Assignment II

INF0 4110 – CLOUD COMPUTING

STEVEN SAITO - 100296655

Contents

Introduction 3
What Steps Did I Take To Set up Network 3
    On Host Machine 3
    On Debian Instance 4
    Problems I encountered 5
Overview 6
Instances 7
Network Topology graph 9
Networks 10
Routers 11
Floating IPs 12
The Website 13
References 15

Introduction

For this assignment we needed to create a website on a TryStack instance. The TryStack link kept pointing me to OpenStack and wanted me to use DevStack. DevStack required a lot of configurations and was one error after another just to set up the environment.

So, I found and used MicroStack. MicroStack is a single OpenStack deployment environment which can run on a workstation. I downloaded and used MicroStack on an Ubuntu VM.

This network as far as I am aware, cannot be access outside of the Virtual Machine. So I do not have an IP address for my website in that sense. The IP addresses associated to it were 10.20.20.125 and 192.168.222.240 (the floating IP address).

Right now the usernames and passwords are all the defaults.

- debian@<<THE IP ADDRESS>> and no password
- cirros@<<THE IP ADDRESS>> and no password
- the username is "admin" and the password is "keystone"

What Steps Did I Take To Set up Network

On Host Machine

Sudo apt-get update

Sudo apt-get upgrade

sudo snap install microstack --classic--beta

sudo microstack.init –auto

microstack.openstack catalog list

microstack.launch cirros --name awesome

ssh -i $HOME/.ssh/id_microstack cirros@10.20.20.151

sudo apt-get install cloud-init

Download admin-openrc from Horizon (Openstack web GUI in the top right)

Download a cloud image of debian in .qcow2 format

- Go to the file where you downloaded these items

source admin-openrc.sh

sudo snap install openstackclients

- You may be prompted to enter more like –classic after the above command and another

microstack.openstack catalog list

openstack image create --public --disk-format qcow2 --container-format bare --file **<<CHANGE TO NAME OF FILE>>** --property key=value **<<CHANGE TO NAME FOR IMAGE>>**

openstack image list

microstack.launch deboan --name debooan --f 2

ssh -i $HOME/.ssh/id_microstack debian@10.20.20.125

- I found sometimes I could not connect to the instance. So stop/restarting it sometimes helped

sudo iptables -t nat -A POSTROUTING -s 10.20.20.1/24 ! -d 10.20.20.1/24 -j MASQUERADE

sudo sysctl net.ipv4.ip_forward=1

sudo snap install lxd

sudo lxd init –auto

sudo snap remove lxd

On Debian Instance

cd ~

curl -sL https://deb.nodesource.com/setup_6.x -o nodesource_setup.sh

nano nodesource_setup.sh

sudo bash nodesource_setup.sh

sudo apt-get install nodejs

sudo apt-get install build-essential

cd ~

nano hello.js

Place in hello.js

```
#!/usr/bin/env nodejs
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(8080, 'localhost');
console.log('Server running at http://localhost:8080/');
```

chmod +x ./hello.js

nvm use 8

nvm install 8

- Use node -v OR npm -v to check versions and if they work

node hello.js

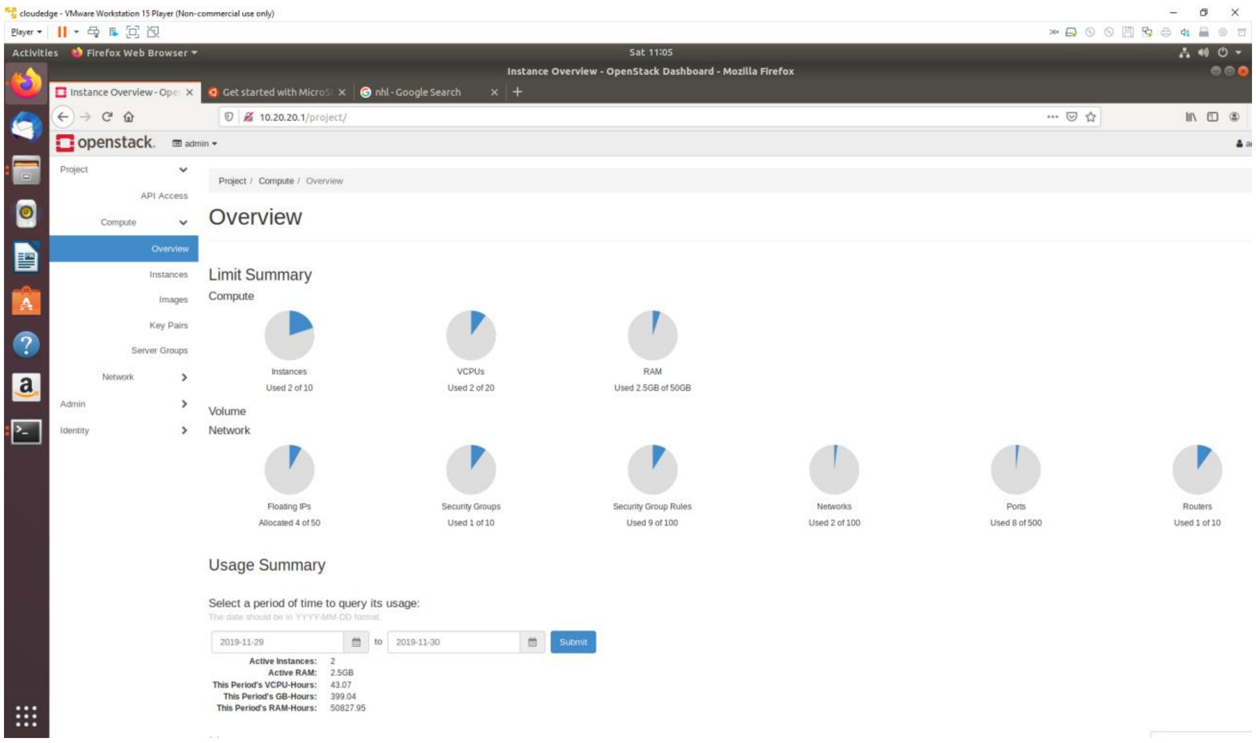
```
Output
Server running at http://localhost:8080/
```

