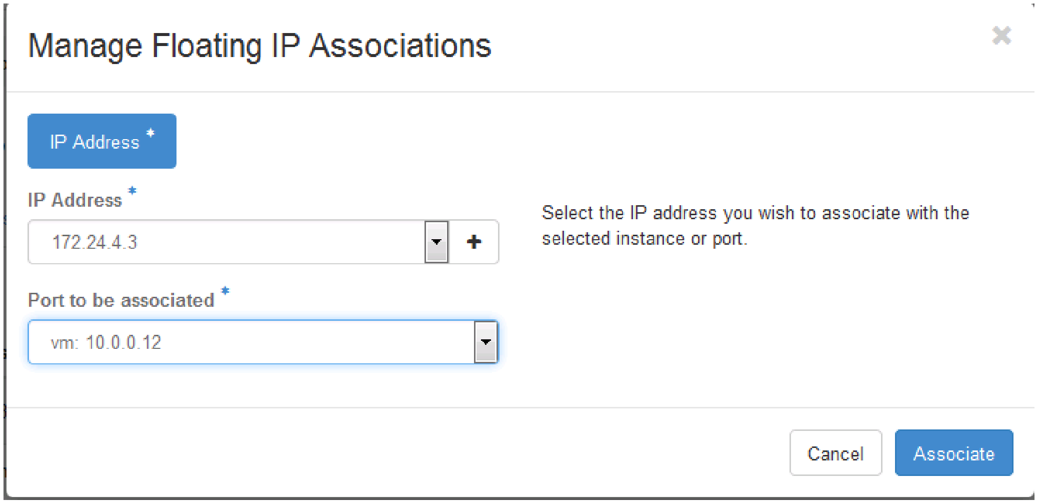
- Let's go back to web console and assign the DevVM floating IP so we can ssh into it from our host.
  - Go to "Access & Security" section and "Floating IPs" tab and "Allocate IP to Project".



□	IP Address	Mapped Fixed IP Address	Pool	Status	Actions
□	172.24.4.3	-	public	Down	Associate

