

# OpenStack Project Report

## Introduction

---

The aim of the project is to achieve the following goals on Openstack single-node setup.

- Get familiar with OpenStack Horizon
- Provision new instances with the OpenStack Dashboard
- Connect to the newly provisioned VM using VNC web console
- Create a shared network and connect 2 instances over it
- Create a router to connect the shared network to public network
- Terminate the newly provisioned VM

## Motivation

---

The motivation of our project is to explore concepts of virtualization studied in classroom and apply them using the freely available open-source project - Openstack. This will help us understand complex theory of hypervisors, IaaS, KVM, QEMU, Libvirt and Openstack practically in a sandboxed playground.

## Details of Project

---

### 1. Hardware

Processor	6 cores
RAM	16 GB
Storage	Around 100GB to spare

## 2. Software

OpenStack is a cloud computing platform enabling the provision of IaaS services. OpenStack constitutes resources such as compute, storage and network resources. Compute resources are allocated in the form of virtual machines (aka instances). Storage resources are allocated in the form of virtual disks (aka volumes). Network resources are allocated in the form of virtual switches, routers and subnets for instance.








## 3. Necessary Components

- Network interface cards on parent OS:

Name	[00000010] Intel(R) Wireless-AC 9560 160MHz
Adapter Type	Ethernet 802.3
Product Type	Intel(R) Wireless-AC 9560 160MHz
Installed	Yes
PNP Device ID	PCI\VEN_8086&DEV_A370&SUBSYS_00348086&REV_10\3&...

Name	[00000011] Realtek PCIe GbE Family Controller
Adapter Type	Ethernet 802.3
Product Type	Realtek PCIe GbE Family Controller
Installed	Yes
PNP Device ID	PCI\VEN_10EC&DEV_8168&SUBSYS_38CD17AA&REV_15\0...

- Ubuntu Live Server 20.04.4 (or any other OS from the below list):

Name	Type	Size
 CentOS-7-x86_64-Minimal-1511	Disc Image File	6,17,472 KB
 Fedora-Workstation-Live-x86_64-35-1.2	Disc Image File	19,62,240 ...
 ubuntu-16.04.6-server-amd64	Disc Image File	8,93,952 KB
 ubuntu-16.04.7-server-amd64	Disc Image File	9,01,120 KB
 ubuntu-18.04.6-live-server-amd64	Disc Image File	9,92,256 KB
 ubuntu-20.04.3-desktop-amd64	Disc Image File	29,99,936 ...
 ubuntu-20.04.4-live-server-amd64	Disc Image File	13,00,480 ...

# Implementation

Below is the sub-sections:

1. Setting up the environment
2. Setting up a shared network with router
3. Configure 2 hosts on the network
4. Pinging the hosts from each other

## 1. Setting up the environment

Following the official Devstack documentation, we first created a stack user with sudo access.

```
sudo useradd -s /bin/bash -d /opt/stack -m stack
```

Cloning latest stable version of OpenStack from DevStack repository:

Series	Status	Initial Release Date	Next Phase	EOL Date
<a href="#">Zed</a>	<a href="#">Development</a>	2022-10-05 <i>estimated (schedule)</i>	<a href="#">Maintained</a> <i>estimated 2022-10-05</i>	
<a href="#">Yoga</a>	<a href="#">Maintained</a>	2022-03-30	<a href="#">Extended Maintenance</a> <i>estimated 2023-09-30</i>	

After setting up passwords for admin and databases, we installed Openstack using:

```
./stack.sh
```

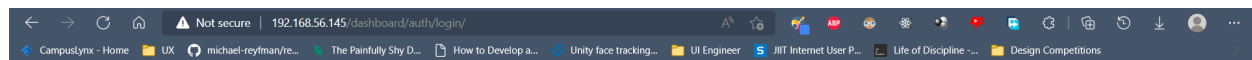
```
git_timed      684
apt-get-update 20
test_with_retry 3
async_wait     99
osc            232
-----
Unaccounted time  85
*****
Total runtime    2710
*****
=====
Async summary
=====
Time spent in the background minus waits: 368 sec
Elapsed time: 2710 sec
Time if we did everything serially: 3078 sec
Speedup: 1.15579

This is your host IP address: 192.168.56.139
This is your host IPv6 address: ::1
Horizon is now available at http://192.168.56.139/dashboard
Keystone is serving at http://192.168.56.139/identity/
The default users are: admin and demo
The password: admin

Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: v0.9.4
Change: 194790df855155a9c304e2cccd70b956392e208bc Merge "Clean up unified limits configuration for no
va and glance" 2022-03-10 19:23:28 +0000
OS Version: Ubuntu 20.04 focal
stack@openstack:~/devstack$ _
```

Successful Openstack installation

A screenshot of the OpenStack Horizon login form. The form has a white background with a red OpenStack logo at the top. Below the logo is the text "openstack.". Underneath is the heading "Log in". There are two input fields: "User Name" and "Password". The "Password" field has a toggle icon for visibility. At the bottom right of the form is a blue "Sign In" button.

Horizon on private IP 192.169.56.145

### ▼ Transferring rc file:

Tried to run an openstack client command and received the following error:

Missing value auth-url required for auth plugin password

We had not setup our openstack cli environment variables (openrc variables) for the project. It can be downloaded from the Web GUI at top right. Then execute:

```
$ bash admin-openrc.sh
$ source admin-openrc.sh
```

To transfer this file, we can use FTP server, SSH, or manually type the file into our Openstack VM. Due to limitations of network setup in our case, we hosted it on Github and used `git` with raw feature to clone it into the VM.

