

# OpenStack 安装手册

Baohua Yang, IBM CRL

[baohyang@cn.ibm.com](mailto:baohyang@cn.ibm.com)

v0.2: 2013-04-08

基于 grizzly+Ubuntu 12.04

v0.1: 2013-04-07

开始整体结构。

## 1.1 网络-Quantum

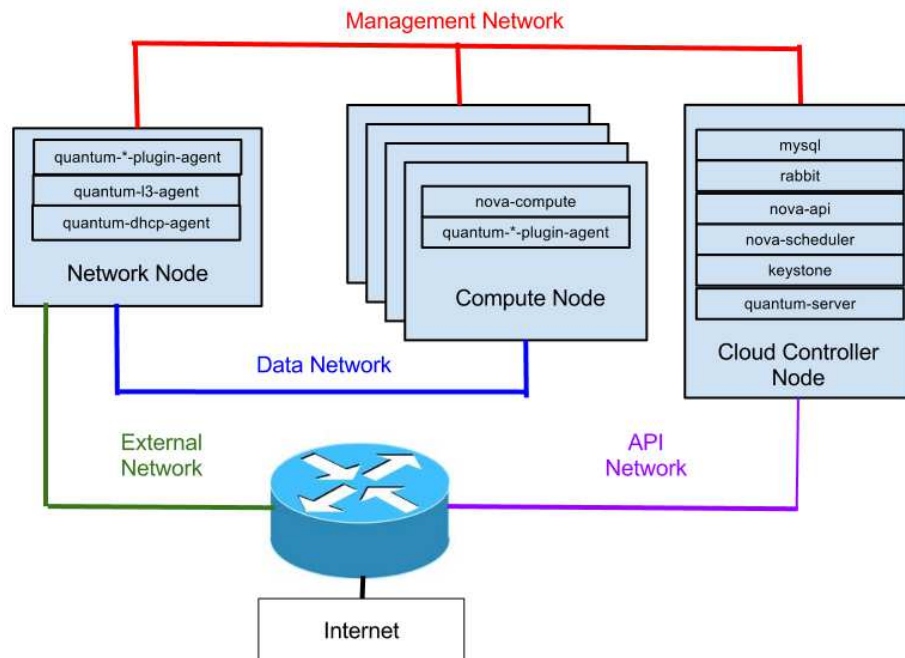
标准的 Quantum 安装包括 4 个独立的数据中心网络，如图表 1 所示。

**管理网络：**用于在 OpenStack 各个组件之间的内部通信。该网络的 IP 地址只能在数据中心内部可访问到。

**数据网络：**用于虚拟机数据传输，该网络的 IP 地址可根据使用的 Quantum 插件来灵活配置。

**外部网络：**用于提供一些带有 Internet 访问的虚拟机。该网络的 IP 地址必须可以从 Internet 上访问到。

**API 网络：**提供所有的可用 API 给用户。该网络的 IP 地址必须是从 Internet 可访问的。该网络跟外部网络可以用同一个网络。



图表 1 OpenStack 中的四种网络

## 1.2 所需资源

至少需要 3 台物理机或虚拟机。具体信息如所示。

表格 1 所需资源和用途

	控制	网络	计算
主机名	grizzly-controller	grizzly-network	grizzly-compute
提供服务	MySQL, RabbitMQ, Nova, Cinder, Glance, Keystone, Quantum	Quantum-L3-agent, Quantum-DHCP-agent, Quantum Agent with Open-vSwitch	nova-compute, KVM, nova-api, Quantum Agent with Open-vSwitch
最少磁盘数	2	1	1
外部/API 网络	7.7.7.7/24	7.7.7.8/24	无
管理网络	192.168.0.1/24	192.168.0.2/24	192.168.0.3/24
数据网络	无	10.10.10.1/24	10.10.10.2/24
网络端口数	2	3	2

## 1.3 控制节点

### 1.3.1 简介

控制节点提供的服务包括：

Databases (with MySQL)

Queues (with RabbitMQ)

Keystone

Glance

Nova (without nova-compute)

Cinder

Quantum Server (with Open-vSwitch plugin)

Dashboard (with Horizon)

