Table des matières

1. Source

`https://github.com/mseknibilel/OpenStack-Grizzly-Install-Guide`

a Keywords

`Multi node, Grizzly, Quantum, Nova, Keystone, Glance, Horizon, Cinder, OpenVSwitch, KVM, Ubuntu Server 12.04/13.04 (64 bits).`

1 Authors

Bilel Msekni

1. Contributors

Houssem Medhioub
Djamal Zeghlache
Sandeep Raman
Sam Stoelinga Anil Vishnoi
Gangur Hrushikesh

Wana contribute ? Read the guide, send your contribution and get your name listed ;)

2 What is it ?

OpenStack Grizzly Install Guide is an easy and tested way to create your own OpenStack platform.

If you like it, don't forget to star it !

Status : Stable

CHAPITRE 1

Requirements

Node Role	NICs
Control Node	eth0 (10.10.10.51), eth1 (192.168.100.51)
Network Node	eth0 (10.10.10.52), eth1 (10.20.20.52), eth2 (192.168.100.52)
Compute Node	eth0 (10.10.10.53), eth1 (10.20.20.53)

Note 1 : Always use dpkg -s to make sure you are using grizzly packages (version : 2013.1)

Note 2 : This is my current network architecture, you can add as many compute node as you wish.

CHAPITRE 2
Controller Node

1 Preparing Ubuntu

– After you install Ubuntu 12.04 or 13.04 Server 64bits, Go in sudo mode and don't leave it until the end of this guide :
```
sudo su
```
– Add Grizzly repositories [Only for Ubuntu 12.04] :
```
apt-get install -y ubuntu-cloud-keyring
echo deb http://ubuntu-cloud.archive.canonical.com/ubuntu \
precise-updates/grizzly main >> /etc/apt/sources.list.d/grizzly.list
```
– Update your system :
```
apt-get update -y
apt-get upgrade -y
apt-get dist-upgrade -y
```

2 Networking

– Only one NIC should have an internet access :
```
#For Exposing OpenStack API over the internet
auto eth1
iface eth1 inet static
address 192.168.100.51
netmask 255.255.255.0
gateway 192.168.100.1
dns-nameservers 8.8.8.8
# Not internet connected(used for OpenStack management)
auto eth0
iface eth0 inet static
address 10.10.10.51
netmask 255.255.255.0
```
– Restart the networking service :

```
service networking restart
```

3 MySQL & RabbitMQ

– Install MySQL :

```
apt-get install -y mysql-server python-mysqldb
```

– Configure mysql to accept all incoming requests :

```
sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
service mysql restart
```

– Create these databases :

```
mysql -u root -p

#Keystone
CREATE DATABASE keystone;
GRANT ALL ON keystone.* TO 'keystoneUser'@'%' IDENTIFIED BY 'keystonePass';

#Glance
CREATE DATABASE glance;
GRANT ALL ON glance.* TO 'glanceUser'@'%' IDENTIFIED BY 'glancePass';

#Quantum
CREATE DATABASE quantum;
GRANT ALL ON quantum.* TO 'quantumUser'@'%' IDENTIFIED BY 'quantumPass';

#Nova
CREATE DATABASE nova;
GRANT ALL ON nova.* TO 'novaUser'@'%' IDENTIFIED BY 'novaPass';

#Cinder
CREATE DATABASE cinder;
GRANT ALL ON cinder.* TO 'cinderUser'@'%' IDENTIFIED BY 'cinderPass';

quit;
```

4 RabbitMQ

– Install RabbitMQ :

```
apt-get install -y rabbitmq-server
```

– Install NTP service :

```
apt-get install -y ntp
```

5 Others

– Install other services :

```
apt-get install -y vlan bridge-utils
```

– Enable IP_Forwarding :

```
sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/' \
/etc/sysctl.conf

# To save you from rebooting, perform the following
sysctl net.ipv4.ip_forward=1
```

