** **

**OPENTACK TUTORIAL**

Réalisé par :

RAMADAN Mostapha Novembre 2013

**Sommaire**

[**INTRODUCTION** 3](#_Toc371902386)

[**Architecture de service** 4](#_Toc371902387)

[**Les exigences du système** 5](#_Toc371902388)

[**Installation d’Openstack (Private Cloud)** 6](#_Toc371902389)

[**Installation Network Time Protocol (NTP)** 6](#_Toc371902392)

[**Installation de MySQL** 6](#_Toc371902394)

[**Installation et Configuration du “Identity Service”** 6](#_Toc371902395)

[**Installation et Configuration du “Image Service”** 8](#_Toc371902397)

[**Configuration du “Image Service database backend**” 8](#_Toc371902398)

[**Installation OpenStack Compute Service** 8](#_Toc371902399)

[**Creation /etc/sysconfig/network-scripts/ifcfg-br100:** 8](#_Toc371902400)

[**Configuration du SQL Database (MySQL) sur le Cloud Controller** 9](#_Toc371902401)

[**Configuration du SQL Database (PostgreSQL) sur le Cloud Controller** 9](#_Toc371902402)

[**Installation et configuration du Block Storage (Cinder)** 10](#_Toc371902403)

[**Installation du Compute Services** 10](#_Toc371902404)

[**Configuration de OpenStack Compute** 10](#_Toc371902405)

[**Configuration du Database for Compute** 12](#_Toc371902407)

[**Creation du Network for Compute VMs** 12](#_Toc371902408)

[**Verification du Compute Installation** 12](#_Toc371902409)

[**Demarrage du Virtual Machine Instances** 12](#_Toc371902410)

[**Ajout du keypair** 12](#_Toc371902411)

[**Demarrage d’une instance** 13](#_Toc371902412)

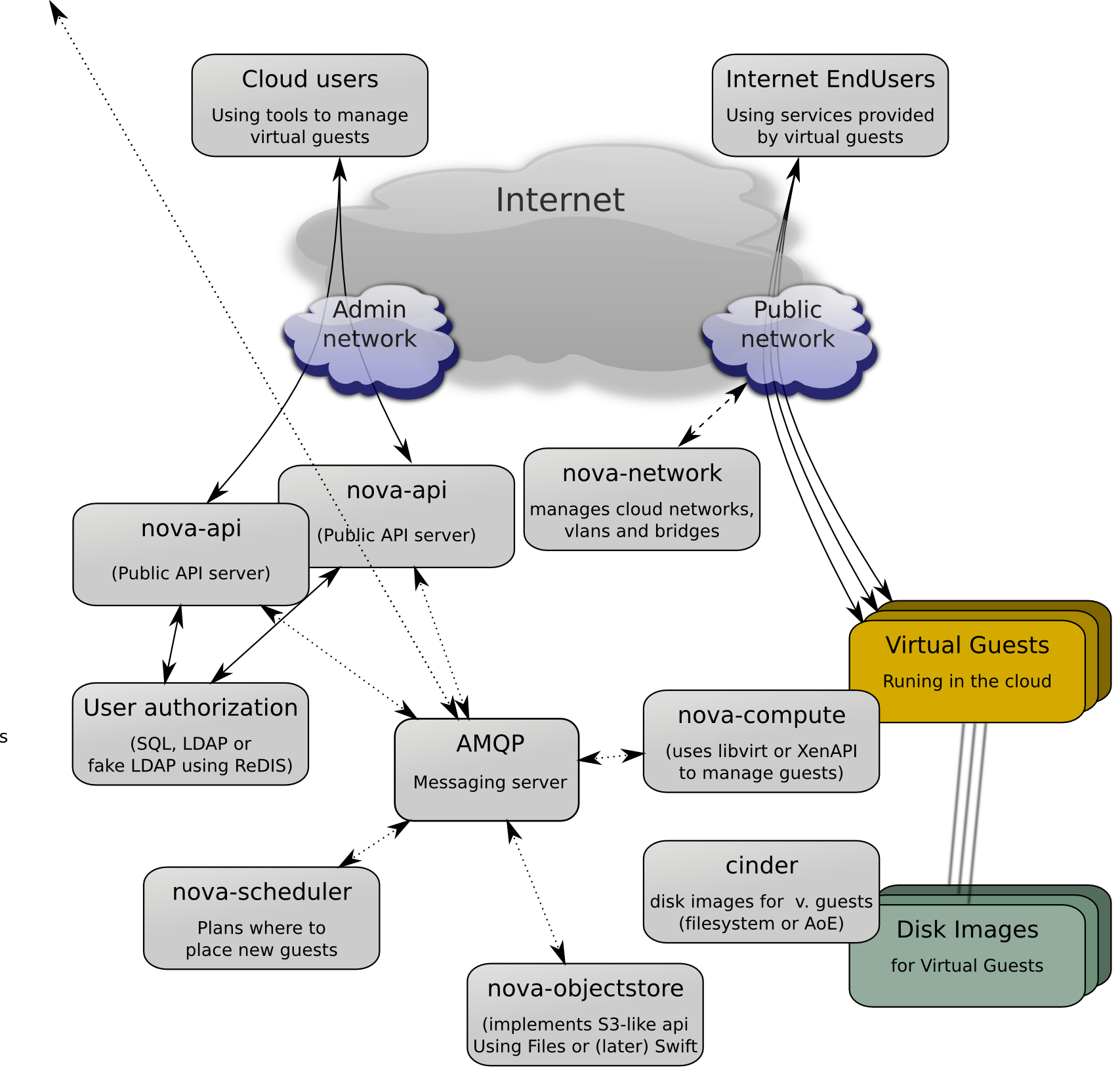
**[Installation et configuration du dashboard](#_Toc371902413)** [13](#_Toc371902413)

