OpenStack Nova 安装手册

作者:	yz
日期:	2011-11-27
版本:	v0.3
网址:	http://bbs.lixnutone.org

目录

实验环境

硬件:

DELL R410(1 台)

CPU：Intel(R) Xeon(R) CPU E5620 @ 2.40GHz * 2

内存：16GB

硬盘：300GB

网卡：Broadcom Corporation NetXtreme II BCM5716 Gigabit Ethernet * 2

DELL R710(1 台)

CPU：Intel(R) Xeon(R) CPU E5606 @ 2.13GHz * 2

内存：32GB

硬盘：250GB

网卡：Broadcom Corporation NetXtreme II BCM5709 Gigabit Ethernet * 4

系统：

Ubuntu Server 11.04 x64

Openstack 版本：

Diablo 4 release（2011.3）

架构部署

机器型号/主机名	外网 IP	内网 IP	作用
R410/r410-control1	60.12.206.111	192.168.1.2	控制节点
R710/r710-compute1	60.12.206.99	192.168.1.3	计算节点 1
实例网段为 10.0.0.0/24，floating ip 为 60.12.206.114,实例网段桥接在内网网卡上，网络模式采用 FlatDHCP			

服务器系统安装

1. Ubuntu server 11.04 x64 使用默认安装方式
2. 服务器外网使用 eth0
3. 服务器内网使用 eth1
4. 除 apache 及 noVNC 外，所有服务均监听内网 IP

控制节点安装

NTP 时钟服务安装

1. 安装 NTP 时钟同步服务器
   apt-get install -y ntp ntpdate

2. 同步时间
   /etc/init.d/ntp stop
   ntpdate ntp.api.bz

3. 编辑/etc/ntp.conf，将文件内容替换为如下：
   restrict 127.0.0.1
   restrict 192.168.1.0 mask 255.255.255.0 nomodify
   server ntp.api.bz
   server 127.127.1.0 # local clock
   fudge 127.127.1.0 stratum 10
   driftfile /var/lib/ntp/drift

4. 重启 ntp 服务
   /etc/init.d/ntp restart

MYSQL 数据库服务安装

1. 预设 MYSQL 数据库服务 root 密码为 openstack
   cat << MYSQL_PASSWORD |debconf-set-selections
   mysql-server-5.1 mysql-server/root_password password openstack
   mysql-server-5.1 mysql-server/root_password_again password openstack
   MYSQL_PASSWORD

2. 安装 MYSQL 数据库服务
   apt-get install -y mysql-server

3. 更改 MYSQL 数据库服务监听内网网卡 IP
   sed -i '/bind-address/s/127.0.0.1/192.168.1.2/g' /etc/mysql/my.cnf

4. 重启 MYSQL 数据库服务
   /etc/init.d/mysql restart

5. 检测服务是否正常启动
   通过 netstat -ltunp 查看是否有 tcp 3306 端口监听
   如果没有正常启动请查看/var/log/mysql 下相关 log 排错

RABBITMQ 消息队列服务安装

1. 安装 RABBITMQ 消息队列服务

   apt-get install -y rabbitmq-server

2. 更改 RABBITMQ 消息队列服务 guest 用户默认密码为 openstack

   rabbitmqctl change_password guest openstack

NOVA 服务安装

1. 导入所需更新源

   echo 'deb http://ppa.launchpad.net/openstack-release/2011.3/ubuntu natty main' >>/etc/apt/sources.list

2. 导入服务密钥

   apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 94CA80414F1043F6495425C37D21C2EC3D1B4472

3. 更新 APT 源列表

   apt-get update

4. nova-api、nova-network、nova-objectstore、nova-scheduler 服务安装

   apt-get install -y nova-api nova-network nova-objectstore nova-scheduler

GLANCE 镜像存储服务安装

1. 安装 glance

   apt-get install -y glance

KEYSTONE、noVNC、Dashboard 服务相关依赖包安装

1. APT 安装相关包

   apt-get install -y python-dev libxml2-dev libxslt1-dev libsasl2-dev libldap2-dev libsqlite3-dev libssl-dev python-pip swig git python-dateutil apache2 libapache2-mod-wsgi python-numpy

2. pip 安装相关包

   pip install passlib sqlalchemy-migrate prettytable glance python-cloudfiles nose==1.0.0 Django==1.3 django-nose==0.1.2 django-registration==0.7 django-mailer mox nosexcover

KEYSTONE 认证服务安装

1. 下载 keystone 认证服务程序

   cd /opt

   git clone https://github.com/cloudbuilders/keystone.git

cd keystone
git checkout diablo
cd ..