Problems I encountered

- I encountered problems with the installation of the Devstack image, and that is why I transitioned to MicroStack
- When manually trying to associate a floating IP address, there were non available. So I used the CLI command to create an instance and it would set everything up automatically.
- I needed the OpenStack RC file to upload an image to the network. But the GUI command also said that the file was too big to process. So I used the CLI command with the OpenStack RC authentication and it allowed me.
- When using the CLI command to create an instance, by default it choose a micro size instance, so I needed to add an argument to choose a larger size instance
- Sometimes there were problems when trying to connect to instance. Restarting the instance sometimes helped
- The instance had problems connecting to the internet and so I had to create another routing table entry so that it could. Sometimes this worked.
- The node application had problems starting. It is suggested that this was due to compatibility problems. So I installed a previous version of node and npm

Overview

This is the Overview page. This shows a summary of the cloud network that I have created. Showing the number of Instances, Floating IPs, or Routers that I have set up.



Overview

Limit Summary

Compute

Instances	VCPUs	RAM
Used 2 of 10	Used 2 of 20	Used 2.5GB of 50GB

Volume

Network

Floating IPs	Security Groups	Security Group Rules	Networks	Ports	Routers
Allocated 4 of 50	Used 1 of 10	Used 9 of 100	Used 2 of 100	Used 8 of 500	Used 1 of 10

Usage Summary

Select a period of time to query its usage:
The date should be in YYYY-MM-DD format.

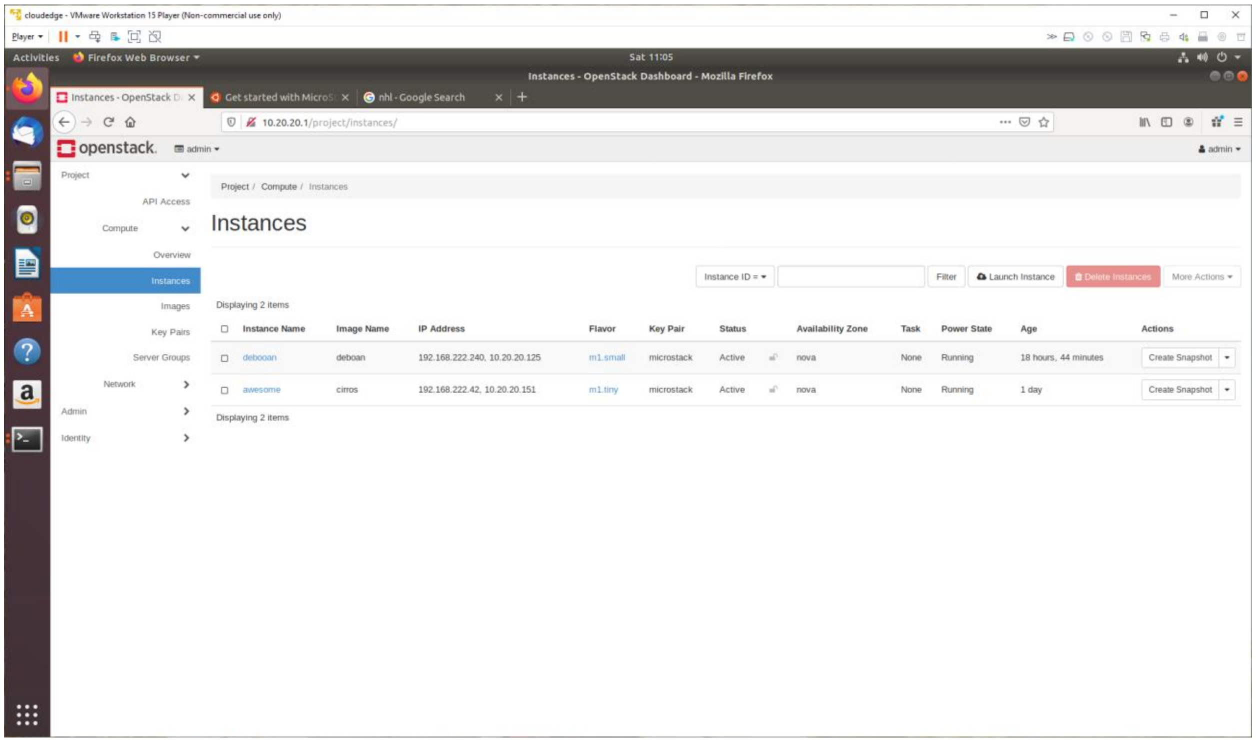
2019-11-29 to 2019-11-30

```
         Active Instances:  2
               Active RAM:  2.5GB
 This Period's VCPU-Hours:  43.07
   This Period's GB-Hours:  399.04
  This Period's RAM-Hours:  50827.95
```

Instances

This image shows the different instances that I have created. The “awesome” instance is a cirros virtual machine. Cirros is a testing OS, that we can use to verify if the cloud network has been set up correctly. It cannot host a website though.