[**REFERENCES** 15](#_Toc371902414)

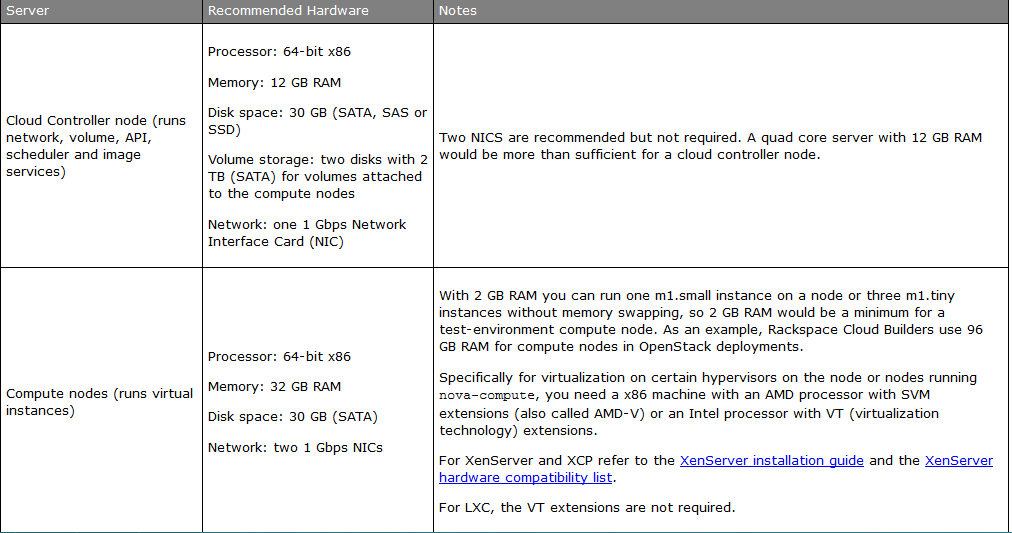
# **INTRODUCTION**

OpenStack est une collaboration mondiale de développeurs et des technologues de « cloud computing » produisant la plate-forme de « cloud computing » open source omniprésente pour les « public and private cloud ». Le projet vise à fournir des solutions pour tous les types de « clouds » ​​en étant simple à mettre en œuvre, hautement évolutive et riche en fonctionnalités. La technologie consiste en une série de projets interdépendants offrant divers composants pour une solution d'infrastructure « cloud ».