2. 安装 keystone 认证服务
   cd keystone
   python setup.py install
   python setup.py develop

3. 建立 keystone 认证服务数据库
   mysql -uroot -popenstack -e 'create database keystone'

4. 为 keystone 认证服务数据库建立访问所需用户名
   mysql -uroot -popenstack -e "grant select,insert,update,delete,create,drop,index,alter on keystone.* to keystone@'localhost' identified by 'keystone'"

5. 建立 keystone 认证服务启动所需用户
   useradd -s /bin/bash -g nogroup -m -d /var/log/keystone keystone

6. 建立 keystone 认证服务配置文件存放路径
   mkdir /etc/keystone

7. 生成 keystone 认证服务配置文件
   cp /opt/keystone/etc/keystone.conf /etc/keystone/
   编辑/etc/keystone/keystone.conf， 更改如下：
   default_store = sqlite
   service_host = 0.0.0.0
   admin_host = 0.0.0.0
   sql_connection = sqlite:///keystone.db
   更改为
   #default_store = sqlite
   service_host = 192.168.1.2
   admin_host = 192.168.1.2
   sql_connection = mysql://keystone:keystone@localhost/keystone

8. 生成 keystone 认证服务数据
   编辑/etc/keystone/keystone_data.sh，添加如下内容：
   #!/bin/bash
   # 建立 tenant 名为 admin
   keystone-manage $* tenant add admin
   # 建立属于 admin tenant 的用户名为 admin 密码为 openstack 的用户
   keystone-manage $* user add admin openstack admin
   # 建立管理员规则
   keystone-manage $* role add Admin

```
# 建立 keystone 管理员规则
keystone-manage $* role add KeystoneAdmin
# 建立 keystone 服务管理员规则
keystone-manage $* role add KeystoneServiceAdmin
# 将管理员规则赋予 admin 用户
keystone-manage $* role grant Admin admin
# 将 keystone 管理员规则赋予 admin 用户
keystone-manage $* role grant KeystoneAdmin admin
# 将 keystone 服务管理员规则赋予 admin 用户
keystone-manage $* role grant KeystoneServiceAdmin admin
# 添加 nova compute 服务
keystone-manage $* service add nova compute "Nova Compute Service"
# 添加 glance image 服务
keystone-manage $* service add glance image "Glance Image Service"
# 添加 keystone identity 服务
keystone-manage $* service add keystone identity "Keystone Identity Service"
# 添加 nova compute 访问点
keystone-manage $* endpointTemplates add RegionOne nova http://192.168.1.2:8774/v1.1/%tenant_id% http://192.168.1.2:8774/v1.1/%tenant_id% http://192.168.1.2:8774/v1.1/%tenant_id% 1 1
# 添加 glance image 访问点
keystone-manage $* endpointTemplates add RegionOne glance http://192.168.1.2:9292/v1.1/%tenant_id% http://192.168.1.2:9292/v1.1/%tenant_id% http://192.168.1.2:9292/v1.1/%tenant_id% 1 1
# 添加 keystone identity 访问点
keystone-manage $* endpointTemplates add RegionOne keystone http://192.168.1.2:5000/v2.0 http://192.168.1.2:35357/v2.0 http://192.168.1.2:5000/v2.0 1 1
# 为 tenant 为 admin 及 admin 用户建立一个名为 openstack，过期时间为 2015 年 2 月 5 日 0 点的 token
keystone-manage $* token add openstack admin admin 2015-02-05T00:00
# 为 tenant 为 admin 及 admin 用户建立一个类型为 EC2 的证书，其 key 和 secret 分别为 admin 用户的用户名和密码
keystone-manage $* credentials add admin EC2 'admin' 'openstack' admin
```

9. 建立 keystone 认证服务启动脚本配置文件

在/etc/init/下建立名为 keystone.conf 的文件，内容如下：

```
description "Keystone API server"
author "Soren Hansen <soren@linux2go.dk>"

start on (local-filesystems and net-device-up IFACE!=lo)
stop on runlevel [016]

respawn

exec su -c "keystone --config-file=/etc/keystone/keystone.conf --log-dir=/var/log/keystone --log-file=keystone.log" keystone
```

10. 建立 keystone 认证服务启动脚本

    ln -sv /lib/init/upstart-job /etc/init.d/keystone

11. 启动 keystone 认证服务

    /etc/init.d/keystone start

12. 验证 keystone 服务是否正常启动

    通过 netstat -ltunp 检测是否有 tcp 5000 和 35357 端口的监听，如果没有，请查看/var/log/keystone 下的相关日志排错

OPENSTACK.COMPUTE 扩展库安装

1. 下载 openstack.compute 扩展库

   cd /opt

   git clone https://github.com/jacobian/openstack.compute.git

   cd openstack.compute

   git checkout master

   cd ..

2. 安装 openstack.compute 扩展库

   cd openstack.compute

   python setup.py install

   python setup.py develop

OPENSTACKX 扩展库安装

1. 下载 openstackx 扩展库

   cd /opt

   git clone https://github.com/cloudbuilders/openstackx.git

   cd openstackx

   git checkout diablo

   cd ..

