Submitted By: Annie Jain

Sap Id: 500083967

Roll No: R214220179

Batch: B3 Hons

CLOUD APPLICATION DEVELOPMENT

OPENSTACK LAB EXPERIMENT – 06

OBJECTIVE: Deploying and Managing Virtual Machines with Different Operating Systems in OpenStack.

Introduction:

OpenStack is a cloud operating system that provides a range of services to help organizations build and manage private and public clouds. One of the primary services provided by OpenStack is the ability to deploy and manage virtual machines (VMs) running on a variety of different operating systems (Oss). In this lab report, we will explore how to deploy and manage VMs with different Oss using OpenStack.

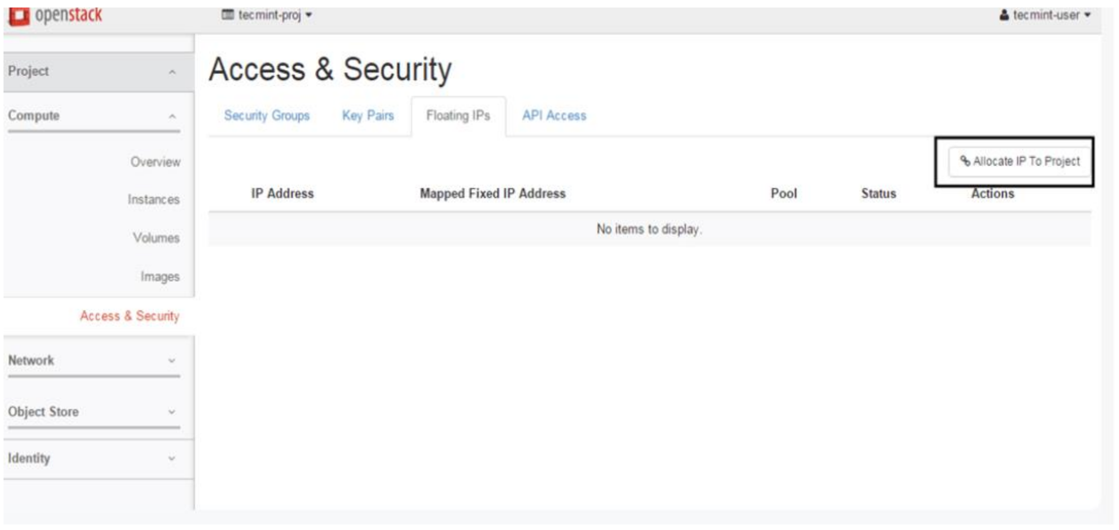
Prerequisites:

To follow along with this lab, you will need access to an OpenStack cloud and an account with sufficient permissions to create and manage VMs. Additionally, you will need an SSH client installed on your local machine to connect to the VMs.

Deploying VMs with Different OSs:

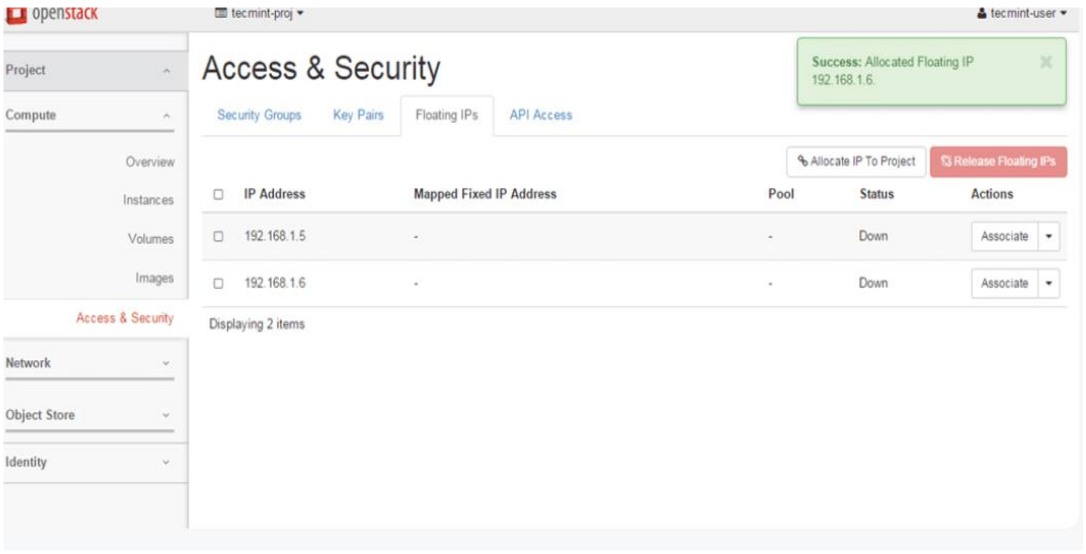
1. Launch an Ubuntu VM:
    - Log in to the OpenStack dashboard and navigate to the "Instances" tab.
    - Click "Launch Instance" and select "Ubuntu" as the image source.
    - Choose a flavor and specify any other required settings, such as security groups and SSH key pairs.
    - Click "Launch" to create the VM.