### **Architecture de service**



### **Les exigences du système**



### **Installation d’Openstack (Private Cloud)**

J’ai réalisé l’installation sur RedHat 6.3.

Serveur : HP server

Hard Disk - 1Tb

Memory/RAM - 16 GB

3.0 Dual Core processor

2 GB Graphical Display

## J’ai utilisé  openstack-dashboard (Horizon).

Le « OpenStack Dashboard » est une application Django qui s'exécute derrière un serveur web Apache par défaut. Il utilise memcache pour le cache de session par défaut. Un client VNC Web appelé novnc est utilisé pour fournir l'accès à la VNC consoles associées aux instances de KVM exécutées.

#### Installation Network Time Protocol (NTP)

**Pour conserver tous les services en synchronisation entre plusieurs machines, vous devez installer le NTP.**

$ sudo yum install ntp

**Configurer le serveur NTP sur votre nœud de façon à ce qu'il reçoit des données en modifiant le fichier ntp.conf et le redémarrage du service. En tant que root:**

$ sudo service ntpd start

$ sudo chkconfig ntpd on

#### Installation de MySQL

# yum install mysql mysql-server MySQL-python

# service mysqld start

#### Installation et Configuration du “Identity Service”

## Installez le service d'identité sur n'importe quel serveur qui est accessible aux autres serveurs que vous comptez utiliser pour les services de OpenStack, en tant que root:

$ yum install openstack-utils openstack-keystone python-keystoneclient

**Supprimez le fichier keystone.db créé dans le répertoire / var / lib / keystone.**

$ sudo rm /var/lib/keystone/keystone.db

**Configurer la base de données Keystone avec la commande openstack-db**

$ sudo openstack-db --init --service keystone

**Pour créer manuellement la base de données, démarrez mysql en exécutant:**

$ mysql -u root -p

**Entrez le mot de passe de l'utilisateur root mysql**   
  
**Pour configurer la base de données MySQL, créer la base de données du keystone.**

mysql> CREATE DATABASE KEYSTONE

**Créez un utilisateur MySQL pour la base de données du KEYSTONE nouvellement créée qui a le plein contrôle sur la base du KEYSTONE**

mysql> GRANT ALL ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY '*[YOUR\_KEYSTONEDB\_PASSWORD]*';

mysql> GRANT ALL ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY '*[YOUR\_KEYSTONEDB\_PASSWORD]*';

mysql> quit

**Pour changer la banque de données de MySQL, modifiez la ligne définissant la connection dans / etc / Keystone / keystone.conf comme ceci:**

connection = mysql://keystone:*[YOUR\_KEYSTONEDB\_PASSWORD]*@192.168.206.130/keystone

admin\_token = 012345SECRET99TOKEN012345

$ export ADMIN\_TOKEN=$(openssl rand -hex 10)

$ sudo openstack-config --set /etc/keystone/keystone.conf DEFAULT admin\_token $ADMIN\_TOKEN

$ sudo service openstack-keystone start && sudo chkconfig openstack-keystone on

**Enfin, initialiser la nouvelle base de donnée du Keystone, en tant que root:**

# keystone-manage db\_sync

#### Installation et Configuration du “Image Service”

**Installez le « image service », en tant que root:**

# yum install openstack-glance

### **Configuration du “Image Service database backend**”

$ mysql -u root -p

**Pour configurer la base de données MySQL, créer « Glance Database » :**

mysql> CREATE DATABASE glance;

**Créez un utilisateur MySQL pour « glance database » nouvellement créée qui a le plein contrôle sur la base de données.**

mysql> GRANT ALL ON glance.\* TO 'glance'@'%' IDENTIFIED BY '*[YOUR\_GLANCEDB\_PASSWORD]*';

mysql> GRANT ALL ON glance.\* TO 'glance'@'localhost' IDENTIFIED BY '*[YOUR\_GLANCEDB\_PASSWORD]*';