So I had to upload another image to use. I choose Debian. The other instance named “debooan”, is the instance that will host my website.



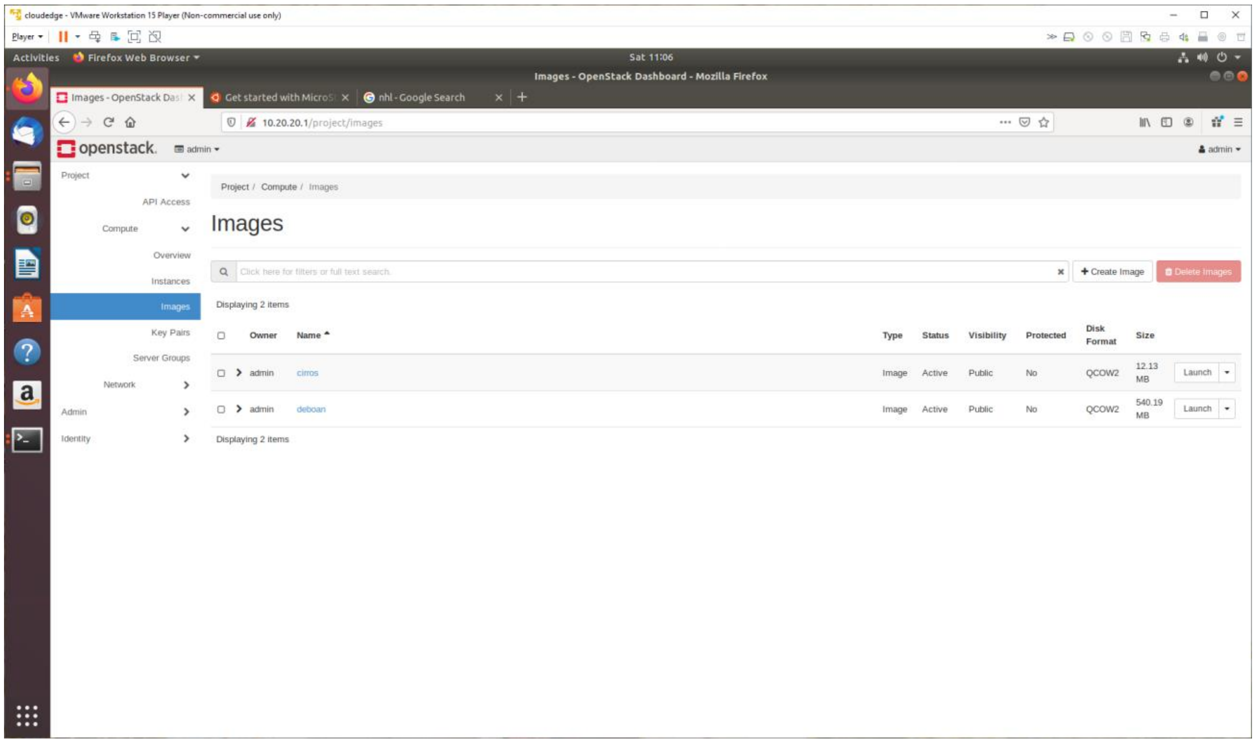
Instances

Displaying 2 items

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
debooan	deboan	192.168.222.240, 10.20.20.125	m1.small	microstack	Active	nova	None	Running	18 hours, 44 minutes	Create Snapshot
awesome	cirros	192.168.222.42, 10.20.20.151	m1.tiny	microstack	Active	nova	None	Running	1 day	Create Snapshot

Displaying 2 items

This image shows all the images that I have in my Openstack environment. There is the Cirros image and the Debian image, which I misspelt as deboan.



