[**Ubuntu 12.04 Openstack Essex 安装步骤（单节点）**](http://blog.csdn.net/sweetburden2011/article/details/7859081)

**8月13日，第一次修改：增加安装截图；**

**8月22日，第二次修改：修改由于转换格式所产生的一些小的错误**

**8月23日，第三次修改：添加一些漏出的步骤（错误提供者：bruce）**

**一、系统准备**

**1.       安装ubuntu系统**

注意分区部分：系统分区是需先格盘，分出5个区，3个主分区，2个逻辑分区；

以150G硬盘为例：

Primary1：100G   挂载到目录/root下；

Primary2：20G    挂载到目录/nova-volume下；

Primary3:  2G      更改分区类型为Swap，作为系统的交换分区；

Logical 1： 15G    挂载到目录/swift下；

Logical 2： 10G    可不进行挂载，也可挂载到/home

**2.       登录系统，获取root权限；**（本文档中，主机名为template）

**3.       检查是否联网：**如ping[www.baidu.com](http://www.baidu.com/)，确认可以联网进行下一步。

**4.       系统更新：**apt-get update

                                   apt-get upgrade

**5.       配置网络**

Eth0: 192.168.1.105

Eth1: 不需要设置IP

编辑 /etc/network/interfaces，在末尾添加以下3行，需要注意一点是对于ubuntu12.04的DNS设置。

auto eth1

iface eth1 inet manual

up ifconfig eth1 up

重启网络服务：

root@template:~# /etc/init.d/networking restart

检查网卡状态：

# mii-tool

http://img.my.csdn.net/uploads/201208/13/1344827452_8416.png

**6.       NTP服务器**

对于单节点来说，NTP服务器，但不是必须的。如果是多台机器的环境，就需要设置所有的节点，将NTP服务指向同一个NTP服务器上。

NTP安装：

# apt-get install ntp

NTP配置：

编辑 /etc/ntp.conf，在 server ntp.ubuntu.com下添加两行

server 127.127.1.0

fudge 127.127.1.0 stratum 10

重启NTP服务：

# service ntp restart

http://img.my.csdn.net/uploads/201208/13/1344827472_3903.png

**7.       ISCSI设置**

这是为测试Nova-volume服务使用的。不过目前nova-volume并不稳定，故简单测试一下就可以。未来的Folsom版本，将会有新的组件来替代Nova-volume。

安装tgt和iscsi客户端：

# apt-get install tgt

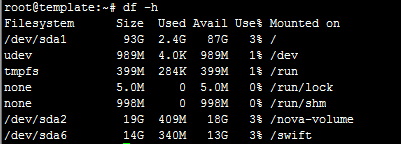
# apt-get install open-iscsi open-iscsi-utils

**8.       Nova-volume**

安装系统的时候，我们已经创建了一个nova-volume的分区，首先对它进行umount操作，再创建一个volume，名字为nova-volumes。nova的默认使用的volume的名字就是叫 nova-volumes. 如果你希望改变，就需要指定在nova.conf 文件里。

通过指令查看系统分区；

# df –h



# umount /dev/sda2

# pvcreate /dev/sda2

http://img.my.csdn.net/uploads/201208/13/1344827460_2184.png

# vgcreate nova-volumes /dev/sda2

http://img.my.csdn.net/uploads/201208/13/1344827476_3862.png

编辑 /etc/fstab ,将nova-volumes开机加载注释掉，否则会导致开机无法启动服务。

# sed -i '/nova-volume/s/^/#/' /etc/fstab

**9.       Bridge** （目前Openstack的网络是通过linux的bridge和iptables来实现的。）

# apt-get install bridge-utils

重启网络：

# /etc/init.d/networking restart

**10.   RabbitMQ和Memcache等**

RabbitMQ是用来做调度使用。Memcache是给Dashboard使用。

# apt-get install rabbitmq-server memcached python-memcache kvm libvirt-bin

**二、数据库**

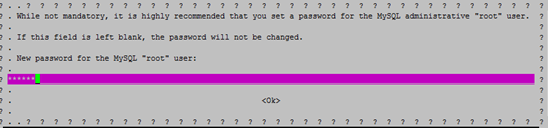
在Openstack组件里，Nova，Keystone, Glance, Horizon,都需要用到数据库。所以我们需要创建相关的数据库和用户。默认Dashboard（horizon）是使用sqlite，文档大部分都是没有修改。生产环境建议改成mysql。

