# OpenStack Folsom Install Guide Documentation

***Release 1.0***

**Zach VanDuyn, Christian Verkerk**

January 25, 2016

Contents

**Version** 1.0

**Source** https://github.com/nimbula/OpenStack-Folsom-Install-Guide

**Keywords** Multi-node OpenStack, Folsom, Nova, Keystone, Glance, Horizon, Cinder, KVM, Ubuntu Server 12.10 (64-bit release).

CHAPTER 1

# Overview

This guide focuses on providing step-by-step instruction to users who are interested in taking a bare-metal server installation to a fully functioning OpenStack cloud. We will avoid using scripts like TryStack and DevStack and will attempt to configure a "vanilla" OpenStack environment. The only scripts used in this tutorial are slight modifications of the existing keystone scripts available on the official OpenStack GitHub repo. (https://github.com/openstack/keystone/blob/master/tools/sample_data.sh)

## 1.1 Who should read this guide

This guide is for the system administrator who is installing, configuring and managing the OpenStack "Folsom" cluster infrastructure. A reasonable level of familiarity with the following is assumed:

- The Unix command line
- Installing packages
- Basic networking concepts

## 1.2 Special thanks to

Emilien Macchi at eNovance for assistance in merging some portions of this text into the official OpenStack repository.

Bilel Msekni from TELECOM SudParis for allowing me to fork sections of his OpenStack install guide and for the valuable suggestions and input.

## 1.3 Author

Zachary VanDuyn, Technical Marketing Intern at Nimbula

CHAPTER 2

# Getting Started

This guide intentionally uses the *nova-network* package instead of the newly released *quantum*. This decision was made in order to reduce the setup time for a basic network configuration. Although the next release plans to freeze nova-network development, the team responsible for overseeing OpenStack networking (Thierry, Vish, Dan) have decided that they will "...continue to support nova-network as it currently exists in Folsom".

You can read more about their decision here.

## 2.1 Hardware Requirements

The following are recommended hardware requirements for both the controller and compute nodes.

### 2.1.1 Controller Node

(runs network, volume, API, scheduler and image services)

- **Processor**: 64-bit x86
- **Memory**: 12GB of RAM
- **Disk Space**: 30GB (SATA, SAS, SSD)
- **Volume Storage**: two disks with 2TB (SATA) for volumes attached to the compute nodes
- **Network**: one 1GB NIC

### 2.1.2 Compute Node(s)

(runs virtual instances)

- **Processor**: 64-bit x86
- **Memory**: 32GB of RAM
- **Disk Space**: 30GB (SATA, SAS, SSD)
- **Network**: one 1GB NIC

CHAPTER 3

# Basic Configuration

CHAPTER 4

# OpenStack Install

## 4.1 Control Node Install

### 4.1.1 Updating your system

- After you complete the Ubuntu 12.10 installation, go into superuser mode and stay there until this tutorial concludes:

```
sudo su
```

- Update your system:

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

### 4.1.2 Install and configure the MySQL & RabbitMQ

- Install MySQL:

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

### 4.1.3 Install and configure the NTP service

- Install the NTP service:

```
apt-get install ntp
```

- Configure the NTP server to synchronize between your compute node(s) and the controller node:

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

### 4.1.4 Install VLAN, Bridge-Utils, and setup IP Forwarding

- Install the VLAN and Bridge-Utils services:

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

- Enable IP_Forwarding
  - by uncommenting net.ipv4.ip_forward=1 in /etc/sysctl.conf:

```
vi /etc/sysctl.conf
```

  - or alternatively, to avoid editing any files:

```
echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/99-openstack-ipv4-forwarding.conf
service procps start
```

- Now, run systcl with the updated configuration:

```
sysctl -p
```

### 4.1.5 Install and configure Keystone

- Install the Keystone identity service:

```
apt-get install keystone
```

- Create a new MySQL database for Keystone:

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

- Adapt the connection attribute in the /etc/keystone/keystone.conf to our newly created database.:

```
vi /etc/keystone/keystone.conf
#and edit the connection field to show
connection = mysql://keystoneUser:keystonePass@10.32.14.232/keystone
```

- Restart the identity service then synchronize the database:

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

- Use mseknibilel's scripts
  - With Quantum:

```
wget https://raw.github.com/nimbula/OpenStack-Folsom-Install-guide/master/Keystone_Scripts/W
wget https://raw.github.com/nimbula/OpenStack-Folsom-Install-guide/master/Keystone_Scripts/W
```

  - Without Quantum:

```
wget https://raw.github.com/nimbula/OpenStack-Folsom-Install-guide/master/Keystone_Scripts/W
wget https://raw.github.com/nimbula/OpenStack-Folsom-Install-guide/master/Keystone_Scripts/W
```

- Change the mode for both files:

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

- In the keystone_basic.sh script, change the $HOST_IP variable to your X.X.X.232 address
- In the keystone_endpoints_basic.sh script, change the $HOST_IP, $EXT_HOST_IP, & $MYSQL_HOST variables to your X.X.X.232 address and then execute the scripts.
- **Note: Double check your work here, screwing up keystone can be a pain to recover from**:

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

- The keystone_basic.sh script has no output, but keystone_endpoints_basic.sh should kick out something similar to this:

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |    OpenStack Compute Service     |
|      id     | 2801693507a44570a7439245b20ea0cd |
|     name    |               nova               |
|     type    |             compute              |
+-------------+----------------------------------+
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |     OpenStack Volume Service     |
|      id     | b80f524c06464c0c8af80942a1c94f78 |
|     name    |              cinder              |
|     type    |              volume              |
+-------------+----------------------------------+
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |     OpenStack Image Service      |
|      id     | 9326c1e4d4bc4e748bd8387fa5279bd0 |
|     name    |              glance              |
|     type    |              image               |
+-------------+----------------------------------+
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |        OpenStack Identity        |
|      id     | 7fd27d54ac7c476cb36ef7d0002b9fda |
|     name    |             keystone             |
|     type    |             identity             |
+-------------+----------------------------------+
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |      OpenStack EC2 service       |
|      id     | 7ce8ae8b16774c3f82e0eeecea60520a |
```

```
|     name    |               ec2                |
|     type    |               ec2                |
+-------------+----------------------------------+
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |   OpenStack Networking service   |
|      id     | 8777783c2f9f4ae3a3a6a501833ab021 |
|     name    |             quantum              |
|     type    |             network              |
+-------------+----------------------------------+
+-------------+-------------------------------------------+
|   Property  |                   Value                   |
+-------------+-------------------------------------------+
|   adminurl  | http://10.32.14.232:8774/v2/$(tenant_id)s |
|      id     |      ecfcff81220c45ce9f13ca000f1c4fa7     |
| internalurl | http://10.32.14.232:8774/v2/$(tenant_id)s |
|  publicurl  | http://10.32.14.232:8774/v2/$(tenant_id)s |
|    region   |                 RegionOne                 |
|  service_id |      2801693507a44570a7439245b20ea0cd     |
+-------------+-------------------------------------------+
+-------------+-------------------------------------------+
|   Property  |                   Value                   |
+-------------+-------------------------------------------+
|   adminurl  | http://10.32.14.232:8776/v1/$(tenant_id)s |
|      id     |      420959377cde408a865445b0ea743a19     |
| internalurl | http://10.32.14.232:8776/v1/$(tenant_id)s |
|  publicurl  | http://10.32.14.232:8776/v1/$(tenant_id)s |
|    region   |                 RegionOne                 |
|  service_id |      b80f524c06464c0c8af80942a1c94f78     |
+-------------+-------------------------------------------+
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
|   adminurl  |   http://10.32.14.232:9292/v2    |
|      id     | f2c0b4ea7bed4a8aa2d44b140df73a0d |
| internalurl |   http://10.32.14.232:9292/v2    |
|  publicurl  |   http://10.32.14.232:9292/v2    |
|    region   |            RegionOne             |
|  service_id | 9326c1e4d4bc4e748bd8387fa5279bd0 |
+-------------+----------------------------------+
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
|   adminurl  |  http://10.32.14.232:35357/v2.0  |
|      id     | ef0fb3dfa5f74f70a2059dd015e7743d |
| internalurl |  http://10.32.14.232:5000/v2.0   |
|  publicurl  |  http://10.32.14.232:5000/v2.0   |
|    region   |            RegionOne             |
|  service_id | 7fd27d54ac7c476cb36ef7d0002b9fda |
+-------------+----------------------------------+
+-------------+-----------------------------------------+
|   Property  |                  Value                  |
+-------------+-----------------------------------------+
|   adminurl  | http://10.32.14.232:8773/services/Admin |
|      id     |     e5a40371df6e47e79dc78bb61591fc87    |
| internalurl | http://10.32.14.232:8773/services/Cloud |
|  publicurl  | http://10.32.14.232:8773/services/Cloud |
```