6 Keystone

– Start by the keystone packages :

```
apt-get install -y keystone
```

– Adapt the connection attribute in the /etc/keystone/keystone.conf to the new database :

```
connection = mysql://keystoneUser:keystonePass@10.10.10.51/keystone
```

– Restart the identity service then synchronize the database :

```
service keystone restart
keystone-manage db_sync
```

– Fill up the keystone database using the two scripts available in the Scripts folder
Modify the **HOST_IP** and **EXT_HOST_IP** variables before executing the scripts

```
wget https://raw.github.com/mseknibilel/OpenStack-Grizzly-Install-Guide/\
 OVS_MultiNode/KeystoneScripts/keystone_basic.sh
wget https://raw.github.com/mseknibilel/OpenStack-Grizzly-Install-Guide/\
 OVS_MultiNode/KeystoneScripts/keystone_endpoints_basic.sh

chmod +x keystone_basic.sh
chmod +x keystone_endpoints_basic.sh

./keystone_basic.sh
./keystone_endpoints_basic.sh
```

– Create a simple credential file and load it so you won't be bothered later :

```
nano creds
```

```
#Paste the following:
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin_pass
export OS_AUTH_URL="http://192.168.100.51:5000/v2.0/"

# Load it:
source creds
```

– To test Keystone, we use a simple CLI command :

```
keystone user-list
```

7 Glance

– We Move now to Glance installation :

```
apt-get install -y glance
```

– Update /etc/glance/glance-api-paste.ini with :

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:\
    filter_factory
delay_auth_decision = true
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service_pass
```

– Update the /etc/glance/glance-registry-paste.ini with :

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:\
    filter_factory
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service_pass
```

– Update /etc/glance/glance-api.conf with :

```
sql_connection = mysql://glanceUser:glancePass@10.10.10.51/glance
```

– And :

```
[paste_deploy]
```

```
flavor = keystone
```

– Update the /etc/glance/glance-registry.conf with :

```
sql_connection = mysql://glanceUser:glancePass@10.10.10.51/glance
```

– And :

```
[paste_deploy]
flavor = keystone
```

– Restart the glance-api and glance-registry services :

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

– Synchronize the glance database :

```
glance-manage db_sync
```

– To test Glance, upload the cirros cloud image directly from the internet :

```
glance image-create --name myFirstImage --is-public true \
--container-format bare --disk-format qcow2 --location \
https://launchpad.net/cirros/trunk/0.3.0/+download/\
cirros-0.3.0-x86_64-disk.img
```

– Now list the image to see what you have just uploaded :

```
glance image-list
```

8 Quantum

– Install the Quantum server and the OpenVSwitch package collection :

```
apt-get install -y quantum-server
```

– Edit the OVS plugin configuration file /etc/quantum/plugins/openvswitch/ovs_quantum_plugin.ini with :

```
#Under the database section
[DATABASE]
sql_connection = mysql://quantumUser:quantumPass@10.10.10.51/quantum

#Under the OVS section
[OVS]
tenant_network_type = gre
tunnel_id_ranges = 1:1000
enable_tunneling = True
```

– Edit /etc/quantum/api-paste.ini :

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:\
    filter_factory
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
```

```
admin_tenant_name = service
admin_user = quantum
admin_password = service_pass
```

– Update the /etc/quantum/quantum.conf :

```
[keystone_authtoken]
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = quantum
admin_password = service_pass
signing_dir = /var/lib/quantum/keystone-signing
```

– Restart the quantum server :

```
service quantum-server restart
```

9 Nova

– Start by installing nova components :

```
apt-get install -y nova-api nova-cert novnc nova-consoleauth \
nova-scheduler nova-novncproxy nova-doc nova-conductor
```

– Now modify authtoken section in the /etc/nova/api-paste.ini file to this :

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:\
    filter_factory
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = service_pass
signing_dirname = /tmp/keystone-signing-nova
# Workaround for https://bugs.launchpad.net/nova/+bug/1154809
auth_version = v2.0
```