**1.      mysql安装**

Openstack都是Python写的，所以你需要python-mysqldb，安装过程，会提示你输入mysql的root的密码。

# apt-get installmysql-server python-mysqldb

提示输入密码的界面：



**2.      mysql配置**

编辑/etc/mysql/my.cnf,允许网络访问mysql

#bind-address          =127.0.0.1

bind-address            = 0.0.0.0

**3.      重启mysql服务**

# service mysql restart

http://img.my.csdn.net/uploads/201208/13/1344827469_7826.png

**4.      创建相关数据库：**

# mysql -u root –ppassword

mysql> CREATE DATABASE nova;

mysql> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' IDENTIFIED BY 'password';

mysql> CREATE DATABASE glance;

mysql> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' IDENTIFIED BY 'password';

mysql> CREATE DATABASE keystone;

mysql> GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%'IDENTIFIED BY 'password';

mysql> CREATE DATABASE horizon;

mysql> GRANT ALL PRIVILEGES ON horizon.\* TO 'horizon'@'%'IDENTIFIED BY 'password';

mysql> quit;

**三、Keystone**

Keystone是Openstack的核心，所有的组件，都需要通过keystone进行认证和授权。

|  |  |  |
| --- | --- | --- |
| 租    户 | 用    户 | 密    码 |
| admin | admin | password |
| service | nova | password |
|  | glance | password |
|  | swift | password |

**1.      keystone安装**

# apt-get install keystone python-keystone python-mysqldb python-keystoneclient

**2.      keystone配置**

编辑/etc/keystone/keystone.conf，需要修改

•keystone的默认token是ADMIN，我这里修改alfred

 •默认是采用sqlite连接，我们需要改成mysql



http://img.my.csdn.net/uploads/201208/13/1344827492_7349.png

重启服务：

# service keystone restart

http://img.my.csdn.net/uploads/201208/13/1344827466_6149.png

同步keystone数据库：

# keystone-manage db\_sync

keystone的数据库，需要导入数据和endpoint，你可以参考官方文档，一步一步用命令行导入。为了方便，你也可以直接使用下面2个脚本来进行全部的设置：

*1.keystone\_data.sh导入用户信息*

#wget <http://www.chenshake.com/wp-content/uploads/2012/07/keystone_data.sh_.txt>

#mv keystone\_data.sh\_.txt keystone\_data.sh

对于keystone\_data.sh 脚本，下载完成后对它进行修改，默认登陆dashboard的密码是：chenshake，Token是chenshake。我这里登录密码修改为：password，Token为alfred；修改keystone\_data.sh的第1、3两行：

ADMIN\_PASSWORD=${ADMIN\_PASSWORD:-password}

export SERVICE\_TOKEN="alfred"

#chmod +x keystone\_data.sh

运行脚本：

#./keystone\_data.sh

没任何输出，就表示正确，可以通过下面命令检查：

#echo $?

http://img.my.csdn.net/uploads/201208/13/1344827449_2725.png

此输出表示脚本运行正常。***千万不要重复运行脚本！！！***

*2.endpoints.sh 设置endpoint*

#wget <http://www.chenshake.com/wp-content/uploads/2012/07/endpoints.sh_.txt>

#mv endpoints.sh\_.txt endpoints.sh

#chmod +x endpoints.sh

这个脚本运行，需要使用不少参数：

#./endpoints.sh -m 192.168.1.105 -u keystone -D keystone -p password -T alfred -K 192.168.1.105 -R RegionOne -E "http://localhost:35357/v2.0" -S 192.168.1.105

正常运行，会输出以下内容：



设置环境变量：建议写成一个openrc文件，方便之后的多次使用：

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=password

export OS\_AUTH\_URL=http://localhost:5000/v2.0/

运行一下：

. ./openrc

检查当前的环境变量：

#export | grep OS\_

declare-x OS\_AUTH\_URL="http://localhost:5000/v2.0/"