Images

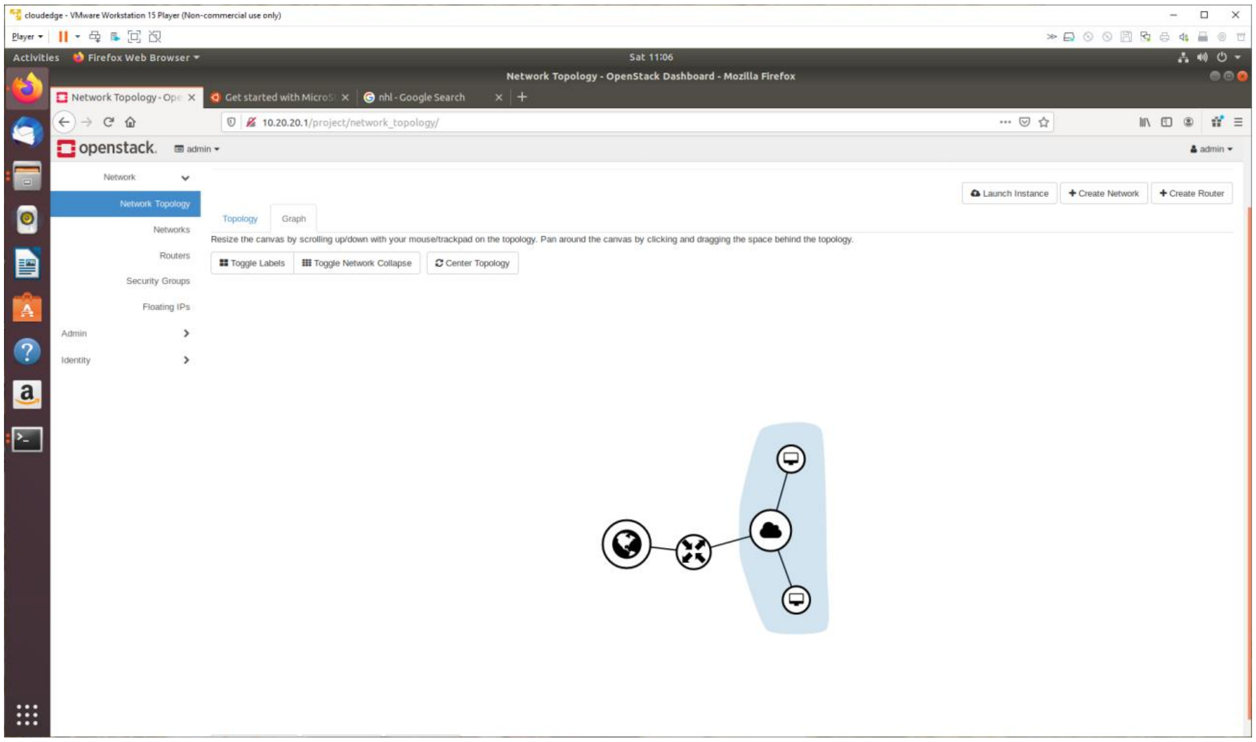
Displaying 2 items

	Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size	
	admin	cirros	Image	Active	Public	No	QCOW2	12.13 MB	Launch
	admin	deboan	Image	Active	Public	No	QCOW2	540.19 MB	Launch

Displaying 2 items

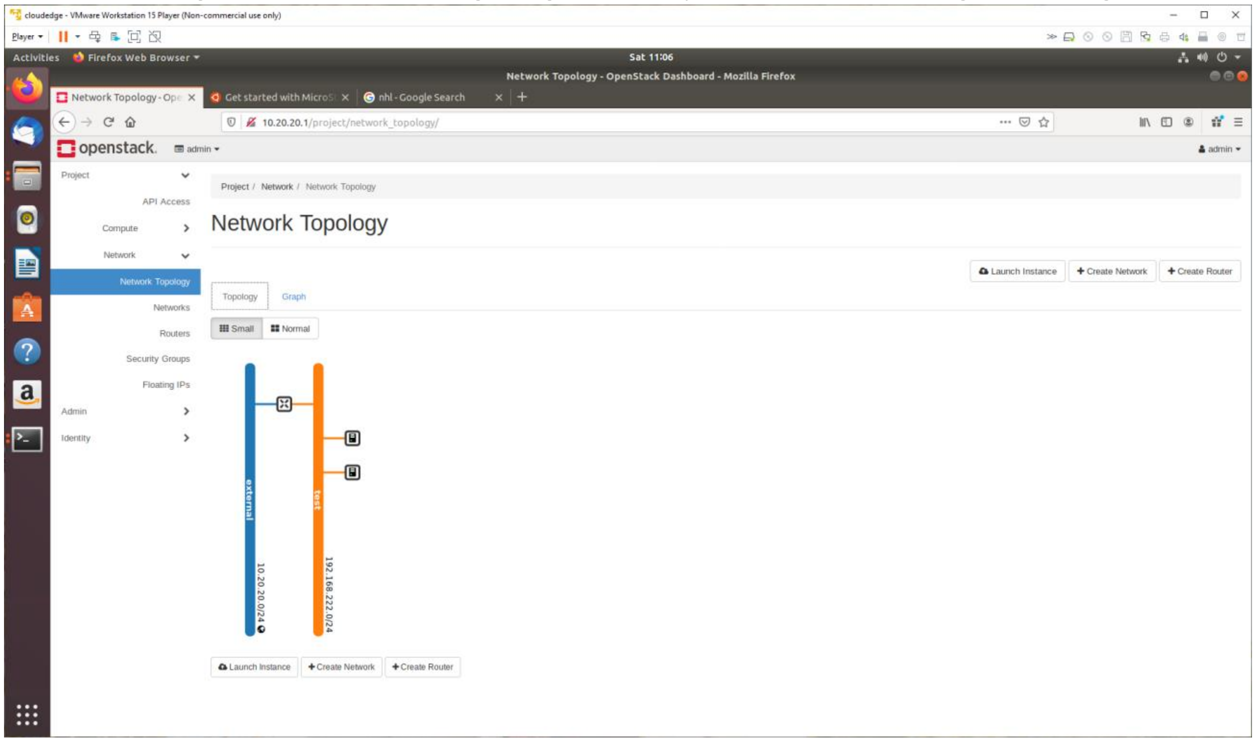
Network Topology graph

This is the network topology graph. It is a visual representation of what the network that we have created looks like. If we hover over each of the items, we can see what they are called, what they are and their associated IP address.