Displaying 1 item

- ○ Click “Associate”, then



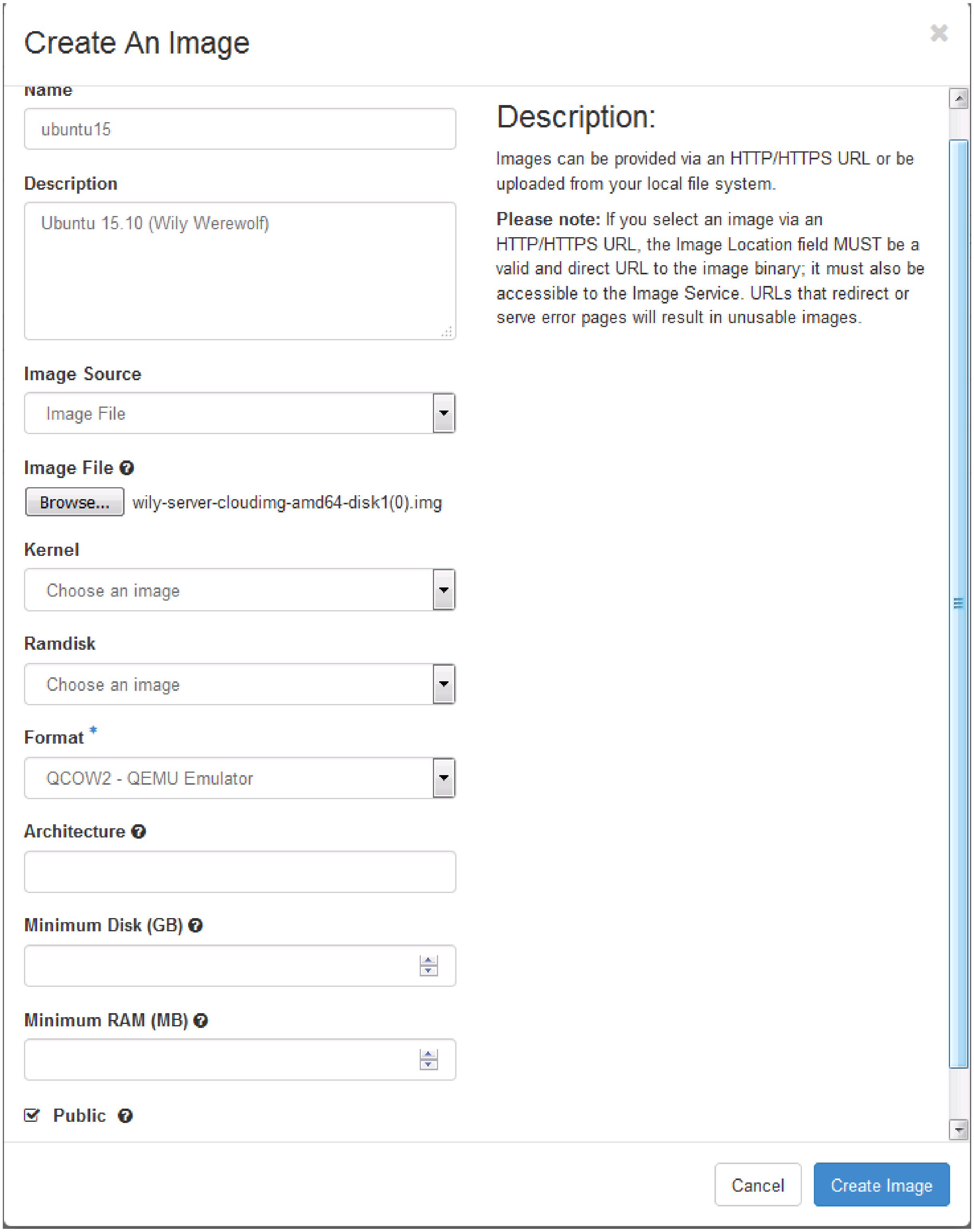
IP Address *: 172.24.4.3

Port to be associated *: vm: 10.0.0.12

Select the IP address you wish to associate with the selected instance or port.

- ○ Now, depending on your host, DevVM is accessible from your host with IP of 172.24.4.3. **(8) Try to ping it and take a screenshot.** Most likely, it will fail and if it does, you are encouraged to fix it (Hint: think about network topology and routing)
- ○ END OF LAB2. Extra credit for the rest

- ▪ Now in your host machine, in openstack directory make a new directory called os-image, then in that directory run
  `$wget http://cloud-images.ubuntu.com/wily/current/wily-server-cloudimg-amd64-disk1.img`
  This command download the Ubuntu 15.10 image (you can download any other images), which we are about to add to our devstack environment.
- ▪ When download is completed go to web console again, **but this time log in as admin**. Go to Image section and press “Create Image” the name it as ubuntu15, for “Image Source” select “Image File” then select the the *.img file that you have just downloaded in openstack/os-image. For “Format” select QCOW2. Also, make sure the image is **Public**.
  Now, we have added a new image of Ubuntu 15.



Create An Image

Name: ubuntu15

Description: Ubuntu 15.10 (Wily Werewolf)

Image Source: Image File

Image File: wily-server-cloudimg-amd64-disk1(0).img

Kernel: Choose an image

Ramdisk: Choose an image

Format *: QCOW2 - QEMU Emulator

Architecture:

Minimum Disk (GB):

Minimum RAM (MB):

☑ Public

Description:

Images can be provided via an HTTP/HTTPS URL or be uploaded from your local file system.

Please note: If you select an image via an HTTP/HTTPS URL, the Image Location field MUST be a valid and direct URL to the image binary; it must also be accessible to the Image Service. URLs that redirect or serve error pages will result in unusable images.