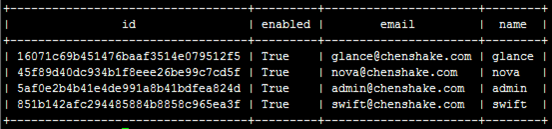
declare-x OS\_PASSWORD="password"

declare-x OS\_TENANT\_NAME="admin"

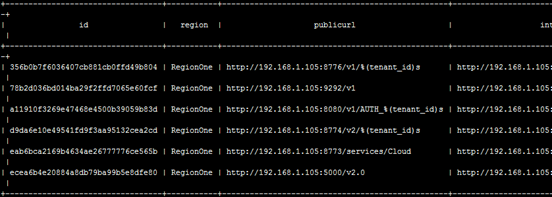
declare-x OS\_USERNAME="admin"

测试keystone：

#keystone user-list



#keystone endpoint-list



可以使用下面命令来查看结果：

#keystone tenant-list

#keystone role-list

**四、Glance**

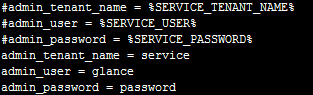
Glance是提供镜像管理服务，可以理解成一个中间件，后面的存储可以是本地存储，也可以使用swift存储。

**1.      glance安装**

# apt-get install glance glance-api glance-client glance-common glance-registry python-glance

**2.      glance配置**

编辑 /etc/glance/glance-api-paste.ini，/etc/glance/glance-registry-paste.ini,两个文件，都是修改文档最后3行



编辑/etc/glance/glance-registry.conf，改成使用mysql验证:

#sql\_connection = sqlite:////var/lib/glance/glance.sqlite

sql\_connection = mysql://glance:password@192.168.1.105/glance

编辑/etc/glance/glance-registry.conf 和 /etc/glance/glance-api.conf，都在文件末尾添加两行

[paste\_deploy]

flavor = keystone

同步glance数据库:

# glance-manage version\_control 0

# glance-manage db\_sync

http://img.my.csdn.net/uploads/201208/13/1344828180_8881.png

重启glance服务:

# service glance-api restart&& service glance-registry restart

http://img.my.csdn.net/uploads/201208/13/1344828204_3069.png

设置永久生效环境变量

为了简化日后的操作，这里设置永久的环境变量，修改 /etc/profile，在末尾添加下面内容：

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=password

exportOS\_AUTH\_URL=http://localhost:5000/v2.0/

**3.      glance测试**

# glance index      没有结果输出就表示正常，因为还没有安装镜像；

CirrOS

下载Image：

我们下载CirrOS的image作为测试使用，只有10M。如果是ubuntu官方的image，220M，并且ubuntu官方的image，都是需要使用密钥登陆。

#wget <https://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-disk.img>

上传image：

# glance add name=cirros-0.3.0-x86\_64 is\_public=true container\_format=bare disk\_format=qcow2 </root/cirros-0.3.0-x86\_64-disk.img

http://img.my.csdn.net/uploads/201208/13/1344828200_2974.png

Cirros，是可以使用用户名和密码登陆，也可以使用密钥登陆

user: cirros        password: cubswin:)

   Ubuntu官方image

下载image：

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

上传image：

# glance add name="Ubuntu 12.04 cloudimg amd64" is\_public=truecontainer\_format=ovf disk\_format=qcow2 </root/precise-server-cloudimg-amd64-disk1.img

user：ubuntu    password:只能使用密钥登陆

查看image：

# glance index

http://img.my.csdn.net/uploads/201208/13/1344828197_2940.png

**五、Nova**

**1.      nova安装**

# apt-get install nova-api nova-cert nova-common nova-objectstore nova-scheduler nova-volume nova-consoleauth novnc python-nova python-novaclient nova-compute nova-compute-kvm nova-network

