

## Contents

- 1 Cisco OpenStack Edition: Folsom Manual Installation
  - ◆ 1.1 Introduction
  - ◆ 1.2 Dependencies
    - ◇ 1.2.1 Critical Reminders
    - ◇ 1.2.2 Operating System
    - ◇ 1.2.3 Nodes
    - ◇ 1.2.4 Network
  - ◆ 1.3 Installation
    - ◇ 1.3.1 Install the Controller Node (control03)
      - 1.3.1.1 Preparing Ubuntu 12.04
      - 1.3.1.2 Networking
      - 1.3.1.3 Time Synchronization
      - 1.3.1.4 MySQL & RabbitMQ
      - 1.3.1.5 Keystone Installation
      - 1.3.1.6 Glance Installation
      - 1.3.1.7 Quantum Installation
      - 1.3.1.8 Nova Installation
      - 1.3.1.9 Horizon Installation
    - ◇ 1.3.2 Install the Network Node (control02)
      - 1.3.2.1 Preparing Ubuntu 12.04
      - 1.3.2.2 Networking
      - 1.3.2.3 Time Synchronization
      - 1.3.2.4 Quantum Installation
    - ◇ 1.3.3 Install the Compute Node (compute01)
      - 1.3.3.1 Preparing Ubuntu 12.04
      - 1.3.3.2 Networking
      - 1.3.3.3 Time Synchronization
      - 1.3.3.4 KVM Installation
      - 1.3.3.5 Quantum Installation
      - 1.3.3.6 Nova Installation
    - ◇ 1.3.4 Your First Instance
  - ◆ 1.4 Credits

# Cisco OpenStack Edition: Folsom Manual Installation

## Introduction

There are two common ways of installing OpenStack, manually or via automation. There is much focus on the full automation of OpenStack deployments using automation tools such as Puppet, Chef, JuJu and others. While these tools offer great advantages over manual configuration, they do hide the under-workings from those who need to learn the installation flow and setup steps that are happening behind the scenes of an OpenStack. This document can be used by those who want to learn more about the OpenStack Folsom installation process. The document covers the following OpenStack software components:

- Nova (Compute Service)
- Glance (Image Service)
- Keystone (Identity Service)
- Horizon (OpenStack Dashboard Web User Interface)
- Quantum (Network Service)

## Dependencies

### Critical Reminders

The most common issues that people have in manual OpenStack deployments are: basic mistakes with typos, incorrect IP addresses in configuration files and incorrect passwords in configuration files. To save you from many troubleshooting steps later down the road, ENSURE that you are double and triple checking the configuration files and commands when an account, password and/or IP address are used. You will likely be using your own IP addressing and passwords in your setup and it is critical to ensure you get them right on each node.

The password used in this setup = Cisco123. Every account, service and configuration file uses this one password. You will want to change this in your setup and you certainly want to use a strong password and a different password for each account/service if this system is going into production.

### Operating System

The operating system used for this installation is Ubuntu 12.04 LTS (Precise).

### Nodes

This document uses three physical servers (Cisco UCS B or C-series) to serve the roles of Controller, Compute, Network. Additional Compute Nodes can be deployed by repeating the steps in the Compute Node Installation Section. The three distinct node types used in this document are:

- **Controller Node**
  - ◆ Runs Nova API, Nova Cert, Nova Consoleauth, Nova Novncproxy, Nova Scheduler, Novnc, Quantum Server, Quantum Plugin OVS, Glance API/Registry, and Keystone services.
  - ◆ Provides control plane functionality for managing the OpenStack environment.
- **Compute Node**
  - ◆ Runs Nova Compute, Quantum Plugin OVS, and OVS Plugin Agent services.
  - ◆ Provides the hypervisor role for running Nova instances (Virtual Machines).
- **Network Node**

- ◆ Runs Quantum DHCP, Quantum L3 Agent, Quantum Plugin OVS, OVS Plugin Agent, DNSMASQ (Base and Util) services.
- ◆ Provides network services such as DHCP, network access, and routing for Nova instances running on the Compute Node.

## Network

The network design referenced in this document uses 3 logically (VLAN) separated networks. While VLAN's are used for accessing the OpenStack Nodes, connectivity within Tenants uses Quantum Open vSwitch (OVS) GRE tunnels. GRE is easier to deploy, especially in larger environments and does not suffer from the scalability limitations of VLAN's. The networks are defined below:

- **Management and CIMC (Cisco Integrated Management Controller for UCS) Network**
  - ◆ This network is used to perform management functions against the node. Examples include SSH to the nodes and the controller node hosting Horizon would listen for incoming connections on this network. The network is also used for lights-out management using the CIMC interface of the UCS servers.
  - ◆ An IP address for each node is required for this network.
  - ◆ This network typically employs private ([RFC1918](#)) IP addressing.
- **Public/API Network**
  - ◆ This network is used for assigning Floating IP addresses to instances for communicating outside of the OpenStack Cloud.
  - ◆ The Nova Metadata Service used for injecting information into instances (i.e. SSH keys) is attached to this network on the Controller Node.
  - ◆ The Controller Node and Network Node will have an interface attached to this network.
  - ◆ An IP address for the Controller Node is required for this network.
  - ◆ This network typically employs publicly routable IP addressing if external NAT'ing is not used upstream towards the Internet edge (**Note:** in this document all IP addressing for all interfaces comes out of various private addressing blocks)
- **Data Network (AKA: Private Network or Instance Network)**
  - ◆ This network is used for providing connectivity to OpenStack Instances (Virtual Machines). A separate data network allows us to segment Tenant data traffic from management traffic.
  - ◆ Node interfaces are attached to this network for terminating OVS GRE tunnels.
  - ◆ In this document, each node is assigned an interface and IP address from this network.
  - ◆ This network typically employs private ([RFC1918](#)) IP addressing.