2. Launch a CentOS VM:
    - Follow the same steps as for the Ubuntu VM, but select "CentOS" as the image source instead.
3. Launch a Windows Server VM:
    - Launching a Windows Server VM requires a different process than for Linux-based VMs. You will need to upload a Windows Server image to your OpenStack cloud and configure the necessary drivers and settings.
    - First, download a Windows Server image from the Microsoft website and convert it to the qcow2 format using the "qemu-img" tool.
    - Upload the image to your OpenStack cloud using the "glance" command-line tool or the dashboard.
    - Create a new flavor for the VM with appropriate specifications for running Windows Server.
    - Launch the VM and configure the necessary drivers and settings, such as enabling Remote Desktop access and installing any required software.
4. Connect to the VMs:
    - Once the VMs are launched, you can connect to them using an SSH client.
    - For Linux-based VMs, you can connect using the IP address assigned to the VM.
    - For the Windows Server VM, you will need to connect using Remote Desktop.
    - You can also configure network settings to allow external access to the VMs, such as opening ports for HTTP or HTTPS traffic.



Access & Security

IP Address	Mapped Fixed IP Address	Pool	Status	Actions
No items to display.				



Access & Security

Success: Allocated Floating IP 192.168.1.6.

Security Groups | Key Pairs | Floating IPs | API Access

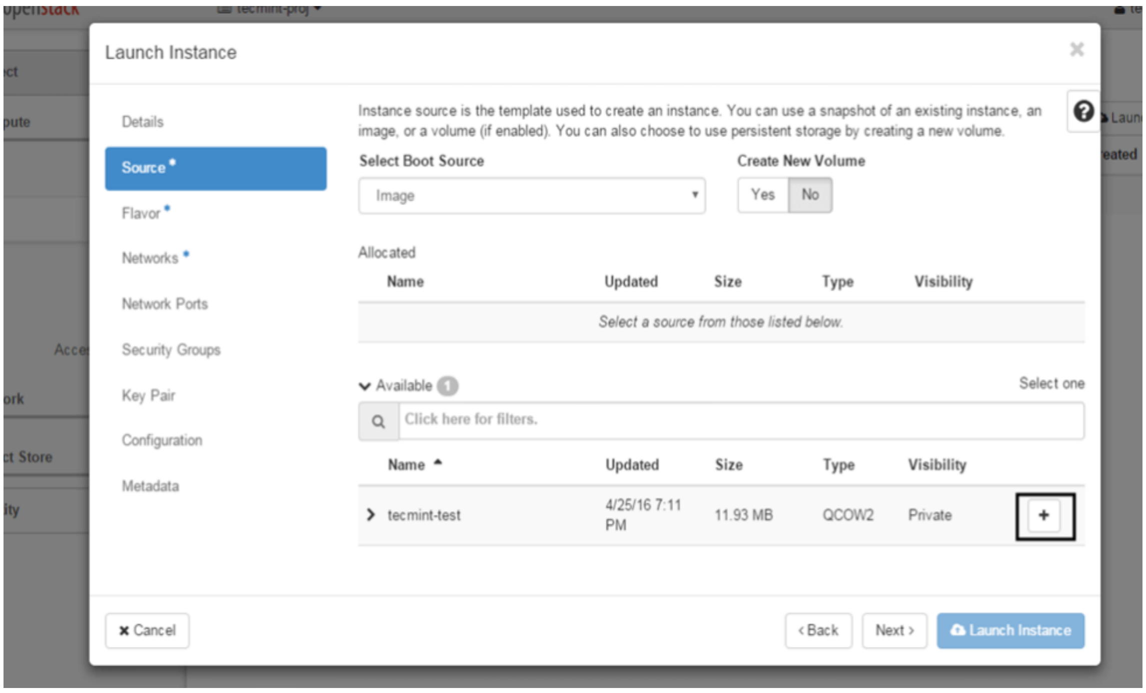
Allocate IP To Project | Release Floating IPs

