Hướng dẫn cài đặt Openstack Icehouse

(Trung tâm ICSE - Bách Khoa Hà Nội)

Trong bài hướng dẫn này chúng ta sẽ tiến hành cài Openstack trên 1 node. Sau đó, bạn có thể thêm các node compute khác vào.

Mục lục

Chú thích:

Màu A: Lệnh

Màu B: Nội dung file config cần chỉnh sửa

Tài liệu có tham khảo từ 2 bài viết:

https://github.com/fornyx/OpenStack-Havana-Install-Guide/blob/master/OpenStack-Havana-Install-Guide.rst#21-preparing-ubuntu

https://github.com/vietstacker/icehouse-aio-ubuntu

I. Requirements

Install Openstack on Ubuntu server 12.04 (64 bit)

Change hosts

nano /etc/hosts

127.0.0.1 change to 10.1.3.160

Config network

nano /etc/network/interfaces

```
#For Exposing OpenStack API over the internet
auto eth0
iface eth0 inet static
address 192.168.50.160
netmask 255.255.255.0
gateway 192.168.50.1
dns-nameservers 8.8.8.8

#Not internet connected(used for OpenStack management)
auto eth1
iface eth1 inet static
address 10.1.3.160
netmask 255.255.255.0
```

service networking restart

II. Install

Add Icehouse repositories

apt-get install python-software-properties

add-apt-repository cloud-archive:icehouse

Update system

apt-get -y update && apt-get -y upgrade && apt-get -y dist-upgrade

1. 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

# Install RabbitMQ:
apt-get install -y rabbitmq-server

# Install NTP service:
apt-get install -y ntp
```

2. Databases

```
# Download
wget https://raw2.github.com/Ch00k/OpenStack-Havana-Install-Guide/master/populate_database.sh

# Change ip in file populate_database.sh:
nano populate_database.sh

# Run
sh populate_database.sh
```

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
```

3. Keystone

```
# Start by the keystone packages:
apt-get install -y keystone

#Verify your keystone is running:
service keystone status

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

```
connection = mysql://keystone:openstacktest@10.1.3.160/keystone

# Remove Keystone SQLite database:
rm /var/lib/keystone/keystone.db

# 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 of this git repository:

# Modify the HOST_IP and EXT_HOST_IP variables before executing the scripts
wget https://raw2.github.com/Ch00k/OpenStack-Havana-Install-Guide/master/keystone_basic.sh
wget https://raw2.github.com/Ch00k/OpenStack-Havana-Install-Guide/master/keystone_endpoints_basic.sh

# Edit your ip, user, pass 2 file keystone_basic.sh and keystone_endpoints_basic.sh
nano keystone_basic.sh
nano keystone_endpoints_basic.sh

# Run:
sh keystone_basic.sh
sh keystone_endpoints_basic.sh

# Create a simple credential file and load it so you won't be bothered later:
nano/vi keystone_source

#Paste the following:
        export OS_TENANT_NAME=admin
        export OS_USERNAME=admin
        export OS_PASSWORD=openstacktest
        export OS_AUTH_URL="http://192.168.50.160:5000/v2.0/"

# Load it:
source keystone_source

# To test Keystone, just use a simple CLI command:
keystone user-list
```

4. Glance

We Move now to Glance installation:
apt-get install -y glance

Verify your glance services are running:
service glance-api status
service glance-registry status

Sửa file /etc/glance/glance-api.conf với nội dung:

```
[DEFAULT]
default_store = file
bind_host = 0.0.0.0
bind_port = 9292
log_file = /var/log/glance/api.log
backlog = 4096
workers = 1
registry_host = 0.0.0.0
registry_port = 9191
registry_client_protocol = http
rabbit_host = localhost
rabbit_port = 5672
rabbit_use_ssl = false
rabbit_userid = guest
rabbit_password = guest
rabbit_virtual_host = /
rabbit_notification_exchange = glance
rabbit_notification_topic = notifications
rabbit_durable_queues = False
qpid_notification_exchange = glance
qpid_notification_topic = notifications
```

qpid_hostname = localhost

qpid_port = 5672

qpid_username =

qpid_password =

qpid_sasl_mechanisms =

qpid_reconnect_timeout = 0

qpid_reconnect_limit = 0

qpid_reconnect_interval_min = 0

qpid_reconnect_interval_max = 0

qpid_reconnect_interval = 0

qpid_heartbeat = 5

qpid_protocol = tcp

qpid_tcp_nodelay = True

filesystem_store_datadir = /var/lib/glance/images/

swift_store_auth_version = 2

swift_store_auth_address = 127.0.0.1:5000/v2.0/

swift_store_user = jdoe:jdoe

swift_store_key = a86850deb2742ec3cb41518e26aa2d89

swift_store_container = glance

swift_store_create_container_on_put = False

swift_store_large_object_size = 5120

swift_store_large_object_chunk_size = 200

swift_enable_snet = False

s3_store_host = 127.0.0.1:8080/v1.0/

s3_store_access_key = <20-char AWS access key>

s3_store_secret_key = <40-char AWS secret key>

s3_store_bucket = <lowercased 20-char aws access key>glance

s3_store_create_bucket_on_put = False

sheepdog_store_address = localhost

```
sheepdog_store_port = 7000
sheepdog_store_chunk_size = 64
delayed_delete = False
scrub_time = 43200
scrubber_datadir = /var/lib/glance/scrubber
image_cache_dir = /var/lib/glance/image-cache/
[database]
# sqlite_db = /var/lib/glance/glance.sqlite
backend = sqlalchemy
connection = mysql://glance:$MYSQL_PASS@192.168.50.160/glance
[keystone_authtoken]
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = $ADMIN_PASS
[paste_deploy]
flavor=keystone
[store_type_location_strategy]
```

Sửa file /etc/glance/glance-registry.conf với nội dung:

```
[DEFAULT]
bind_host = 0.0.0.0
bind_port = 9191
log_file = /var/log/glance/registry.log
backlog = 4096
api_limit_max = 1000
limit_param_default = 25
[database]
```

```
backend = sqlalchemy
connection = mysql://glance:$MYSQL_PASS@192.168.50.160/glance
[keystone_authtoken]
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = $ADMIN_PASS
[paste_deploy]
flavor=keystone
```

Remove Glance's SQLite database:
rm /var/lib/glance/glance.sqlite

Restart the glance-api and glance-registry services:
service glance-api restart; service glance-registry restart

Synchronize the glance database:
glance-manage db_sync

Restart the services again to take into account the new modifications:
service glance-registry restart; service glance-api restart

To test Glance, upload the cirros cloud image and Ubuntu cloud image:
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

(mind you will be able to access VMs created with such image with the following credentials: user:cirros passwd: cubswin:))

wget http://cloud-images.ubuntu.com/precise/current/precise-server-cloudimg-amd64-disk1.img

glance add name="Ubuntu 12.04 cloudimg amd64" is_public=true container_format=ovf disk_format=qcow2 < ./precise-server-cloudimg-amd64-disk1.img

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

5. Neutron

5.1. OpenVSwitch

Install the openVSwitch:
apt-get install -y openvswitch-controller openvswitch-switch openvswitch-datapath-dkms

Restart openVSwitch:
service openvswitch-switch restart

Create the bridges:
#br-int will be used for VM integration
ovs-vsctl add-br br-int

#br-ex is used to make VMs to access the internet
ovs-vsctl add-br br-ex

5.2. OpenVSwitch

This will guide you to setting up the br-ex interface. Edit the eth1 in /etc/network/interfaces to become like this:

```
# VM internet Access
auto eth0
iface eth0 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 eth1 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 eth0

If you want to get internet connection back, you can assign the eth1's IP address to the br-ex in the /etc/network/interfaces file:

```
auto br-ex
iface br-ex inet static
```

address 192.168.60.160
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.50.1

If you want IMMEDIATELY want your FULL networking features back I suggest:
reboot
source keystone_source

5.3. Neutron-*

Install the Neutron components:
apt-get install -y neutron-server neutron-plugin-openvswitch neutron-plugin-openvswitch-agent dnsmasq neutron-dhcp-agent neutron-l3-agent neutron-metadata-agent