– Modify the /etc/nova/nova.conf like this :

```
[DEFAULT]
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/run/lock/nova
verbose=True
api_paste_config=/etc/nova/api-paste.ini
```

```
compute_scheduler_driver=nova.scheduler.simple.SimpleScheduler
rabbit_host=10.10.10.51
nova_url=http://10.10.10.51:8774/v1.1/
sql_connection=mysql://novaUser:novaPass@10.10.10.51/nova
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf

# Auth
use_deprecated_auth=false
auth_strategy=keystone

# Imaging service
glance_api_servers=10.10.10.51:9292
image_service=nova.image.glance.GlanceImageService

# Vnc configuration
novnc_enabled=true
novncproxy_base_url=http://192.168.100.51:6080/vnc_auto.html
novncproxy_port=6080
vncserver_proxyclient_address=10.10.10.51
vncserver_listen=0.0.0.0

# Network settings
network_api_class=nova.network.quantumv2.api.API
quantum_url=http://10.10.10.51:9696
quantum_auth_strategy=keystone
quantum_admin_tenant_name=service
quantum_admin_username=quantum
quantum_admin_password=service_pass
quantum_admin_auth_url=http://10.10.10.51:35357/v2.0
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSInterfaceDriver
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver

#Metadata
service_quantum_metadata_proxy = True
quantum_metadata_proxy_shared_secret = helloOpenStack
metadata_host = 10.10.10.51
metadata_listen = 10.10.10.51
metadata_listen_port = 8775

# Compute #
```

```
compute_driver=libvirt.LibvirtDriver

# Cinder #
volume_api_class=nova.volume.cinder.API
osapi_volume_listen_port=5900
```

– Synchronize your database :

```
nova-manage db sync
```

– Restart nova-* services :

```
cd /etc/init.d/; for i in $( ls nova-* ); do sudo service $i restart; done
```

– Check for the smiling faces on nova-* services to confirm your installation :

```
nova-manage service list
```

10 Cinder

– Install the required packages :

```
apt-get install -y cinder-api cinder-scheduler cinder-volume \
iscsitarget open-iscsi iscsitarget-dkms
```

– Configure the iscsi services :

```
sed -i 's/false/true/g' /etc/default/iscsitarget
```

– Restart the services :

```
service iscsitarget start
service open-iscsi start
```

– Configure /etc/cinder/api-paste.ini like the following :

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:
    filter_factory
service_protocol = http
service_host = 192.168.100.51
service_port = 5000
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = cinder
admin_password = service_pass
signing_dir = /var/lib/cinder
```

– Edit the /etc/cinder/cinder.conf to :

```
[DEFAULT]
rootwrap_config=/etc/cinder/rootwrap.conf
sql_connection = mysql://cinderUser:cinderPass@10.10.10.51/cinder
```

```
api_paste_config = /etc/cinder/api-paste.ini
iscsi_helper=ietadm
volume_name_template = volume-%s
volume_group = cinder-volumes
verbose = True
auth_strategy = keystone
iscsi_ip_address=10.10.10.51
```

– Then, synchronize your database :

```
cinder-manage db sync
```

– Finally, don't forget to create a volumegroup and name it cinder-volumes :

```
dd if=/dev/zero of=cinder-volumes bs=1 count=0 seek=2G
losetup /dev/loop2 cinder-volumes
fdisk /dev/loop2
#Type in the followings:
n
p
1
ENTER
ENTER
t
8e
w
```

– Proceed to create the physical volume then the volume group :

```
pvcreate /dev/loop2
vgcreate cinder-volumes /dev/loop2
```

Note : Beware that this volume group gets lost after a system reboot. Click Here to know how to load it after a reboot

– Restart the cinder services :

```
cd /etc/init.d/; for i in $( ls cinder-* ); do sudo service $i restart;
done
```

– Verify if cinder services are running :

```
cd /etc/init.d/; for i in $( ls cinder-* ); do sudo service $i status;
done
```

11 Horizon

– To install horizon, proceed like this :

