Concordia Institute of Information Systems Engineering (CIISE)

INSE6620: Cloud Computing Security and Privacy
Final Project Report

Submitted to

Professor Dr. Lingyu Wang

Submitted by:

Name	Student ID
Dhruv Haribhakti	40172725
Mahima Shukla	40206690
Tonmoy Banik	40185216
Aashika Lakhani	40198282
B M Fahmid Jahur Shuvo	40189484
Md Yeasin Arafat	40181574
Katha Sengupta	40202955
Simanta Sen	40187190

1. INTRODUCTION

Cloud computing is the complete process of performing computation as well as delivering, storing, accessing data through internet. The resources can be as networks, servers, storage, applications, and services. The cloud service is being provided by three service models by the cloud providers – IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). IaaS is the service model where the cloud provider offers the complete computing resources for the computational, storing data purposes whereas PaaS offers the platform and SaaS offers the software service. [1]

Cloud computing is getting popular day by day. Some of the most popular cloud providers are Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP). AWS has 34% of the market share whereas Microsoft and GCP as 22% and 9.5% respectively. [2] We intended to perform the implementation of a small virtual network deploying a security tool inside the network using OpenStack. OpenStack is an open-source cloud computing platform that allows to manage storage, computing, and network resources.

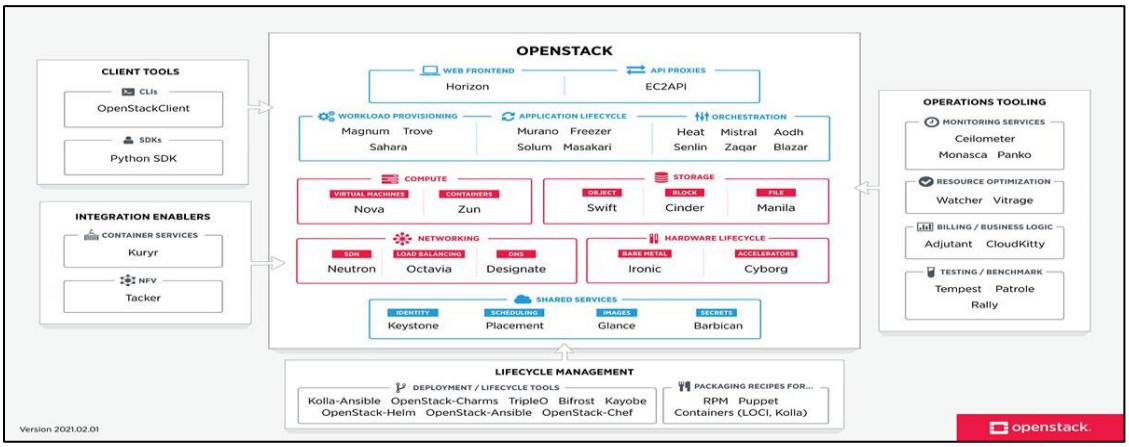

Fig 1: Map of OpenStack resources and functions

2. IMPLEMENTATION

2.1 Installation of OpenStack

For installing the OpenStack, we have used the Host Network Operating System of **Ubuntu Server 16.04 LTS** and created Controller, Compute and Block Node. We have taken the Network type of Management Network and Provider Network for the Network Environment. We have named the interfaces as eth0, eth1 and eth2 for the Management, Provider, and Internet.

Network Type	**CIDR**	**Gateway**	**Default Interface**
Management Network	10.0.0.0/24	10.0.0.1	eth0
Provider Network	203.0.113.0/24	203.0.113.1	eth1

2.2 Installation of Controller

To install Controller, we will set up the Virtual Machine (VM) and second Bare Metal Server. For setting up the VM we used VCPU core of 2, RAM of 6 GB, Primary Disk of 20 GB. For Bare Metal Server we used CPU of 4 cores, RAM of 32GB and Primary Disk of 512GB.

In Virtual Box Host-Only Network Ethernet Adapter 2, we should manually configure to IPv4 Address 10.0.0.1 and Net Mask 255.255.255.0 with DHCP disabled. NAT Network Provider Network1 should be changed from CIDR 203.0.113.0/24 DHCP disabled to NAT Network NatNetwork1 CIDR 10.10.10.0/24 DHCP enabled.
Virtual Box Network Name, Host Only Adapter 2, NAT Network ProviderNetwork1 should be set to Promiscuous Mode: allow all NAT Network NatNetwork1

Configure Security, Networking, Install Linux Utilities

Configuring 'sudo' access for kris	**Enabling Network Interfaces**	We need to run 'ifconfig' as superuser to verify settings. Also, to verify connectivity to other hosts, once configured.
sudo su *visudo* adding the following line at the bottom of the file: *kris ALL=(ALL) NOPASSWD:ALL* Now, save, exit and run *sudo su* again to test ***Edit /etc/hosts*** We remove 127.0.1.1 controller, if present and also we need to make sure following lines are present: *10.0.0.11 controller* *10.0.0.31 compute1* *10.0.0.41 block1* ***Edit /etc/default/grub to include:*** GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0" Run command: *update-grub* *reboot*	*sudo su* Edit /etc/network/interfaces We have to make sure that the following Interfaces definitions are present: *auto eth0* *iface eth0 inet static* *address 10.0.0.11* *netmask 255.255.255.0* *dns-nameservers 8.8.8.8* *auto eth1* *iface eth1 inet manual* *up ip link set dev eth1 up* *down ip link set dev eth1 down* *auto eth2* *iface eth2 inet dhcp* Now the system needs to be rebooted.	*Ping -c 3 openstack.org* *ping -c 3 compute1* *ping -c 3 block1* ***Install basic Linux Utilities*** The following command has been used. *sudo su* *apt update* *apt install vim glances curl* *apt upgrade -y*

Installation and Configuration of Network Time Protocol:
At first, we have to installl and configure the components with the following commands:
sudo su
`apt install chrony`

Edit /etc/chrony/chrony.conf:
set **server** to Orgaznization's NTP Server,
set **allow** to **10.0.0.0/24**
save and quit
Restart of the **chrony** service:
`service chrony restart`

Verify: *chronyc sources*

Install Basic OpenStack Packages:

We have used the following command for the installation of the Basic OpenStack Packages:

```
sudo su
apt install software-properties-common
add-apt-repository cloud-archive:yoga
apt update && apt dist-upgrade
reboot
apt install python-openstackclient
```