Verify all Neutron components are running:
cd /etc/init.d/; for i in $(ls neutron-*); do sudo service $i status; cd; done

Sửa file /etc/neutron/neutron.conf với nội dung sau:

```
[DEFAULT]
state_path = /var/lib/neutron
lock_path = \$state_path/lock
core_plugin = ml2
service_plugins = router
auth_strategy = keystone
allow_overlapping_ips = True
rpc_backend = neutron.openstack.common.rpc.impl_kombu

rabbit_host = 192.168.50.160
rabbit_password = $ADMIN_PASS
rabbit_userid = guest

notification_driver = neutron.openstack.common.notifier.rpc_notifier
notify_nova_on_port_status_changes = True
```

```
notify_nova_on_port_data_changes = True
nova_url = http://192.168.50.160:8774/v2
nova_admin_username = nova
nova_admin_tenant_id = $SERVICE_ID
nova_admin_password = $ADMIN_PASS
nova_admin_auth_url = http://192.168.50.160:35357/v2.0

[quotas]

[agent]
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf

[keystone_authtoken]
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = neutron
admin_password = $ADMIN_PASS
signing_dir = \$state_path/keystone-signing

[database]
connection = mysql://neutron:$MYSQL_PASS@192.168.50.160/neutron
[service_providers]
service_provider=LOADBALANCER:Haproxy:neutron.services.loadbalancer.drivers.haproxy.
plugin_driver.HaproxyOnHostPluginDriver:default
service_provider=VPN:openswan:neutron.services.vpn.service_drivers.ipsec.IPsecVPNDriver:
default
```

Sửa file /etc/neutron/plugins/ml2/ml2_conf.ini với nội dung sau:

```
[ml2]
type_drivers = gre
tenant_network_types = gre
mechanism_drivers = openvswitch

[ml2_type_flat]

[ml2_type_vlan]

[ml2_type_gre]
tunnel_id_ranges = 1:1000

[ml2_type_vxlan]

[securitygroup]
enable_security_group = True
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver

[ovs]
local_ip = $LOCAL_IP
tunnel_type = gre
enable_tunneling = True
```

Sửa file /etc/neutron/metadata_agent.ini với nội dung sau:

```
[DEFAULT]
verbose = True
auth_url = http://localhost:5000/v2.0
auth_region = RegionOne
admin_tenant_name = service
admin_user = neutron
admin_password = $ADMIN_PASS
nova_metadata_ip = 192.168.50.160
metadata_proxy_shared_secret = $METADATA_SECRET
```

Sửa file /etc/neutron/dhcp_agent.ini với nội dung sau:

```
verbose = True
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
use_namespaces = True
```

Sửa file /etc/neutron/l3_agent.ini với nội dung sau:

```
[DEFAULT]
verbose = True
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
use_namespaces = True
```

Remove Neutron's SQLite database:

```
rm /var/lib/neutron/neutron.sqlite
```

Restart all neutron services:

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

and check status:

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

then check all neutron agents, hopefully you'll enjoy smiling faces :-)

```
neutron agent-list
```

6. Nova

6.1. KVM

Make sure that your hardware enables virtualization:

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

it could be necessary to do:

```
sudo modprobe kvm_intel
kvm-ok
```

Finally you should get:

INFO: /dev/kvm exists
KVM acceleration can be used

Let's go for KVM installation:
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 and dbus to load the new values:
service dbus restart && service libvirt-bin restart

then check status:
service dbus status && service libvirt-bin status

6.2. 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 nova-compute-kvm

Kiểm tra tất cả các dịch vụ nova:
cd /etc/init.d/; for i in $(ls nova-*); do service $i status; cd; done