2. 安装 openstackx 扩展库

   cd openstackx

   python setup.py install

   python setup.py develop

PYTHON-NOVACLIENT 扩展库安装

1. 下载 python-novaclient 扩展库

   cd /opt

   git clone https://github.com/cloudbuilders/python-novaclient.git

   cd python-novaclient

git checkout diablo

cd ..

2. 安装 python-novaclient 扩展库

cd python-novaclient

python setup.py install

python setup.py develop

QUANTUM 模块安装

1. 下载 quantum 扩展库

cd /opt

git clone https://github.com/openstack/quantum.git

cd quantum

git checkout stable/diablo

cd ..

2. 安装 quantum 扩展库

cd quantum

python setup.py install

python setup.py develop

OPENSTACK-DASHBOARD 控制面板安装

1. 下载 openstack-dashboard 控制面板

cd /opt

git clone https://github.com/openstack/openstack-dashboard.git

cd openstack-dashboard

git checkout master

cd ..

2. 安装 openstack-dashboard 控制面板

cd openstack-dashboard/django-openstack

python setup.py install

python setup.py develop

cd ..

cd openstack-dashboard

python setup.py install

python setup.py develop

3. 建立 openstack-dashboard 控制面板数据库

mysql -uroot -popenstack -e 'create database dashboard'

4. 为 openstack-dashboard 控制面板数据库建立访问所需用户名

mysql -uroot -popenstack -e "grant select,insert,update,delete,create,drop,index,alter on dashboard.* to dashboard@'localhost' identified by 'dashboard'"

5. 配置 openstack-dashboard 控制面板

cd /opt/openstack-dashboard/openstack-dashboard/local

cp local_settings.py.example local_settings.py

编辑 local_settings.py，更改如下内容：

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(LOCAL_PATH, 'dashboard_openstack.sqlite3'),
    },
}
```

更改为

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'dashboard',
        'USER': 'dashboard',
        'PASSWORD': 'dashboard',
        'HOST': 'localhost',
        'PORT': '3306',
    },
}
```

```
OPENSTACK_KEYSTONE_URL = http://localhost:5000/v2.0/
OPENSTACK_KEYSTONE_ADMIN_URL = http://localhost:35357/v2.0
OPENSTACK_ADMIN_TOKEN = "999888777666"
```

更改为

```
OPENSTACK_KEYSTONE_URL = "http://192.168.1.2:5000/v2.0/"
OPENSTACK_KEYSTONE_ADMIN_URL = "http://192.168.1.2:35357/v2.0"
OPENSTACK_ADMIN_TOKEN = "openstack"
```

6. 配置 apache

mkdir /opt/openstack-dashboard/.blackhole

chown -R www-data:www-data /opt/openstack-dashboard

编辑/etc/apache2/sites-available/default 文件，将内容替换为如下：

```
<VirtualHost *:80>
    WSGIScriptAlias / /opt/openstack-dashboard/openstack-dashboard/dashboard/wsgi/django.wsgi
    WSGIDaemonProcess dashboard user=www-data group=www-data processes=3 threads=10
    SetEnv APACHE_RUN_USER www-data
    SetEnv APACHE_RUN_GROUP www-data
    WSGIProcessGroup dashboard

    DocumentRoot /opt/openstack-dashboard/.blackhole/
```

```
    Alias /media /opt/openstack-dashboard/openstack-dashboard/media

    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>

    <Directory /opt/openstack-dashboard/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ErrorLog /var/log/apache2/error.log
    LogLevel warn
    CustomLog /var/log/apache2/access.log combined
</VirtualHost>
```

7. 建立 openstack-dashboard 控制面板数据库结构

   /opt/openstack-dashboard/openstack-dashboard/dashboard/manage.py syncdb

8. 重启 apache 服务

   /etc/init.d/apache restart

9. 验证 openstack-dashboard 控制面板

   首先通过 netstat -ltunp 查看 80 端口的监听，其次通过浏览器访问 web 服务是否可以看到如下界面：

如不成功请查看/var/log/apache/下错误日志

noVNC 服务安装

1. 下载 noVNC 服务

   cd /opt

   git clone https://github.com/cloudbuilders/noVNC.git

cd noVNC

git checkout diablo

cd ..