SQL Database – MariaDB:

We have used the follwing command for the installation and configuration of the Package:

```
sudo su
apt install mariadb-server python-pymysql
```

To create and edit MariaDB configuration file: /etc/mysql/mariadb.conf.d/99-openstack.cnf
Used this following 7 lines in the file:

```
[mysqld]
bind-address = 10.0.0.11
default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

Restart of the MariaDB service:

```
service mysql restart
```

Secure the Database Service:

```
mysql_secure_installation
```

Message Queue – RabbitMQ	**Memcached**
Install and Configure Packages: sudo su *apt install rabbitmq-server* Add openstack user: *rabbitmqctl add_user openstack openstack* Configure permissions for openstack user: *rabbitmqctl set_permissions openstack ".*" ".*" ".*"*	Install and Configure Packages: sudo su apt install memcached python-memcache Edit /etc/memcached.conf to define IP address: -l 10.0.0.11 Restart Memcached Service: service memcached restart

Etcd

At this point we created ***etcd*** User and directories using the following lines of the commands:

```
sudo su
groupadd --system etcd
```

```
useradd --home-dir "/var/lib/etcd" --system --shell /bin/false -g etcd etcd
mkdir -p /etc/etcd
chown etcd:etcd /etc/etcd
mkdir -p /var/lib/etcd
chown etcd:etcd /var/lib/etcd
```

Download and installation **etcd** tarball:

```
ETCD_VER=v3.2.7
rm -rf /tmp/etcd && mkdir -p /tmp/etcd
curl -L https://github.com/coreos/etcd/releases/download/${ETCD_VER}/etcd-${ETCD_VER}-linux-amd64.tar.gz -o /tmp/etcd-${ETCD_VER}-linux-amd64.tar.gz
tar xzvf /tmp/etcd-${ETCD_VER}-linux-amd64.tar.gz -C /tmp/etcd --strip-components=1
cp /tmp/etcd/etcd /usr/bin/etcd
cp /tmp/etcd/etcdctl /usr/bin/etcdctl
```

Create and edit the /etc/etcd/etcd.conf.yml file

```
vim /etc/etcd/etcd.conf.yml
```

and put following 9 lines in it:

```
name: controller
data-dir: /var/lib/etcd
initial-cluster-state: 'new'
initial-cluster-token: 'etcd-cluster-01'
initial-cluster: controller=http://10.0.0.11:2380
initial-advertise-peer-urls: http://10.0.0.11:2380
advertise-client-urls: http://10.0.0.11:2379
listen-peer-urls: http://0.0.0.0:2380
listen-client-urls: http://10.0.0.11:2379
```

Create and edit /lib/systemd/system/etcd.service file

```
vim /lib/systemd/system/etcd.service
```

and put following 13 lines in it:

```
[Unit]
After=network.target
Description=etcd - highly-available key value store

[Service]
LimitNOFILE=65536
Restart=on-failure
Type=notify
ExecStart=/usr/bin/etcd --config-file /etc/etcd/etcd.conf.yml
User=etcd

[Install]
WantedBy=multi-user.target
```

Enable and start **etcd** Service:

```
systemctl enable etcd
systemctl start etcd
```

Installation of Keystone - Identity Management

Configuring SQL Database for Keystone with these commands:

```
sudo su
mysql
CREATE DATABASE keystone;
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY 'openstack';
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'openstack';
EXIT;
```

Install and Configure Packages:

Run these commands: `sudo su`

Install required packages + crudini to edit .conf files

```
apt install keystone apache2 libapache2-mod-wsgi crudini -y
```

Configure Keystone database access, as set above

```
crudini --set /etc/keystone/keystone.conf database connection mysql+pymysql://keystone:openstack@controller/keystone
```

Set Fernet Token Provider

```
crudini --set /etc/keystone/keystone.conf token provider fernet
```

Populate Identity Service Database

```
su -s /bin/sh -c "keystone-manage db_sync" keystone
```

Initialize Fernet Repositories

```
keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

Bootstrap Identity Service

```
keystone-manage bootstrap --bootstrap-password openstack --bootstrap-admin-url http://controller:35357/v3/ --bootstrap-internal-url http://controller:5000/v3/ --bootstrap-public-url http://controller:5000/v3/ --bootstrap-region-id RegionOne
```

Configuration of Apache Server:

Configuration of Apache Server:
Edit /etc/apache2/apache2.conf and add following line:

```
ServerName controller
```

Now we need restart the apache2 service as:
Service>apache2>restart

Configureing OpenStack Client Environment Scripts. To Create admin-openrc Script (in Primary User's Home Directory, for example), we must Insert following lines:

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
```

Verifying the Keystone operation

```
. admin-openrc
openstack token issue
```

Creating Projects, Users and Roles with these commands:

```
. admin-openrc
# Create a service Project
openstack project create --domain default --description "Service Project" service
# Create a demo Project
```

```
export OS_USERNAME=admin
export OS_PASSWORD=openstack
export
OS_AUTH_URL=http://controller:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

We Inserted following lines to Create demo-openrc Scriptd:

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=demo
export OS_USERNAME=demo
export OS_PASSWORD=openstack
export
OS_AUTH_URL=http://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

```
openstack project create --domain
default --description "Demo Project"
demo
# Create a demo User
openstack user create --domain
default --password openstack demo
# Create a user Role
openstack role create user
# Add the user role to User demo in
Project demo
openstack role add --project demo --
user demo user
```

Verification of User demo with these following commands:

```
. demo-openrc
openstack token issue
```

Installation of Glance - Image Service

Configuration of SQL Database for Glance through these following commands:

```
sudo su
mysql
CREATE DATABASE glance;
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' IDENTIFIED BY
'openstack';
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' IDENTIFIED BY 'openstack';
EXIT;
```

Creating glance User

```
. admin-openrc
openstack user create --domain default --password openstack glance
```

Adding admin role to User glance in Project service

```
openstack role add --project service --user glance admin
```

Creating glance Service

```
openstack service create --name glance --description "OpenStack Image" image
```

Creating glance Service Endpoints

```
openstack endpoint create --region RegionOne image public
http://controller:9292
openstack endpoint create --region RegionOne image internal
http://controller:9292
openstack endpoint create --region RegionOne image admin http://controller:9292
```