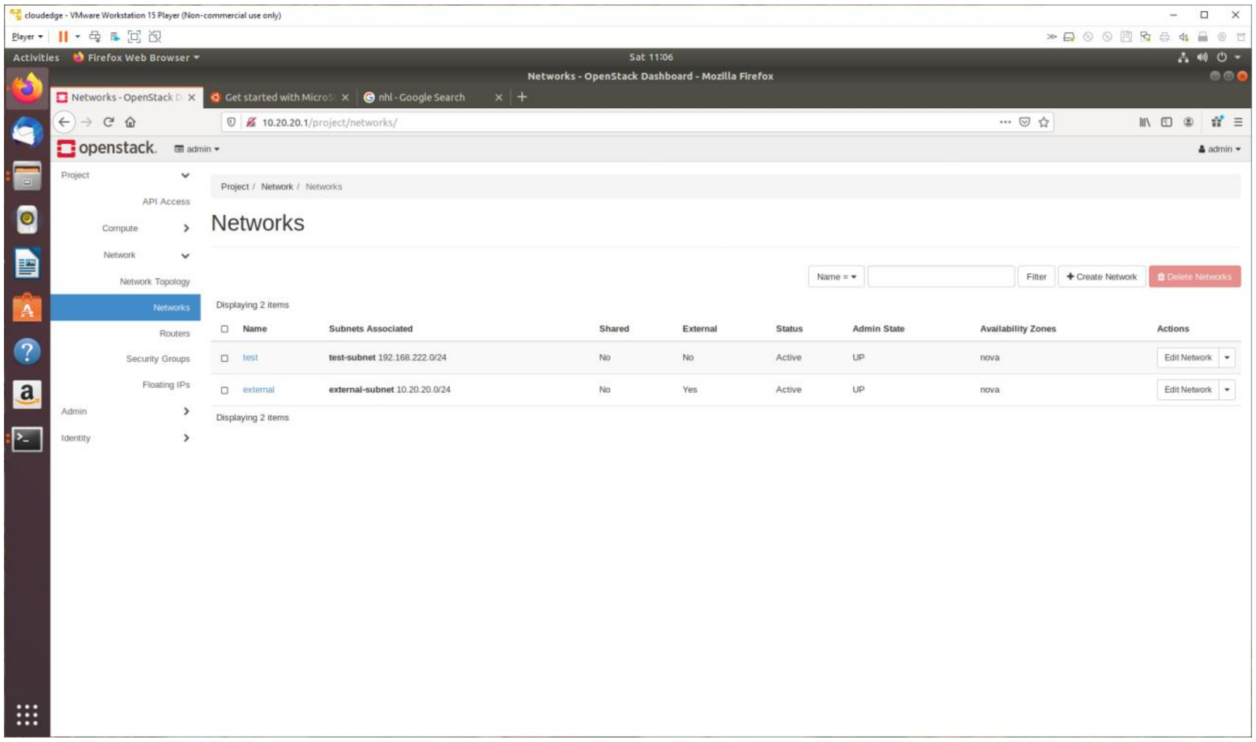
Resize the canvas by scrolling up/down with your mouse/trackpad on the topology. Pan around the canvas by clicking and dragging the space behind the topology.

This second image is a flat network image diagram but represents the same thing as the image above.



Networks

This image shows the networks that we have created. One is the “test” network which is our internal network in the environment. The “external” is the network that associates with the outside world or the edge of the network.