### 1.3.2 基本服务

#### 1.3.2.1 操作系统

##### 1.3.2.1.1 安装系统

时间域：UTC

主机名：controller

包：openssh-server

安装系统完成后，需要重启系统。

### 1.3.2.1.2 使用 Cloud Archive 进行升级

因为 Ubuntu 12.04 中自带的 OpenStack 版本为旧的 Folsom，所以使用 Ubuntu Cloud Archive 进行必要的升级和更新。

执行

```
apt-get install ubuntu-cloud-keyring
```

修改/etc/apt/sources.list.d/cloud-archive.list：

```
deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-updates/grizzly main
```

升级系统，并重启。

```
apt-get update && apt-get upgrade
```

### 1.3.2.1.3 配置网络

配置静态网络接口，添加管理、外部/API 网络，修改/etc/network/interfaces 文件：

```
# Management Network
auto eth0
    iface eth0 inet static
        address 192.168.0.1
        netmask 255.255.255.0

# API + Public Network
auto eth1
    iface eth1 inet static
        address 7.7.7.7
        netmask 255.255.255.0
        gateway 7.7.7.1
        dns-nameservers 8.8.8.8
```

关闭 IP 欺骗过滤（避免多网卡下的丢包），修改/etc/sysctl.conf 文件：

```
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.default.rp_filter = 0
```

重启网络服务：

```
service networking restart
```

添加控制节点、网络节点和计算节点的主机名和 IP 对应到本地/etc/hosts 中。

```
127.0.0.1      localhost
127.0.1.1      controller
192.168.0.1     controller
192.168.0.2     network
192.168.0.3     compute
```

#### 1.3.2.1.4 安装 NTP 包

安装 NTP 包

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

配置/etc/ntp.conf 文件

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

重启 NTP 服务

```
service ntp restart
```

#### 1.3.2.2 MySQL 数据库服务

安装相关的包。

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

安装后默认只允许本地访问 MySQL，需要修改配置文件/etc/mysql/my.cnf，允许从任意地址都可以访问 MySQL。

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

重启服务。

```
service mysql restart
```

创建需要的数据库、用户和权限。

```
mysql -u root -password <<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';
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'192.168.0.2' \
IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'192.168.0.3' \
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';
GRANT ALL PRIVILEGES ON quantum.* TO 'quantum'@'192.168.0.2' \
IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON quantum.* TO 'quantum'@'192.168.0.3' \
IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
EOF
```

### 1.3.2.3 RabbitMQ 消息服务

安装 rabbitmq 包。

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

修改默认密码。

```
rabbitmqctl change_password guest password
```

### 1.3.3 Keystone

安装 keystone 包。

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

修改配置文件，指定数据库等信息，`/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

(...)
```

重启 keystone 服务，在数据库中创建对应的表。

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

加载环境变量，将命令写入 `novarc` 文件，并在 `.bashrc` 中指定运行 `novarc` 文件，让变量自动加载。

```
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
```

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

下载 [data script](#)，向 keystone 的数据库中写入用户、租户、服务等信息。

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

下载 [endpoint script](#)，创建各个 endpoint。

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

若需要手动指定管理网络，则在执行 endpoint\_script 时给定参数。

```
./keystone-endpoints.sh -K <ip address of the management network>
```

## 1.3.4 Glance

### 1.3.4.1 安装包

执行

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

### 1.3.4.2 配置 Glance

修改配置文件，指定用户、密码、数据库等信息，修改/etc/glance/glance-api.conf 文件和 /etc/glance/glance-registry.conf 文件，修改

```
sql_connection = mysql://glance:password@localhost/glance
admin_tenant_name = service
admin_user = glance
```

```
admin_password = password
```

对于 glance-api 文件，修改

```
notifier_strategy = rabbit
rabbit_password = password
```

重启 glance 服务，命令为

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

在数据库中创建对应的表

```
glance-manage db_sync
```

为了测试 glance 服务安装成功，可以下载并导入 Ubuntu 12.04 的 UEC 镜像文件。

```
glance image-create \
    --location
http://uec-images.ubuntu.com/releases/12.04/release/ubuntu-12.04-server-cloudimg
-amd64-disk1.img \
    --is-public true --disk-format qcow2 --container-format bare --name "Ubuntu"
```

检查镜像是否在导入成功，可以列出可用的镜像文件名。

```
glance image-list
```

ID	Name	Disk Format	Container Format	Status
0d2664d3-cda9-4937-95b2-909ecf8ea362	Ubuntu	qcow2	bare	active

## 1.3.5 Nova

### 1.3.5.1 安装包

执行