Install and Configure Packages with these following commands:

```
apt update -y
apt install glance -y

Configure /etc/glance/glance-api.conf Parameters
Run following commands:
# Configure database access for glance
crudini --set /etc/glance/glance-api.conf database connection
mysql+pymysql://glance:openstack@controller/glance
# Configure Identity Service access
crudini --set /etc/glance/glance-api.conf keystone_authtoken auth_uri
http://controller:5000
crudini --set /etc/glance/glance-api.conf keystone_authtoken auth_url
http://controller:35357
crudini --set /etc/glance/glance-api.conf keystone_authtoken memcached_servers
controller:11211
crudini --set /etc/glance/glance-api.conf keystone_authtoken auth_type password
crudini --set /etc/glance/glance-api.conf keystone_authtoken
project_domain_name default
crudini --set /etc/glance/glance-api.conf keystone_authtoken user_domain_name
default
crudini --set /etc/glance/glance-api.conf keystone_authtoken project_name
service
crudini --set /etc/glance/glance-api.conf keystone_authtoken username glance
crudini --set /etc/glance/glance-api.conf keystone_authtoken password openstack
crudini --set /etc/glance/glance-api.conf paste_deploy flavor keystone
# Configure Glance to store Images on Local Filesystem
crudini --set /etc/glance/glance-api.conf glance_store stores "file,http"
crudini --set /etc/glance/glance-api.conf glance_store default_store file
crudini --set /etc/glance/glance-api.conf glance_store
filesystem_store_datadir /var/lib/glance/images/
```

Configure /etc/glance/glance-registry.conf Parameters with these following commands:

```
# Configure database access for glance
crudini --set /etc/glance/glance-registry.conf database connection
mysql+pymysql://glance:openstack@controller/glance
# Configure Identity Service access
crudini --set /etc/glance/glance-registry.conf keystone_authtoken auth_uri
http://controller:5000
crudini --set /etc/glance/glance-registry.conf keystone_authtoken auth_url
http://controller:35357
crudini --set /etc/glance/glance-registry.conf keystone_authtoken
memcached_servers controller:11211
crudini --set /etc/glance/glance-registry.conf keystone_authtoken auth_type
password
crudini --set /etc/glance/glance-registry.conf keystone_authtoken
project_domain_name default
```

```
crudini --set /etc/glance/glance-registry.conf keystone_authtoken
user_domain_name default
crudini --set /etc/glance/glance-registry.conf keystone_authtoken project_name
service
crudini --set /etc/glance/glance-registry.conf keystone_authtoken username
glance
crudini --set /etc/glance/glance-registry.conf keystone_authtoken password
openstack
crudini --set /etc/glance/glance-registry.conf paste_deploy flavor keystone
```

Populate the Image Service Database with these following commands:

```
su -s /bin/sh -c "glance-manage db_sync" glance
```

Restart of glance Services

```
service glance-registry restart
service glance-api restart
```

Verify Glance Operation with these following commands:

```
. admin-openrc
wget http://download.cirros-cloud.net/0.3.5/cirros-0.3.5-x86_64-disk.img
openstack image create cirros3.5 --file cirros-0.3.5-x86_64-disk.img --disk-
format qcow2 --container-format bare --public
openstack image list
```

Install & Configure Nova (Compute Service) Controller

Configure SQL Databases for Nova with these following commands:

```
sudo su
mysql
CREATE DATABASE nova_api;
CREATE DATABASE nova;
CREATE DATABASE nova_cell0;
GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'localhost' IDENTIFIED BY
'openstack';
GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'%' IDENTIFIED BY 'openstack';
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' IDENTIFIED BY 'openstack';
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY 'openstack';
GRANT ALL PRIVILEGES ON nova_cell0.* TO 'nova'@'localhost' IDENTIFIED BY
'openstack';
GRANT ALL PRIVILEGES ON nova_cell0.* TO 'nova'@'%' IDENTIFIED BY 'openstack';
```

Create Compute Service User and add admin role in service Project with these following commands:

```
. admin-openrc
openstack user create --domain default --password openstack nova
openstack role add --project service --user nova admin
```

Create Compute Service & Endpoints with these following commands:

```
. admin-openrc
openstack service create --name nova --description "OpenStack Compute" compute
openstack endpoint create --region RegionOne compute public
http://controller:8774/v2.1
openstack endpoint create --region RegionOne compute internal
http://controller:8774/v2.1
openstack endpoint create --region RegionOne compute admin
http://controller:8774/v2.1
```

Create Placement Service User and add admin role in service Project with these following commands:

```
. admin-openrc
openstack user create --domain default --password openstack placement
openstack role add --project service --user placement admin
```

Create Placement Service & Endpoints with these following commands:

```
. admin-openrc
openstack service create --name placement --description "Placement API"
placement
openstack endpoint create --region RegionOne placement public
http://controller:8778
openstack endpoint create --region RegionOne placement internal
http://controller:8778
openstack endpoint create --region RegionOne placement admin
http://controller:8778
```

Install Nova Controller Packages with these following commands:

```
sudo su
apt install -y nova-api nova-conductor nova-consoleauth nova-novncproxy nova-
scheduler nova-placement-api
```

Configure MySQL & RabbitMQ parameters in /etc/nova/nova.conf with these following commands:

```
crudini --set /etc/nova/nova.conf api_database connection
mysql+pymysql://nova:openstack@controller/nova_api
crudini --set /etc/nova/nova.conf database connection
mysql+pymysql://nova:openstack@controller/nova
crudini --set /etc/nova/nova.conf DEFAULT transport_url
rabbit://openstack:openstack@controller
```

Configure Identity Service access with these following commands:

```
crudini --set /etc/nova/nova.conf api auth_strategy keystone
crudini --set /etc/nova/nova.conf keystone_authtoken auth_uri
http://controller:5000
crudini --set /etc/nova/nova.conf keystone_authtoken auth_url
http://controller:35357
crudini --set /etc/nova/nova.conf keystone_authtoken memcached_servers
controller:11211
```

```
crudini --set /etc/nova/nova.conf keystone_authtoken auth_type password
crudini  --set  /etc/nova/nova.conf  keystone_authtoken  project_domain_name
default
crudini --set /etc/nova/nova.conf keystone_authtoken user_domain_name default
crudini --set /etc/nova/nova.conf keystone_authtoken project_name service
crudini --set /etc/nova/nova.conf keystone_authtoken username nova
crudini --set /etc/nova/nova.conf keystone_authtoken password openstack
```