	IP Address	Mapped Fixed IP Address	Pool	Status	Actions
☐	192.168.1.5	-	-	Down	Associate
☐	192.168.1.6	-	-	Down	Associate

Displaying 2 items

Managing VMs with Different OSs:

1. Managing Linux-based VMs:
   - You can manage Linux-based VMs using SSH or the OpenStack dashboard.
   - Using SSH, you can perform tasks such as updating packages, installing software, and configuring network settings.
   - Using the dashboard, you can view and manage the VM's status, view console output, and manage security groups and networking settings.
2. Managing Windows Server VMs:
   - You can manage Windows Server VMs using Remote Desktop or the OpenStack dashboard.
   - Using Remote Desktop, you can perform tasks such as installing software, configuring network settings, and managing users and permissions.
   - Using the dashboard, you can view and manage the VM's status, view console output, and manage security groups and networking settings.



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]

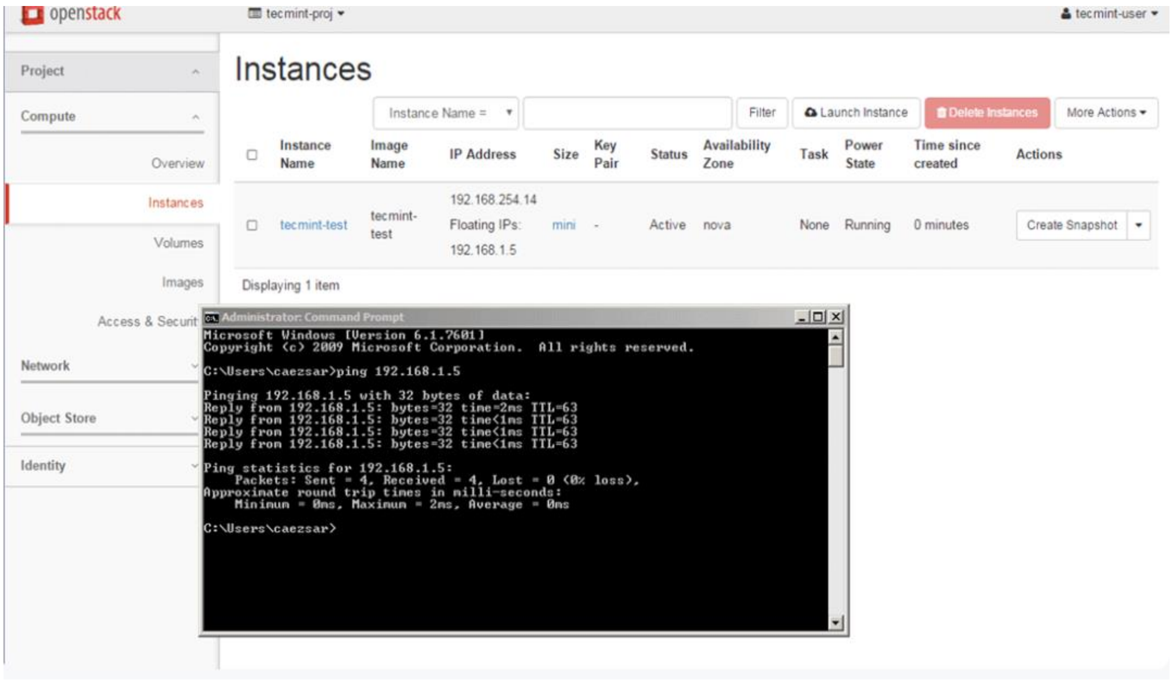
Allocated

Name	Updated	Size	Type	Visibility
Select a source from those listed below.				