## 2. Setting up a shared network with router

We can create all the network setup through the Horizon dashboard using the user interface.

### Create Router ✕

**Router Name**

☒ **Enable Admin State** ?

**External Network**

public ▼

☒ **Enable SNAT**

**Availability Zone Hints** ?

**Description:**

Creates a router with specified parameters.

Enable SNAT will only have an effect if an external network is set.

Cancel

Create Router

## Add Interface



### Subnet \*

shared: 192.168.233.0/24 (shared-subnet) ▼

### IP Address (optional) ?

### Description:

You can connect a specified subnet to the router.

If you don't specify an IP address here, the gateway's IP address of the selected subnet will be used as the IP address of the newly created interface of the router. If the gateway's IP address is in use, you must use a different address which belongs to the selected subnet.

Cancel

Submit

## 3. Configure 2 hosts on the network

### Launch Instance



#### Details

Source \*

Flavor \*

Networks \*

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

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.

#### Project Name

admin

#### Instance Name \*

instance\_1

#### Description

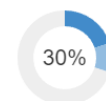
#### Availability Zone

nova ▼

#### Count \*

1

Total Instances  
(10 Max)



■ 2 Current Usage  
■ 1 Added  
■ 7 Remaining

✕ Cancel

< Back

Next >

Launch Instance

Resource allocation to both hosts:

Name	Updated	Size	Format	Visibility	
➤ cirros-0.5.2-x86_64-disk	5/15/22 11:00 PM	15.55 MB	QCOW2	Public	↓

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
➤ m1.nano	1	128 MB	1 GB	1 GB	0 GB	Yes	↓

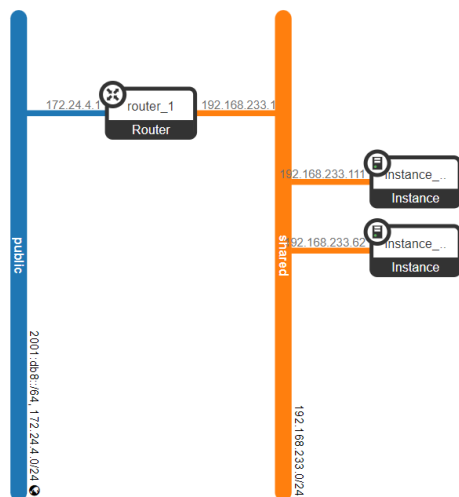
Network	Subnets Associated	Shared	Admin State	Status	
↕ 1 ➤ shared	shared-subnet	Yes	Up	Active	↓

Active instances created:

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
<input type="checkbox"/>	instance_2	cirros-0.5.2-x86_64-disk	192.168.233.111	m1.nano	-	Active	nova	None	Running	1 hour, 25 minutes	Create Snapshot
<input type="checkbox"/>	instance_1	cirros-0.5.2-x86_64-disk	192.168.233.62	m1.nano	-	Active	nova	None	Running	1 hour, 25 minutes	Create Snapshot

Displaying 2 items

Final network topology:



## 4. Pinging the hosts from each other

Note the IP addresses of the instances: `192.169.233.62` and `192.168.233.111`.

```
Connected to QEMU (instance-00000002) Send CtrlAltDel
$ ping 192.168.233.62
PING 192.168.233.62 (192.168.233.62): 56 data bytes
64 bytes from 192.168.233.62: seq=0 ttl=64 time=19.870 ms
64 bytes from 192.168.233.62: seq=1 ttl=64 time=2.183 ms
64 bytes from 192.168.233.62: seq=2 ttl=64 time=1.170 ms
64 bytes from 192.168.233.62: seq=3 ttl=64 time=1.394 ms
64 bytes from 192.168.233.62: seq=4 ttl=64 time=3.526 ms
64 bytes from 192.168.233.62: seq=5 ttl=64 time=1.212 ms
64 bytes from 192.168.233.62: seq=6 ttl=64 time=1.170 ms
64 bytes from 192.168.233.62: seq=7 ttl=64 time=1.216 ms
```

```
Connected to QEMU (instance-00000001) Send CtrlAltDel
$ ping 192.168.233.111
PING 192.168.233.111 (192.168.233.111): 56 data bytes
64 bytes from 192.168.233.111: seq=0 ttl=64 time=5.482 ms
64 bytes from 192.168.233.111: seq=1 ttl=64 time=2.333 ms
64 bytes from 192.168.233.111: seq=2 ttl=64 time=3.119 ms
64 bytes from 192.168.233.111: seq=3 ttl=64 time=2.491 ms
64 bytes from 192.168.233.111: seq=4 ttl=64 time=1.677 ms
64 bytes from 192.168.233.111: seq=5 ttl=64 time=1.750 ms
64 bytes from 192.168.233.111: seq=6 ttl=64 time=1.084 ms
-
```



# References

---

- [1] Vinoth Kumar Selvaraj, “Openstack Bootcamp”, November 2017
- [2] Openstack Documentation, 2021, “Configure live migrations”,  
<<https://docs.openstack.org/nova/pike/admin/configuring-migrations.html>>
- [3] Openstack Documentation, 2021, “Live-migrate instances”  
<<https://docs.openstack.org/nova/pike/admin/live-migration-usage.html>>