Configure support for Networking Service with these following commands:

```
crudini --set /etc/nova/nova.conf DEFAULT my_ip 10.0.0.11
crudini --set /etc/nova/nova.conf DEFAULT use _neutron True
crudini      --set      /etc/nova/nova.conf      DEFAULT      firewall_driver
nova.virt.firewall.NoopFirewallDriver
```

Configure vnc proxy on Controller Node with these following commands:

```
crudini --set /etc/nova/nova.conf vnc enabled True
crudini --set /etc/nova/nova.conf vnc vncserver_listen 10.0.0.11
crudini --set /etc/nova/nova.conf vnc vncserver_proxyclient_address 10.0.0.11
```

Configure Glance location with this following command:

```
crudini --set /etc/nova/nova.conf glance api_servers http://controller:9292
```

Configure Lock Path for Oslo Concurrency with this following command:

```
crudini --set /etc/nova/nova.conf oslo_concurrency lock_path /var/lib/nova/tmp
```

Configure Placement API with these following commands:

```
crudini --set /etc/nova/nova.conf placement os_region_name RegionOne
crudini --set /etc/nova/nova.conf placement project_domain_name Default
crudini --set /etc/nova/nova.conf placement project_name service
crudini --set /etc/nova/nova.conf placement auth_type password
crudini --set /etc/nova/nova.conf placement user_domain_name Default
crudini --set /etc/nova/nova.conf placement auth_url http://controller:35357/v3
crudini --set /etc/nova/nova.conf placement username placement
crudini --set /etc/nova/nova.conf placement password openstack
```

Remove log_dir parameter in DEFAULT section with this following command:

```
crudini --del /etc/nova/nova.conf DEFAULT log_dir
```

Populate nova_api Database with these following commands:

```
sudo su
su -s /bin/sh -c "nova-manage api_db sync" nova
```

Register cell0 Database with this following command:

```
su -s /bin/sh -c "nova-manage cell_v2 map_cell0" nova
```

Create cell1 Cell with this following command:

su -s /bin/sh -c "nova-manage cell_v2 create_cell --name=cell1 --verbose" nova

Populate nova Database with this following command:

```
su -s /bin/sh -c "nova-manage db sync" nova
```

Verify configuration of Cells with this following command: `nova-manage cell_v2 list_cells`

Restart Services with these following commands:

```
service nova-api restart
service nova-consoleauth restart
service nova-scheduler restart
service nova-conductor restart
service nova-novncproxy restart
```

Install and Configure Nova on Compute Node(s)

Discovering the Compute Nodes with this following command:

```
su -s /bin/sh -c "nova-manage cell_v2 discover_hosts --verbose" nova
```

Verify Compute Service Installation with these following commands:

```
. admin-openrc
openstack compute service list
openstack catalog list
openstack image list
nova-status upgrade check
```

Install Neutron (Network Service) on Controller Node

Creating Neutron SQL Database with these following commands:

```
sudo su
mysql
CREATE DATABASE neutron;
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' IDENTIFIED BY
'openstack';
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' IDENTIFIED BY 'openstack';
EXIT;
```

Creating neutron User and add admin Role in service Project with these following commands:

```
. admin-openrc
openstack user create --domain default --password openstack neutron
openstack role add --project service --user neutron admin
```

Create Neutron Service and Endpoints with these following commands:

```
openstack service create --name neutron --description "OpenStack Networking"
network
```

```
openstack endpoint create --region RegionOne network public
http://controller:9696
openstack endpoint create --region RegionOne network internal
http://controller:9696
openstack endpoint create --region RegionOne network admin
http://controller:9696
```

Install Neutron Packages with these following commands:

```
sudo su
apt install -y neutron-server neutron-plugin-ml2 neutron-linuxbridge-agent
neutron-l3-agent neutron-dhcp-agent  neutron-metadata-agent
```

Configure SQL Database and RabbitMQ access for Neutron with these following commands:

```
crudini --set /etc/neutron/neutron.conf database connection
mysql+pymysql://neutron:openstack@controller/neutron
crudini --set /etc/neutron/neutron.conf DEFAULT transport_url
rabbit://openstack:openstack@controller
```

Enable the Modular Layer 2 (ML2) plug-in, router service, and overlapping IP addresses with these following commands:

```
crudini --set /etc/neutron/neutron.conf DEFAULT core_plugin ml2
crudini --set /etc/neutron/neutron.conf DEFAULT service_plugins router
crudini --set /etc/neutron/neutron.conf DEFAULT allow_overlapping_ips true
```

Configure Identity Service access with these following commands:

```
crudini --set /etc/neutron/neutron.conf api auth_strategy keystone
crudini --set /etc/neutron/neutron.conf keystone_authtoken auth_uri
http://controller:5000
crudini --set /etc/neutron/neutron.conf keystone_authtoken auth_url
http://controller:35357
crudini --set /etc/neutron/neutron.conf keystone_authtoken memcached_servers
controller:11211
crudini --set /etc/neutron/neutron.conf keystone_authtoken auth_type password
crudini --set /etc/neutron/neutron.conf keystone_authtoken project_domain_name
default
crudini --set /etc/neutron/neutron.conf keystone_authtoken user_domain_name
default
crudini --set /etc/neutron/neutron.conf keystone_authtoken project_name service
crudini --set /etc/neutron/neutron.conf keystone_authtoken username neutron
crudini --set /etc/neutron/neutron.conf keystone_authtoken password openstack
```

Configure Networking to notify Compute of network topology changes with the following commands:

```
crudini --set /etc/neutron/neutron.conf DEFAULT
notify_nova_on_port_status_changes true
crudini --set /etc/neutron/neutron.conf DEFAULT
notify_nova_on_port_data_changes true
```

Configure Nova access with these following commands:

```
crudini --set /etc/neutron/neutron.conf nova auth_url http://controller:35357
crudini --set /etc/neutron/neutron.conf nova auth_type password
crudini --set /etc/neutron/neutron.conf nova project_domain_name default
crudini --set /etc/neutron/neutron.conf nova user_domain_name default
crudini --set /etc/neutron/neutron.conf nova region_name RegionOne
crudini --set /etc/neutron/neutron.conf nova project_name service
crudini --set /etc/neutron/neutron.conf nova username nova
crudini --set /etc/neutron/neutron.conf nova password openstack
```