```
|    region    |                RegionOne                 |
|  service_id  |     7ce8ae8b16774c3f82e0eeecea60520a     |
+--------------+--------------------------------------+
+--------------+----------------------------------+
|   Property   |              Value               |
+--------------+----------------------------------+
|   adminurl   |    http://10.32.14.232:9696/     |
|      id      | 8396e30ecbf14f3d9bd97d489f7407ea |
| internalurl  |    http://10.32.14.232:9696/     |
|  publicurl   |    http://10.32.14.232:9696/     |
|    region    |            RegionOne             |
|  service_id  | 8777783c2f9f4ae3a3a6a501833ab021 |
+--------------+----------------------------------+
```

- Let's create our OpenStack credential file and load it so we won't be bothered later:

```
vi creds
```

- Paste the following text:

```
export OS_NO_CACHE=1
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin_pass
export OS_AUTH_URL="http://10.32.14.232:5000/v2.0/"
```

- Load the file:

```
source creds
```

- Let's just do a quick test to see if Keystone is up:

```
apt-get install curl openssl
curl http://10.32.14.232:35357/v2.0/endpoints -H 'x-auth-token: ADMIN' | python -m json.tool
```

- It should kick out something like this:

```
{
"endpoints": [
{
"adminurl": "http://10.32.14.232:8776/v1/$(tenant_id)s",
"id": "0fe6ddf16ce344989adb22a644befa48",
"internalurl": "http://10.32.14.232:8776/v1/$(tenant_id)s",
"publicurl": "http://10.32.14.232:8776/v1/$(tenant_id)s",
"region": "RegionOne",
"service_id": "d0a8dbeac60845aaa1fa043c23177d5e"
},
{
"adminurl": "http://10.32.14.232:35357/v2.0",
"id": "7811cbbf4c3042f1a6b97d19a9ceace5",
"internalurl": "http://10.32.14.232:5000/v2.0",
"publicurl": "http://10.32.14.232:5000/v2.0",
"region": "RegionOne",
"service_id": "00685df9e085427a97837892622ca4b2"
},
{
"adminurl": "http://10.32.14.232:8774/v2/$(tenant_id)s",
"id": "826a7b77f108414ea4be8eb06d3b0c96",
"internalurl": "http://10.32.14.232:8774/v2/$(tenant_id)s",
"publicurl": "http://10.32.14.232:8774/v2/$(tenant_id)s",
```