Available 1 — Select one

Click here for filters.

Name	Updated	Size	Type	Visibility	
tecmint-test	4/25/16 7:11 PM	11.93 MB	QCOW2	Private	+



Instances

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
☐	tecmint-test	tecmint-test	192.168.254.14 Floating IPs: 192.168.1.5	mini	-	Active	nova	None	Running	0 minutes	Create Snapshot

Displaying 1 item

Administrator: Command Prompt

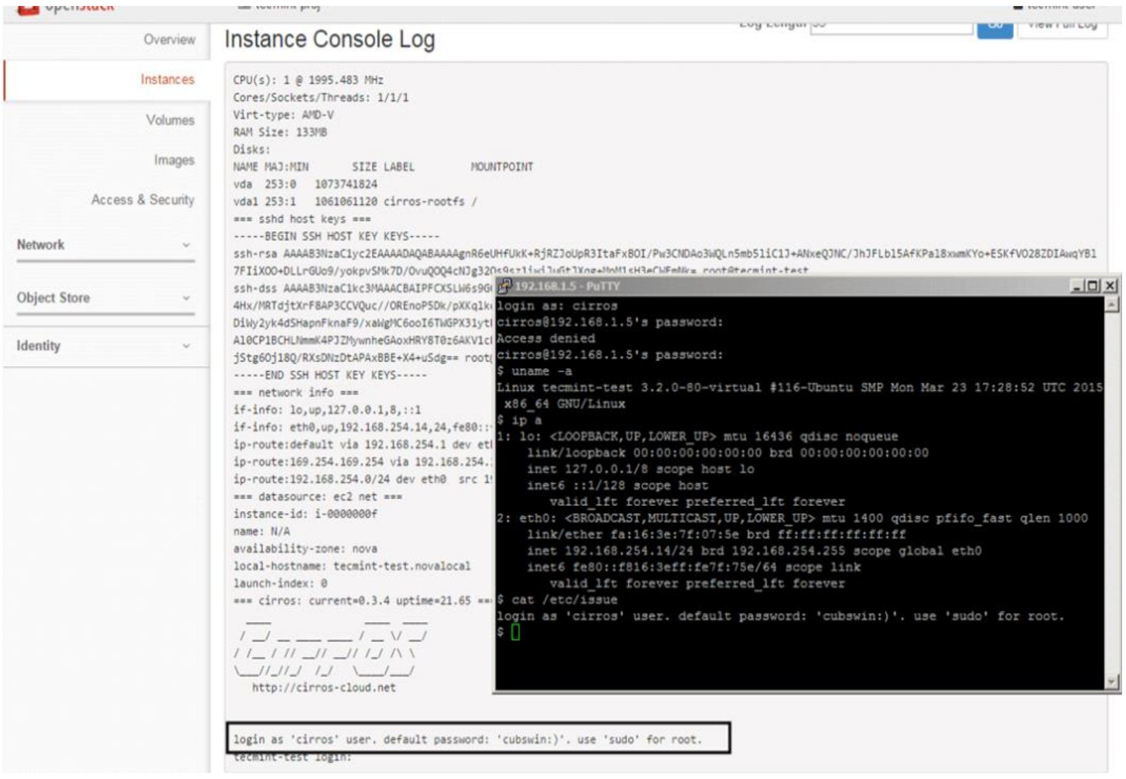
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\caezsar>ping 192.168.1.5

Pinging 192.168.1.5 with 32 bytes of data:
Reply from 192.168.1.5: bytes=32 time=2ms TTL=63
Reply from 192.168.1.5: bytes=32 time<1ms TTL=63
Reply from 192.168.1.5: bytes=32 time<1ms TTL=63
Reply from 192.168.1.5: bytes=32 time<1ms TTL=63