Configure ML2 Plugin with these following commands:

```
# Enable flat, VLAN and VXLAN Networks
crudini --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 type_drivers flat,vlan,vxlan
# Enable VXLAN Self-service Networks
crudini --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 tenant_network_types vxlan
# Enable Linux Bridge and L2Population mechanisms
crudini --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 mechanism_drivers linuxbridge,l2population
# Enable Port Security Extenstion Driver
crudini --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 extension_drivers port_security
# Configure provider Virtual Network as flat Network
crudini --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2_type_flat flat_networks provider
# Configure VXLAN Network Identifier Range for Self-service Networks
crudini --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2_type_vxlan vni_ranges 1:1000
# Enable ipset to increase efficiency of Security Group Rules
crudini --set /etc/neutron/plugins/ml2/ml2_conf.ini securitygroup enable_ipset true
```

Configure the Linux Bridge Agent with these following commands:

```
# Configure provider Virtual Network mapping to Physical Interface
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini linux_bridge physical_interface_mappings provider:eth1
# Enable VXLAN for Self-service Networks, configure IP address of the Management Interface handling VXLAN traffic
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini vxlan enable_vxlan true
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini vxlan local_ip 10.0.0.11
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini vxlan l2_population true
# Enable security groups and configure the Linux bridge iptables firewall driver
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini securitygroup enable_security_group true
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini securitygroup firewall_driver neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
```

Configure the Layer-3 Agent with this following command:

```
crudini --set /etc/neutron/l3_agent.ini DEFAULT interface_driver linuxbridge
```

Configure the DHCP Agent with these following commands:

```
crudini --set /etc/neutron/dhcp_agent.ini DEFAULT interface_driver linuxbridge
crudini --set /etc/neutron/dhcp_agent.ini DEFAULT dhcp_driver
neutron.agent.linux.dhcp.Dnsmasq
crudini --set /etc/neutron/dhcp_agent.ini DEFAULT enable_isolated_metadata true
```

Configure Metadata Agent with these Run following commands:

```
crudini --set /etc/neutron/metadata_agent.ini DEFAULT nova_metadata_host
controller
crudini --set /etc/neutron/metadata_agent.ini DEFAULT
metadata_proxy_shared_secret openstack
```

Configure Compute Service to use Neutron with these following commands:

```
crudini --set /etc/nova/nova.conf neutron url http://controller:9696
crudini --set /etc/nova/nova.conf neutron auth_url http://controller:35357
crudini --set /etc/nova/nova.conf neutron auth_type password
crudini --set /etc/nova/nova.conf neutron project_domain_name default
crudini --set /etc/nova/nova.conf neutron user_domain_name default
crudini --set /etc/nova/nova.conf neutron region_name RegionOne
crudini --set /etc/nova/nova.conf neutron project_name service
crudini --set /etc/nova/nova.conf neutron username neutron
crudini --set /etc/nova/nova.conf neutron password openstack
crudini --set /etc/nova/nova.conf neutron service_metadata_proxy true
crudini --set /etc/nova/nova.conf neutron metadata_proxy_shared_secret
openstack
```

Populate Neutron Database with these following Command:

```
su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf --
config-file /etc/neutron/plugins/ml2/ml2_conf.ini upgrade head" neutron
```

Restart the Compute API Service with this following command: `service nova-api restart`

Restart Networking Services with these following Commands:

```
service neutron-server restart
service neutron-linuxbridge-agent restart
service neutron-dhcp-agent restart
service neutron-metadata-agent restart
service neutron-l3-agent restart
```

Installation of Neutron on Compute Nodes

Verify Installation with following commands:

```
. admin-openrc
openstack network agent list
```

Install Cinder - Block Storage Service on "&BLOCK1_NAME&" Node"

Install Cinder Block Storage Service on Controller Node

Creating Cinder Database with these following commands:

```
sudo su
mysql
CREATE DATABASE cinder;
GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'localhost' IDENTIFIED BY
'openstack';
GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'%' IDENTIFIED BY 'openstack';
EXIT;
```

Creating cinder User and Add admin Role in service Project with these following commands:

```
. admin-openrc
openstack user create --domain default --password openstack cinder
openstack role add --project service --user cinder admin
```

Create cinderv2 and cinderv3 Services and their Endpoints with these following commands:

```
openstack service create --name cinderv2 --description "OpenStack Block
Storage" volumev2
openstack service create --name cinderv3 --description "OpenStack Block
Storage" volumev3
openstack endpoint create --region RegionOne volumev2 public
http://controller:8776/v2/%\(project_id\)s
openstack endpoint create --region RegionOne volumev2 internal
http://controller:8776/v2/%\(project_id\)s
openstack endpoint create --region RegionOne volumev2 admin
http://controller:8776/v2/%\(project_id\)s
openstack endpoint create --region RegionOne volumev3 public
http://controller:8776/v3/%\(project_id\)s
openstack endpoint create --region RegionOne volumev3 internal
http://controller:8776/v3/%\(project_id\)s
openstack endpoint create --region RegionOne volumev3 admin
http://controller:8776/v3/%\(project_id\)s
```

Install Packages with this following command: apt install -y cinder-api cinder-scheduler

Configure Database and RabbitMQ Access with following commands:

```
crudini --set /etc/cinder/cinder.conf database connection
mysql+pymysql://cinder:openstack@controller/cinder
crudini --set /etc/cinder/cinder.conf DEFAULT transport_url
rabbit://openstack:openstack@controller
```

Configure Identity Service Access with these following commands:

```
crudini --set /etc/cinder/cinder.conf DEFAULT auth_strategy keystone
crudini --set /etc/cinder/cinder.conf keystone_authtoken auth_uri
http://controller:5000
crudini --set /etc/cinder/cinder.conf keystone_authtoken auth_url
http://controller:35357
```

```
crudini --set /etc/cinder/cinder.conf keystone_authtoken memcached_servers
controller:11211
crudini --set /etc/cinder/cinder.conf keystone_authtoken auth_type password
crudini --set /etc/cinder/cinder.conf keystone_authtoken project_domain_name
default
crudini --set /etc/cinder/cinder.conf keystone_authtoken user_domain_name
default
crudini --set /etc/cinder/cinder.conf keystone_authtoken project_name service
crudini --set /etc/cinder/cinder.conf keystone_authtoken username cinder
crudini --set /etc/cinder/cinder.conf keystone_authtoken password openstack
```