**Figure 1** is used to help visualize the setup and to act as a reference for configuration steps later on in the document. We highly recommend that you print the diagram so you can easily reference it throughout the installation process. A summary of the network topology is as follows:

- Controller Node
  - ◆ Hostname: control03
  - ◆ Single physical NIC used to logically separate three networks
    - ◇ eth0 connects to the Management/CIMC network which is on VLAN220 (VLAN 220 is the Native VLAN on the upstream Layer 2 switch)
      - eth0 IP address: 192.168.220.43
    - ◇ eth0.221 connects to the Public/API network on VLAN221
      - eth0.221 IP address: 192.168.221.43
    - ◇ eth0.223 connects to the Data network
      - eth0.223 IP address: 10.0.0.43
    - ◇ CIMC 0 connects to the Management/CIMC network
      - CIMC 0 IP address: 192.168.220.13
- Compute Node

- ◆ Hostname: compute01
- ◆ Single physical NIC used to logically separate three networks
  - ◇ eth0 connects to the Management/CIMC network which is on VLAN220 (VLAN 220 is the Native VLAN on the upstream Layer 2 switch)
    - eth0 IP address: 192.168.220.51
  - ◇ eth0.223 connects to the Data network
    - eth0.223 IP address: 10.0.0.51
  - ◇ CIMC 0 connects to the Management/CIMC network
    - CIMC 0 IP address: 192.168.220.4
- Network Node
  - ◆ Hostname: control02
  - ◆ Single physical NIC used to logically separate three networks
    - ◇ eth0 connects to the Management/CIMC network which is on VLAN220 (VLAN 220 is the Native VLAN on the upstream Layer 2 switch)
      - eth0 IP address: 192.168.220.42
    - ◇ eth0.221 connects to the Public/API network on VLAN221
      - eth0.221 No IP address is set for this interface (see notes later in document on OVS/Quantum setup)
    - ◇ eth0.223 connects to the Data network
      - eth0.223 IP address: 10.0.0.42
    - ◇ CIMC 0 connects to the Management/CIMC network
      - CIMC 0 IP address: 192.168.220.3



- Other Network Services
  - ◆ DNS: In this setup an external DNS server is used for name resolution for OpenStack nodes and external name resolution.
  - ◆ NTP: In this setup an external NTP server(s) is used for time synchronization
  - ◆ Physical Network Switches: Each node in this setup is physically attached to a Cisco Nexus switch acting as a Top-of-Rack access layer device. Trunking is configured on each interface connecting to the eth0 NIC of each node. **Note:** Upstream routers/aggregation layer switches

will most likely be terminating the Layer-3 VLAN interfaces. If these interfaces are deployed in a redundant fashion with a First Hop Redundancy Protocol like HSRP or VRRP, then you need to be careful of the IP addresses assigned to the physical L3 switches/routers as they may conflict with the IP address of the Quantum router on the public subnet (usually assigned .3 address). For example, if you are using HSRP and you have .1 as the standby IP address, .2 as the first L3 switch IP and .3 as the second L3 switch IP, you will receive a duplicate IP address error on the second L3 switch. This can be worked around by using high-order IPs on your upstream L3 device or altering the Quantum subnet configuration at the time of creation to have an IP starting range higher than the physical switches/routers are using (i.e. .4 and higher).

## Installation

The installation of the nodes will be in the following order:

1. Controller Node
2. Network Node
3. Compute Node

### Install the Controller Node (control03)

#### Preparing Ubuntu 12.04

Install Ubuntu 12.04 (AMD 64-bit) from CD/ISO or automated install (i.e. kickstart). You can reference Section 4 in the [Build Node Guide](#) if you are unfamiliar with the Ubuntu Precise installation process. Use the networking information from the beginning of the document to configure your network properties. Select ssh-server as the only additional package.