NOVA 服务配置

1. 建立 nova 服务数据库

   mysql -uroot -popenstack -e ‘create database nova’

2. 为 nova 服务数据库建立访问所需用户名

   mysql -uroot -popenstack -e “grant select,insert,update,delete,create,drop,index,alter on nova.* to nova@'192.168.1.%' identified by 'nova'”

3. 配置 nova 服务

   cp /opt/keystone/examples/paste/nova-api-paste.ini /etc/nova/api-paste.ini

   编辑/etc/nova/api-paste.init，更改如下内容：

   service_host = 127.0.0.1

   auth_host = 127.0.0.1

   auth_uri = http://127.0.0.1:5000/

   admin_token = 999888777666

   更改为

   service_host = 192.168.1.2

   auth_host = 192.168.1.2

   auth_uri = http://192.168.1.2:5000/

   admin_token = openstack

   编辑/etc/nova.conf，更改为如下内容

   #general

   --logdir=/var/log/nova

   --state_path=/var/lib/nova

   --lock_path=/var/lock/nova

   --verbose=True

   --use_syslog=False

   #nova-objectstore

   --use_s3=True

   --s3_host=192.168.1.2

   --s3_port=3333

   #rabbit

   --rabbit_host=192.168.1.2

   --rabbit_port=5672

   --rabbit_password=openstack

   #ec2

   --ec2_listen=192.168.1.2

   --ec2_listen_port=8773

#osapi
--osapi_listen=192.168.1.2
--osapi_listen_port=8774
--osapi_extensions_path=/opt/openstackx/extensions
--api_paste_config=/etc/nova/api-paste.ini
#db
--sql_connection=mysql://nova:nova@192.168.1.2/nova
--sql_idle_timeout=600
--sql_max_retries=3
--sql_retry_interval=3
#glance
--glance_host=192.168.1.2
--glance_api_servers=192.168.1.2:9292
--image_service=nova.image.glance.GlanceImageService
#nova-network
--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--network_manager=nova.network.manager.FlatDHCPManager
--linuxnet_interface_driver=nova.network.linux_net.LinuxBridgeInterfaceDriver

4. 建立 nova 数据库结构
   nova-manage db sync

5. 建立名为 private，ip 地址范围为 10.0.0.0/24，网络 id 为 1，主机数 256 个，桥接在 eth1 网卡，桥接卡名称为 br1 的实例网络段，并启用多 nova-network
   nova-manage network create private 10.0.0.0/24 1 256 --bridge=br1 --bridge_interface=eth1 --multi_host='T'

6. 建立可分配的 floating ip
   nova-manage floating create 60.12.206.114

7. 重启相关服务
   /etc/init.d/nova-api restart
   /etc/init.d/nova-network restart
   /etc/init.d/nova-objectstore restart
   /etc/init.d/nova-scheduler restart

8. 检测相关服务是否启动成功
   查看/var/log/nova/nova-api.log 最下方是否有如下输出：
   2011-11-28 00:44:29,390 INFO nova.wsgi [-] Started ec2 on 192.168.1.2:8773
   2011-11-28 00:44:29,390 INFO nova.wsgi [-] Started osapi on 192.168.1.2:8774
   并通过 netstat -ltunp 查看是否有 tcp 8773 和 8774 的端口监听

   查看/var/log/nova/nova-network.log 最下方是否有如下输出：

2011-11-28 00:46:05,519 INFO nova.rpc [-] Connected to AMQP server on 192.168.1.2:5672
2011-11-28 00:46:05,520 DEBUG nova [-] Creating Consumer connection for Service network from (pid=7592) start /usr/lib/python2.7/dist-packages/nova/service.py:153
通过命令 nova-manage service list 查看是否有如下输出：
nova-network r410-control1 nova enabled :-) 2011-11-27 16:48:36

查看/var/log/nova/nova-objectstore.log 最下方是否有如下输出：
2011-11-28 00:46:46,017 INFO nova.wsgi [-] Started S3 Objectstore on 192.168.1.2:3333
并通过 netstat -ltunp 查看是否有 tcp 333 的端口监听

查看/var/log/nova/nova-scheduler.log 最下方是否有如下输出：
2011-11-28 00:47:59,789 INFO nova.rpc [-] Connected to AMQP server on 192.168.1.2:5672
2011-11-28 00:47:59,790 DEBUG nova [-] Creating Consumer connection for Service scheduler from (pid=7805) start /usr/lib/python2.7/dist-packages/nova/service.py:153
通过命令 nova-manage service list 查看是否有如下输出：
nova-scheduler r410-control1 nova enabled :-) 2011-11-27 16:48:40

如上述有哪些服务没有成功启动请查看相关/var/log/nova 下相关 log 排错

GLANCE 镜像存储服务配置

1. 建立 glance 镜像存储服务数据库
   mysql -uroot -popenstack -e 'create database glance'

2. 为 glance 镜像存储服务数据库建立访问所需用户名
   mysql -uroot -popenstack -e "grant select,insert,update,delete,create,drop,index,alter on glance.* to glance@'localhost' identified by 'glance'"