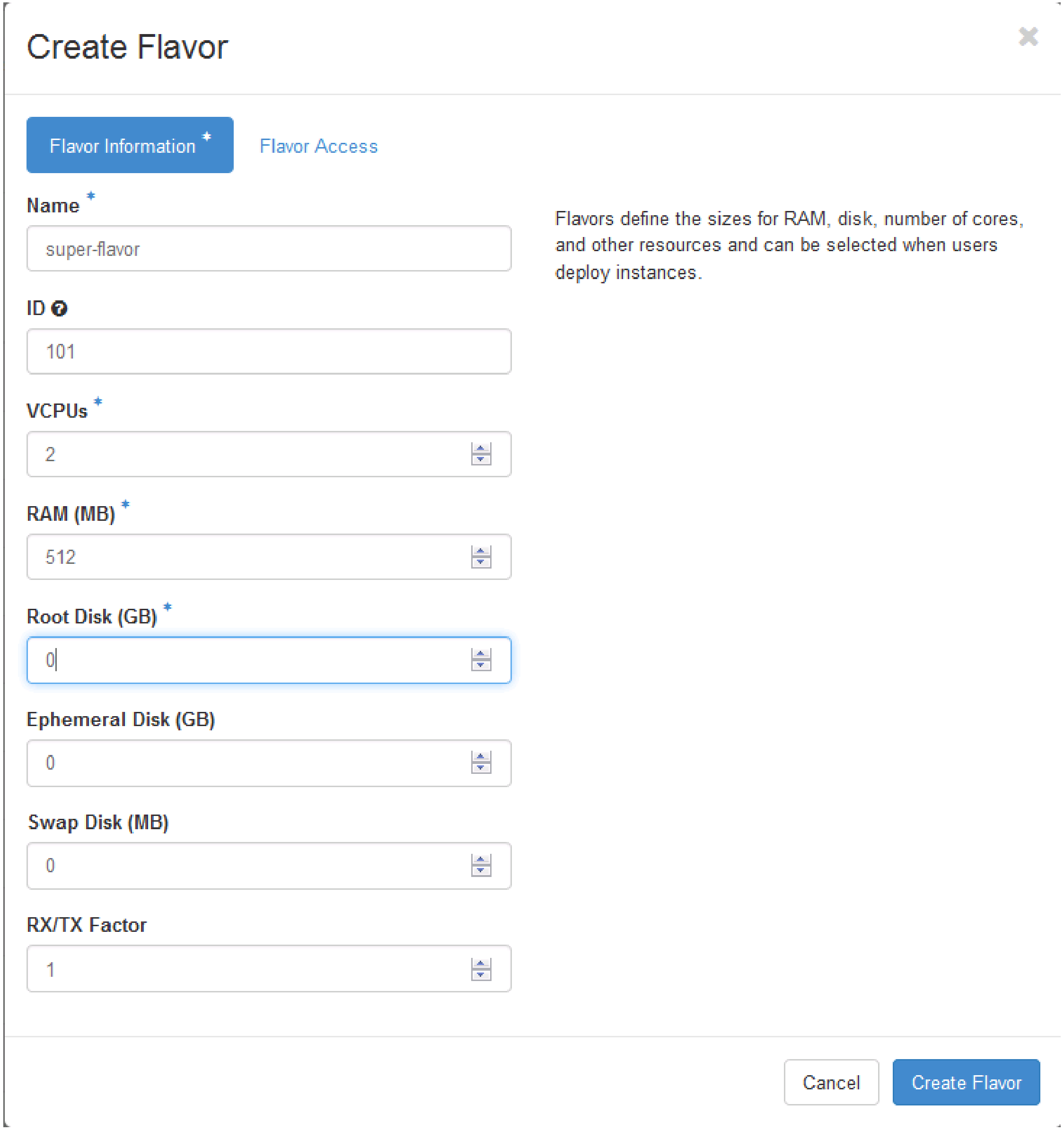
Optional go to VVM:

- Run `$glance image-list`, you must see your new image 'ubuntu15'.
  There are three kinds of disks displayed that need explanation:
  - AMI - Amazon Machine Image
  - ARI - Amazon Ramdisk Image
  - AKI - Amazon Kernel Image

  The AMI file is a readonly filesystem which is what DevStack uses to create the cloud instance. The ARI and AKI files are used by the AMI to create and initialize the cloud instance.

- Run `$nova flavor-list`
  In DevStack the size of a cloud instance is referred to as its "flavor". The smallest flavor available is "m1.nano" which has 64 MB of RAM allocated to it. This is the size we will use. Note that this "flavor" has an ID of "**42**" in the screenshot above.
- You can more information on web, some references:
  http://docs.openstack.org/user-guide/cli_cheat_sheet.html

- In here we create a new flavor called "super-flavor"