- Use sudo mode or run from root account for the entire installation:

```
sudo su
```

- Add the Cisco Edition repository::

```
vi /etc/apt/sources.list.d/cisco-openstack-mirror_folsom.list
# cisco-openstack-mirror_folsom
deb ftp://ftpeng.cisco.com/openstack/cisco folsom main
deb-src ftp://ftpeng.cisco.com/openstack/cisco folsom main
```

- You will receive the following error when trying to run apt-get update:

```
GPG error: ftp://ftpeng.cisco.com folsom InRelease:
The following signatures couldn't be verified because the public key is not available:
NO_PUBKEY E8CC67053ED3B199
```

- Add the apt key to remove the error:

```
gpg --keyserver hkp://pgpkeys.mit.edu --recv-keys E8CC67053ED3B199
gpg --armor --export E8CC67053ED3B199 | apt-key add -
```

- **Note:** If you have issues using the pgpkeys.mit.edu site you can use keyserver.ubuntu.com instead:

```
gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys E8CC67053ED3B199
```

- Update your system:

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

### Networking

- Our implementation uses VLANs for network separation. Make sure the VLAN package is installed and your network switches have been configured for VLANs. Otherwise, replicate the network setup using physical interfaces:

```
apt-get install vlan
```

- Controller Node (control03) /etc/network/interfaces:

```
# The loopback network interface
auto lo iface lo inet loopback

#Management Network
auto eth0
iface eth0 inet static
    address 192.168.220.43
    netmask 255.255.255.0
    gateway 192.168.220.1
    dns-nameservers 192.168.220.254
    dns-search dmz-pod2.lab

#VM Network with OVS in tunnel mode

auto eth0.223
iface eth0.223 inet static
    vlan-raw-device eth0
    address 10.0.0.43
    netmask 255.255.255.0

#Public/API Network: Bridged Interface

auto eth0.221
iface eth0.221 inet static
    vlan-raw-device eth0
    address 192.168.221.43
    netmask 255.255.255.0
```

### Time Synchronization

- Install NTP:

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

- Configure NTP:

```
sed -i 's/server ntp.ubuntu.com/server ntp.ubuntu.com\nserver 127.127.1.0\nfudge 127.127.1.0 strat\nservice ntp restart
```

### MySQL & RabbitMQ

- Install MySQL. **Note:** You will be prompted for the root password for MySQL. Document this password as it will be needed later when creating databases for OpenStack services:

```
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
```

- Create a RabbitMQ user account that will be used by OpenStack services:

```
rabbitmqctl add_user openstack_rabbit_user Cisco123
```

- Create the RabbitMQ vhost used by Quantum:

```
rabbitmqctl add_vhost /quantum
```

- Set the permissions for the new RabbitMQ user account:

```
rabbitmqctl set_permissions -p / openstack_rabbit_user ".*" ".*" ".*"
rabbitmqctl set_permissions -p /quantum openstack_rabbit_user ".*" ".*" ".*"
```

### Keystone Installation

- Install Keystone:

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

- Create a MySQL database for Keystone (use the root password that was created during original MySQL install). **Note:** All services and DB accounts will use Cisco123:

```
mysql -u root -p
CREATE DATABASE keystone;
GRANT ALL ON keystone.* TO 'keystone_admin'@'%' IDENTIFIED BY 'Cisco123';
quit;
```

- Edit the /etc/keystone/keystone.conf to the new database:

```
[DEFAULT]
admin_token = keystone_admin_token

[sql]
connection = mysql://keystone_admin:Cisco123@192.168.220.43/keystone
```

- Test whether MySQL is listening on 192.168.220.43 for the Keystone database:

```
mysql -h192.168.220.43 -ukeystone_admin -pCisco123 keystone
```

- Restart the identity service then synchronize the database:

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

- Download the [Keystone data script](#)(Provided by Emilien Macchi) and edit the following fields in the script:

```
ADMIN_PASSWORD=${ADMIN_PASSWORD:-Cisco123}
```

## Cisco\_OpenStack\_Edition:\_Folsom\_Manual\_Install

```
export SERVICE_TOKEN="keystone_admin_token"
export SERVICE_ENDPOINT="http://192.168.220.43:35357/v2.0/"
SERVICE_TENANT_NAME=${SERVICE_TENANT_NAME:-services}
```

- Run the script to populate the Keystone database with data (users, tenants, services)

```
./keystone-data.sh
```

- Download the [Keystone endpoint script](#) (Provided by Emilien Macchi) and create the OpenStack service endpoints:

```
# MySQL definitions
MYSQL_USER=keystone_admin
MYSQL_DATABASE=keystone
MYSQL_HOST=192.168.220.43
MYSQL_PASSWORD=Cisco123

# Keystone definitions
KEYSTONE_REGION=RegionOne
SERVICE_TOKEN=keystone_admin_token
SERVICE_ENDPOINT="http://192.168.220.43:35357/v2.0"

# other definitions. This should be your Controller Node IP address.
MASTER="192.168.220.43"
```

- Run the script to populate the Keystone database with service endpoints. **Note:** If you logout or reboot after running the keystone-data.sh script then you must re-export the following before running the keystone-endpoints.sh script:

```
export SERVICE_TOKEN="keystone_admin_token"
export SERVICE_ENDPOINT="http://192.168.220.43:35357/v2.0/"

./keystone-endpoints.sh
```

- Create a credential file and load it so you don not have to provide credentials for every CLI command. **Note:** This needs to be created on each node that you will run OpenStack commands from:

```
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=Cisco123
export OS_AUTH_URL="http://192.168.220.43:5000/v2.0/"
export OS_AUTH_STRATEGY=keystone
export SERVICE_TOKEN=keystone_admin_token
export SERVICE_ENDPOINT=http://192.168.220.43:35357/v2.0/
```

