# OpenStack Folsom Guide

Guide for Ubuntu Precise

Emilien Macchi

## Table of Contents



# Introduction

I'm writing this document a few weeks before Folsom stable release. I could not resist to share my experience with the community.

This document helps anyone who wants to deploy Folsom of OpenStack for development purpose.

**Table 1. Architecture and informations**

| | **controller** | **compute** |
|---|---|---|
| Managment Network | 192.168.0.1/24 | 192.168.0.2/24 |
| Hostname | folsom-controller | folsom-compute |
| Services | MySQL, RabbitMQ, Nova, Glance, Keystone, Quantum | nova-compute, KVM, nova-api |

### Note

That's a basic architecture, of course you can add many compute nodes as you want.

Since Folsom code has not been release into stable Ubuntu Packages, we are going to use "Folsom Testing Packages" which are built from master for each component.

# Requirements

You need at least two machines (virtual or physical) with 2 NIC. You need also to download Ubuntu 12.04 (LTS).

### Note

Run all commands as the root user

# Controller Node

## Operating System

1. Install Ubuntu with this parameters :

   - Time zone : **UTC**
   - Hostname : **folsom-controller**
   - Packages : **OpenSSH-Server**

   After OS Installation, reboot the server .

2. Add the repository and upgrade Ubuntu :

   ```
   apt-get install -y python-software-properties
   add-apt-repository ppa:openstack-ubuntu-testing/folsom-trunk-testing
   add-apt-repository ppa:openstack-ubuntu-testing/folsom-deps-staging
   apt-get update && apt-get -y dist-upgrade
   ```

   Reboot the server.

3. Configure the network :

   - Edit **/etc/network/interfaces** file :

   ```
   # Management Network
   auto eth0
       iface eth0 inet static
       address 192.168.0.1
       netmask 255.255.255.0
       gateway 192.168.0.254
       dns-nameservers 8.8.8.8

   # Bridged Network
   auto eth1
       iface eth1 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
   ```

   Then, restart network service :

   ```
   service networking restart
   ```

   

   ### Note

   If **eth1** is connected to a Switch, it should be in tagged mode.

   - Enable **IP forwarding** :

   ```
   sed -i -r 's/^\s*#(net\.ipv4\.ip_forward=1.*)/\1/' /etc/sysctl.conf
   echo 1 > /proc/sys/net/ipv4/ip_forward
   ```

- Edit the **/etc/hosts** file and add **folsom-controller** & **folsom-compute** hostnames with correct IP.

4. Install & Configure NTP :

   - Install the package :

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

   - Configure **/etc/ntp.conf** file :

     ```
     server ntp.ubuntu.com iburst
     server 127.127.1.0
     fudge 127.127.1.0 stratum 10
     ```

   - Restart the service :

     ```
     service ntp restart
     ```

# MySQL

1. Install the packages :

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

2. Allow connection from the network :

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

3. Restart the service :

   ```
   service mysql restart
   ```

4. Create Databases, Users, Rights :

   ```
   mysql -u root -ppassword <<EOF
   CREATE DATABASE nova;
   GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' \
       IDENTIFIED BY 'password';
   GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'192.168.0.1' \
       IDENTIFIED BY 'password';
   CREATE DATABASE cinder;
   GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'localhost' \
       IDENTIFIED BY 'password';
   CREATE DATABASE glance;
   GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' \
       IDENTIFIED BY 'password';
   CREATE DATABASE keystone;
   GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' \
       IDENTIFIED BY 'password';
   CREATE DATABASE quantum;
   GRANT ALL PRIVILEGES ON quantum.* TO 'quantum'@'localhost' \
       IDENTIFIED BY 'password';
   FLUSH PRIVILEGES;
   EOF
   ```

# RabbitMQ

Install the packages :

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

# Keystone

1. Install the packages :

```
apt-get -y install keystone python-keystone python-keystoneclient
```

2. Edit **/etc/keystone/keystone.conf** :

```
[DEFAULT]
admin_token = password
bind_host = 0.0.0.0
public_port = 5000
admin_port = 35357
compute_port = 8774
verbose = True
debug = True
log_file = keystone.log
log_dir = /var/log/keystone
log_config = /etc/keystone/logging.conf

[sql]
connection = mysql://keystone:password@localhost:3306/keystone
idle_timeout = 200

[identity]
driver = keystone.identity.backends.sql.Identity

[catalog]
driver = keystone.catalog.backends.sql.Catalog

(...)
```

3. Restart Keystone and create the tables in the database :

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

4. Load environment variables :

   - Create **novarc** file :

```
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=password
export OS_AUTH_URL="http://localhost:5000/v2.0/"
export SERVICE_ENDPOINT="http://localhost:35357/v2.0";
export SERVICE_TOKEN=password
```

   - Export the variables :

```
source novarc
echo "source novarc">>.bashrc
```

5. Fill Keystone database with datas :

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

6. Create Endpoints :

```
./keystone-endpoints.sh
```

# Glance

1. Install the packages :

```
apt-get -y install glance glance-api python-glanceclient glance-common
```

2. Configure Glance :

- Edit **/etc/glance-api-paste.ini** and **/etc/glance-registry-paste.ini** files and modify :

```
admin_tenant_name = service
admin_user = glance
admin_password = password
```

- Edit **/etc/glance-api.conf** and **/etc/glance-registry.conf** files and modify :

```
sql_connection = mysql://glance:password@localhost/glance
[paste_deploy]
flavor = keystone
```