```
    "region": "RegionOne",
    "service_id": "1a7bd347252049d9921703d45c1182dc"
    },
    {
    "adminurl": "http://10.32.14.232:9696/",
    "id": "b0974d6c9bbb4f2cab281f3ff5bcd412",
    "internalurl": "http://10.32.14.232:9696/",
    "publicurl": "http://10.32.14.232:9696/",
    "region": "RegionOne",
    "service_id": "fc2b6886fd8241448d4f3b0c9a960bf0"
    },
    {
    "adminurl": "http://10.32.14.232:9292/v2",
    "id": "c49d46bc5a62445ea60dc568abc954bb",
    "internalurl": "http://10.32.14.232:9292/v2",
    "publicurl": "http://10.32.14.232:9292/v2",
    "region": "RegionOne",
    "service_id": "574f359c07fc449ab6b0b4fad42b2df9"
    },
    {
    "adminurl": "http://10.32.14.232:8773/services/Admin",
    "id": "d50733db9848451596c84b782906cba1",
    "internalurl": "http://10.32.14.232:8773/services/Cloud",
    "publicurl": "http://10.32.14.232:8773/services/Cloud",
    "region": "RegionOne",
    "service_id": "a0fdb3cd3a234cada512ba0a75a6df56"
    }
    ]
    }
```

### 4.1.6 Install and configure Glance

- Now, let's continue by installing the image storage service (Glance):

```
apt-get install glance
```

- Let's create a new MySQL database for Glance:

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

- Next, replace the existing filter:authtoken section in /etc/glance/glance-api-paste.ini with:

```
vi /etc/glance/glance-api-paste.ini

[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_host = 10.32.14.232
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service_pass
```

- Then, update /etc/glance/glance-registry-paste.ini with:

```
vi /etc/glance/glance-registry-paste.ini

[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_host = 10.32.14.232
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service_pass
```

- Open /etc/glance/glance-api.conf and update with the following:

```
vi /etc/glance/glance-api.conf

sql_connection = mysql://glanceUser:glancePass@10.32.14.232/glance

[paste_deploy]
flavor = keystone
```

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

```
vi /etc/glance/glance-registry.conf

sql_connection = mysql://glanceUser:glancePass@10.32.14.232/glance

[paste_deploy]
flavor = keystone
```

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

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

- Sync databases:

```
glance-manage db_sync
```

- **Note: You'll probably get a warning, reminding you that 'useexisting' is deprecated. That's normal, don't worry about it.**
- Restart the services again to take into account the new modifications

    service glance-registry restart; service glance-api restart

- Now, let's test the Glance installation by installing the cirros cloud image from the Launchpad mirror:

```
mkdir images
cd images
wget https://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-disk.img
glance image-create --name NimbulaTest --is-public true --container-format bare --disk-format qc
```

- That last command should produce output similar to:

```
+------------------+--------------------------------------+
| Property | Value |
+------------------+--------------------------------------+
| checksum | 50bdc35edb03a38d91b1b071afb20a3c |
| container_format | bare |
| created_at | 2012-12-04T21:52:49 |
| deleted | False |
| deleted_at | None |
```

```
| disk_format | qcow2 |
| id | 9f045abf-3aa4-40d9-a9e1-7ab7bfa3e1ef |
| is_public | True |
| min_disk | 0 |
| min_ram | 0 |
| name | NimbulaTest |
| owner | b302c28c0f0e4d2f8f4d99553fc3971f |
| protected | False |
| size | 9761280 |
| status | active |
| updated_at | 2012-12-04T21:52:50 |
+----------------+--------------------------------------+
```

- Now let's make sure it uploaded, by using glance's image list:

```
glance image-list
```

- It should return something like this:

```
+--------------------------------------+-------------+-------------+------------------+---------
| ID                                   | Name        | Disk Format | Container Format | Size
+--------------------------------------+-------------+-------------+------------------+---------
| 74cec29b-76a1-4e89-8060-f0e2623ae5bf | NimbulaTest | qcow2       | bare             | 9761280
+--------------------------------------+-------------+-------------+------------------+---------
```

### 4.1.7 Setup networking

- Now, time to install bridge-utils:

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

- Reconfigure /etc/network/interfaces:

```
vi /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto br100
iface br100 inet static
        address 10.32.14.232
        netmask 255.255.255.0
        network 10.32.14.0
        broadcast 10.32.14.255
        gateway 10.32.14.1
        # dns-* options are implemented by the resolvconf package, if installed
        dns-nameservers 172.16.0.16
        dns-search mtv.nimbula.org
        bridge_ports eth0
        bridge_stp off
        bridge_maxwait 0
        bridge_fd 0
```

- Ensure that you setup the bridge and then restart networking:

```
sudo brctl addbr br100; sudo /etc/init.d/networking restart
```

### 4.1.8 Install and configure Nova

- Time to install Nova (and some other packages):

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

- We are also going to remove the Quantum endpoint and service, the script we ran earlier assumes that we will use Quantum instead of nova-networks, and having both endpoints on the same installation can cause some serious conflicts:
- First, get the endpoint ID:

```
keystone endpoint-list | grep 9696
```

- It will return output similar to this:

```
| b0974d6c9bbb4f2cab281f3ff5bcd412 | RegionOne | http://192.168.161.232:9696/ | http://192.168.1
```

- Grab the ID from the output and then remove that endpoint:

```
keystone endpoint-delete b0974d6c9bbb4f2cab281f3ff5bcd412
```

- Next, find the Quantum service ID:

```
keystone service-list | grep quantum
```

- It will return output similar to this:

```
| 9e3b400f6531414c93262644f20cfda1 | quantum  |   network    | OpenStack Networking Service |
```

- Grab the ID from the output and then remove that service:

```
keystone service-delete 9e3b400f6531414c93262644f20cfda1
```

- Prepare a MySQL database for Nova:

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

- Now, let's modify the authtoken section in the /etc/nova/api-paste.ini file:

```
[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_host = 10.32.14.232
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = service_pass
signing_dirname = /tmp/keystone-signing-nova
```

- Next up is the /etc/nova/nova.conf file:

```
[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
scheduler_driver=nova.scheduler.simple.SimpleScheduler
s3_host=10.32.14.232
ec2_host=10.32.14.232
ec2_dmz_host=10.32.14.232
rabbit_host=10.32.14.232
cc_host=10.32.14.232
metadata_host=10.32.14.232
metadata_listen=0.0.0.0
nova_url=http://10.32.14.232:8774/v1.1/
sql_connection=mysql://novaUser:novaPass@10.32.14.232/nova
ec2_url=http://10.32.14.232:8773/services/Cloud
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf

# Auth
use_deprecated_auth=false
auth_strategy=keystone
keystone_ec2_url=http://10.32.14.232:5000/v2.0/ec2tokens
# Imaging service
glance_api_servers=10.32.14.232:9292
image_service=nova.image.glance.GlanceImageService

# Vnc configuration
novnc_enabled=true
novncproxy_base_url=http://10.32.14.232:6080/vnc_auto.html
novncproxy_port=6080
vncserver_proxyclient_address=10.32.14.232
vncserver_listen=0.0.0.0

# NETWORK
network_manager=nova.network.manager.FlatDHCPManager
force_dhcp_release=True
dhcpbridge_flagfile=/etc/nova/nova.conf
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
# Change my_ip to match each host
my_ip=10.32.14.232
public_interface=br100
vlan_interface=eth0
flat_network_bridge=br100
flat_interface=eth0
#Note the different pool, this will be used for instance range
fixed_range=10.33.14.0/24

# Compute #
compute_driver=libvirt.LibvirtDriver

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

- Now, sync your database:

```
nova-manage db sync
```

- **Note: You may get some debug output mentioning 'nova.db.sqlalchemy.migration'. That's normal, don't worry about it.**
- Restart all of your nova-* services:

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

- Make sure all of your services are up and happy:

```
nova-manage service list
```

- You should get something like this:

```
Binary           Host                                 Zone             Status     State Updated_
nova-cert        folsom-1                             nova             enabled    :-)   2012-11-
nova-consoleauth folsom-1                             nova             enabled    :-)   2012-11-
nova-scheduler   folsom-1                             nova             enabled    :-)   2012-11-
nova-network     folsom-1                             nova             enabled    :-)   2012-11-
```

- **Note: You may get some debug output mentioning 'nova.db.sqlalchemy.migration'. That's normal, don't worry about it.**