Create Flavor

Name * super-flavor

ID 101

VCPUs * 2

RAM (MB) * 512

Root Disk (GB) * 0

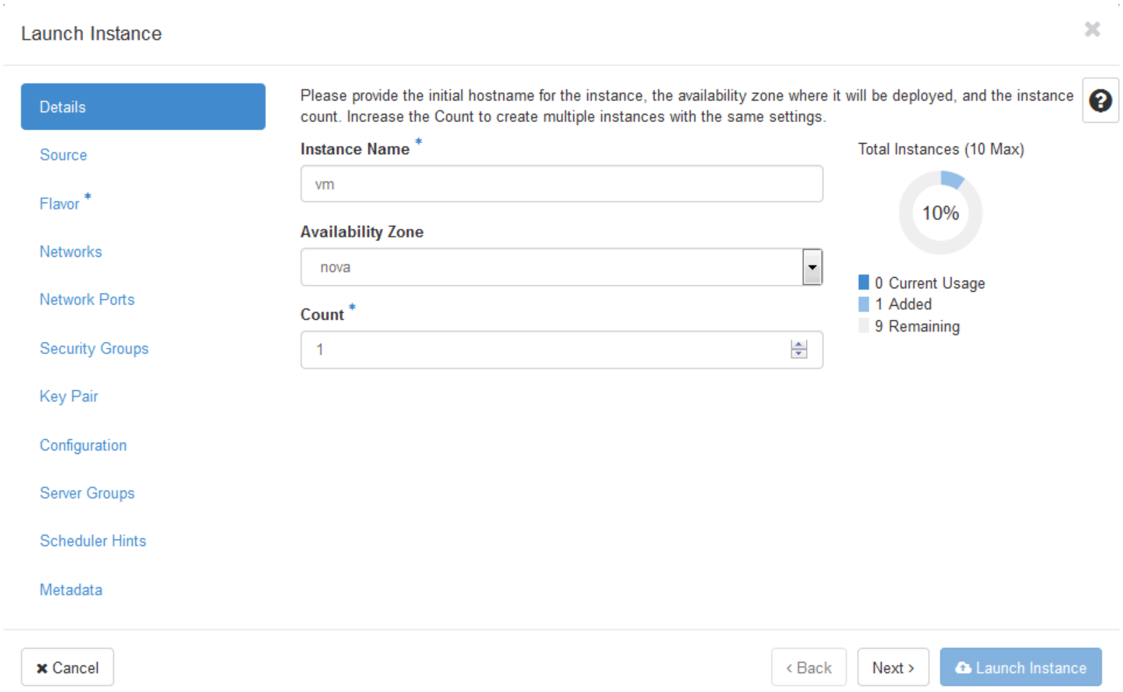
Ephemeral Disk (GB) 0

Swap Disk (MB) 0

RX/TX Factor 1

Flavors define the sizes for RAM, disk, number of cores, and other resources and can be selected when users deploy instances.

- Now again **log in as demo**. Go to Image section and lunch 'ubuntu15', instead of 'vm' call your DevVM <Your First Name>vm (e.g. johnvm).



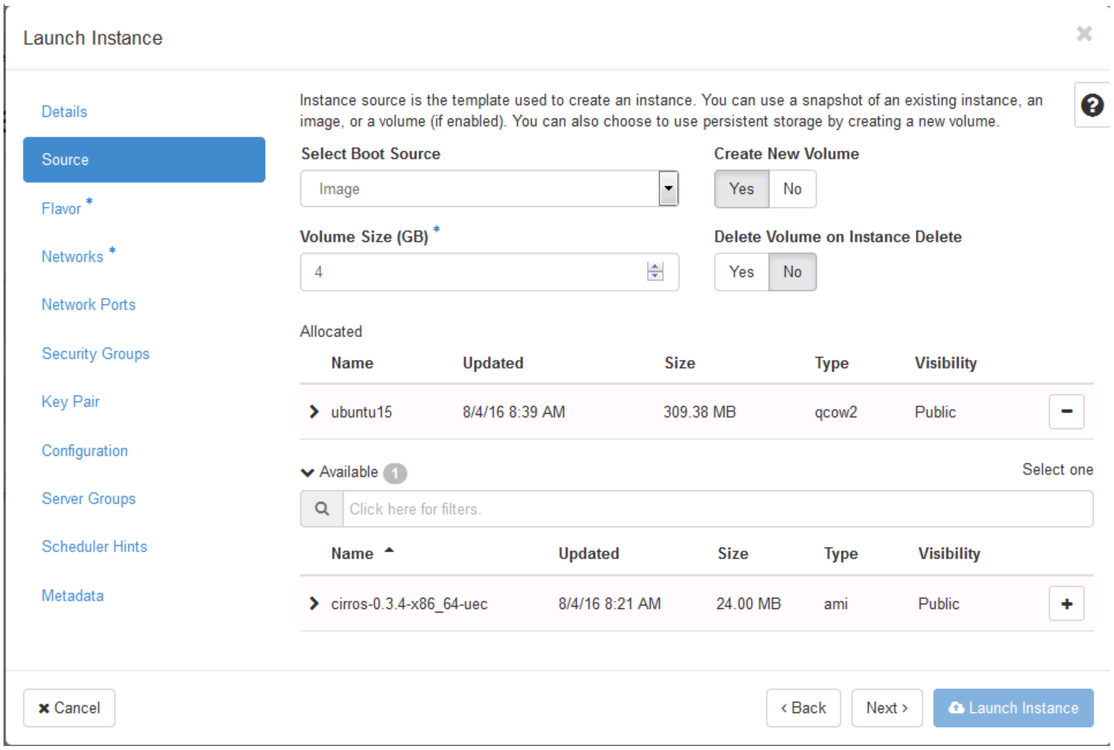
Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name *	vm
Availability Zone	nova
Count *	1