```
apt-get install -y openstack-dashboard memcached
```

– If you don't like the OpenStack ubuntu theme, you can remove the package to disable it :

```
dpkg --purge openstack-dashboard-ubuntu-theme
```

– Reload Apache and memcached :

```
service apache2 restart; service memcached restart
```

CHAPITRE 3

Network Node

1 Preparing the Node

- After you install Ubuntu 12.04 or 13.04 Server 64bits, Go in sudo mode :

```
sudo su
```

- Add Grizzly repositories [Only for Ubuntu 12.04] :

```
apt-get install -y ubuntu-cloud-keyring
echo deb http://ubuntu-cloud.archive.canonical.com/ubuntu\
precise-updates/grizzly main >> /etc/apt/sources.list.d/grizzly.list
```

- Update your system :

```
apt-get update -y
apt-get upgrade -y
apt-get dist-upgrade -y
```

- Install ntp service :

```
apt-get install -y ntp
```

- Configure the NTP server to follow the controller node :

```
#Comment the ubuntu NTP servers
sed -i 's/server 0.ubuntu.pool.ntp.org/#server 0.ubuntu.pool.ntp.org/g' \
/etc/ntp.conf
sed -i 's/server 1.ubuntu.pool.ntp.org/#server 1.ubuntu.pool.ntp.org/g' \
/etc/ntp.conf
sed -i 's/server 2.ubuntu.pool.ntp.org/#server 2.ubuntu.pool.ntp.org/g' \
/etc/ntp.conf
sed -i 's/server 3.ubuntu.pool.ntp.org/#server 3.ubuntu.pool.ntp.org/g' \
/etc/ntp.conf

#Set the network node to follow up your conroller node
sed -i 's/server ntp.ubuntu.com/server 10.10.10.51/g' /etc/ntp.conf

service ntp restart
```

- Install other services :

```
apt-get install -y vlan bridge-utils
```

– Enable IP_Forwarding :

```
sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/'\
/etc/sysctl.conf

# To save you from rebooting, perform the following
sysctl net.ipv4.ip_forward=1
```

2 Networking

– 3 NICs must be present :

```
# OpenStack management
auto eth0
iface eth0 inet static
address 10.10.10.52
netmask 255.255.255.0

# VM Configuration
auto eth1
iface eth1 inet static
address 10.20.20.52
netmask 255.255.255.0

# VM internet Access
auto eth2
iface eth2 inet static
address 192.168.100.52
netmask 255.255.255.0
```

3 OpenVSwitch (Part1)

– Install the openVSwitch :

```
apt-get install -y openvswitch-switch openvswitch-datapath-dkms
```

– Create the bridges :

```
#br-int will be used for VM integration
ovs-vsctl add-br br-int

#br-ex is used to make to VM accessible from the internet
ovs-vsctl add-br br-ex
```

4 Quantum

- Install the Quantum openvswitch agent, l3 agent and dhcp agent :

```
apt-get -y install quantum-plugin-openvswitch-agent quantum-dhcp-agent \
quantum-l3-agent quantum-metadata-agent
```

- Edit /etc/quantum/api-paste.ini :

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:
    filter_factory
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = quantum
admin_password = service_pass
```

- Edit the OVS plugin configuration file /etc/quantum/plugins/openvswitch/ovs_quantum_plugin.ini with :

```
#Under the database section
[DATABASE]
sql_connection = mysql://quantumUser:quantumPass@10.10.10.51/quantum

#Under the OVS section
[OVS]
tenant_network_type = gre
tunnel_id_ranges = 1:1000
integration_bridge = br-int
tunnel_bridge = br-tun
local_ip = 10.20.20.52
enable_tunneling = True
```