mysql> quit

#### Installation OpenStack Compute Service

**Preconfiguration du reseau**

# ip link set eth0 promisc on

### **Creation /etc/sysconfig/network-scripts/ifcfg-br100:**

DEVICE=br100

TYPE=Bridge

ONBOOT=yes

DELAY=0

BOOTPROTO=static

IPADDR=192.168.100.1

NETMASK=255.255.255.0

**Installation bridge-utils:**

$ sudo yum install bridge-utils

$ sudo brctl addbr br100

$ sudo /etc/init.d/networking restart

#### Configuration du SQL Database (MySQL) sur le Cloud Controller

mysql -u root –p

mysql> CREATE DATABASE nova

mysql> GRANT ALL ON nova.\* TO 'nova'@'%' IDENTIFIED BY '*[YOUR\_NOVADB\_PASSWORD]*';

mysql> GRANT ALL ON nova.\* TO 'nova'@'localhost' IDENTIFIED BY '*[YOUR\_NOVADB\_PASSWORD]*';

mysql> quit

#### Configuration du SQL Database (PostgreSQL) sur le Cloud Controller

$ sudo yum install postgresql postgresql-server

**sudo su – postgres**

postgres> psql

postgres=# CREATE USER novadbadmin;

postgres=# ALTER USER novadbadmin WITH PASSWORD '*[YOUR\_NOVADB\_PASSWORD]*';

postgres=# CREATE DATABASE nova;

postgres=# GRANT ALL PRIVILEGES ON DATABASE nova TO novadbadmin;

postgres=# \q

postgres> exit

**Maintenant, nous devons installer les paquets qui aideront Nova accéder à la base de données.**

$ sudo yum install python-sqlalchemy python-psycopg2

**Configurer le fichier / etc / nova / fichier nova.conf, afin de s'assurer qu'elle sait utiliser la base de données PostgreSQL**

sql\_connection = postgres://novadbadmin:[*[YOUR\_NOVADB\_PASSWORD]*]@127.0.0.1/nova

#### Installation et configuration du Block Storage (Cinder)

$ sudo yum install openstack-cinder openstack-cinder-doc \

iscsi-initiator-utils scsi-target-utils

#### Installation du Compute Services

$ sudo yum install openstack-nova

#### Configuration de OpenStack Compute

# groupadd nova

# usermod -g nova nova

# chown -R nova:nova /etc/nova

# chmod 640 /etc/nova/nova.conf

## /etc/nova/nova.conf:

auth\_strategy=keystone

network\_manager=nova.network.manager.FlatDHCPManager

fixed\_range=192.168.100.0/24

public\_interface=eth0

flat\_interface=eth0

flat\_network\_bridge=br100

[DEFAULT]

# LOGS/STATE

verbose=True

logdir=/var/log/nova

state\_path=/var/lib/nova

lock\_path=/var/lock/nova

rootwrap\_config=/etc/nova/rootwrap.conf

# SCHEDULER

compute\_scheduler\_driver=nova.scheduler.filter\_scheduler.FilterScheduler

# VOLUMES

# configured in cinder.conf

# DATABASE

sql\_connection=mysql://nova:yourpassword@192.168.206.130/nova

# COMPUTE

libvirt\_type=qemu

compute\_driver=libvirt.LibvirtDriver

instance\_name\_template=instance-%08x

api\_paste\_config=/etc/nova/api-paste.ini

# COMPUTE/APIS: if you have separate configs for separate services

# this flag is required for both nova-api and nova-compute

allow\_resize\_to\_same\_host=True

# APIS

osapi\_compute\_extension=nova.api.openstack.compute.contrib.standard\_extensions

ec2\_dmz\_host=192.168.206.130

s3\_host=192.168.206.130

# QPID

rpc\_backend=nova.rpc.impl\_qpid

qpid\_hostname=192.168.206.130

# GLANCE

image\_service=nova.image.glance.GlanceImageService

glance\_api\_servers=192.168.206.130:9292

# NETWORK

network\_manager=nova.network.manager.FlatDHCPManager

force\_dhcp\_release=True

dhcpbridge\_flagfile=/etc/nova/nova.conf

firewall\_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver

# Change my\_ip to match each host

my\_ip=192.168.206.130

public\_interface=eth100

vlan\_interface=eth0

flat\_network\_bridge=br100

flat\_interface=eth0

fixed\_range=192.168.100.0/24

# NOVNC CONSOLE

novncproxy\_base\_url=http://192.168.206.130:6080/vnc\_auto.html

# Change vncserver\_proxyclient\_address and vncserver\_listen to match each compute host

vncserver\_proxyclient\_address=192.168.206.130

vncserver\_listen=192.168.206.130

# AUTHENTICATION

auth\_strategy=keystone

[keystone\_authtoken]