3. 配置 glance 镜像存储服务
   cp /opt/keystone/examples/paste/glance-api.conf /etc/glance/glance-api.conf
   cp /opt/keystone/examples/paste/glance-registry.conf /etc/glance/glance-registry.conf
   编辑/etc/glance/glance-api.conf，更改如下内容：
   bind_host = 0.0.0.0
   registry_host = 0.0.0.0
   rabbit_password = guest
   service_host = 127.0.0.1
   auth_host = 127.0.0.1
   auth_uri = http://127.0.0.1:5000/
   admin_token = 999888777666
   更改为
   bind_host = 192.168.1.2
   registry_host = 192.168.1.2

rabbit_password = openstack
service_host = 192.168.1.2
auth_host = 192.168.1.2
auth_uri = http://192.168.1.2:5000/
admin_token = openstack

编辑/etc/glance/glance-registry.conf，更改如下内容：
bind_host = 0.0.0.0
sql_connection = sqlite:///glance.sqlite
service_host = 127.0.0.1
auth_host = 127.0.0.1
auth_uri = http://127.0.0.1:5000/
admin_token = 999888777666
更改为
bind_host = 192.168.1.2
sql_connection = mysql://glance:glance@localhost/glance
service_host = 192.168.1.2
auth_host = 192.168.1.2
auth_uri = http://192.168.1.2:5000/
admin_token = openstack

4. 重启相关服务
/etc/init.d/glance-api restart
/etc/init.d/glance-registry restart

5. 检测服务是否成功启动
通过命令 netstat -ltunp 查看是否有 tcp 9191 和 9292 端口监听
如果没有启动成功请查看/var/log/glance 下相关 log 排错

6. 通过 glance 上传镜像
glance add -H 192.168.1.2 -A openstack name=win2k3 is_public=true < win2k3.img

noVNC 服务配置

1. 配置 noVNC 服务
向/etc/nova/nova.conf 添加如下内容：
#nova-vncproxy
--vnc_enabled=True
--vncproxy_url=http://60.12.206.111:6080
--vncproxy_wwwroot=/opt/noVNC
--vncproxy_manager=nova.vnc.auth.VNCProxyAuthManager

将计算节点 ip 和主机名对应关系添加到/etc/hosts 文件内

2. 建立 noVNC 服务启动程序软链接

   ln -sv /opt/noVNC/utils/nova-wsproxy.py /usr/bin/nova-wsproxy

3. 建立 noVNC 服务启动脚本配置文件

   在/etc/init/下建立名为 nova-vncproxy.conf 文件，内容如下：

   description "Nova VNC proxy"

   author "Vishvananda Ishaya <vishvananda@gmail.com>"

   start on (filesystem and net-device-up IFACE!=lo)

   stop on runlevel [016]

   exec su -c "nova-wsproxy 6080 --web /opt/noVNC --flagfile=/etc/nova/nova.conf" nova

4. 建立 noVNC 启动脚本

   ln -sv /lib/init/upstart-job /etc/init.d/nova-vncproxy

5. 重启相关服务

   /etc/init.d/nova-api restart

   /etc/init.d/nova-vncproxy start

6. 检测服务是否启动成功

   通过 netstat -ltunp 查看是否有 tcp 6080 端口监听

   如没启动成功请以前台模式启动并查找问题

计算节点安装

NTP 时钟同步配置

1. 安装 NTP 相关命令包

   apt-get install -y ntpdate

   跟控制节点同步时间并写入硬件

   ntpdate 192.168.1.2

   hwclock –w

2. 将时间同步添加到计划任务

   echo ' 30 8 * * * root /usr/sbin/ntpdate 192.168.1.2;hwclock –w' >>/etc/crontab

NOVA 服务安装

1. 导入所需更新源

   IT 运维专家网------“我为人人，人人为我！知识源于分享，源于交流！”---LinuxTone 技术交流分享社区！

echo 'deb http://ppa.launchpad.net/openstack-release/2011.3/ubuntu natty main' >>/etc/apt/sources.list

2. 导入服务密钥

apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 94CA80414F1043F6495425C37D21C2EC3D1B4472