- Update /etc/quantum/metadata_agent.ini :

```
# The Quantum user information for accessing the Quantum API.
auth_url = http://10.10.10.51:35357/v2.0
auth_region = RegionOne
admin_tenant_name = service
admin_user = quantum
admin_password = service_pass

# IP address used by Nova metadata server
nova_metadata_ip = 10.10.10.51
```

```
# TCP Port used by Nova metadata server
nova_metadata_port = 8775

metadata_proxy_shared_secret = helloOpenStack
```

– Make sure that your rabbitMQ IP in /etc/quantum/quantum.conf is set to the controller node :

```
rabbit_host = 10.10.10.51

#And update the keystone_authtoken section

[keystone_authtoken]
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = quantum
admin_password = service_pass
signing_dir = /var/lib/quantum/keystone-signing
```

– Restart all the services :

```
cd /etc/init.d/; for i in $( ls quantum-* );
do sudo service $i restart; done
```

5 OpenVSwitch (Part2)

– Edit the eth2 in /etc/network/interfaces to become like this :

```
# VM internet Access
auto eth2
iface eth2 inet manual
up ifconfig $IFACE 0.0.0.0 up
up ip link set $IFACE promisc on
down ip link set $IFACE promisc off
down ifconfig $IFACE down
```

– Add the eth2 to the br-ex :

```
#Internet connectivity will be lost after this step but\
this won't affect OpenStack's work
ovs-vsctl add-port br-ex eth2
```

If you want to get internet connection back, you can assign the eth2's IP address to the br-ex in the /etc/network interfaces file.

CHAPITRE 4

Compute Node

1 Preparing the Node

– After you install Ubuntu 12.04 or 13.04 Server 64bits, Go in sudo mode :

```
sudo su
```

– Add Grizzly repositories [Only for Ubuntu 12.04] :

```
apt-get install -y ubuntu-cloud-keyring
echo deb http://ubuntu-cloud.archive.canonical.com/ubuntu\
precise-updates/grizzly main >> /etc/apt/sources.list.d/grizzly.list
```

– Update your system :

```
apt-get update -y
apt-get upgrade -y
apt-get dist-upgrade -y
```

– Install ntp service :

```
apt-get install -y ntp
```

– Configure the NTP server to follow the controller node :

```
#Comment the ubuntu NTP servers
sed -i 's/server 0.ubuntu.pool.ntp.org/#server 0.ubuntu.pool.ntp.org/g' \
/etc/ntp.conf
sed -i 's/server 1.ubuntu.pool.ntp.org/#server 1.ubuntu.pool.ntp.org/g' \
/etc/ntp.conf
sed -i 's/server 2.ubuntu.pool.ntp.org/#server 2.ubuntu.pool.ntp.org/g' \
/etc/ntp.conf
sed -i 's/server 3.ubuntu.pool.ntp.org/#server 3.ubuntu.pool.ntp.org/g' \
/etc/ntp.conf

#Set the compute node to follow up your conroller node
sed -i 's/server ntp.ubuntu.com/server 10.10.10.51/g' /etc/ntp.conf

service ntp restart
```

– Install other services :

```
apt-get install -y vlan bridge-utils
```

– Enable IP_Forwarding :

```
sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/' \
/etc/sysctl.conf

# To save you from rebooting, perform the following
sysctl net.ipv4.ip_forward=1
```

2 Networking

– Perform the following :

```
# OpenStack management
auto eth0
iface eth0 inet static
address 10.10.10.53
netmask 255.255.255.0

# VM Configuration
auto eth1
iface eth1 inet static
address 10.20.20.53
netmask 255.255.255.0
```

3 KVM

– make sure that your hardware enables virtualization :

```
apt-get install -y cpu-checker
kvm-ok
```

– Normally you would get a good response. Now, move to install kvm and configure it :

```
apt-get install -y kvm libvirt-bin pm-utils
```

– Edit the cgroup_device_acl array in the /etc/libvirt/qemu.conf file to :

```
cgroup_device_acl = [
"/dev/null", "/dev/full", "/dev/zero",
"/dev/random", "/dev/urandom",
"/dev/ptmx", "/dev/kvm", "/dev/kqemu",
"/dev/rtc", "/dev/hpet","/dev/net/tun"
]
```