Project / Network / Networks

Networks

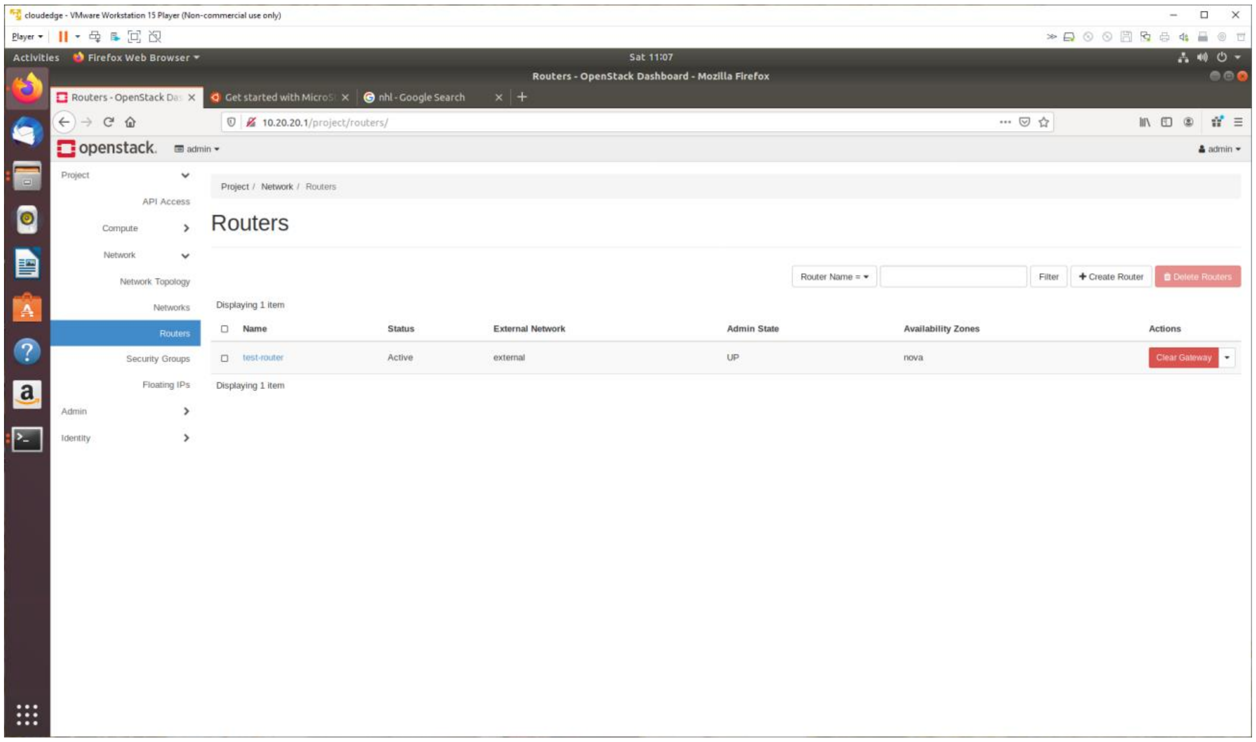
Displaying 2 items

Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
test	**test-subnet** 192.168.222.0/24	No	No	Active	UP	nova	Edit Network
external	**external-subnet** 10.20.20.0/24	No	Yes	Active	UP	nova	Edit Network

Displaying 2 items

Routers

Here you can see the test router that I have created for my network. This router helps connect the different networks that I have created.



Routers

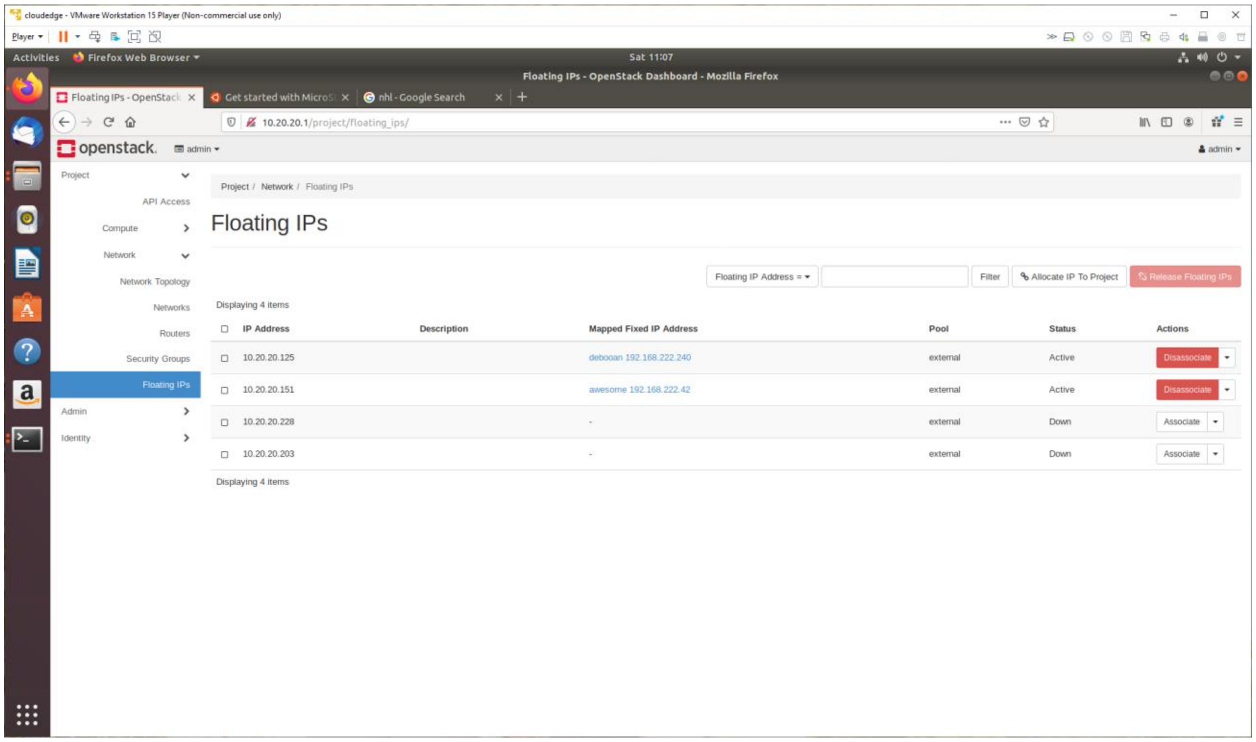
Displaying 1 item

Name	Status	External Network	Admin State	Availability Zones	Actions
test-router	Active	external	UP	nova	Clear Gateway

Displaying 1 item

Floating IPs

The image below shows the various floating IP address that I have created to associate with the different instances and items in my network.