如果你希望控制节点，不安装计算服务，那么 nova-compute nova-compute-kvmnova-network 这3个包不安装就可以。

**2.      nova配置**

编辑 /etc/nova/api-paste.ini , 修改末尾3行

#admin\_tenant\_name = %SERVICE\_TENANT\_NAME%

#admin\_user = %SERVICE\_USER%

#admin\_password = %SERVICE\_PASSWORD%

admin\_tenant\_name = service

admin\_user = nova

admin\_password = 123456

编辑/etc/nova/nova.conf 文件，下面是我的nova.conf文件的配置。需要注意的一点是Essex版本的nova，配置文件不需要前面加上—。日后我会逐步调整nova.conf文件，让他更加可读和方便大家理解。

[DEFAULT]

###### LOGS/STATE

#verbose=True

verbose=False

###### AUTHENTICATION

auth\_strategy=keystone

###### SCHEDULER

#--compute\_scheduler\_driver=nova.scheduler.filter\_scheduler.FilterScheduler

scheduler\_driver=nova.scheduler.simple.SimpleScheduler

###### VOLUMES

volume\_group=nova-volumes

volume\_name\_template=volume-%08x

iscsi\_helper=tgtadm

iscsi\_ip\_prefix=192.168.22

###### DATABASE

sql\_connection=mysql://nova:password@192.168.1.105/nova

###### COMPUTE

#libvirt\_type=kvm

libvirt\_type=qemu

connection\_type=libvirt

instance\_name\_template=instance-%08x

api\_paste\_config=/etc/nova/api-paste.ini

allow\_resize\_to\_same\_host=True

libvirt\_use\_virtio\_for\_bridges=true

start\_guests\_on\_host\_boot=true

resume\_guests\_state\_on\_host\_boot=true

###### APIS

osapi\_compute\_extension=nova.api.openstack.compute.contrib.standard\_extensions

allow\_admin\_api=true

s3\_host=192.168.1.105

cc\_host=192.168.1.105

###### RABBITMQ

rabbit\_host=192.168.1.105

###### GLANCE

image\_service=nova.image.glance.GlanceImageService

glance\_api\_servers=192.168.1.105:9292

###### NETWORK

network\_manager=nova.network.manager.FlatDHCPManager

force\_dhcp\_release=True

dhcpbridge\_flagfile=/etc/nova/nova.conf

dhcpbridge=/usr/bin/nova-dhcpbridge

firewall\_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver

public\_interface=eth0

flat\_interface=eth1

flat\_network\_bridge=br100

fixed\_range=192.168.22.0/24

multi\_host=true

###### NOVNC CONSOLE

novnc\_enabled=true

novncproxy\_base\_url= http://192.168.1.105:6080/vnc\_auto.html

vncserver\_proxyclient\_address=192.168.1.105

vncserver\_listen=192.168.1.105

########Nova

logdir=/var/log/nova

state\_path=/var/lib/nova

lock\_path=/var/lock/nova

#####MISC

use\_deprecated\_auth=false

root\_helper=sudo nova-rootwrap

由于使用的机器cpu不支持kvm虚拟技术，所以采用固有的qemu进行实验。监测cpu时候支持kvm虚拟技术的方法：

# kvm-ok

1如果出现下面的内容，则kvm可用；

http://img.my.csdn.net/uploads/201208/13/1344828914_4313.png

开启kvm服务，方法及详细步骤见官方安装文档。

2如果出现下面的内容，则kvm不可用；

http://img.my.csdn.net/uploads/201208/13/1344828910_4290.png

修改nova-compute.conf，如下：

--libvirt\_type=qemu

设置目录权限：

# chown -R nova:nova /etc/nova

重启所有服务：

由于服务数量比较多,创建一个脚本 restart.sh来重启所有服务。

#!/bin/bash

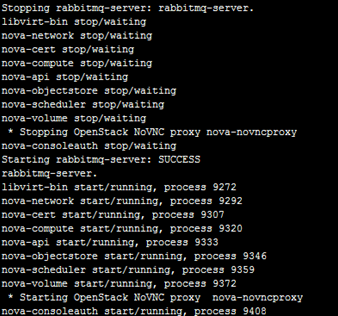
for a in rabbitmq-server libvirt-bin nova-network nova-cert nova-compute nova-api nova-objectstorenova-scheduler nova-volume novnc nova-consoleauth; do service "$a" stop; done

for a in rabbitmq-server libvirt-bin nova-network nova-cert nova-compute nova-api nova-objectstorenova-scheduler nova-volume novnc nova-consoleauth; do service "$a" start; done