```
apt-get install nova-api nova-cert nova-common nova-conductor \  
nova-scheduler python-nova python-novaclient nova-consoleauth novnc \  
nova-novncproxy
```

### 1.3.5.2 配置 Nova

修改配置文件，指定用户、密码、数据库等信息，修改/etc/nova/api-paste.ini 文件，修改

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

```
=====  
[composite:osapi_volume]  
use = call:nova.api.openstack.urlmap:urlmap_factory  
/: osvolumeverSIONs  
/v1: openstack_volume_api_v1  
=====  
  
=====  
[composite:openstack_volume_api_v1]  
use = call:nova.api.auth:pipeline_factory  
noauth = faultwrap sizelimit noauth ratelimit osapi_volume_app_v1  
keystone = faultwrap sizelimit authToken keystonecontext ratelimit  
osapi_volume_app_v1  
keystone_nolimit = faultwrap sizelimit authToken keystonecontext  
osapi_volume_app_v1  
=====  
  
=====  
[app:osapi_volume_app_v1]  
paste.app_factory = nova.api.openstack.volume:APIRouter.factory  
=====  
  
=====  
[pipeline:osvolumeverSIONs]  
pipeline = faultwrap osvolumeverSIONapp  
  
[app:osvolumeverSIONapp]
```

```
paste.app_factory = nova.api.openstack.volume.versions:Versions.factory
```

```
=====
```

编辑/etc/nova/nova.conf 文件，修改

```
[DEFAULT]
```

```
# MySQL Connection #
```

```
sql_connection=mysql://nova:password@192.168.0.1/nova
```

```
# nova-scheduler #
```

```
rabbit_password=password
```

```
scheduler_driver=nova.scheduler.filter_scheduler.FilterScheduler
```

```
# 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
```

```
ec2_private_dns_show_ip=True
```

```
dmz_cidr=169.254.169.254/32
```

```
ec2_dmz_host=192.168.0.1
```

```
metadata_host=192.168.0.1
```

```
metadata_listen=0.0.0.0
```

```
enabled_apis=ec2,osapi_compute,metadata
```

```
# Networking #
```

```
network_api_class=nova.network.quantumv2.api.API
```

```
quantum_url=http://192.168.0.1:9696
```

```
quantum_auth_strategy=keystone
```

```
quantum_admin_tenant_name=service
```

```
quantum_admin_username=quantum
```

```
quantum_admin_password=password
```

```
quantum_admin_auth_url=http://192.168.0.1:35357/v2.0
```

```
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
```

```
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSIfaceDriver
```

```
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver

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

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

# novnc #
novnc_enable=true
novncproxy_base_url=http://7.7.7.7:6080/vnc_auto.html
vncserver_proxycient_address=192.168.0.1
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 /etc/nova/rootwrap.conf
verbose=true
```

在数据库中创建 Nova 表：

```
nova-manage db sync
```

重启 Nova 服务：

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

## 1.3.6 Cinder

### 1.3.6.1 安装包

```
apt-get install -y cinder-api cinder-scheduler cinder-volume iscsitarget \
open-iscsi iscsitarget-dkms python-cinderclient linux-headers-`uname -r`
```

### 1.3.6.2 配置和启动 iSCSI 服务

```
sed -i 's/false/true/g' /etc/default/iscsitarget  
service iscsitarget start  
service open-iscsi start
```

### 1.3.6.3 配置 Cinder

编辑/etc/cinder/cinder.conf 文件，修改

```
[DEFAULT]  
sql_connection = mysql://cinder:password@localhost:3306/cinder  
rabbit_password = password
```

编辑/etc/cinder/api-paste.ini 文件，修改

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

在第二块硬盘上创建卷：

```
fdisk /dev/sdb  
  
[Create a Linux partition]  
  
pvcreate /dev/sdb1  
vgcreate cinder-volumes /dev/sdb1
```

在数据库中创建 Cinder 的表：

```
cinder-manage db sync
```

重启服务：

```
service cinder-api restart  
service cinder-scheduler restart  
service cinder-volume restart
```

## 1.3.7 Quantum

### 1.3.7.1 安装包

安装 quantum-server 包：