– Delete default virtual bridge :

```
virsh net-destroy default
virsh net-undefine default
```

– Enable live migration by updating /etc/libvirt/libvirtd.conf file :

```
listen_tls = 0
listen_tcp = 1
auth_tcp = "none"
```

– Edit libvirtd_opts variable in /etc/init/libvirt-bin.conf file :

```
env libvirtd_opts="-d -l"
```

– Edit /etc/default/libvirt-bin file :

```
libvirtd_opts="-d -l"
```

– Restart the libvirt service to load the new values :

```
service libvirt-bin restart
```

4 OpenVSwitch

– Install the openVSwitch :

```
apt-get install -y openvswitch-switch openvswitch-datapath-dkms
```

– Create the bridges :

```
#br-int will be used for VM integration
ovs-vsctl add-br br-int
```

5 Quantum

– Install the Quantum openvswitch agent :

```
apt-get -y install quantum-plugin-openvswitch-agent
```

– Edit the OVS plugin configuration file /etc/quantum/plugins/openvswitch/ovs_quantum_plugin.ini with :

```
#Under the database section
[DATABASE]
sql_connection = mysql://quantumUser:quantumPass@10.10.10.51/quantum

#Under the OVS section
[OVS]
tenant_network_type = gre
tunnel_id_ranges = 1:1000
integration_bridge = br-int
tunnel_bridge = br-tun
local_ip = 10.20.20.53
enable_tunneling = True
```

– Make sure that your rabbitMQ IP in /etc/quantum/quantum.conf is set to the controller node :

```
rabbit_host = 10.10.10.51

#And update the keystone_authtoken section

[keystone_authtoken]
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = quantum
admin_password = service_pass
signing_dir = /var/lib/quantum/keystone-signing
```

– Restart all the services :

```
service quantum-plugin-openvswitch-agent restart
```

6 Nova

– Install nova's required components for the compute node :

```
apt-get install -y nova-compute-kvm
```

– Now modify authtoken section in the /etc/nova/api-paste.ini file to this :

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:\
    filter_factory
auth_host = 10.10.10.51
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = service_pass
signing_dirname = /tmp/keystone-signing-nova
# Workaround for https://bugs.launchpad.net/nova/+bug/1154809
auth_version = v2.0
```

– Edit /etc/nova/nova-compute.conf file :

```
[DEFAULT]
libvirt_type=kvm
libvirt_ovs_bridge=br-int
libvirt_vif_type=ethernet
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
```

```
libvirt_use_virtio_for_bridges=True
```

– Modify the /etc/nova/nova.conf like this :

```
[DEFAULT]
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/run/lock/nova
verbose=True
api_paste_config=/etc/nova/api-paste.ini
compute_scheduler_driver=nova.scheduler.simple.SimpleScheduler
rabbit_host=10.10.10.51
nova_url=http://10.10.10.51:8774/v1.1/
sql_connection=mysql://novaUser:novaPass@10.10.10.51/nova
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf

# Auth
use_deprecated_auth=false
auth_strategy=keystone

# Imaging service
glance_api_servers=10.10.10.51:9292
image_service=nova.image.glance.GlanceImageService

# Vnc configuration
novnc_enabled=true
novncproxy_base_url=http://192.168.100.51:6080/vnc_auto.html
novncproxy_port=6080
vncserver_proxyclient_address=10.10.10.53
vncserver_listen=0.0.0.0

# Network settings
network_api_class=nova.network.quantumv2.api.API
quantum_url=http://10.10.10.51:9696
quantum_auth_strategy=keystone
quantum_admin_tenant_name=service
quantum_admin_username=quantum
quantum_admin_password=service_pass
quantum_admin_auth_url=http://10.10.10.51:35357/v2.0
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
linuxnet_interface_driver=nova.network.\
    linux_net.LinuxOVSInterfaceDriver
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
```