运行脚本：

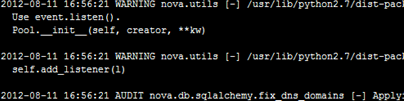
# chmod +x restart.sh

# ./restart.sh



同步数据库：

# nova-manage db sync



看到上面的输出，检查nova数据库里已经有相应的表，就表示正确。

创建Fix IP

FIX IP，就是分配给虚拟机的实际IP地址，这些数据都会写入数据库：

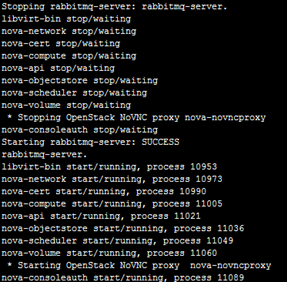
# nova-manage network createprivate --fixed\_range\_v4=192.168.22.0/24 --num\_networks=1 --bridge=br100 --bridge\_interface=eth1 --network\_size=256 --multi\_host=T

创建floating IP

所谓Floating IP，是亚马逊EC2的定义。简单说，就是公网的IP。他其实是通过类似防火墙类似，做一个映射。实际上是通过iptables来实现映射。

# nova-manage floating create --ip\_range=192.168.1.122/27

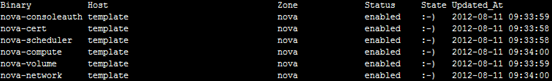
重启nova服务：



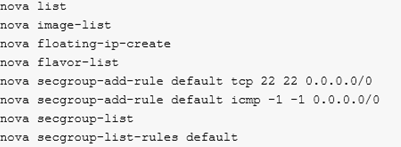
**3.      nova测试**

可以尝试用下面命令去检查nova的状况：

# nova-manage service list



下面命令可以参考：



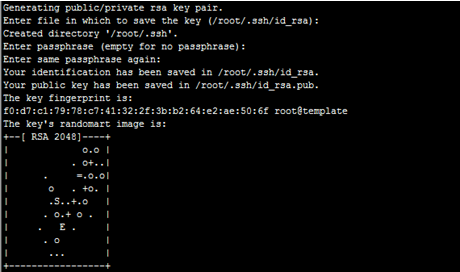
**六、创建一个VM**

**1.      创建第一个VM**

以下是用命令行创建vm的方法，你可以跳过：

1创建密钥：

#ssh-keygen

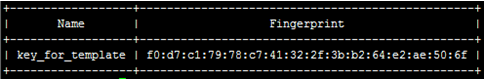


2上传密钥到数据库：

#nova keypair-add --pub\_key .ssh/id\_rsa.pub key\_for\_template

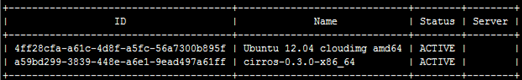
3查看到上传的key：

#nova keypair-list

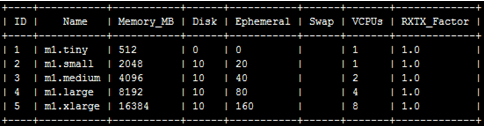


4开始创建虚拟机

#nova image-list

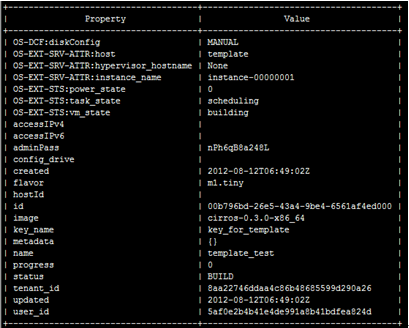


#nova flavor-list



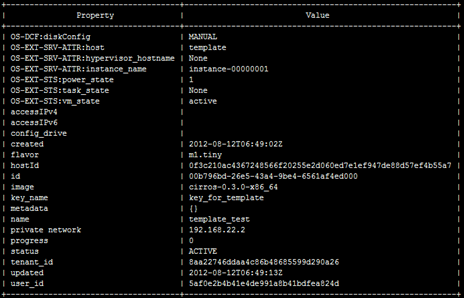
创建虚拟机：

#nova boot --flavor 1 --image a59bd299-3839-448e-a6e1-9ead497a61ff --key\_namekey\_for\_template template\_test