- Restart Glance services :

```
service glance-api restart && service glance-registry restart
```

- Create Glance tables into the database :

```
glance-manage db_sync
```

- Download and import Ubuntu 12.04 UEC Image :

```
wget http://goo.gl/UQfmj
tar xzvf ubuntu-12.04-server-cloudimg-amd64.tar.gz
glance add name="Ubuntu 12.04 LTS" is_public=true container_format=ovf \
    disk_format=qcow2 < precise-server-cloudimg-amd64.img
rm -rf precise-server-cloudimg-amd64* ubuntu-12.04-server-cloudimg-amd64.tar
```

- Check if the image has been introduced in the index :

```
glance image-list

+--------+------------------+-------------+------------------+------------+-
| ID     | Name             | Disk Format | Container Format | Size       |
+--------+------------------+-------------+------------------+------------+-
| 9a17961| Ubuntu 12.04 LTS | qcow2       | ovf              | 1476395008 |
+--------+------------------+-------------+------------------+------------+-
```

# Nova

1. Install the packages :

```
apt-get -y install nova-api nova-cert nova-common nova-network nova-scheduler
    python-nova python-novaclient nova-consoleauth novnc
```

2. Configure Nova :

- Edit **/etc/nova/api-paste.ini**file and modify :

```
admin_tenant_name = service
admin_user = nova
admin_password = password
```

- Edit **/etc/nova/nova.conf**file and modify :

```
[DEFAULT]

# MySQL Connection #
sql_connection=mysql://nova:password@192.168.0.1/nova

# nova-scheduler #
rabbit_host=localhost
scheduler_driver=nova.scheduler.simple.SimpleScheduler

# nova-api #
cc_host=192.168.0.1
auth_strategy=keystone
s3_host=192.168.0.1
ec2_host=192.168.0.1
nova_url=http://192.168.0.1:8774/v1.1/
ec2_url=http://192.168.0.1:8773/services/Cloud
keystone_ec2_url=http://192.168.0.1:5000/v2.0/ec2tokens
api_paste_config=/etc/nova/api-paste.ini
allow_admin_api=true
use_deprecated_auth=false
dmz_cidr=169.254.169.254/32
ec2_dmz_host=169.254.169.254
metadata_host=169.254.169.254

# Quantum #
network_api_class=nova.network.quantumv2.api.API

# Cinder #
volume_api_class=cinder.volume.api.API

# Glance #
glance_api_servers=192.168.0.1:9292
image_service=nova.image.glance.GlanceImageService

# novnc #
novnc_enable=true
novncproxy_base_url=http://192.168.0.1:6080/vnc_auto.html
vncserver_proxyclient_address=127.0.0.1
vncserver_listen=0.0.0.0

# Misc #
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
root_helper=sudo nova-rootwrap
verbose=true
```

- Create Nova tables into the database :

```
nova-manage db sync
```

- Restart Nova services :

```
service nova-api restart
service nova-cert restart
service consoleauth restart
```

```
service nova-network restart
service nova-scheduler restart
service novnc restart
```

# Quantum

1. Install the packages :

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

   If you have an error like :

   ```
   E: Sub-process /usr/bin/dpkg returned an error code (1)
   ```

   You should force the configuration :

   ```
   dpkg --force-all --configure -a
   ```

2. Configure Quantum :

   - Edit **/etc/quantum/quantum.conf**file and modify :

     ```
     core_plugin = quantum.plugins.openvswitch.ovs_quantum_plugin.OVSQuantumPlugi
     auth_strategy = keystone
     rabbit_host=localhost
     ```

   - Edit **/etc/quantum/plugins/openvswitch/ovs_quantum_plugin.ini**file and modify :

     ```
     sql_connection = mysql://quantum:password@localhost:3306/quantum
     network_vlan_ranges = default:1000:2999
     tunnel_id_ranges =
     integration_bridge = br-int
     bridge_mappings = default:br-eth1
     ```

   - Edit **/etc/quantum/api-paste.ini**file and modify :

     ```
     admin_tenant_name = service
     admin_user = quantum
     admin_password = password
     ```

# Open-vSwitch

1. Install the packages :

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

2. Configure Open-vSwitch :

   - Create the Integration Dridge :

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

   - Create the physical Network Bridge :

     ```
     ovs-vsctl add-br br-eth1
     ovs-vsctl add-port br-eth1 eth1
     ```

# Cinder

1. Install the packages :

```
apt-get install -y cinder-api cinder-scheduler cinder-volume
```

2. Configure Cinder :

- Edit **/etc/cinder/cinder.conf**file and modify :

```
[DEFAULT]
rootwrap_config = /etc/cinder/rootwrap.conf
sql_connection = mysql://cinder:password@localhost:3306/cinder
volume_group = cinder
```

- Edit **/etc/cinder/api-paste.ini**file and modify :

```
admin_tenant_name = service
admin_user = cinder
admin_password = password
```

## Horizon

Since all dependences have not been yet introduced into Ubuntu 12.04, we need to download **python-openstack-auth** from Quantal .

Install the packages :

```
apt-get -y install python-django
wget http://mirror.ovh.net/ubuntu//pool/main/p/python-django-openstack-auth/pyt
dpkg -i python-openstack-auth*.deb
apt-get -f install
apt-get -y install apache2 libapache2-mod-wsgi openstack-dashboard memcached py
```

# Compute Node

## Operating System

Need to be written

## Hypervisor

Need to be written

## Nova

Need to be written

## Open-vSwitch

Need to be written

# Credits

Written by Emilien Macchi.