```
apt-get install quantum-server
```

### 1.3.7.2 配置服务

编辑/etc/quantum/quantum.conf 文件，添加插件、认证和队列信息，修改：

```
core_plugin = \  
    quantum.plugins.openvswitch.ovs_quantum_plugin.OVSQuantumPluginV2  
auth_strategy = keystone  
fake_rabbit = False  
rabbit_password = password
```

编辑/etc/quantum/plugins/openvswitch/ovs\_quantum\_plugin.ini 文件，添加数据库信息，修改：

```
[DATABASE]  
sql_connection = mysql://quantum:password@localhost:3306/quantum  
[OVS]  
tenant_network_type = gre  
tunnel_id_ranges = 1:1000  
enable_tunneling = True
```

编辑/etc/quantum/api-paste.ini 文件，添加用户名、密码等信息，修改：

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

### 1.3.7.3 启动服务

```
service quantum-server start
```

### 1.3.8 Horizon

安装包:

```
apt-get install apache2 libapache2-mod-wsgi openstack-dashboard \
    memcached python-memcache
```

此时，控制节点安装完毕，可以通过 web 来访问 horizon 了。地址为 `http://<controller_node>/horizon`，可以通过 `admin/password` 或者 `demo/password` 来进行认证。

## 1.4 网络节点

### 1.4.1 简介

网络节点主要提供如下服务:

Virtual Bridging (Open-vSwitch + Quantum Agent) with tunneling

DHCP Server (Quantum DHCP Agent)

Virtual Routing (Quantum L3 Agent)

### 1.4.2 基本服务

#### 1.4.2.1 操作系统

##### 1.4.2.1.1 安装系统

时间域: UTC

主机名: `grizzly-network`

包: `openssh-server`

安装系统完成后，需要重启系统。

##### 1.4.2.1.2 使用 Cloud Archive 进行升级

因为 Ubuntu 12.04 中自带的 OpenStack 版本为旧的 Folsom，所以使用 Ubuntu Cloud Archive 进行必要的升级和更新。

执行

```
apt-get install ubuntu-cloud-keyring
```

修改 `/etc/apt/sources.list.d/cloud-archive.list` :

```
deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-updates/grizzly main
```

升级系统，并重启。

```
apt-get update && apt-get upgrade
```

#### 1.4.2.1.3 配置网络

配置静态网络接口，添加管理、数据和桥网络信息，编辑/etc/network/interfaces 文件：

```
# Management Network
auto eth0
    iface eth0 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    gateway 192.168.0.254
    dns-nameservers 8.8.8.8

# Data Network
auto eth1
    iface eth1 inet static
    address 10.10.10.1
    netmask 255.255.255.0

# Public Bridge
auto eth2
    iface eth2 inet manual
    up ifconfig $IFACE 0.0.0.0 up
    up ip link set $IFACE promisc on
    down ifconfig $IFACE down
```

开启 IP 转发，关闭 IP 欺骗过滤（避免多网卡下的丢包），修改/etc/sysctl.conf 文件：

```
net.ipv4.ip_forward=1
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.default.rp_filter = 0
```

重启网络服务：

```
service networking restart
```

添加控制节点、网络节点和计算节点的主机名和 IP 对应到本地/etc/hosts 中。

```
127.0.0.1      localhost
127.0.1.1      grizzly-network
192.168.0.1     grizzly-controller
192.168.0.2     grizzly-network
192.168.0.3     grizzly-compute
```

#### 1.4.2.1.4 安装 NTP 包

安装 NTP 包

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

配置/etc/ntp.conf 文件，指定控制节点（192.168.0.1）为 NTP 服务器：

```
server 192.168.0.1
```

重启 NTP 服务

```
service ntp restart
```

### 1.4.3 网络服务

#### 1.4.3.1 OpenvSwitch

安装包，包括 quantum 的 openvswitch-agent 插件、dhcp 和 L3 的代理。

```
apt-get install quantum-plugin-openvswitch-agent \
quantum-dhcp-agent quantum-l3-agent
```

启动 OpenvSwitch:

```
service openvswitch-switch start
```

创建虚拟网桥:

```
ovs-vsctl add-br br-int
ovs-vsctl add-br br-ex
ovs-vsctl add-port br-ex eth2
```