Ping statistics for 192.168.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\Users\caezsar>
```

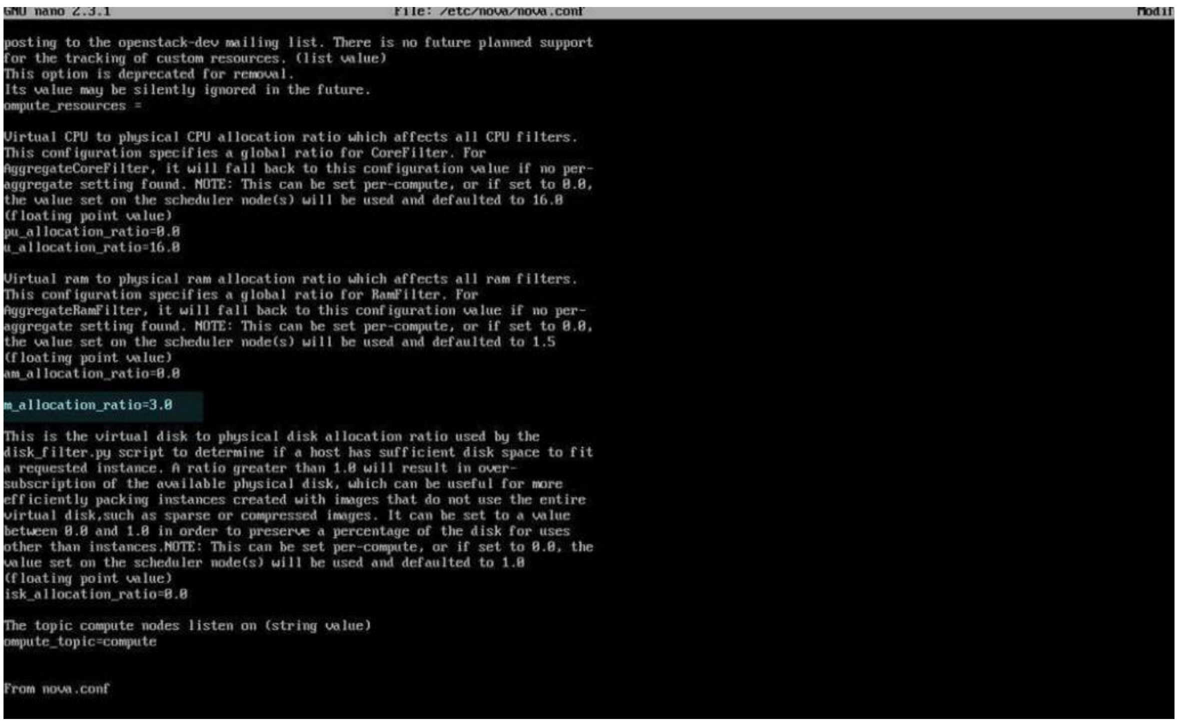


Instance Console Log

```
CPU(s): 1 @ 1995.483 MHz
Cores/Sockets/Threads: 1/1/1
Virt-type: AMD-V
RAM Size: 133MB
Disks:
NAME MAJ:MIN       SIZE LABEL         MOUNTPOINT
vda  253:0   1073741824
vda1 253:1   1061061120 cirros-rootfs /
=== sshd host keys ===
-----BEGIN SSH HOST KEY KEYS-----
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgnR6eUHfUkK+RjRZJoUpR3ItaFxBOI/Pw3CNDAo3WQLn5mb51iC1J+ANxeQJNC/JhJFLb15AfKPa18xwmKYo+ESKfVO28ZDIAwqYB1
7FIiXOO+DLLrGUo9/yokpvSMk7D/OvuQOQ4cNJg320s9sz1iwi1uGtJXog+MnM1sH3eCWEmNk= root@tecmint-test
ssh-dss AAAAB3NzaC1kc3MAAACBAIPFCXSLW6s9G
4Hx/MRTdjtXrF8AP3CCVQuc//OREnoP5Dk/pXKqlk
DiWy2yk4dSHapnFknaF9/xaWgMC6ooI6TWGPX31yt
A10CP1BCHLNmmK4PJZMywnheGAoxHRY8T0z6AKV1c
jStg6Oj18Q/RXsDNzDtAPAxBBE+X4+uSdg== root
-----END SSH HOST KEY KEYS-----
=== network info ===
if-info: lo,up,127.0.0.1,8,::1
if-info: eth0,up,192.168.254.14,24,fe80::
ip-route:default via 192.168.254.1 dev et
ip-route:169.254.169.254 via 192.168.254.
ip-route:192.168.254.0/24 dev eth0  src 1
=== datasource: ec2 net ===
instance-id: i-0000000f
name: N/A
availability-zone: nova
local-hostname: tecmint-test.novalocal
launch-index: 0
=== cirros: current=0.3.4 uptime=21.65 ==
  ____               ____  ____
 / __/ __ ____ ____ / __ \/ __/
/ /__ / // __// __// /_/ /\ \
\___//_//_/  /_/   \____/___/
   http://cirros-cloud.net