- Load it:

```
source openrc
```

- To test Keystone, install curl and use a curl request :

```
apt-get install curl openssl -y
curl -d '{"auth": {"tenantName": "admin", "passwordCredentials":{"username": "admin", "password": "application/json"}' http://192.168.220.43:35357/v2.0/tokens | python -mjson.tool
```

- Or you can use the Keystone client to verify the configuration:



```
keystone user-list
keystone tenant-list
keystone service-list
keystone endpoint-list
```

### Glance Installation

- Install Glance packages:

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

- Create a MySQL database for Glance (use the root password that was created during original MySQL install):

```
mysql -u root -p
CREATE DATABASE glance;
GRANT ALL ON glance.* TO 'glance'@'%' IDENTIFIED BY 'Cisco123';
quit;
```

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

```
[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_host = 192.168.220.43
auth_port = 35357
auth_protocol = http
admin_tenant_name = services
admin_user = glance
admin_password = Cisco123
```

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

```
[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_host = 192.168.220.43
auth_port = 35357
auth_protocol = http
admin_tenant_name = services
admin_user = glance
admin_password = Cisco123
```

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

```
sql_connection = mysql://glance:Cisco123@192.168.220.43/glance

[paste_deploy]
flavor = keystone
```

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

```
sql_connection = mysql://glance:Cisco123@192.168.220.43/glance

[paste_deploy]
flavor = keystone
```

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

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

## Cisco\_OpenStack\_Edition:\_Folsom\_Manual\_Install

- Synchronize the glance database (You may get a message about deprecation - you can ignore):

```
glance-manage db_sync
```

- Restart the services again to take into account the new modifications:

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

- Upload an image to Glance. Start by downloading the Ubuntu Precise cloud image to the Controller node and then uploading it to Glance:

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

```
glance add name="precise" is_public=true container_format=ovf disk_format=qcow2 < precise-server-c
```

- Verify that the image is available through Glance:

```
glance image-list
```

## Quantum Installation

- Install the Quantum Server on the Controller Node:

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

- Create a database (use root password that was created during original MySQL install):

```
mysql -u root -p
CREATE DATABASE quantum;
GRANT ALL ON quantum.* TO 'quantum'@'%' IDENTIFIED BY 'Cisco123';
quit;
```

- Edit `/etc/quantum/quantum.conf`. As part of the configuration, we will disable overlapping IP address support. This is needed to support the Nova metadata service and/or Nova security groups. More can be found at [Quantum Limitations](#):

```
allow_overlapping_ips = False
fake_rabbit = False
rabbit_virtual_host=/quantum
rabbit_userid=openstack_rabbit_user
rabbit_password=Cisco123
rabbit_host=192.168.220.43
rabbit_port=5672
```

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

```
#Under the database section
[DATABASE]
sql_connection = mysql://quantum:Cisco123@192.168.220.43/quantum

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

```
tenant_network_type=gre
```

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

```
[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_host=192.168.220.43
auth_port = 35357
auth_protocol = http
admin_tenant_name=services
admin_user=quantum
admin_password=Cisco123
```

- Restart the quantum server:

```
service quantum-server restart
```

### Nova Installation

- Start by installing nova software packages:

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

- Prepare a MySQL database for Nova (use the root password that was created during original MySQL install):

```
mysql -u root -p
CREATE DATABASE nova;
GRANT ALL ON nova.* TO 'nova'@'%' IDENTIFIED BY 'Cisco123';
quit;
```

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

```
[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_host = 192.168.220.43
auth_port = 35357
auth_protocol = http
auth_uri = http://192.168.220.43:35357/v2.0
admin_tenant_name = services
admin_user = nova
admin_password = Cisco123
```

- Replace the contents of the /etc/nova/nova.conf with the following - take note that the IP address of the "metadata\_listen" variable is the "control03" Controller Node eth0.221 interface in the diagram:

```
[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
ec2_listen=192.168.220.43
rabbit_port=5672
rabbit_virtual_host=/
rabbit_password=Cisco123
rabbit_userid=openstack_rabbit_user
rabbit_host=192.168.220.43
metadata_listen=192.168.221.43
```

## Cisco\_OpenStack\_Edition:\_Folsom\_Manual\_Install

```
sql_connection=mysql://nova:Cisco123@192.168.220.43/nova
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf

# Auth
use_deprecated_auth=false
auth_strategy=keystone

# Imaging service
glance_api_servers=192.168.220.43:9292
image_service=nova.image.glance.GlanceImageService

# VNC configuration
novncproxy_port=6080
novncproxy_host=0.0.0.0
novnc_enabled=true
novncproxy_base_url=http://192.168.220.43:6080/vnc_auto.html

# Network settings
network_api_class=nova.network.quantumv2.api.API
quantum_url=http://192.168.220.43:9696
quantum_auth_strategy=keystone
quantum_admin_auth_url=http://192.168.220.43:35357/v2.0
quantum_admin_password=Cisco123
quantum_admin_username=quantum
quantum_admin_tenant_name=services
libvirt_use_virtio_for_bridges=True
connection_type=libvirt
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybirdOVSBridgeDriver
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSIfaceDriver
```

- Synchronize the Nova database (You may get a DEBUG message - You can ignore this):