Floating IPs

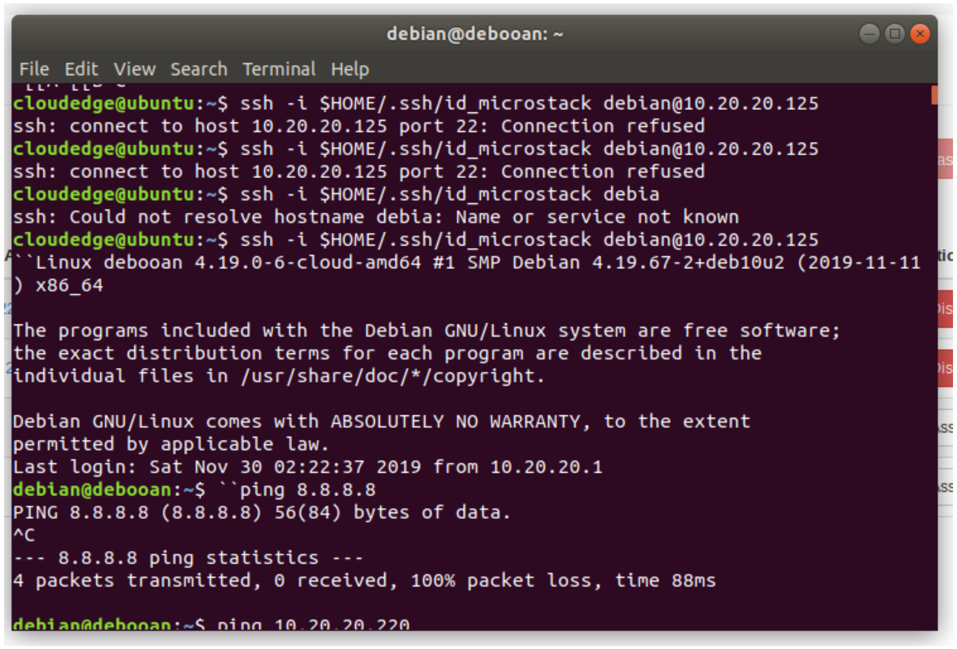
Displaying 4 items

IP Address	Description	Mapped Fixed IP Address	Pool	Status	Actions
10.20.20.125		debooan 192.168.222.240	external	Active	Disassociate
10.20.20.151		awesome 192.168.222.42	external	Active	Disassociate
10.20.20.228		-	external	Down	Associate
10.20.20.203		-	external	Down	Associate

Displaying 4 items

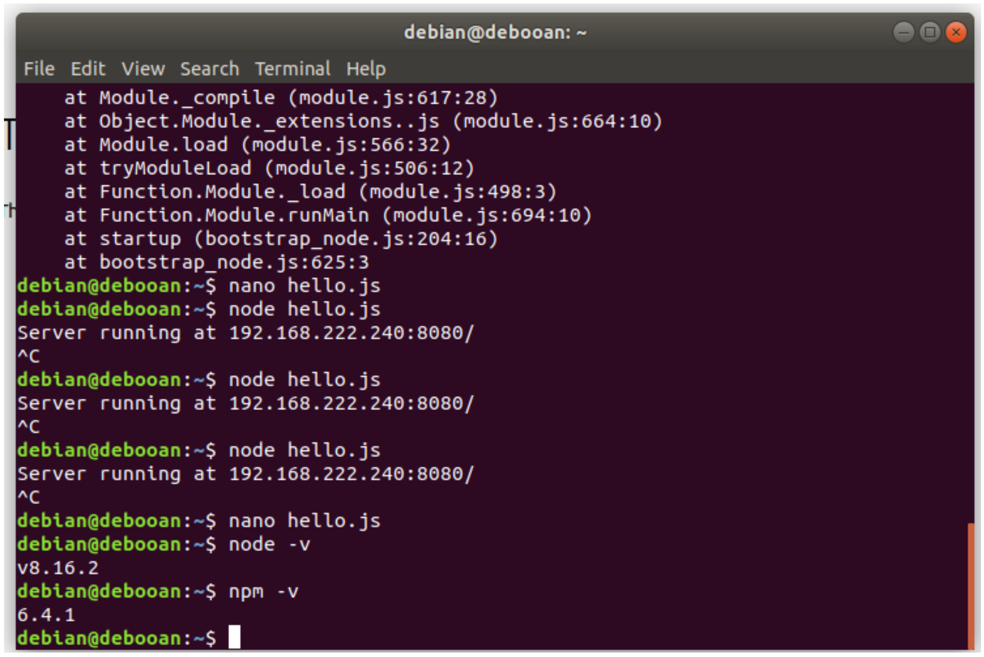
The Website

This image shows me using an SSH connetion to connect to my debian instance.