3. 更新 APT 源列表

apt-get update

4. nova-network、nova-compute 服务安装

apt-get install -y nova-network nova-compute

NOVA 服务配置

1. 配置 nova 服务

编辑/etc/nova.conf，更改为如下内容

```
#general
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/var/lock/nova
--verbose=True
--use_syslog=False
#nova-objectstore
--use_s3=True
--s3_host=192.168.1.2
--s3_port=3333
#rabbit
--rabbit_host=192.168.1.2
--rabbit_port=5672
--rabbit_password=openstack
#ec2
--ec2_host=192.168.1.2
--ec2_port=8773
--ec2_url=http://192.168.1.2:8773/services/Cloud
#osapi
--osapi_host=192.168.1.2
--osapi_port=8774
#db
--sql_connection=mysql://nova:nova@192.168.1.2/nova
--sql_idle_timeout=600
--sql_max_retries=3
--sql_retry_interval=3
#glance
--glance_host=192.168.1.2
```

```
--glance_api_servers=192.168.1.2:9292
--image_service=nova.image.glance.GlanceImageService
#libvirt
--connection_type=libvirt
--libvirt_type=kvm
--snapshot_image_format=qcow2
--use_cow_image=True
--libvirt_use_virtio_for_bridges=True
#nova-scheduler
--scheduler_driver=nova.scheduler.multi.MultiScheduler
--max_cores=48
--start_guests_on_host_boot=True
--resume_guests_state_on_host_boot=True
#nova-network
--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--network_manager=nova.network.manager.FlatDHCPManager
--linuxnet_interface_driver=nova.network.linux_net.LinuxBridgeInterfaceDriver
--fixed_range=10.0.0.0/24
--flat_interface=br1
--flat_network_bridge=eth1
--flat_network_dhcp_start=10.0.0.2
--floating_range=60.12.206.114
--multi_host=true
--public_interface=eth0
--force_dhcp_release=true
--use_ipv6=False
```

2. 启动相关服务

```
/etc/init.d/nova-network restart
/etc/init.d/nova-compute restart
```

3. 检测服务是否启动成功

通过命令 netstat –ntap 查看是否有类似如下连接状态：

```
tcp        0      0 192.168.1.3:26342       192.168.1.2:5672        ESTABLISHED 29466/python
tcp        0      0 192.168.1.3:19757       192.168.1.2:3306        ESTABLISHED 29466/python
tcp        0      0 192.168.1.3:27483       192.168.1.2:5672        ESTABLISHED 29510/python
tcp        0      0 192.168.1.3:4423        192.168.1.2:3306        ESTABLISHED 29510/python
tcp        0      0 118.26.228.117:59878    211.101.24.8:56527      ESTABLISHED 29817/2
tcp        0      0 192.168.1.3:9542        192.168.1.2:3306        ESTABLISHED 29510/python
tcp        0      0 192.168.1.3:4422        192.168.1.2:3306        TIME_WAIT   -
tcp        0      0 192.168.1.3:26340       192.168.1.2:5672        ESTABLISHED 29510/python
tcp        0      0 192.168.1.3:4424        192.168.1.2:3306        ESTABLISHED 29510/python
tcp        0      0 192.168.1.3:26328       192.168.1.2:5672        ESTABLISHED 29466/python
```

查看/var/log/nova/nova-network.log 最下方是否有如下输出：

2011-11-28 00:46:05,519 INFO nova.rpc [-] Connected to AMQP server on 192.168.1.2:5672

2011-11-28 00:46:05,520 DEBUG nova [-] Creating Consumer connection for Service network from (pid=7592) start /usr/lib/python2.7/dist-packages/nova/service.py:153

查看/var/log/nova/nova-compute.log 最下方是否有如下输出：

2011-11-28 17:06:24,491 INFO nova.rpc [-] Connected to AMQP server on 192.168.1.2:5672

2011-11-28 17:06:24,492 DEBUG nova [-] Creating Consumer connection for Service compute from (pid=31197) start /usr/lib/python2.7/dist-packages/nova/service.py:153

通过在控制节点执行 nova-manage service list 结果是否有如下输出(红字)：

Binary Host Zone Status State Updated_At

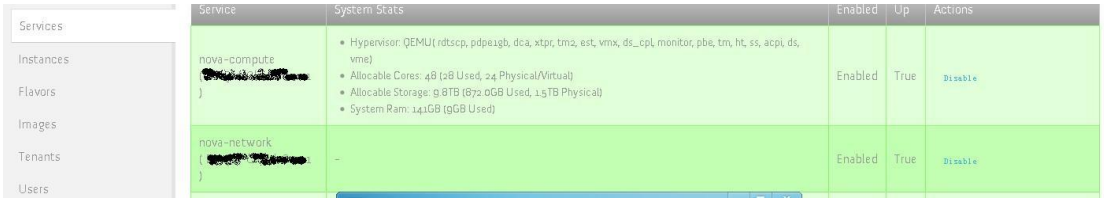
nova-scheduler r410-control1 nova enabled :-) 2011-11-28 09:07:21

nova-network r410-control1 nova enabled :-) 2011-11-28 09:07:21

nova-compute r710-compute1 nova enabled :-) 2011-11-28 09:07:14

nova-network r710-compute1 nova enabled :-) 2011-11-28 09:07:22

通过管理员登陆 dashboard 在 SYSTEM PANEL 面板通过左侧 Services 标签查看是否有计算节点 nova-compute 和 nova-network 服务，并且颜色是否为绿色，如图：