Total Instances (10 Max)

10%

- 0 Current Usage
- 1 Added
- 9 Remaining



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

Select Boot Source	Image
Create New Volume	[✓ Yes] No
Volume Size (GB) *	4
Delete Volume on Instance Delete	Yes [✓ No]

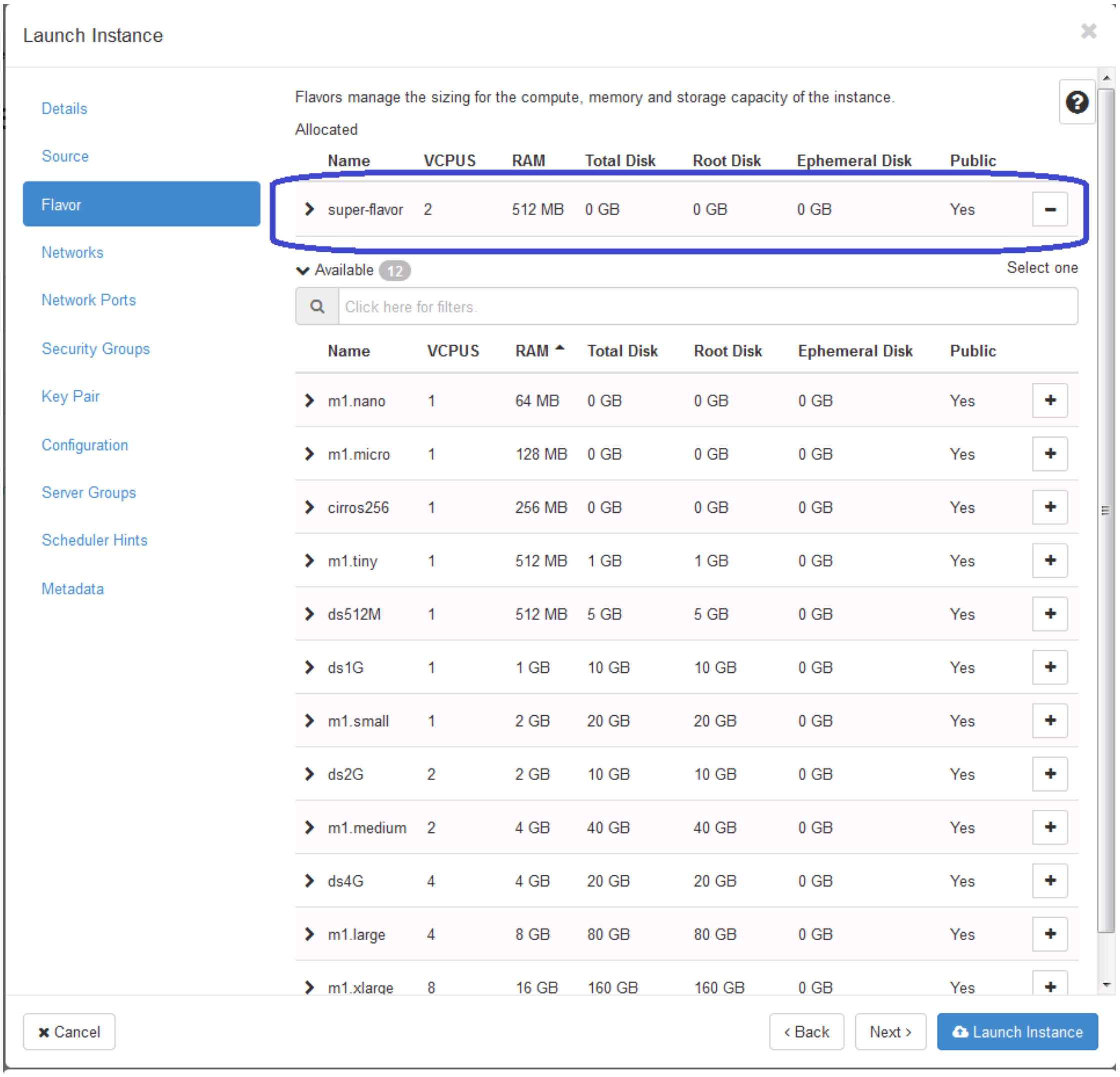
Allocated

Name	Updated	Size	Type	Visibility	
ubuntu15	8/4/16 8:39 AM	309.38 MB	qcow2	Public	−

Available 1 Select one

Click here for filters.

Name	Updated	Size	Type	Visibility	
cirros-0.3.4-x86_64-uec	8/4/16 8:21 AM	24.00 MB	ami	Public	+



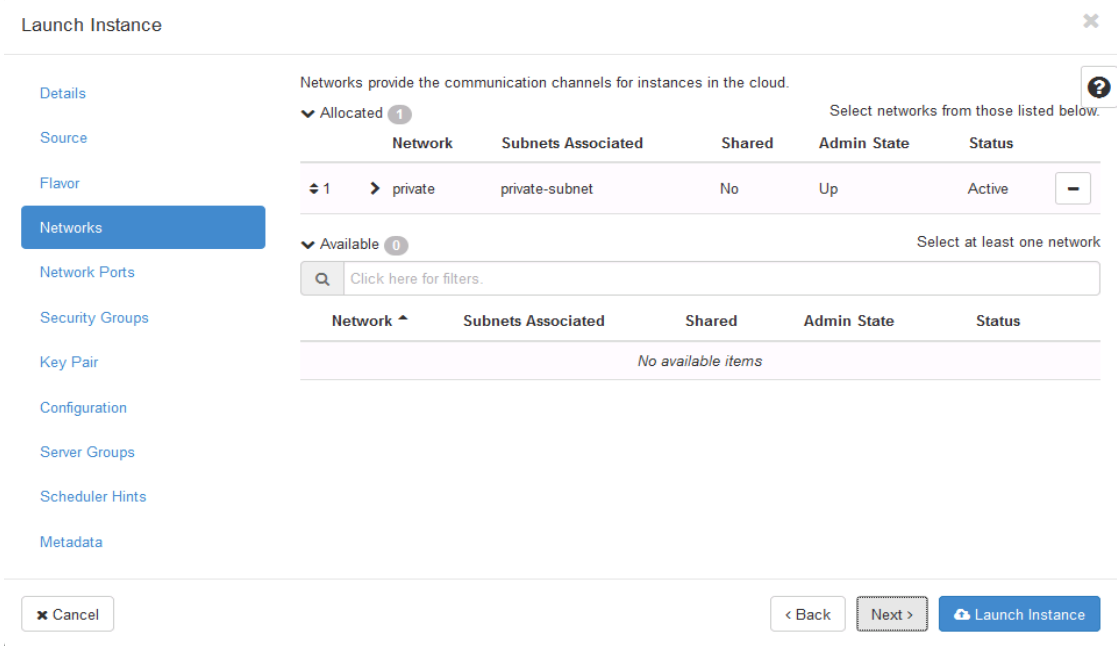
Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
super-flavor	2	512 MB	0 GB	0 GB	0 GB	Yes

Available 12 — Select one

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.nano	1	64 MB	0 GB	0 GB	0 GB	Yes
m1.micro	1	128 MB	0 GB	0 GB	0 GB	Yes
cirros256	1	256 MB	0 GB	0 GB	0 GB	Yes
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes
ds512M	1	512 MB	5 GB	5 GB	0 GB	Yes
ds1G	1	1 GB	10 GB	10 GB	0 GB	Yes
m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes
ds2G	2	2 GB	10 GB	10 GB	0 GB	Yes
m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes
ds4G	4	4 GB	20 GB	20 GB	0 GB	Yes
m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes
m1.xlarge	8	16 GB	160 GB	160 GB	0 GB	Yes



