

# Hướng dẫn cài đặt Openstack Havana

## (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.

### Mục lục

- I. Requirements
- II. Install
  - Add Havana repositories
  - Update system
  - 1. MySQL & RabbitMQ
  - 2. Databases
  - Others
  - 3. Keystone
  - 4. Glance
  - 5. Neutron
    - 5.1. OpenVSwitch
    - 5.2. OpenVSwitch
    - 5.3. Neutron-\*
  - 6. Nova
    - 6.1. KVM
    - 6.2. Nova-\*
  - 7. Cinder
  - 8. Horizon
- III. Create VMs

### **Chú thích:**

Màu A: Lệnh

Màu B: Nội dung file config cần chỉnh sửa

Bài viết dựa trên hướng dẫn của tác giả Fornyx tại:

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

`apt-get install python-software-properties`

`add-apt-repository cloud-archive:havana`

### # 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
```

# Update the /etc/glance/glance-api-paste.ini and /etc/glance/glance-registry-paste.ini with:

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = 10.1.3.160
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = openstacktest
```

# Update /etc/glance/glance-api.conf and /etc/glance/glance-registry.conf with:

```
[DEFAULT]
sql_connection = mysql://glance:openstacktest@10.1.3.160/glance

[keystone_authtoken]
auth_host = 10.1.3.160
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = openstacktest

[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`

# Edit /etc/neutron/api-paste.ini

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = 10.1.3.160
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = neutron
admin_password = openstacktest
```

# Edit the OVS plugin configuration file /etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini with:

```
#Under the database section
[DATABASE]
sql_connection=mysql://neutron:openstacktest@10.1.3.160/neutron

#Under the OVS section
[OVS]
tenant_network_type = gre
enable_tunneling = True
tunnel_id_ranges = 1:1000
integration_bridge = br-int
tunnel_bridge = br-tun
local_ip = 10.1.3.160

#Firewall driver for realizing neutron security group function
[SECURITYGROUP]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
```

# Update /etc/neutron/metadata\_agent.ini:

```
# The Neutron user information for accessing the Neutron API.
auth_url = http://10.1.3.160:35357/v2.0
auth_region = RegionOne
admin_tenant_name = service
admin_user = neutron
admin_password = openstacktest

# IP address used by Nova metadata server
nova_metadata_ip = 10.1.3.160

# TCP Port used by Nova metadata server
```



```
nova_metadata_port = 8775
```

```
metadata_proxy_shared_secret = helloOpenStack
```

# Edit your /etc/neutron/neutron.conf:

```
#RabbitMQ IP
```

```
rabbit_host = 10.1.3.160
```

```
[keystone_authtoken]
```

```
auth_host = 10.1.3.160
```

```
auth_port = 35357
```

```
auth_protocol = http
```

```
admin_tenant_name = service
```

```
admin_user = neutron
```

```
admin_password = openstacktest
```

```
signing_dir = /var/lib/neutron/keystone-signing
```

```
[DATABASE]
```

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

# Edit your /etc/neutron/l3\_agent.ini:

```
[DEFAULT]
```

```
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
```

```
use_namespaces = True
```

```
external_network_bridge = br-ex
```

```
signing_dir = /var/cache/neutron
```

```
admin_tenant_name = service
```

```
admin_user = neutron
```

```
admin_password = openstacktest
```

```
auth_url = http://10.1.3.160:35357/v2.0
```

```
l3_agent_manager = neutron.agent.l3_agent.L3NATAgentWithStateReport
```

```
root_helper = sudo neutron-rootwrap /etc/neutron/rootwrap.conf
```

```
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
```

# Edit your /etc/neutron/dhcp\_agent.ini:

```
[DEFAULT]
```

```
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
```

```
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
```

```
use_namespaces = True
```

```
signing_dir = /var/cache/neutron
admin_tenant_name = service
admin_user = neutron
admin_password = openstacktest
auth_url = http://10.1.3.160:35357/v2.0
dhcp_agent_manager = neutron.agent.dhcp_agent.DhcpAgentWithStateReport
root_helper = sudo neutron-rootwrap /etc/neutron/rootwrap.conf
state_path = /var/lib/neutron
```

# 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`

# Check the status of all nova-services:

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

# 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 = 10.1.3.160
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = openstacktest
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=10.1.3.160
nova_url=http://10.1.3.160:8774/v1.1/
sql_connection=mysql://nova:openstacktest@10.1.3.160/nova
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf
```

```
# Auth
use_deprecated_auth=false
auth_strategy=keystone
```

```
# Imaging service
glance_api_servers=10.1.3.160:9292
image_service=nova.image.glance.GlanceImageService
```

```
# Vnc configuration
novnc_enabled=true
novncproxy_base_url=http://192.168.50.160:6080/vnc_auto.html
novncproxy_port=6080
vncserver_proxycient_address=10.1.3.160
vncserver_listen=0.0.0.0
```

```

# Network settings
network_api_class=nova.network.neutronv2.api.API
neutron_url=http://10.1.3.160:9696
neutron_auth_strategy=keystone
neutron_admin_tenant_name=service
neutron_admin_username=neutron
neutron_admin_password=openstacktest
neutron_admin_auth_url=http://10.1.3.160:35357/v2.0
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSIfaceDriver
#If you want Neutron + Nova Security groups
#firewall_driver=nova.virt.firewall.NoopFirewallDriver
#security_group_api=neutron
#If you want Nova Security groups only, comment the two lines above and uncomment line -1-.
#-1-firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver

#Metadata
service_neutron_metadata_proxy = True
neutron_metadata_proxy_shared_secret = helloOpenStack
metadata_host = 10.1.3.160
metadata_listen = 0.0.0.0
metadata_listen_port = 8775

# Compute #
compute_driver=libvirt.LibvirtDriver

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

# Edit the /etc/nova/nova-compute.conf:
[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

```

# 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 volume group 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:

```
pvccreate /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](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
```