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

admin\_tenant\_name = service

admin\_user = nova

admin\_password = nova

signing\_dirname = /tmp/keystone-signing-nova

#### Configuration du Database for Compute

# nova-manage db sync

#### Creation du Network for Compute VMs

nova-manage network create private --fixed\_range\_v4=192.168.100.0/24 --bridge\_interface=br100

#### Verification du Compute Installation

# nova-manage service list

#### Demarrage du Virtual Machine Instances

#### **ACTIVER SSH and ICMP (ping)**

$ nova secgroup-add-rule default tcp 22 22 0.0.0.0/0

$ nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0

#### 

#### Ajout du keypair

Le service Compute peut injecter une clé publique SSH sur un compte à l'instance, en supposant que l'image de la machine virtuelle utilisée supporte cela. Pour ajouter une paire de clés pour le service Compute, utilisez la commande ci-dessous. Cette commande peut être utilisée soit pour générer une nouvelle paire de clés, ou pour télécharger une clé publique existante.

$ nova keypair-add --pub\_key ~/.ssh/id\_rsa.pub mykey

#### Demarrage d’une instance

Pour démarrer une instance, nous devons spécifier une saveur, aussi connu comme un type d'instance, qui indique la taille d'une instance.

$ nova flavor-list

+----+-----------+-----------+------+-----------+------+-------+-------------+

| ID | Name | Memory\_MB | Disk | Ephemeral | Swap | VCPUs | RXTX\_Factor |

+----+-----------+-----------+------+-----------+------+-------+-------------+

| 1 | m1.tiny | 512 | 1 | N/A | 0 | 1 | |

| 2 | m1.small | 2048 | 20 | N/A | 0 | 1 | |

| 3 | m1.medium | 4096 | 40 | N/A | 0 | 2 | |

| 4 | m1.large | 8192 | 80 | N/A | 0 | 4 | |

| 5 | m1.xlarge | 16384 | 160 | N/A | 0 | 8 | |

Nous devons également spécifier l'image :

$ nova image-list

Enfin vous deviez être capable de d’accéder a l’instance créée à travers ssh.

#### Installation et configuration du dashboard

yum install memcached python-memcached mod\_wsgi openstack-dashboard

#### Pour configurer http pour le Dashboard:

Indiquez l'hôte de votre OpenStack Identity Service Endpoint dans le fichier / etc / openstack-dashboard /local\_settings.py avec le réglage OPENSTACK\_HOST.

OPENSTACK\_HOST = "127.0.0.1"

OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v2.0" % OPENSTACK\_HOST

OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "Member"

# service apache2 restart

# service memcached restart

#### Création d’un VM Client à travers Openstack Horizon Dashboard

Connectez-vous pour Horizon utilisant administrateur

Etapes pour créer un client :

* Create a new project – say demo project
* Create a new user and assign the privileges for user to create instances with the above created project
* You can use pre-existing images for creating clients on openstack or you can upload the images in openstack using glance index command from command line –
* glance image-create --name 'Fedora 19 x86\_64' --disk-format qcow2 --container-format bare --is-public true \ --copy-from <http://cloud.fedoraproject.org/fedora-19.x86_64.qcow2>
* Select the existing flavour or create a new flavour.
* Flavour is the name where you can tune hardware requirements for your system. For Ex. Hard Disk, Memory, CPU
* Now login using the newly created user.
* After login create a “Security keypair” so that the user can access the client machine using security group.
* Save the keypair to your machine.
* Click launching the image using appropriate image and flavour .
* After successful launching click on the image to access VNC console.
* You can also ssh the newly created instance using the IP from openstack server.

# **REFERENCES**

www.openstack.org