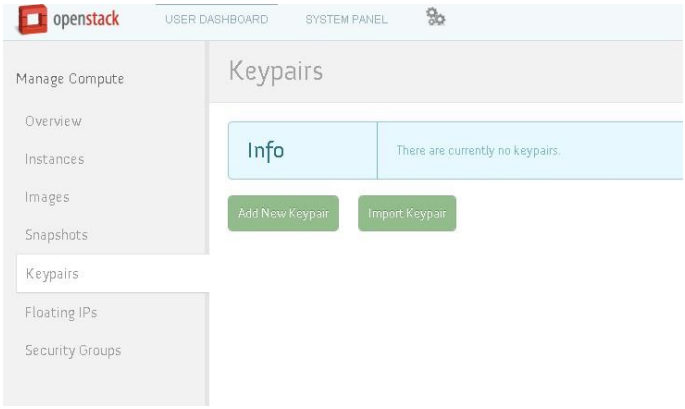


如上述有哪些服务没有成功启动请查看相关/var/log/nova 下相关 log 排错

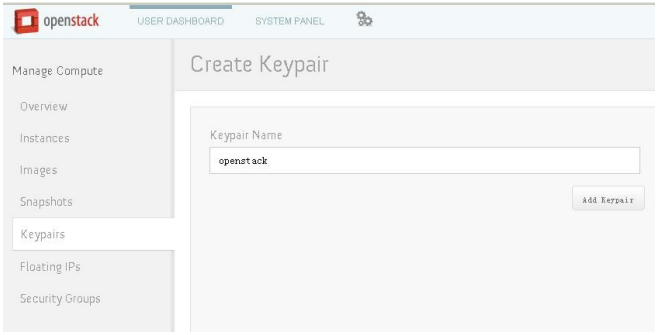
DASHBOARD 使用基础

建立 Keypairs

通过 USER DASHBOARD 面板左侧 Keypairs 标签，点击 Add New Keypair，如图：



输入 keypair 名字，这里假名为 openstack,点击 Add Keypair 按钮，如图：



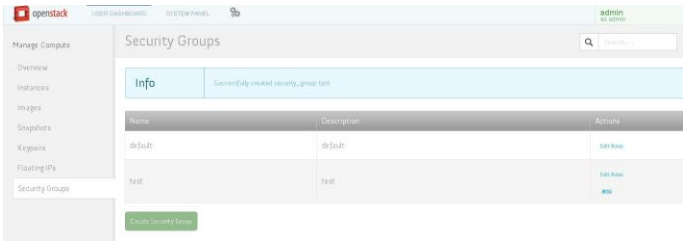
此后会要求下载一个 pem 文件，可以通过这个文件登陆启动的系统

建立安全组

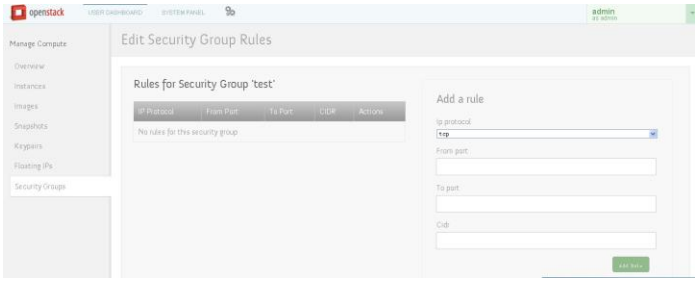
通过 USER DASHBOARD 面板左侧 Security Groups 标签，点击 Create Security Group，如图：

输入 name，在 Name 输入 test，Description 输入 test，点击 Create Security Group，如图：

建立成功后会自动跳转回 Security Groups 标签，可以看到我们建立的新安全组 test，如图：



点击我们创建的安全组的 Edit Rules，进入如下界面，如图：



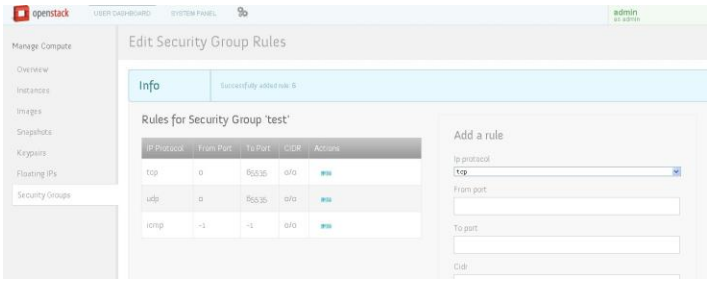
我们默认放所有，规则如下

Ip protocol：tcp，From port：0，To port：65535，Cidr：0/0

Ip protocol：udp，From port：0，To port：65535，Cidr：0/0

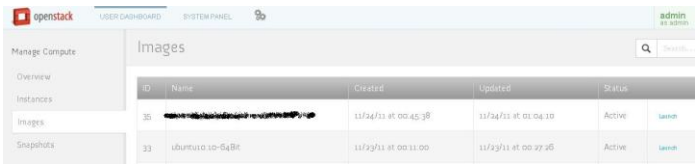
Ip protocol：icmp，From port：-1，To port：-1，Cidr：0/0