Configure my_ip Parameter and Lock Path with these following commands:

```
crudini --set /etc/cinder/cinder.conf DEFAULT my_ip 10.0.0.11
crudini --set /etc/cinder/cinder.conf oslo_concurrency lock_path
/var/lib/cinder/tmp
```

Populate Block Storage Database with following command:

```
su -s /bin/sh -c "cinder-manage db sync" cinder
```

Configure Compute Service to use Cinder with this following command:

```
crudini --set /etc/nova/nova.conf cinder os_region_name RegionOne
```

Restart Services with these following commands:

```
service nova-api restart
service cinder-scheduler restart
service apache2 restart
```

Verify Cinder Operation with these following commands:

```
. admin-openrc
openstack volume service list
```

Install Horizon Dashboard

Installation of the Packages with these following commands:

```
sudo su
apt install -y openstack-dashboard
```

Edit /etc/openstack-dashboard/local_settings.py to include following settings:

```
OPENSTACK_HOST = "controller"

SESSION_ENGINE =
'django.contrib.sessions.backends.cache'

CACHES = {
    'default': {
```

OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"

OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"

Edit /etc/apache2/conf-available/openstack-dashboard.conf to include following line:

WSGIApplicationGroup %{GLOBAL}

Reload Web Server Configuration with this following command:

```
service apache2 reload
```

'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache', 'LOCATION': 'controller:11211', } } OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True OPENSTACK_API_VERSIONS = { "identity": 3, "image": 2, "volume": 2, }	Verify Horizon Operation by pointing Web Browser to http://10.0.0.11/horizon

2.3 Compute1 Installation

To install Compute1, we will set up the Virtual Machine (VM) and second Bare Metal Server. For setting up the VM we used VCPU core of 1, RAM of 4 GB, Primary Disk of 10 GB. For Bare Metal Server we used CPU of 8 cores, RAM of 32GB and Primary Disk of 512GB.

In Virtual Box Host-Only Network Ethernet Adapter 2, we should manually configure to IPv4 Address 10.0.0.1 and Net Mask 255.255.255.0 with DHCP disabled. NAT Network ProviderNetwork1 should be changed from CIDR 203.0.113.0/24 DHCP disabled to NAT Network NatNetwork1 CIDR 10.10.10.0/24 DHCP enabled.

Virtual Box Network Name Host Only Adapter 2 NAT Network ProviderNetwork1 should be set to Promiscuous Mode: allow all NAT Network NatNetwork1. The Network Interfaces should be as in the below chart.

For Installing the Ubuntu Operating System, We have chosen the following steps throughout the process.

Configure Security, Networking, Install Linux Utilities

Configure 'sudo' access for
`sudo su`
`visudo`
add following line at the bottom of the file:
kris ALL=(ALL) NOPASSWD:ALL
save, exit and run sudo su again to test
 	
Edit /etc/hosts
Remove 127.0.1.1 compute1, if present
Make sure following lines are present:
10.0.0.11 controller
10.0.0.31 compute1
10.0.0.41 block1
 	
Edit /etc/default/grub to include:
GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"
	Reboot the system
	Run 'ifconfig' as superuser to verify settings.
	Verify connectivity to other hosts, once configured
	`ping -c 3 openstack.org`
	`ping -c 3 controller`
	`ping -c 3 block1`
	
	Install basic Linux Utilities
	Run following commands:
	`sudo su`
	`apt update`
	`apt install vim glances curl`
	`apt upgrade -y`
	
	Install and Configure Network Time Protocol

Run command:
```
update-grub
reboot
```

Enable Network Interfaces
```
sudo su
```
Edit /etc/network/interfaces
Make sure following Interfaces definitions are present:

```
auto eth0
iface eth0 inet static
  address 10.0.0.31
  netmask 255.255.255.0
  dns-nameservers 8.8.8.8
auto eth1
iface eth1 inet manual
  up ip link set dev eth1 up
  down ip link set dev eth1 down
auto eth2
iface eth2 inet dhcp
```

Install and Configure Components
```
sudo su
apt install chrony
```

Edit /etc/chrony/chrony.conf:
set server to controller
server controller iburst
comment out pool 2.debian.pool.ntp.org offline iburst line.
save and quit
Restart chrony service:
```
service chrony restart
```

Verify:
```
chronyc sources
```

Installation of Basic OpenStack Packages

```
sudo su
apt install software-properties-common
add-apt-repository cloud-archive:yoga
apt update && apt dist-upgrade
reboot
apt install python-openstackclient
```

Installation of Keystone Identity Service on Controller Node

Install Glance Image Service on Controller Node

Install Compute Service on Compute Node

Install Nova Compute Package with these following commands:
```
sudo su
apt update
apt install -y nova-compute crudini
```

Configure RabbitMQ access with this following command:

```
crudini --set /etc/nova/nova.conf DEFAULT transport_url rabbit://openstack:openstack@controller
```

Configure Identity Service access
Run following commands:

```
crudini --set /etc/nova/nova.conf api auth_strategy keystone
crudini --set /etc/nova/nova.conf keystone_auth auth_uri http://controller:5000
```

Configure vnc Remote Console access on Compute Node with these following commands:
crudini --set /etc/nova/nova.conf vnc enabled True
```
crudini --set /etc/nova/nova.conf vnc vncserver_listen 0.0.0.0
crudini --set /etc/nova/nova.conf vnc vncserver_proxyclient_address 10.0.0.31

crudini --set /etc/nova/nova.conf vnc novncproxy_base_url http://10.0.0.11:6080/vnc_auto.html
```

Configure Glance location with this following command:
```
crudini --set /etc/nova/nova.conf glance api_servers http://controller:9292
```

```
crudini    --set    /etc/nova/nova.conf
keystone_auth                  auth_url
http://controller:35357
crudini    --set    /etc/nova/nova.conf
keystone_auth         memcached_servers
controller:11211
crudini    --set    /etc/nova/nova.conf
keystone_auth auth_type password
crudini    --set    /etc/nova/nova.conf
keystone_auth       project_domain_name
default
crudini    --set    /etc/nova/nova.conf
keystone_auth user_domain_name default
crudini    --set    /etc/nova/nova.conf
keystone_auth project_name service
crudini    --set    /etc/nova/nova.conf
keystone_auth username nova
crudini    --set    /etc/nova/nova.conf
keystone_auth password openstack
```