login as 'cirros' user. default password: 'cubswin:)'. use 'sudo' for root.
tecmint-test login:
```

```
login as: cirros
cirros@192.168.1.5's password:
Access denied
cirros@192.168.1.5's password:
$ uname -a
Linux tecmint-test 3.2.0-80-virtual #116-Ubuntu SMP Mon Mar 23 17:28:52 UTC 2015
 x86_64 GNU/Linux
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1400 qdisc pfifo_fast qlen 1000
    link/ether fa:16:3e:7f:07:5e brd ff:ff:ff:ff:ff:ff
    inet 192.168.254.14/24 brd 192.168.254.255 scope global eth0
    inet6 fe80::f816:3eff:fe7f:75e/64 scope link
       valid_lft forever preferred_lft forever
$ cat /etc/issue
login as 'cirros' user. default password: 'cubswin:)'. use 'sudo' for root.
$
```



```
GNU nano 2.3.1                    File: /etc/nova/nova.conf

posting to the openstack-dev mailing list. There is no future planned support
for the tracking of custom resources. (list value)
This option is deprecated for removal.
Its value may be silently ignored in the future.
ompute_resources =

Virtual CPU to physical CPU allocation ratio which affects all CPU filters.
This configuration specifies a global ratio for CoreFilter. For
AggregateCoreFilter, it will fall back to this configuration value if no per-
aggregate setting found. NOTE: This can be set per-compute, or if set to 0.0,
the value set on the scheduler node(s) will be used and defaulted to 16.0
(floating point value)
pu_allocation_ratio=0.0
u_allocation_ratio=16.0

Virtual ram to physical ram allocation ratio which affects all ram filters.
This configuration specifies a global ratio for RamFilter. For
AggregateRamFilter, it will fall back to this configuration value if no per-
aggregate setting found. NOTE: This can be set per-compute, or if set to 0.0,
the value set on the scheduler node(s) will be used and defaulted to 1.5
(floating point value)
am_allocation_ratio=0.0

m_allocation_ratio=3.0

This is the virtual disk to physical disk allocation ratio used by the
disk_filter.py script to determine if a host has sufficient disk space to fit
a requested instance. A ratio greater than 1.0 will result in over-
subscription of the available physical disk, which can be useful for more
efficiently packing instances created with images that do not use the entire
virtual disk,such as sparse or compressed images. It can be set to a value
between 0.0 and 1.0 in order to preserve a percentage of the disk for uses
other than instances.NOTE: This can be set per-compute, or if set to 0.0, the
value set on the scheduler node(s) will be used and defaulted to 1.0
(floating point value)
isk_allocation_ratio=0.0

The topic compute nodes listen on (string value)
ompute_topic=compute


From nova.conf
```

Conclusion:

Deploying and managing VMs with different OSs in OpenStack is a straightforward process that can provide significant flexibility and cost savings for organizations. By using OpenStack to manage VMs, organizations can centralize their cloud infrastructure and easily deploy and manage VMs with different OSs, allowing them to meet a wide range of computing needs.