```
ip link set up br-ex
```

### 1.4.3.2 Quantum

#### 1.4.3.2.1 配置服务

编辑/etc/quantum/l3\_agent.ini，添加认证和管理的相关信息，修改

```
auth_url = http://192.168.0.1:35357/v2.0
admin_tenant_name = service
admin_user = quantum
admin_password = password
metadata_ip = 192.168.0.1
use_namespaces = False
```

编辑/etc/quantum/api-paste.ini，修改

```
auth_host = 192.168.0.1
admin_tenant_name = service
admin_user = quantum
admin_password = password
```

编辑/etc/quantum/quantum.conf，修改

```
core_plugin = \
    quantum.plugins.openvswitch.ovs_quantum_plugin.OVSQuantumPluginV2
auth_strategy = keystone
fake_rabbit = False
rabbit_host = 192.168.0.1
rabbit_password = password
```

编辑/etc/quantum/plugins/openvswitch/ovs\_quantum\_plugin.ini，修改

```
[DATABASE]
sql_connection = mysql://quantum:password@192.168.0.1:3306/quantum
[OVS]
tenant_network_type = gre
tunnel_id_ranges = 1:1000
enable_tunneling = True
integration_bridge = br-int
```

```
tunnel_bridge = br-tun  
local_ip = 10.10.10.1
```

编辑/etc/quantum/dhcp\_agent.ini，修改

```
use_namespaces = False
```

### 1.4.3.2.2 启动服务

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

## 1.4.4 虚拟网络

### 1.4.4.1 创建虚拟网络

创建 novarc 文件，写入加载环境变量命令，并在.bashrc 中调用。

```
export OS_TENANT_NAME=admin  
export OS_USERNAME=admin  
export OS_PASSWORD=password  
export OS_AUTH_URL="http://192.168.0.1:5000/v2.0/"  
export SERVICE_ENDPOINT="http://192.168.0.1:35357/v2.0/"  
export SERVICE_TOKEN=password
```

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

下载 [Quantum\\_script](#)，选用“为私有网络提供路由”情景，根据配置的 IP 来调整脚本，并执行。

### 1.4.4.2 L3 配置

获取并复制对外网络的 ID。

```
quantum net-list
```

编辑/etc/quantum/l3\_agent.ini，粘贴上获取的 ID。

```
gateway_external_network_id = ID
```

获取并复制路由器 ID。

```
quantum router-list
```

编辑/etc/quantum/l3\_agent.ini，粘贴上获取的 ID。

```
router_id = ID
```

重启 L3 代理服务。

```
service quantum-l3-agent restart
```

## 1.5 计算节点

### 1.5.1 简介

计算节点主要提供：

Hypervisor (KVM)

Nova-compute

Quantum OVS agent

### 1.5.2 基本服务

#### 1.5.2.1 操作系统

##### 1.5.2.1.1 安装系统

时间域：UTC

主机名：grizzly-compute

包：openssh-server

安装系统完成后，需要重启系统。

##### 1.5.2.1.2 使用 Cloud Archive 进行升级

因为 Ubuntu 12.04 中自带的 OpenStack 版本为旧的 Folsom，所以使用 Ubuntu Cloud Archive 进行必要的升级和更新。

执行

```
apt-get install ubuntu-cloud-keyring
```

修改/etc/apt/sources.list.d/cloud-archive.list :

```
deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-updates/grizzly main
```

升级系统，并重启。

```
apt-get update && apt-get upgrade
```

### 1.5.2.1.3 配置网络

配置静态网络接口，添加管理、数据网络信息，编辑/etc/network/interfaces 文件：

```
# Management Network
auto eth0
    iface eth0 inet static
        address 192.168.0.3
        netmask 255.255.255.0
        gateway 192.168.0.254
        dns-nameservers 8.8.8.8

# Data Network
auto eth1
    iface eth1 inet static
        address 10.10.10.2
        netmask 255.255.255.0
```

关闭 IP 欺骗过滤（避免多网卡下的丢包），修改/etc/sysctl.conf 文件：

```
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.default.rp_filter = 0
```

重启网络服务：

```
service networking restart
```

添加控制节点、网络节点和计算节点的主机名和 IP 对应到本地/etc/hosts 中。

```
127.0.0.1      localhost
127.0.1.1      grizzly-compute
192.168.0.1    grizzly-controller
192.168.0.2    grizzly-network
192.168.0.3    grizzly-compute
```

#### 1.5.2.1.4 安装 NTP 包

安装 NTP 包

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

配置/etc/ntp.conf 文件，指定控制节点（192.168.0.1）为 NTP 服务器：