```
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
```

- Also check that nova-api is running:

```
service nova-api status
```

### Horizon Installation

- To install horizon, proceed like this :

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

- Reload Apache and memcached:

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

- Access Horizon by using the following URL in your web browser. Use **admin/Cisco123** for your login credentials **Note:** A reboot may be needed for a successful login:

<http://192.168.220.43/horizon>

## Install the Network Node (control02)

### Preparing Ubuntu 12.04

Install Ubuntu 12.04 (AMD 64-bit) from CD/ISO or automated install (i.e. kickstart). You can reference Section 4 in the [Build Node Guide](#) if you are unfamiliar with the Ubuntu Precise installation process. Use the networking information from the beginning of the document to configure your network properties. Select ssh-server as the only additional package.

- Use sudo mode or run from root account for the entire installation:

```
sudo su
```

- Add the Cisco Edition repository::

```
vi /etc/apt/sources.list.d/cisco-openstack-mirror_folsom.list
# cisco-openstack-mirror_folsom
deb http://128.107.252.163/openstack/cisco folsom main
deb-src http://128.107.252.163/openstack/cisco folsom main
```

- You will receive the following error when trying to run apt-get update:

```
GPG error: http://128.107.252.163 folsom InRelease:
The following signatures couldn't be verified because the public key is not available: NO_PUBKEY E
```

- Add the apt key to remove the error:

```
gpg --keyserver hkp://pgpkeys.mit.edu --recv-keys E8CC67053ED3B199
gpg --armor --export E8CC67053ED3B199 | apt-key add -
```

- **Note:** If you have issues using the pgpkeys.mit.edu site you can use keyserver.ubuntu.com instead:

```
gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys E8CC67053ED3B199
```

- Update your system:

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

### Networking

- Our implementation uses VLANs for network separation. Make sure you have the VLAN package installed and your network switches have been configured for VLANs:

```
apt-get install vlan
```

- Network Node (control02) /etc/network/interfaces. **Note:** The Public/API facing NIC (eth0.221) on the Network node does not have an IP address assigned:

```
# The loopback network interface
auto lo
iface lo inet loopback

# Management Network
auto eth0
```

```
iface eth0 inet static
address 192.168.220.42
netmask 255.255.255.0
gateway 192.168.220.1
dns-nameservers 192.168.220.254
dns-search dmz-pod2.lab

# VM Network with OVS in tunnel mode
auto eth0.223
iface eth0.223 inet static
vlan-raw-device eth0
address 10.0.0.42
netmask 255.255.255.0

# Public/API Network: Bridged Interface
auto eth0.221
iface eth0.221 inet manual
vlan-raw-device eth0
up ifconfig $IFACE 0.0.0.0 up
up ip link set $IFACE promisc on
down ifconfig $IFACE down
```

### Time Synchronization

- Install NTP:

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

- Configure NTP:

```
sed -i 's/server ntp.ubuntu.com/server ntp.ubuntu.com\nserver 127.127.1.0\nfudge 127.127.1.0 strat
service ntp restart
```

### Quantum Installation

- Install the Quantum openvswitch plugin, openvswitch agent, l3\_agent, and dhcp\_agent:

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

- Quantum dhcp\_agent uses dnsmasq by default. Verify that dnsmasq is installed:

```
dpkg -l | grep dnsmasq
ii dnsmasq-base          2.59-4          Small caching DNS proxy and DHCP
ii dnsmasq-utils        2.59-4          Utilities for manipulating DHCP
```

- If dnsmasq-base and dnsmasq-utils packages are not installed then install them manually:

```
apt-get install -y dnsmasq-base
apt-get install -y dnsmasq-utils
```

- The Network Node requires that an OVS bridge named "br-int" and "br-ex" exist, and that "br-ex" is associated with the Public/API interface (eth0.221 in this setup). In order for the below commands to take, you must reboot the Network Node at this point. If you don't and you attempt to add the bridges, you will receive errors related to db.sock. Once the node is rebooted create the bridges. To create them, run:

```
ovs-vsctl add-br br-int
ovs-vsctl add-br br-ex
ovs-vsctl add-port br-ex eth0.221
```

## Cisco\_OpenStack\_Edition:\_Folsom\_Manual\_Install

- If you still get db.sock errors, you will have to use the following workaround for this [bug](#):

```
kernel_version=`cat /proc/version | cut -d " " -f3`  
apt-get install -y dkms openvswitch-switch openvswitch-datapath-dkms linux-headers-$kernel_version  
apt-get autoremove openvswitch-datapath-dkms  
apt-get install -y dkms openvswitch-switch openvswitch-datapath-dkms linux-headers-$kernel_version  
/etc/init.d/openvswitch-switch restart
```

- Now add the bridges (only if the above workaround was needed). If you are unsure if the bridges need to be added, then use the `ovs-vsctl list-br` command to list the configured bridges:

```
ovs-vsctl add-br br-int  
ovs-vsctl add-br br-ex  
ovs-vsctl add-port br-ex eth0.221
```

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