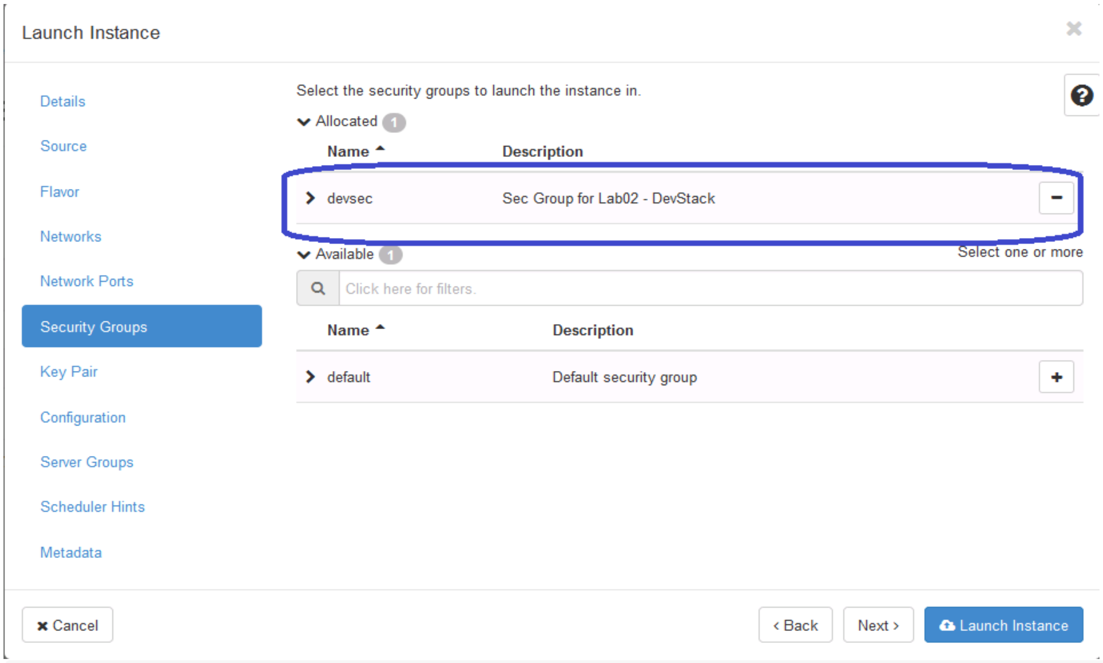
Networks provide the communication channels for instances in the cloud.

Allocated 1 — Select networks from those listed below.

	Network	Subnets Associated	Shared	Admin State	Status
1	private	private-subnet	No	Up	Active

Available 0 — Select at least one network

Network	Subnets Associated	Shared	Admin State	Status
No available items				



Launch Instance

Select the security groups to launch the instance in.

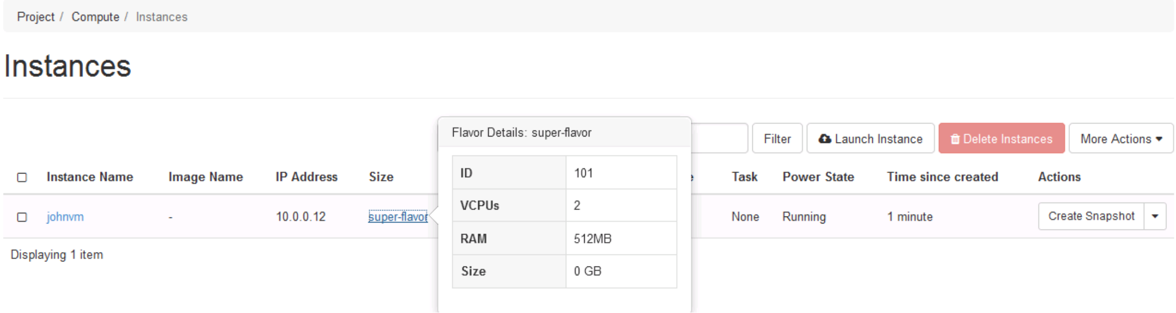
Allocated 1

Name	Description
devsec	Sec Group for Lab02 - DevStack

Available 1 · Select one or more

Click here for filters.