添加完毕后，如图：

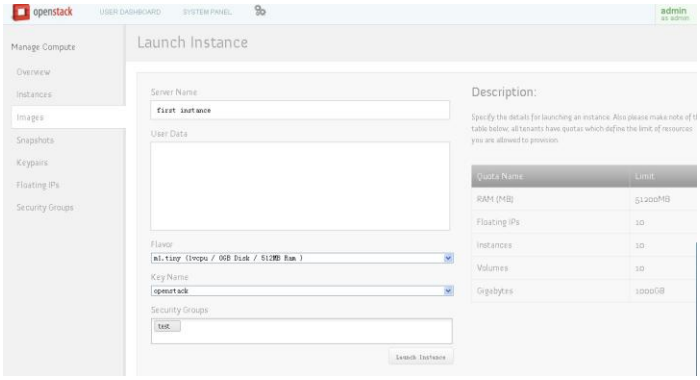


启动实例

通过 USER DASHBOARD 面板左侧 Images 标签，在已上传的镜像后点击 Launch，如图：

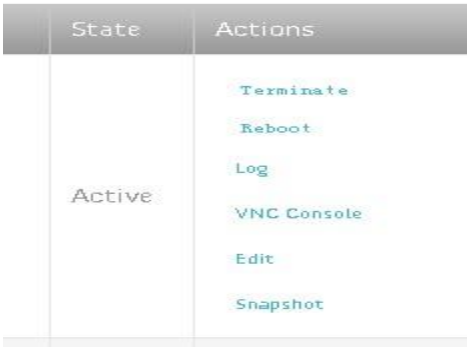


输入实例名称，这里假设为 first instance，通过 Flavor 下拉列表选择你要启动的实例配置，通过 Key Name 下拉列表选择你已有的 keypair ,通过 Securtiy Group 列表框选择我们建立的安全组 test ,点击 Launche Instance，如图：



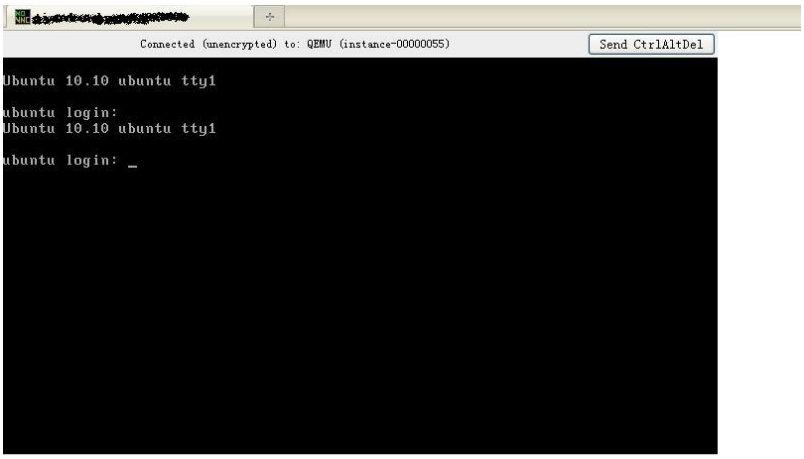
此后通过 USER DASHBOARD 面板左侧 Instances 标签 ,可以看到你刚刚启动的实例 ,实例刚启动再状态栏 Build，如图：

当实例状态变为 Active 后，我们可以通过 vnc 连接，如图：



通过 VNC 连接实例

通过 USER DASHBOARD 面板左侧 Instances 标签，找到我们启动的 first instance 实例->Actions 下的 VNC Console 链接，会新打开一个窗口，如图：



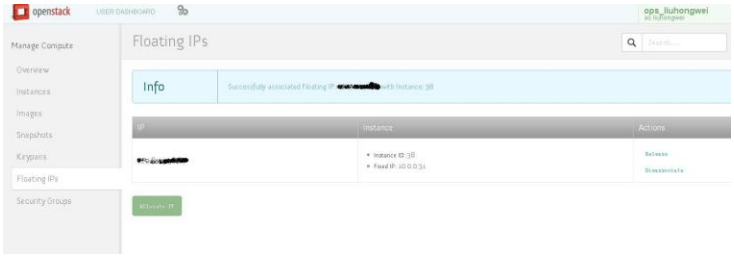
通过此窗口我们就可以访问启动的实例了

为实例分配外网 IP

通过 USER DASHBOARD 面板左侧 Floating IPs 标签，点击 Allocate IP 按钮，将出现一个可用外网 IP，如图：

点击 Associate Floating IP 链接，进入如下界面，Floating IP 是要分配的 IP，Instance 下拉列表选择你要讲此 IP 给予哪个实例，最后点击 Associate IP，如图：

成功后会跳转到 Floating IPs 标签，可以查看到我们已经分配完成，如图：



接下来就可以通过 ssh 连接你的实例了