```
server 192.168.0.1
```

重启 NTP 服务

```
service ntp restart
```

### 1.5.3 管理程序

#### 1.5.3.1 安装包

安装 kvm 相关包：

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

#### 1.5.3.2 配置 libvirt

编辑/etc/libvirt/qemu.conf 文件，添加

```
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"]
```

禁用 KVM 默认的虚拟桥：

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

允许在线的迁移:

编辑/etc/libvirt/libvirtd.conf 文件:

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

编辑文件中的 libvirtd\_opts 变量:

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

编辑/etc/default/libvirt-bin 文件:

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

### 1.5.3.3 重启 libvirt

```
service libvirt-bin restart
```

## 1.5.4 Nova

### 1.5.4.1 安装包

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

### 1.5.4.2 配置 Nova

编辑/etc/nova/api-paste.ini 文件, 添加用户、密码等信息, 修改:

```
auth_host = 192.168.0.1
admin_tenant_name = service
admin_user = nova
admin_password = password
```

编辑/etc/nova/nova-compute.conf 文件, 修改:

```
[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
```

编辑/etc/nova/nova.conf 文件，修改：

```
[DEFAULT]

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

# nova-scheduler #
rabbit_host=192.168.0.1
rabbit_password=password
scheduler_driver=nova.scheduler.filter_scheduler.FilterScheduler

# 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
ec2_private_dns_show_ip=True
dmz_cidr=169.254.169.254/32
ec2_dmz_host=192.168.0.1
metadata_host=192.168.0.1
metadata_listen=0.0.0.0
enabled_apis=metadata

# Networking #
network_api_class=nova.network.quantumv2.api.API
quantum_url=http://192.168.0.1:9696
quantum_auth_strategy=keystone
quantum_admin_tenant_name=service
```

```
quantum_admin_username=quantum
quantum_admin_password=password
quantum_admin_auth_url=http://192.168.0.1:35357/v2.0
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
connection_type=libvirt

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

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

# novnc #
novnc_enable=true
novncproxy_base_url=http://7.7.7.7:6080/vnc_auto.html
vncserver_proxyclient_address=192.168.0.3
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 /etc/nova/rootwrap.conf
verbose=true
```

重启服务：

```
service nova-compute restart
```



## 1.5.5 Quantum

### 1.5.5.1 OpenvSwitch

安装包。

```
apt-get install -y openvswitch-switch
```

启动 OpenvSwitch 服务。

```
service openvswitch-switch start
```

配置虚拟网桥。

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

### 1.5.5.2 Quantum

安装包。

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

编辑/etc/quantum/quantum.conf 文件，修改

```
core_plugin = \  
    quantum.plugins.openvswitch.ovs_quantum_plugin.OVSQuantumPluginV2  
auth_strategy = keystone  
fake_rabbit = False  
rabbit_host = 192.168.0.1  
rabbit_password = password
```

编辑/etc/quantum/plugins/openvswitch/ovs\_quantum\_plugin.ini 文件，修改

```
[DATABASE]  
sql_connection = mysql://quantum:password@192.168.0.1:3306/quantum  
[OVS]  
tenant_network_type = gre  
tunnel_id_ranges = 1:1000  
integration_bridge = br-int  
tunnel_bridge = br-tun
```

```
local_ip = 10.10.10.2
enable_tunneling = True
```

启动代理：

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

## 1.6 创建虚拟机

安装完成后，可以通过 OpenStack 的 API 或者 Horizon 提供的 web 界面（地址为 <http://192.168.0.1/horizon>）来管理 IaaS，认证信息为 demo/password。

编辑安全组 “Default”，允许 ICMP 和 SSH。

创建个人口令对。

找到 Instance，并执行 “Launch Instance”，产生一个新的虚拟机。

获取虚拟机的外网 ID 和端口 ID。

```
quantum net-list -- --router:external True
quantum port-list -- --device_id <vm-uuid>
```

创建绑定到虚拟机虚拟端口的 IP，并路由到外网。

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

## 1.7 参考资料

OpenStack Basic Installation, <http://docs.openstack.org/folsom/basic-install/content/index.html>。