```
[filter:authtoken]  
paste.filter_factory = keystone.middleware.auth_token:filter_factory  
auth_host=192.168.220.43  
auth_port = 35357  
auth_protocol = http  
admin_tenant_name=services  
admin_user=quantum  
admin_password=Cisco123
```

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

```
#Under the database section  
[DATABASE]  
sql_connection=mysql://quantum:Cisco123@192.168.220.43/quantum
```

- Edit the following under the OVS section. **Note:** Ensure the "local\_ip" value is correct. In this case it is the eth0.223 IP address on the Network node. If this is wrong, GRE and, therefore, Quantum won't work correctly:

```
[OVS]  
enable_tunneling = True  
tunnel_id_ranges = 1:1000  
integration_bridge=br-int  
tunnel_bridge = br-tun  
network_vlan_ranges=  
tenant_network_type=gre  
local_ip = 10.0.0.42
```

- Update the `/etc/quantum/l3_agent.ini`. (Ensure that the "metadata\_ip" is the same value set in the "metadata\_listen" entry in the `nova.conf` file on the Controller):

```
auth_url = http://192.168.220.43:35357/v2.0  
auth_region = RegionOne  
admin_tenant_name = services  
admin_user = quantum  
admin_password = Cisco123  
metadata_ip = 192.168.221.43  
metadata_port = 8775  
use_namespaces = True
```

- Update the `/etc/quantum/dhcp_agent.ini`:

```
use_namespaces = True
```

- Also, update your RabbitMQ settings in `/etc/quantum/quantum.conf`:

```
allow_overlapping_ips = False
fake_rabbit = False
rabbit_virtual_host=/quantum
rabbit_userid=openstack_rabbit_user
rabbit_password=Cisco123
rabbit_host=192.168.220.43
rabbit_port=5672
```

- Restart all the services:

```
service quantum-plugin-openvswitch-agent restart
service quantum-dhcp-agent restart
service quantum-l3-agent restart
```

## Install the Compute Node (compute01)

### Preparing Ubuntu 12.04

Install Ubuntu 12.04 (AMD 64-bit) from CD/ISO or automated install (i.e. kickstart). You can reference Section 4 in the [Build Node Guide](#) if you are unfamiliar with the Ubuntu Precise installation process. Use the networking information from the beginning of the document to configure your network properties. Select `ssh-server` as the only additional package.

- Use sudo mode or run from root account for the entire installation:

```
sudo su
```

- Add the Cisco Edition repository::

```
vi /etc/apt/sources.list.d/cisco-openstack-mirror_folsom.list
# cisco-openstack-mirror_folsom
deb http://128.107.252.163/openstack/cisco folsom main
deb-src http://128.107.252.163/openstack/cisco folsom main
```

- You will receive the following error when trying to run `apt-get update`:

```
GPG error: http://128.107.252.163 folsom InRelease:
The following signatures couldn't be verified because the public key is not available: NO_PUBKEY E
```

- Add the apt key to remove the error:

```
gpg --keyserver hkp://pgpkeys.mit.edu --recv-keys E8CC67053ED3B199
gpg --armor --export E8CC67053ED3B199 | apt-key add -
```

- **Note:** If you have issues using the `pgpkeys.mit.edu` site you can use `keyserver.ubuntu.com` instead:

```
gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys E8CC67053ED3B199
```

- Update your system:

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



## Networking

- Our implementation uses VLANs for network separation. Make sure you have the VLAN package installed and your network switches have been configured for VLANs:

```
apt-get install vlan
```

- Compute Node (compute01) /etc/network/interfaces. # Note: The Compute node does not have a NIC attached to the Public/API network (VLAN221):

```
# The loopback network interface
auto lo
iface lo inet loopback

# Management Network
auto eth0
iface eth0 inet static
    address 192.168.220.51
    netmask 255.255.255.0
    gateway 192.168.220.1
    dns-nameservers 192.168.220.254
    dns-search dmz-pod2.lab

# Data Network
auto eth0.223
iface eth0.223 inet static
    vlan-raw-device eth0
    address 10.0.0.51
    netmask 255.255.255.0
```

## Time Synchronization

- Install NTP:

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

- Configure NTP:

```
sed -i 's/server ntp.ubuntu.com/server ntp.ubuntu.com\nserver 127.127.1.0\nfudge 127.127.1.0 strat
service ntp restart
```

## KVM Installation

- Make sure that your hardware supports virtualization:

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

- Normally you would get a good response. If so, install kvm and configure it:

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

- Edit the cgroup\_device\_acl array in the /etc/libvirt/qemu.conf file to add the "/dev/net/tun":

```
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"  
]
```

- Restart the libvirt service to load the new values:

```
service libvirt-bin restart
```

### Quantum Installation

- Install the Quantum openvswitch agent:

```
apt-get -y install quantum-plugin-openvswitch 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://quantum:Cisco123@192.168.220.43/quantum
```

- Under the OVS section edit the following. **Note:** Again be careful to add the correct "local\_ip" entry. In this case it is the IP address for eth0.223 on compute01:

```
[OVS]  
enable_tunneling = True  
tunnel_id_ranges = 1:1000  
integration_bridge=br-int  
tunnel_bridge = br-tun  
network_vlan_ranges=  
tenant_network_type=gre  
local_ip = 10.0.0.51
```

- Edit `/etc/quantum/quantum.conf`:

```
allow_overlapping_ips = False  
fake_rabbit = False  
rabbit_virtual_host=/quantum  
rabbit_userid=openstack_rabbit_user  
rabbit_password=Cisco123  
rabbit_host=192.168.220.43  
rabbit_port=5672
```

- All hosts running quantum-plugin-openvswitch-agent require that an OVS bridge named "br-int" is created. Reboot the Compute node at this point. If you don't and you attempt to add the bridge, you will receive errors related to db.sock. Once the node is rebooted create the bridge. To create it, run:

```
ovs-vsctl add-br br-int
```

- Restart all the services:

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

### Nova Installation

- Install Nova:

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

### KVM Installation

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

```
[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_host = 192.168.220.43
auth_port = 35357
auth_protocol = http
auth_uri = http://192.168.220.43:35357/v2.0
admin_tenant_name = services
admin_user = nova
admin_password = Cisco123
```

- Edit the /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
```

- Replace the contents of the /etc/nova/nova.conf with the contents below. **Note:** Ensure you re-verify the IP address going into metadata\_host, vncserver\_proxyclient\_address, and vncserver\_listen as they are NOT the same as 192.168.220.43 (eth0 on control03). metadata\_host is eth0.221 on control03 (192.168.221.43) and the VNC proxy and listen address are eth0 on compute01 (192.168.220.51):

```
[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
s3_host=192.168.220.43
ec2_host=192.168.220.43
rabbit_port=5672
rabbit_virtual_host=/
rabbit_password=Cisco123
rabbit_userid=openstack_rabbit_user
rabbit_host=192.168.220.43
metadata_host=192.168.221.43
sql_connection=mysql://nova:Cisco123@192.168.220.43/nova
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf
connection_type=libvirt

# Auth
use_deprecated_auth=false
auth_strategy=keystone
keystone_ec2_url=http://192.168.220.43:5000/v2.0/ec2tokens

# Imaging service
glance_api_servers=192.168.220.43:9292
image_service=nova.image.glance.GlanceImageService

# VNC configuration
vnc_enabled=true
vncserver_proxyclient_address=192.168.220.51
novncproxy_base_url=http://192.168.220.43:6080/vnc_auto.html
vncserver_listen=192.168.220.51

# Network settings
```

## Cisco\_OpenStack\_Edition:\_Folsom\_Manual\_Install

```
network_api_class=nova.network.quantumv2.api.API
quantum_url=http://192.168.220.43:9696
quantum_auth_strategy=keystone
quantum_admin_auth_url=http://192.168.220.43:35357/v2.0
quantum_connection_host=localhost
quantum_admin_password=Cisco123
quantum_admin_username=quantum
quantum_admin_tenant_name=services
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSIInterfaceDriver
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver

# Compute
compute_driver=libvirt.LibvirtDriver
```

- Restart all 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 (remember to run "source openrc" first if you rebooted/logged out earlier):

```
nova-manage service list
```

### Your First Instance

- Run the following commands from either the Network Node or Controller Node. If something has to be done on a specific node it will be called out.
- Create a Quantum external network. **Note:** The eth0.221 (Public/API network) interface on the Network Node (control02) should not have an IP assigned:

```
quantum net-create public --router:external=True
```

- Create a Quantum subnet. We are using 192.168.221.0/24 as our external network:

```
quantum subnet-create --name 221-subnet public 192.168.221.0/24
```

- By default, Quantum will use the 2nd available IP address from the subnet for the starting IP allocation address. For example, Quantum will allocate an IP address if you create a router that has a gateway configured or for allocating floating IP's to instances. Use the following command instead of the one above if you need to specifically tell Quantum the starting and ending addresses to use for IP allocation within the subnet:

```
quantum subnet-create --name 221-subnet --allocation-pool start=192.168.221.10,end=192.168.221.250
```

- Create the internal (data) network used by the instances. Create additional networks and associated subnets as needed:

```
quantum net-create net1
quantum subnet-create --name 10-subnet net1 10.10.10.0/24
```

- To specify a DNS server(s) for name resolution, use the --dns\_nameservers list=true <dns\_server\_ip> command option. Here is an example of creating a subnet with the dns\_servers command option:

```
quantum subnet-create --name 10-subnet net1 10.10.10.0/24 --dns_nameservers list=true 192.168.26.1
```

## Cisco\_OpenStack\_Edition:\_Folsom\_Manual\_Install

- Create a virtual router and an associated interface used for the subnet created in the previous step:

```
quantum router-create router1
quantum router-interface-add router1 10-subnet
```

- Connect the virtual router to your external network:

```
quantum router-gateway-set router1 public
```

- **The following commands are entered on the Controller Node.** Instances/VMs gain access to the metadata server locally present on the Controller Node via the external network. To create that necessary connection, perform the following:
- Get the ID of the quantum router:

```
quantum router-list
```

- Copy the id from the command output. Issue the following command and replace router-id with the id found in the output from the previous step:

```
quantum port-list -- --device_id <router-id> --device_owner network:router_gateway
```