### 4.1.9 Install and configure Cinder

- Now, it's time to install Cinder, this new OpenStack project aims at managing the volumes for VM's. It replaces nova-volumes:

```
apt-get install cinder-api cinder-scheduler cinder-volume iscsitarget iscsitarget-dkms
```

- Prepare a MySQL database for Cinder:

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

- Configure api-paste.ini by following this template:

```
vi /etc/cinder/api-paste.ini

[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
service_protocol = http
service_host = 10.32.14.232
service_port = 5000
auth_host = 10.32.14.232
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = cinder
admin_password = service_pass
```

- Open cinder.conf and change it to the following:

```
vi /etc/cinder/cinder.conf

[DEFAULT]
```

```
rootwrap_config=/etc/cinder/rootwrap.conf
sql_connection = mysql://cinderUser:cinderPass@10.32.14.232/cinder
api_paste_confg = /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
```

- Time to synchronize the database, yet again:

```
cinder-manage db sync
```

- Now, let's create a volume group, name it cinder-volumes, and make sure that it persists after a reboot:

```
dd if=/dev/zero of=cinder-volumes bs=1 count=0 seek=2G
losetup /dev/loop2 cinder-volumes
fdisk /dev/loop2
```

- And at the fdisk prompt, enter the following commands:

```
n
p
1
ENTER
ENTER
t
8e
w
```

- Proceed to create the physical volume and then the volume group itself:

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

- Now, let's add this to the rc.local to make sure that we don't lose this volume on a server reboot
- Add the following to the file before the exit 0 line:

```
#the path is wherever you ran the previous two steps + "cinder-volumes"
losetup /dev/loop2 <PATH_TO_VG>
```

### 4.1.10 Install and configure Horizon

- Finally, we are almost done. On to the Horizon interface:

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

- Some users (including myself) have encountered a few bugs when using the default Ubuntu theme. Disable it by doing the following:

```
vi /etc/openstack-dashboard/local_settings.py

#Comment these lines
#Enable the Ubuntu theme if it is present.
#try:
#   from ubuntu_theme import *
```

```
    #except ImportError:
    #   pass
```

- Horizon now requires a reboot to authenticate properly. Reboot and when the machine is ready, start all of your nova services:

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

## 4.2 Compute Node

### 4.2.1 Updating your system

- Now, all we have to do is add a compute node. Log onto the next available node on your cluster. (repeat for as many nodes as you'd like)
- Start by updating your system as root:

```
    sudo su
    apt-get update
    apt-get upgrade
    apt-get dist-upgrade
```

### 4.2.2 Install and configure the NTP service

- Install the NTP service:

```
    apt-get install ntp
```

- Configure the NTP server to follow the controller node:

```
    sed -i 's/server ntp.ubuntu.com/server 10.32.14.232/' /etc/ntp.conf
    service ntp restart
```

### 4.2.3 Setup vlan, bridge-utils, and KVM

- Install other miscellaneous services:

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

- Enable IP_Forwarding
  - by uncommenting net.ipv4.ip_forward=1 in /etc/sysctl.conf:

```
      vi /etc/sysctl.conf
```

  - or alternatively, to avoid editing any files:

```
      echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/99-openstack-ipv4-forwarding.conf
      service procps start
```

- Now, run systcl with the updated configuration:

```
sysctl -p
```

- Next, check if you can install KVM on your machine:

```
apt-get install cpu-checker
```

- Then run:

```
kvm-ok
```

- You should get a response similar to:

```
KVM acceleration can be used
```

- Now that we are all clear, let's install kvm and configure it:

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

- Edit the cgroup_device_acl array in the qemu.conf file to:

```
vi /etc/libvirt/qemu.conf

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 the default virtual bridge:

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

### 4.2.4 Setup live migration

- Enable live migration by uncommenting the listen_tls = 0, listen_tcp = 1, and auth_tcp = "none" fields in the libvirtd.conf file. Don't touch any of the other existing settings:

```
vi /etc/libvirt/libvirtd.conf

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

- Edit libvirtd_opts variable in the libvirt-bin.conf file:

```
vi /etc/init/libvirt-bin.conf
```

- Find env libvirtd_opts and set it to:

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

- Edit the same field in /etc/default/libvirt-bin and again, set it to:

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

- Restart the libvirt service to apply the changes:

```
service libvirt-bin restart
```

### 4.2.5 Install and configure nova-network

- Now, time to install nova-network and bridge-utils:

```
apt-get install nova-network bridge-utils
```

- Now, let's configure our interfaces file similar to our first node. (this time we will just use a different IP):

```
vi /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto br100
iface br100 inet static
     address 10.32.14.234
     netmask 255.255.255.0
     network 10.32.14.0
     broadcast 10.32.14.255
     gateway 10.32.14.1
     # dns-* options are implemented by the resolvconf package, if installed
     dns-nameservers 172.16.0.16
     dns-search mtv.nimbula.org
     bridge_ports eth0
     bridge_stp off
     bridge_maxwait 0
     bridge_fd 0
```

- Now, make sure the br100 is added and restart the networking services:

```
brctl addbr br100; /etc/init.d/networking restart
```

### 4.2.6 Install and configure nova-api and nova-compute

- Now, let's install the compute packages:

```
apt-get install nova-api-metadata nova-compute-kvm
```

- Now, modify the authtoken section in api-paste.ini:

```
vi /etc/nova/api-paste.ini

[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_host = 10.32.14.232
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = service_pass
signing_dirname = /tmp/keystone-signing-nova
```

- Next up, edit the nova-compute.conf:

```
vi /etc/nova/nova-compute.conf

[DEFAULT]
libvirt_type=kvm
```

- Now, time for the good ol' nova.conf again:

```
vi /etc/nova/nova.conf

[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
scheduler_driver=nova.scheduler.simple.SimpleScheduler
s3_host=10.32.14.232
ec2_host=10.32.14.232
ec2_dmz_host=10.32.14.232
rabbit_host=10.32.14.232
cc_host=10.32.14.232
metadata_host=10.32.14.234
metadata_listen=0.0.0.0
nova_url=http://10.32.14.232:8774/v1.1/
sql_connection=mysql://novaUser:novaPass@10.32.14.232/nova
ec2_url=http://10.32.14.232:8773/services/Cloud
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf

# Auth
use_deprecated_auth=false
auth_strategy=keystone
keystone_ec2_url=http://10.32.14.232:5000/v2.0/ec2tokens
# Imaging service
glance_api_servers=10.32.14.232:9292
image_service=nova.image.glance.GlanceImageService

# Vnc configuration
novnc_enabled=true
novncproxy_base_url=http://10.32.14.232:6080/vnc_auto.html
novncproxy_port=6080
vncserver_proxyclient_address=10.32.14.234
vncserver_listen=0.0.0.0

# NETWORK
network_manager=nova.network.manager.FlatDHCPManager
force_dhcp_release=True
dhcpbridge=/usr/bin/nova-dhcpbridge
dhcpbridge_flagfile=/etc/nova/nova.conf
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
# Change my_ip to match each host
my_ip=10.32.14.234
public_interface=br100
vlan_interface=eth0
flat_network_bridge=br100
flat_interface=eth0
#Note the different pool, this will be used for instance range
fixed_range=10.33.14.0/24
```

```
# Compute #
compute_driver=libvirt.LibvirtDriver

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

- Resync your databases:

```
nova-manage db sync
```

- Restart services to update changes:

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

- Check to make sure your services are in a good mood:

```
nova-manage service list
```

- You should now see a mix of services running on multiple nodes:

```
Binary           Host                                 Zone             Status     State Updated_
nova-cert        folsom-1                             nova             enabled    :-)   2012-11-
nova-consoleauth folsom-1                             nova             enabled    :-)   2012-11-
nova-scheduler   folsom-1                             nova             enabled    :-)   2012-11-
nova-network     folsom-1                             nova             enabled    :-)   2012-11-
nova-compute     folsom-2                             nova             enabled    :-)   2012-11-
nova-network     folsom-2                             nova             enabled    :-)   2012-11-
```

- **Note: You may get some debug output mentioning 'nova.db.sqlalchemy.api'. That's normal, don't worry about it.**

### 4.2.7 Setting up projects using Horizon

- Now, log onto OpenStack horizon by visiting the URL: http://10.32.14.232/horizon and logging in with the credentials:

```
username: admin
password: admin_pass
```

- Next, navigate to the "Projects" tab on the bottom left of the landing screen:
- Now, in the new pane go ahead and click the "Create Project" button in the top right. You will be greeted with a modal dialog like so:
- Fill in the fields presented, and don't forget the tabs on top. Make sure you add yourself as a project member:
- Now, your new project should be behind the modal grid. Find your new project and the corresponding row. Copy the "Project ID" to your clipboard, we'll use it in the next step:

### 4.2.8 Creating a network

- We are almost finished. Now it's time to create a network and bind it to that project:
- **Note: Remember to use the instance network**:

```
nova-manage network create --label=NimbulaNetwork --fixed_range_v4=10.33.14.0/24 --bridge=br100
```

- Now, you are done. No seriously. Go to http://10.32.14.232/horizon (or whatever your IP is) and then select your project, find an image, and launch and instance.

CHAPTER 5

# Tips & Tricks

## 5.1 Removing 'error' state instances

So, it's entirely possible that you screw up your network the first time, maybe you give it the wrong IP Pool, or maybe you assign it to the wrong project. Now, all of your instances are in the error state and you can't delete them. Luckily, I've already found two very simple and undocumented processes of removing them.

- Jump on your first node, open up the terminal as root, and plugin the following commands:

```
service nova-network stop
nova-manage project scrub <ProjectName>
nova-manage network list
```

- It should return something like this:

```
id      IPv4                IPv6            start address   DNS1            DNS2            Vlan
3       10.33.14.0/24       None            10.33.14.2      8.8.4.4         None            None
```

- Find the network you want to remove and copy the IPv4 section:

```
nova-manage network delete 10.33.14.0/24
```

- Now, we've got to get rid of those error state instances:

```
nova list | grep ERROR
```

- That command should return a list of all ERROR state instances:

```
+------+------------+--------+--------------------------------+
|  ID  |    Name    | Status |            Networks            |
+------+------------+--------+--------------------------------+
| 1805 | testserver | ERROR  | private=10.4.96.81             |
+------+------------+--------+--------------------------------+
```

- To delete, plug in their names to the following command(s):

  nova reset-state –active <name> nova delete <name>

- If those two commands don't work, we can take more drastic measures:

```
vi DeleteInstances.sh
```

- Paste the following in:

```
#!/bin/bash
mysql -uroot -ppassword << EOF
use nova;
DELETE a FROM nova.security_group_instance_association AS a INNER JOIN nova.instances AS b ON a.
DELETE FROM nova.instance_info_caches WHERE instance_id='$1';
DELETE FROM nova.instances WHERE uuid='$1';
EOF
```

- Save it by hitting ESC, then ":", then x, and hitting Enter - Then make sure it's executable by using the following:

```
chmod +x DeleteInstances.sh
```

- And run it:

```
./DeleteInstances.sh
```

## 5.2 Floating IP setup

- First create a dedicated pool:

```
sudo nova-mange floating create --pool pool_auto_assign --ip_range X.X.X.X/X
```

- Then modify the nova.conf with these flags:

```
vi /etc/nova/nova.conf

default_floating_pool = pool_auto_assign
floating_range = X.X.X.X/X
auto_assign_floating_ip = True
```

- You may also want to increase the floating IP's quota (this is also in the */etc/nova/nova.conf*):

```
quota_floating_ips = 50
```

- Then, we need to restart nova-network:

```
sudo service nova-network restart
```

CHAPTER 6

# Feedback

If you find any errors, or have any ideas on how to improve this guide, open an issue on GitHub.