Sửa file /etc/nova/nova.conf với nội dung như sau:

```
[DEFAULT]
dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
iscsi_helper=tgtadm
libvirt_use_virtio_for_bridges=True
connection_type=libvirt
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf
verbose=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
volumes_path = /var/lib/nova/volumes
enabled_apis = ec2,osapi_compute,metadata

# Khai bao GLANCE
glance_host = 192.168.50.160

# Khai bao RABBITMQ
rpc_backend = rabbit
rabbit_host = 192.168.50.160
rabbit_userid = guest
rabbit_password = $RABBIT_PASS
```

```
# Cau hinh VNC
my_ip = 192.168.50.160
vncserver_listen = 192.168.50.160
vncserver_proxyclient_address = 192.168.50.160
auth_strategy = keystone
novncproxy_base_url = http://192.168.50.160:6080/vnc_auto.html

# Tu dong start may ao khi reboot server openstack
resume_guests_state_on_host_boot=True

#Cho phep dat password cho Instance khi khoi tao
libvirt_inject_password = True
enable_instance_password = True

network_api_class = nova.network.neutronv2.api.API
neutron_url = http://192.168.50.160:9696
neutron_auth_strategy = keystone
neutron_admin_tenant_name = service
neutron_admin_username = neutron
neutron_admin_password = $ADMIN_PASS
neutron_admin_auth_url = http://192.168.50.160:35357/v2.0
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver
security_group_api = neutron
service_neutron_metadata_proxy = true
neutron_metadata_proxy_shared_secret = $METADATA_SECRET

[database]
connection = mysql://nova:$MYSQL_PASS@192.168.50.160/nova
```

```
[keystone_authtoken]
auth_uri = http://192.168.50.160:5000
auth_host = 192.168.50.160
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = $ADMIN_PASS
```

Restart and check nova-* services:
cd /etc/init.d/; for i in $(ls nova-*); do sudo service $i restart; cd /root/;done
cd /etc/init.d/; for i in $(ls nova-*); do sudo service $i status; cd /root/;done
(mind nova cert is ok if it's down: still the db has to be built up!)

Remove Nova's SQLite database:
rm /var/lib/nova/nova.sqlite

Synchronize your database:
nova-manage db sync

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

...and check:
cd /etc/init.d/; for i in $(ls nova-*); do sudo service $i status; cd /root/;done

Hopefully you should enjoy smiling faces on nova-* services to confirm your installation:
nova-manage service list

7. 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

# Start 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 = 192.168.50.160
        service_port = 5000
        auth_host = 10.1.3.160
        auth_port = 35357
        auth_protocol = http
        admin_tenant_name = service
        admin_user = cinder
        admin_password = openstacktest

# Edit the /etc/cinder/cinder.conf to:
        [DEFAULT]
        rootwrap_config=/etc/cinder/rootwrap.conf
        sql_connection = mysql://cinder:openstacktest@10.1.3.160/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
        #osapi_volume_listen_port=5900

# Remove Cinder's SQLite database:

rm /var/lib/cinder/cinder.sqlite

#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.
Restart the cinder services:
cd /etc/init.d/; for i in $( ls cinder-* ); do sudo service $i restart; cd /root/; done

Verify if cinder services are running:
cd /etc/init.d/; for i in $( ls cinder-* ); do sudo service $i status; cd /root/; done
```

8. Horizon

```
# To install horizon, proceed like this
apt-get -y install 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
```

 # You can now access your OpenStack 192.168.50.160/horizon with credentials admin:openstacktest.

III. Create VMs

Việc tạo network và tạo máy ảo bạn hoàn toàn có thể thực hiện thông qua giao diện. Có thể tham khảo 2 bài viết sau:
http://mhst13-06-openstack.blogspot.com/2013/07/huong-dan-su-dung-openstack-dashboard.html
http://mhst13-06-openstack.blogspot.com/2013/07/huong-dan-su-dung-openstack-dashboard_23.html

Có một chú ý là khi ta tạo máy ảo, máy ảo không ra được internet có thể do sai dns. Cách khắc phục:
liệt kê các dải mạng
neutron subnet-list

cấu hình DNS cho dải mạng
neutron subnet-update 44453bfa-14b9-445c-b8f4-33d7ba22455f --dns_nameservers 8.8.8.8 8.8.4.4
Với 44453bfa-14b9-445c-b8f4-33d7ba22455f là id của network máy ảo 10.10.10.0/24