- Copy the IP address from the command output above. Now create your static route on the Controller Node:
- For a permanent static route add the following to /etc/network/interfaces on the Controller Node:

```
up route add -net 10.10.10.0/24 gw <router_ip_address>
```

- Restart networking for the new route to take effect:

```
/etc/init.d/networking restart
```

- Use the route command to verify the route has been added:

```
route -n
```

- For a temporary static route (without having to restart networking), use the following command:

```
route add -net 10.10.10.0/24 gw <router_ip_address>
```

- If you skipped the earlier step of downloading an image and uploading it to glance, do that now:

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

```
glance add name="precise" is_public=true container_format=ovf disk_format=qcow2 < precise-server-c
```

- **On the Network Node** create an SSH keypair and add the public key to Nova. **Note:** Leave the passphrase empty when creating the keypair. You will need to to install the nova client support:

```
apt-get -y install python-novaclient
```

```
ssh-keygen
```

```
cd ~/.ssh/
```

```
nova keypair-add --pub_key id_rsa.pub <key_name>
```

- Example:

```
nova keypair-add --pub_key id_rsa.pub net-key
```

- Before booting the instance, check for the ID of the network we created earlier. **Note:** the <quantum\_net\_id> value will come from the "quantum net-list" command:

```
quantum net-list
```

```
nova boot --image precise --flavor m1.small --key_name <key_name> --nic net-id=<quantum_net_id> <i
```

- Example:

```
nova boot --image precise --flavor m1.small --key_name net-key --nic net-id=f9035744-72a9-42cf-bd4
```

- Watch the status of the instance:

```
nova show <instance_name>
```

- Example:

```
nova show vm1
```

- The instance is booted completely when the OS-EXT-STS:vm\_state is "active". Make note of the IP address of the VM.

- ◆ Alternatively, you can watch the complete log of the VM booting by running:

```
nova console-log --length=25 vm1
```

- Verify connectivity to Instance from Quantum Server (running L3 Agent):
- Get the ID of the Quantum Router

```
quantum router-list
```

- Use "ip netns" commands against the Quantum router to get its IP address list and ping the instance. **Note:** the value of <UUID\_of\_quantum\_router> comes from the output from the "quantum router-list" command and the value of <fixed\_ip\_of\_instance> comes from the "nova show <instance\_name>" command (or console-log):

```
ip netns exec qrouter-<UUID_of_quantum_router> ip addr list
```

```
ip netns exec qrouter-<UUID_of_quantum_router> ping <fixed_ip_of_instance>
```

- Example:

```
ip netns exec qrouter-215445ab-49a7-4a38-992b-e99bc1e26ec2 ip addr list
```

- **From the Network Node, SSH into the instance booted earlier:**

```
ip netns exec qrouter-<UUID_of_quantum_router> ssh ubuntu@<fixed_ip_of_instance>
```

- Example:

```
ip netns exec qrouter-64cde9d5-0a2e-48cd-8b4d-ddd3cfc82c36 ssh ubuntu@10.10.10.3
```

```
The authenticity of host '10.10.10.3 (10.10.10.3)' can't be established. ECDSA key fingerprint is  
Are you sure you want to continue connecting (yes/no)? yes Warning: Permanently added '10.10.10.3'
```

## Cisco\_OpenStack\_Edition:\_Folsom\_Manual\_Install

Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-34-virtual x86\_64)

\* Documentation: <https://help.ubuntu.com/>

System information as of Thu Dec 6 21:04:03 UTC 2012

```
System load: 0.04          Processes:          64
Usage of /:  33.3% of 1.96GB Users logged in:      0
Memory usage: 8%          IP address for eth0: 10.10.10.3
Swap usage:  0%
```

Graph this data and manage this system at <https://landscape.canonical.com/>

0 packages can be updated. 0 updates are security updates.

Get cloud support with Ubuntu Advantage Cloud Guest

<http://www.ubuntu.com/business/services/cloud>

The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the individual files in /usr/share/

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

To run a command as administrator (user "root"), use "sudo <command>". See "man sudo\_root" for details.

ubuntu@vml:~\$

- To test from the Controller Node (or other node besides the Network node), edit the default security group or create a new security group to allow for ICMP and SSH:

```
nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
```

- Boot an instance using the keys you created on the Controller node earlier and then create and associate a Floating IP. **Note:** The value of <ext\_net\_id> comes from the output of the "quantum net-list" command and the value of the <internal\_VM\_port\_id> comes from the "quantum port-list" command for the specific instance you booted earlier:

```
quantum net-list
quantum port-list
quantum floatingip-create <ext_net_id>
quantum floatingip-associate <floatingip_id> <internal_VM_port_id>
```

- Or in a single step:

```
quantum floatingip-create --port_id <internal_VM_port-id> <ext_net_id>
```

- ping and SSH to the address that came back in the "floating\_ip\_address" field:

```
ping <floating_ip_address_of_instance>
ssh ubuntu@<floating_ip_of_instance>
```

## Credits

This work has been based on:

- Bilel Msekni OpenStack Folsom Install Guide [1]
- Emilien Macchi's Folsom guide [2]

- [OpenStack Documentation \[3\]](#)
- [OpenStack Quantum Install \[4\]](#)