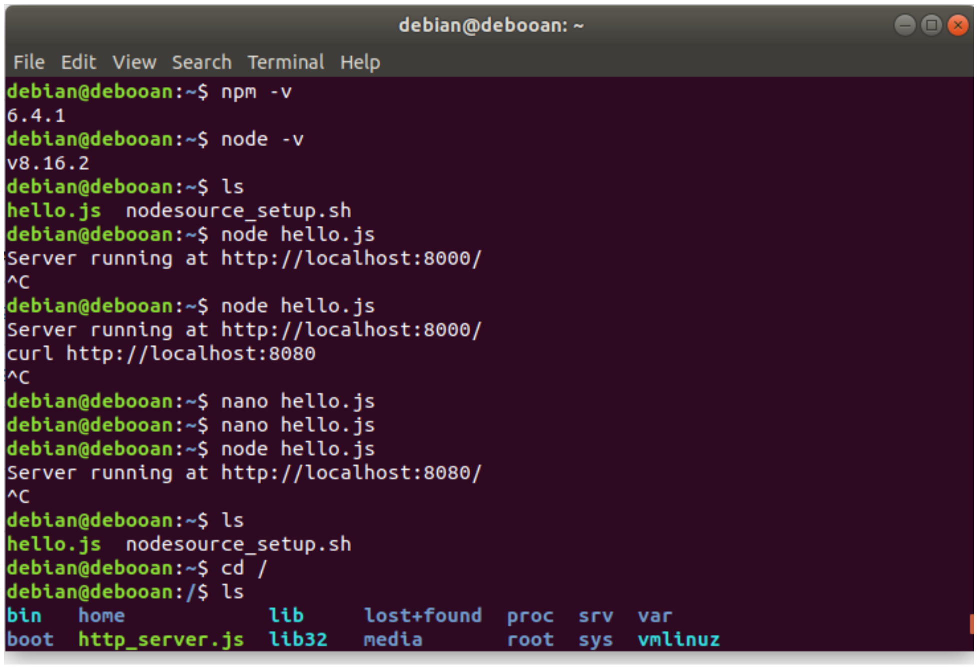
```
cloudedge@ubuntu:~$ ssh -i $HOME/.ssh/id_microstack debian@10.20.20.125
ssh: connect to host 10.20.20.125 port 22: Connection refused
cloudedge@ubuntu:~$ ssh -i $HOME/.ssh/id_microstack debian@10.20.20.125
ssh: connect to host 10.20.20.125 port 22: Connection refused
cloudedge@ubuntu:~$ ssh -i $HOME/.ssh/id_microstack debia
ssh: Could not resolve hostname debia: Name or service not known
cloudedge@ubuntu:~$ ssh -i $HOME/.ssh/id_microstack debian@10.20.20.125
``Linux debooan 4.19.0-6-cloud-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11
) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Nov 30 02:22:37 2019 from 10.20.20.1
debian@debooan:~$ ``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 ---
4 packets transmitted, 0 received, 100% packet loss, time 88ms

debian@debooan:~$ ping 10.20.20.220
```

Here is an image showing that I have installed nodejs and npm onto the instance



```
    at Module._compile (module.js:617:28)
    at Object.Module._extensions..js (module.js:664:10)
    at Module.load (module.js:566:32)
    at tryModuleLoad (module.js:506:12)
    at Function.Module._load (module.js:498:3)
    at Function.Module.runMain (module.js:694:10)
    at startup (bootstrap_node.js:204:16)
    at bootstrap_node.js:625:3
debian@debooan:~$ nano hello.js
debian@debooan:~$ node hello.js
Server running at 192.168.222.240:8080/
^C
debian@debooan:~$ node hello.js
Server running at 192.168.222.240:8080/
^C
debian@debooan:~$ node hello.js
Server running at 192.168.222.240:8080/
^C
debian@debooan:~$ nano hello.js
debian@debooan:~$ node -v
v8.16.2
debian@debooan:~$ npm -v
6.4.1
debian@debooan:~$ 
```

This image shows me initiating the instance with the “node hello.js” command



```
debian@debooan:~$ npm -v
6.4.1
debian@debooan:~$ node -v
v8.16.2
debian@debooan:~$ ls
hello.js  nodesource_setup.sh
debian@debooan:~$ node hello.js
Server running at http://localhost:8000/
^C
debian@debooan:~$ node hello.js
Server running at http://localhost:8000/
curl http://localhost:8080
^C
debian@debooan:~$ nano hello.js
debian@debooan:~$ nano hello.js
debian@debooan:~$ node hello.js
Server running at http://localhost:8080/
^C
debian@debooan:~$ ls
hello.js  nodesource_setup.sh
debian@debooan:~$ cd /
debian@debooan:/$ ls
bin   home            lib     lost+found  proc  srv  var
boot  http_server.js  lib32   media       root  sys  vmlinuz
```

This is as far as I got. I was able to create an OpenStack environment and create my network and an instance. I was able to download node.js on my instance and I was able to create a .js file that had the code to initialize a basic nodejs website. But I was not able to view my website. I was only able to ssh into the instance and make changes to it. But the web hosting was not working.

References

https://opendev.org/x/microstack

https://tutorials.ubuntu.com/tutorial/microstack-get-started#0

https://docs.openstack.org/ocata/user-guide/common/cli-manage-images.html

https://stackoverflow.com/questions/42844649/missing-value-auth-url-required-for-auth-plugin-password

https://wiki.openstack.org/wiki/Horizon/Logs

https://microstack.run/

https://edwardsamuel.wordpress.com/2014/10/25/tutorial-creating-openstack-instance-in-trystack/

https://www.linuxtechi.com/upload-download-cloud-images-in-openstack/

lxdhttps://bugs.launchpad.net/microstack/+bug/1812415

https://www.digitalocean.com/community/tutorials/how-to-set-up-and-use-lxd-on-ubuntu-16-04

https://nodejs.org/en/download/package-manager/