Name	Description
default	Default security group



Instances

Instance Name	Image Name	IP Address	Size	Task	Power State	Time since created	Actions
johnvm	-	10.0.0.12	super-flavor	None	Running	1 minute	Create Snapshot

Displaying 1 item

Flavor Details: super-flavor

ID	101
VCPUs	2
RAM	512MB
Size	0 GB

It takes a while until system boots up. **(9) Run `$nova list`, in VVM and take a screen shot of instances.**

- After 20-25 minutes try to ssh into system (`$ssh –i ~/.ssh/id_rsa ubuntu@10.0.0.x`) and ping 8.8.8.8. **(10) Take screen shot. Upon successful ping then download apache2, associate a floating IP to apache VM and then reboot the instance. When instance is up again use the public IP in browser to see the default apache page, take screen shot of your success.**

  Note 1: *you don't have to reboot the instance – after you install apache2, you can start it by running the following command:*

  $ sudo service apache2 start

  Note 2: *you might fail to access websites from DevVM even though pinging 8.8.8.8 is successful. One of the reasons could be with the name resolution service. Modify the file:*

  ```
  /etc/resolvconf/resolv.conf.d/head
  ```

  so it has the following line: "nameserver 8.8.8.8". After you have updated this file, run the following command:

  ```
  $ sudo service networking restart
  ```

  Now, try to ping google.com .
  (Source: http://askubuntu.com/questions/455338/ping-unknown-host-google-com-but-ips-works-fine)

Deliverables

- Create a {Microsoft Word| PDF} document containing the answers to the test section.
- Name the file <Last Name>_<First Name>_Lab02.{docx|pdf}
- Send the file to me via slack private message

FAQ

What is the difference between DevStack and OpenStack?

DevStack is a developer version of OpenStack designed to be installed on a single machine for an OpenStack developer to play around with. DevStack is a fully functional version of OpenStack (i.e. **everything** you can do in OpenStack can be done in DevStack). The only difference is:

a) It's all on one machine
b) It's (relatively) easy to install and configure.
c) It's slower (because everything is running on one machine)

Why don't we just install OpenStack?

The regular installation of OpenStack requires a minimum of 3 computers (2 servers and one client) with a significant amount of manual configuration. While this would be interesting to configure, the skillset required for the install is more in line with Linux System Administration (which we are not teaching) as opposed to Cloud Computing.

As mentioned before DevStack has the same functionality as OpenStack so besides it running slower, you as a cloud developer will not see any differences working with it.

If you are interested in doing a non-DevStack deployment of OpenStack official Install and Deploy guides can be found here:

http://docs.openstack.org/

Can we use Windows based VM's in OpenStack instead of Linux?

Yes. OpenStack supports the creation and use of cloud instances running Microsoft Windows. Unfortunately due to time constraints this topic will not be covered in this class.

I tried to create multiple cloud instances in DevStack but it only lets me make one (I get an error message when I try to make more.) What is wrong?

This may happen if you use a different "flavor".

As an example, in Section 1 the Vagrant VM instance was configured to run with ~2GB (2048MB) of RAM. DevStack + Linux take up almost 1.3 GB of RAM which leaves ~700MB left over for cloud instances.

The "tiny" instance uses 512MB. If you try to run 2 tiny instances then they collectively will take up 2 x 512 =1024 MB of RAM which is more than what is available.

If you'd like to run multiple cloud instances (and can spare memory) increase the amount of memory allocated to the Vagrant VM instance as you did in Section 1 (however be sure to leave enough for your host OS to run).

References

DevStack: http://devstack.org

OpenStack: http://www.openstack.org/

Vagrant: http://vagrantup.com

AMI Description: http://en.wikipedia.org/wiki/Amazon_Machine_Image

Possible problems and solutions

"No Host IP" error returned by stack.sh:

https://bugs.launchpad.net/devstack/+bug/1644145

stack.sh returns dependency error ("missing", "cannot found" …etc.)

I used "ubuntu/xenial64" instead of "ubuntu/trusty64" to avoid any dependency error. You can use other images of course. The password for xenial64 is in "\.vagrant.d\boxes\ubuntu-VAGRANTSLASH-xenial64\20170704.0.0\virtualbox\Vagrantfile". User name is "ubuntu". In general, you should pick vagrant box with ubuntu 16.04

Creating instance in Horizon return error

Please give your VVM at least 6.5GB of ram. More if you have more to spare.

Cannot create instance with a custom os-image (i.e. Wily, ubuntu…)

Create an instance without volume then create a volume separately (leave image source to default). Finally, attach the volume to the instance. The os-image I use is https://cloud-images.ubuntu.com/trusty/current/

I cannot add rules to a security group in Horizon

Use command line instead

https://docs.openstack.org/python-openstackclient/latest/cli/command-objects/security-group-rule.html#security-group-rule-create

https://docs.openstack.org/user-guide/cli-nova-configure-access-security-for-instances.html