```
#Metadata
service_quantum_metadata_proxy = True
quantum_metadata_proxy_shared_secret = helloOpenStack

# Compute #
compute_driver=libvirt.LibvirtDriver

# Cinder #
volume_api_class=nova.volume.cinder.API
osapi_volume_listen_port=5900
cinder_catalog_info=volume:cinder:internalURL
```

– Restart nova-* services :

```
cd /etc/init.d/; for i in $( ls nova-* );
do sudo service $i restart; done
```

– Check for the smiling faces on nova-* services to confirm your installation :

```
nova-manage service list
```

CHAPITRE 5

Your first VM

To start your first VM, we first need to create a new tenant, user and internal network.

- Create a new tenant :
  ```
  keystone tenant-create --name project_one
  ```
- Create a new user and assign the member role to it in the new tenant (keystone role-list to get the appropriate id) :
  ```
  keystone user-create --name=user_one --pass=user_one \
  --tenant-id $put_id_of_project_one --email=user_one@domain.com
  keystone user-role-add --tenant-id $put_id_of_project_one  \
  --user-id $put_id_of_user_one --role-id $put_id_of_member_role
  ```
- Create a new network for the tenant :
  ```
  quantum net-create --tenant-id $put_id_of_project_one net_proj_one
  ```
- Create a new subnet inside the new tenant network :
  ```
  quantum subnet-create --tenant-id \
  $put_id_of_project_one net_proj_one 50.50.1.0/24
  ```
- Create a router for the new tenant :
  ```
  quantum router-create --tenant-id \
  $put_id_of_project_one router_proj_one
  ```
- Add the router to the running l3 agent (if it wasn't automatically added) :
  ```
  quantum agent-list (to get the l3 agent ID)
  quantum l3-agent-router-add $l3_agent_ID router_proj_one
  ```
- Add the router to the subnet :
  ```
  quantum router-interface-add $put_router_proj_one_id_here \
  $put_subnet_id_here
  ```
- Restart all quantum services :
  ```
  cd /etc/init.d/; for i in $( ls quantum-* );
  do sudo service $i restart; done
  ```
- Create an external network with the tenant id belonging to the admin tenant (keystone tenant-list to get the appropriate id) :
  ```
  quantum net-create --tenant-id $put_id_of_admin_tenant ext_net \
  --router:external=True
  ```
- Create a subnet for the floating ips :

```
quantum subnet-create --tenant-id $put_id_of_admin_tenant \
--allocation-pool start=192.168.100.102,end=192.168.100.126 \
--gateway 192.168.100.1 ext_net 192.168.100.100/24 \
--enable_dhcp=False
```

- Set your router's gateway to the external network :

```
quantum router-gateway-set $put_router_proj_one_id_here $put_id_of_ext_net_
```

- Source creds relative to your project one tenant now :

```
nano creds_proj_one

#Paste the following:
export OS_TENANT_NAME=project_one
export OS_USERNAME=user_one
export OS_PASSWORD=user_one
export OS_AUTH_URL="http://192.168.100.51:5000/v2.0/"

source creds_proj_one
```

- Start by allocating a floating ip to the project one tenant :

```
quantum floatingip-create ext_net
```

- Start a VM :

```
nova --no-cache boot --image $id_myFirstImage --flavor 1 my_first_vm
```

- pick the id of the port corresponding to your VM :

```
quantum port-list
```

- Associate the floating IP to your VM :

```
quantum floatingip-associate $put_id_floating_ip $put_id_vm_port
```

That's it ! ping your VM and enjoy your OpenStack.

CHAPITRE 6

Licensing

CHAPITRE 7

Credits

This work has been based on :

– Bilel Msekni's Folsom Install guide https ://github.com/mseknibilel/OpenStack-Folsom-Install-guide
– OpenStack Grizzly Install Guide (Master Branch) https ://github.com/mseknibilel/OpenStack-Grizzly-Install-Guide