Configure support for Networking Service with these following commands:

```
crudini    --set    /etc/nova/nova.conf
DEFAULT my_ip 10.0.0.31
crudini    --set    /etc/nova/nova.conf
DEFAULT use_neutron True
crudini    --set    /etc/nova/nova.conf
DEFAULT                  firewall_driver
nova.virt.firewall.NoopFirewallDriver
```

Configure Lock Path for Oslo Concurrency with this following command:

```
crudini  --set  /etc/nova/nova.conf
oslo_concurrency          lock_path
/var/lib/nova/tmp
```

Configure Placement API with these following commands:

```
crudini  --set  /etc/nova/nova.conf
placement os_region_name RegionOne
crudini  --set  /etc/nova/nova.conf
placement       project_domain_name
Default
crudini  --set  /etc/nova/nova.conf
placement project_name service
crudini  --set  /etc/nova/nova.conf
placement auth_type password
crudini  --set  /etc/nova/nova.conf
placement user_domain_name Default
crudini  --set  /etc/nova/nova.conf
placement                    auth_url
http://controller:35357/v3
crudini  --set  /etc/nova/nova.conf
placement username placement
crudini  --set  /etc/nova/nova.conf
placement password openstack
```

Remove log_dir parameter in DEFAULT section with this following command:

```
crudini  --del  /etc/nova/nova.conf
DEFAULT log_dir
```

Set-up #1 - Virtual Machines: use QEMU Emulator

Setting up VM with the following command:

="crudini --set /etc/nova/nova-compute.conf libvirt virt_type qemu"

Set-up #2 - Bare Metal Hosts: use KVM

Verify Compute Host Capabilities with these following commands:

```
sudo su
kvm-ok
uname -m
```

Install KVM & Utilities with this following command:

```
apt-get install -y qemu-kvm libvirt-bin bridge-utils
```

Verify KVM Installation with this following command:

```
virsh list --all
```

Run following command:

```
crudini --set /etc/nova/nova-compute.conf libvirt virt_type kvm
```

Restart Nova Compute service with this following command: `service nova-compute restart`

Discover Compute Node on Controller Node

Install Neutron on Controller Node
Install Neutron on Compute Node

Install Packages with these following commands:

```
sudo su
apt update
apt install -y neutron-linuxbridge-agent
```

Configure RabbitMQ access with this following command:

```
crudini --set /etc/neutron/neutron.conf DEFAULT transport_url
rabbit://openstack:openstack@controller
```

Configure Indetity Service Accesss with these following commands:

```
crudini --set /etc/neutron/neutron.conf DEFAULT auth_strategy keystone
crudini --set /etc/neutron/neutron.conf keystone_authtoken auth_uri
http://controller:5000
crudini --set /etc/neutron/neutron.conf keystone_authtoken auth_url
http://controller:35357
crudini --set /etc/neutron/neutron.conf keystone_authtoken memcached_servers
controller:11211
crudini --set /etc/neutron/neutron.conf keystone_authtoken auth_type password
crudini --set /etc/neutron/neutron.conf keystone_authtoken project_domain_name
default
crudini --set /etc/neutron/neutron.conf keystone_authtoken user_domain_name
default
crudini --set /etc/neutron/neutron.conf keystone_authtoken project_name service
crudini --set /etc/neutron/neutron.conf keystone_authtoken username neutron
crudini --set /etc/neutron/neutron.conf keystone_authtoken password openstack
```

Configure the Linux Bridge Agent with these following commands:

```
# Configure provider Virtual Network mapping to Physical Interface
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini linux_bridge
physical_interface_mappings provider:eth1
# Enable VXLAN for Self-service Networks, configure IP address of the Management
Interface handling VXLAN traffic
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini vxlan
enable_vxlan true
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini vxlan local_ip
10.0.0.31
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini vxlan
l2_population true
# Enable security groups and configure the Linux bridge iptables firewall driver
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini securitygroup
enable_security_group true
```

```
crudini --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini securitygroup
firewall_driver neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
```

Configure Compute Service to use Neutron with these following commands:

```
crudini --set /etc/nova/nova.conf neutron url http://controller:9696
crudini --set /etc/nova/nova.conf neutron auth_url http://controller:35357
crudini --set /etc/nova/nova.conf neutron auth_type password
crudini --set /etc/nova/nova.conf neutron project_domain_name default
crudini --set /etc/nova/nova.conf neutron user_domain_name default
crudini --set /etc/nova/nova.conf neutron region_name RegionOne
crudini --set /etc/nova/nova.conf neutron project_name service
crudini --set /etc/nova/nova.conf neutron username neutron
crudini --set /etc/nova/nova.conf neutron password openstack
```

Restart Services with these following commands:

```
service nova-compute restart
service neutron-linuxbridge-agent restart
```

3. Installation of Block1

To install Controller, we will set up the Virtual Machine (VM) and second Bare Metal Server. For setting up the VM we used VCPU core of 1, RAM of 4 GB, Primary Disk of 20 GB. For Bare Metal Server we used CPU of 4 cores, RAM of 24 GB and Primary Disk of 256 GB + 2 TB.

In Virtual Box Host-Only Network Ethernet Adapter 2, we should manually configure to IPv4 Address 10.0.0.1 and Net Mask 255.255.255.0 with DHCP disabled. NAT Network ProviderNetwork1 should be changed from CIDR 203.0.113.0/24 DHCP disabled to NAT Network NatNetwork1 CIDR 10.10.10.0/24 DHCP enabled.

Virtual Box Network Name Host Only Adapter 2 NAT Network ProviderNetwork1 should be set to Promiscuous Mode: allow all NAT Network NatNetwork1. The Network Interfaces should be as in the below chart.

For Installing the Ubuntu Operating System, We have chosen the following steps throughout the process.

Configure Security, Networking, Install Linux Utilities

Configure 'sudo' access for
`sudo su`
`visudo`
add following line at the bottom of the file:
kris ALL=(ALL) NOPASSWD:ALL
save, exit and run sudo su again to test
 	
Edit /etc/hosts
Remove 127.0.1.1 block1, if present
Make sure following lines are present:
10.0.0.11 controller
10.0.0.31 compute1
10.0.0.41 block1
	Reboot the system
	Run 'ifconfig' as superuser to verify settings.
	Verify connectivity to other hosts, once configured
	`ping -c 3 openstack.org`
	`ping -c 3 controller`
	`ping -c 3 compute1`
	
	Install basic Linux Utilities with these following commands:
	`sudo su`
	`apt update`
	`apt install vim glances curl`
	`apt upgrade -y`
	`reboot`

Edit /etc/default/grub to include:
GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"
Run command:
`update-grub`
`reboot`
 	
Enable Network Interfaces
`sudo su`
Edit /etc/network/interfaces
Make sure following Interfaces definitions are present:
`auto eth0`
`iface eth0 inet static`
`  address 10.0.0.41`
`  netmask 255.255.255.0`
`  dns-nameservers 8.8.8.8`
`auto eth2`
`iface eth2 inet dhcp`	

Install and Configure Network Time Protocol	**Install Basic OpenStack Packages**

Install and Configure Components
`sudo su`
`apt install chrony`
 	
Edit /etc/chrony/chrony.conf:
set server to controller
`server controller iburst`
comment out pool 2.debian.pool.ntp.org offline iburst line.
save and quit
Restart chrony service:
service chrony restart
 	
Verify:
`chronyc sources`
	`sudo su`
	`apt install software-properties-common`
	`add-apt-repository cloud-archive:yoga`
	`apt update && apt dist-upgrade`
	`reboot`
	`apt install python-openstackclient`

Install Keystone Identity Service on Controller Node
Install Glance Image Service on Controller Node
Install Compute Service on Controller Node
Install Compute Service on Compute Node
Install Network Service on Controller Node
Install Network Service on Compute Node
Install Block Storage Service on Storage Node

Install Supporting Packages with these following commands:
`sudo su`
`apt update`
	Configure Database and RabbitMQ Access with these following commands:
	`crudini --set /etc/cinder/cinder.conf database connection`

```
apt install -y lvm2 thin-provisioning-tools crudini
```

Verify sdb Disk with this following command:

```
fdisk -l
```

Create LVM Physical Volume /dev/sdb with this following command:

```
pvcreate /dev/sdb
```

Create LVM Volume Group "cinder-volumes" with this following command:

```
vgcreate cinder-volumes /dev/sdb
```

Edit LVM Configuration File /etc/lvm/lvm.conf to include following line in devices section

```
filter = [ "a/sda/", "a/sdb/", "r/.*/"]
```

Install Cinder Packages with this following command:

```
apt install -y cinder-volume
```

```
mysql+pymysql://cinder:openstack@controller/cinder
crudini --set /etc/cinder/cinder.conf DEFAULT transport_url rabbit://openstack:openstack@controller
```

Configure Identity Service Access with these following commands:

```
crudini --set /etc/cinder/cinder.conf DEFAULT auth_strategy keystone
crudini --set /etc/cinder/cinder.conf keystone_authtoken auth_uri = http://controller:5000
crudini --set /etc/cinder/cinder.conf keystone_authtoken auth_url http://controller:35357
crudini --set /etc/cinder/cinder.conf keystone_authtoken memcached_servers controller:11211
crudini --set /etc/cinder/cinder.conf keystone_authtoken auth_type password
crudini --set /etc/cinder/cinder.conf keystone_authtoken project_domain_name default
crudini --set /etc/cinder/cinder.conf keystone_authtoken user_domain_name default
crudini --set /etc/cinder/cinder.conf keystone_authtoken project_name service
crudini --set /etc/cinder/cinder.conf keystone_authtoken username cinder
crudini --set /etc/cinder/cinder.conf keystone_authtoken password openstack
```

Configure my_ip Parameter with this following command:

```
crudini --set /etc/cinder/cinder.conf DEFAULT my_ip 10.0.0.41
```

Configure LVM Backend with these following commands:

```
crudini --set /etc/cinder/cinder.conf lvm volume_driver cinder.volume.drivers.lvm.LVMVolumeDriver
crudini --set /etc/cinder/cinder.conf lvm volume_group cinder-volumes
crudini --set /etc/cinder/cinder.conf lvm iscsi_protocol iscsi
crudini --set /etc/cinder/cinder.conf lvm iscsi_helper tgtadm
```

Enable LVM Backend with this following command:

```
crudini --set /etc/cinder/cinder.conf DEFAULT enabled_backends lvm
```

Configure Location of Image Service and Lock Path with these following commands:

```
crudini --set /etc/cinder/cinder.conf DEFAULT glance_api_servers
http://controller:9292
crudini --set /etc/cinder/cinder.conf oslo_concurrency lock_path
/var/lib/cinder/tmp
```

Restart Services with these following commands:

```
service tgt restart
service cinder-volume restart
```

Lesson Learned:

Initially we have selected Openstack version YOGA and Ubuntu Server 20.04 LTS. But both seems to be resource hungry as we are short on resources. After setup Ubuntu 20.04 LTS and OPENSTACK Yoga in our system, our system could not cope up with the resource hunger. Our System crashed. Later, we select OPENSTACK version PIKE and Ubuntu Server version 16.04 LTS. Besides, during our implementation we found Openstack service unstable (data not persistent in some cases). We have setup the whole system for several iteration. We used Packstack on CentOS. We also used DevStack where data is not persistent. Later, we setup the whole system manually.

3. Conclusion

In This paper, we discussed our learning as well as the different configurations that we implemented. Openstack is designed to manage several computers hosting application servers. These application servers can be executed by fully edged VMs, bare metal hosts. We worked on the most difficult case of application servers running on VMs to manage the necessary storage facilities and virtual network infrastructures. In this work, the problem of evaluating complexity and performance, in terms of virtual networking in cloud computing infrastructures dedicated to Network Function Virtualization deployment was addressed. OpenStack has a component specially dedicated to network service management. An OpenStack-based cloud platform was considered and deeply analyzed to fully understand the architecture of its virtual network infrastructure. Our project study shows that some of the constraints can be reduced by carefully redesigning the virtual network infrastructure and by carefully planning the functions of the virtual network.

4. References

1. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

2. https://dgtlinfra.com/top-10-cloud-service-providers-2022/