查看刚创建的虚拟机：

#nova show template\_test

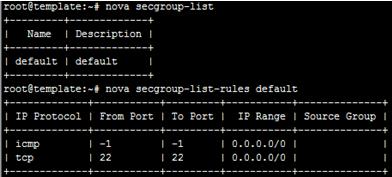


**2.      ssh访问VM**

1）打开防火墙：

# nova secgroup-add-rule default tcp 22 22 0.0.0.0/0

# nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0



2）ssh到虚拟机上：

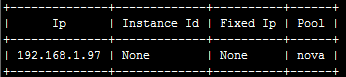
 #ssh -i .ssh/id\_rsa [cirros@192.168.22.2](mailto:cirros@192.168.22.2)

这样就可以登录服务器。

3） Floating IP访问：

申请floating IP：

# nova floating-ip-create



关联vm：

# nova add-floating-iptemplate\_test 192.168.1.97

查看虚拟机：

# nova show template\_test

ssh访问虚拟机：

# ssh -i .ssh/id\_rsa [cirros@192.168.1.97](mailto:cirros@192.168.1.97)

**七、Dashboard**

**1.      dashboard安装**

# apt-get install apache2 libapache2-mod-wsgi openstack-dashboard

**2.      dashboard配置**

默认采用sqlite来存储，直接访问就可以。如果希望改成mysql存储，才需要做下面的配置。

编辑 /etc/openstack-dashboard/local\_settings.py，添加下面内容：

DATABASES= {

   'default': {

       'ENGINE': 'django.db.backends.mysql',

       'NAME': 'horizon',

       'USER': 'horizon',

       'PASSWORD': 'password',

       'HOST': '192.168.1.105',

       'default-character-set': 'utf8'

   },

}

同步数据库：

#/usr/share/openstack-dashboard/manage.py syncdb

重启nova api：

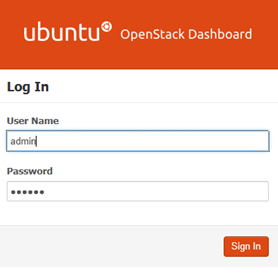
# restart nova-api

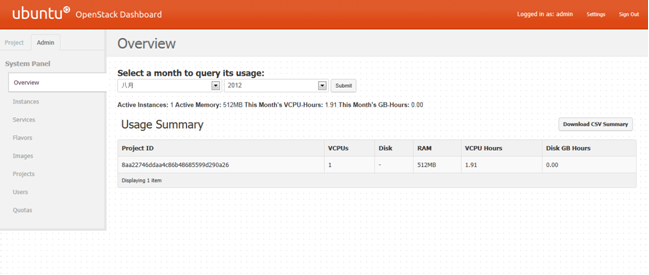
**3.      dashboard测试**

登陆dashobard：

注意：登录dashboard请使用IE浏览器，其他的浏览器可能会导致下载的密钥无法使用！！！

|  |  |  |
| --- | --- | --- |
| Address | User | Password |
| http://192.168.1.105 | admin | password |





**4.      Dashboard创建VM**

对于没有用过aws的用户，其实会比较困惑。不知道如何在dashbaord里设置，可以ssh访问虚拟机。无法ping虚拟机。

不能ping，是因为关闭icmp包响应，不能ssh，是因为防护墙关闭的22端口。

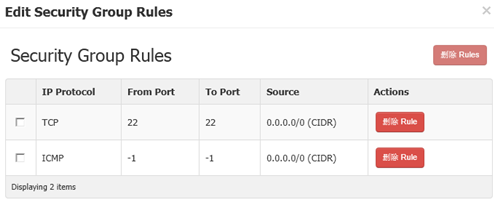
从ubuntu下载的image，只能用密钥登录，不能用密码登录。

1）设置安全组 Security group

安全组，其实就是虚拟机前面的防护墙。默认是关闭所有的包。需要打开防护墙。

为了简单，我们把default的防护墙，端口全部打开。生产环境就不要这样操作。

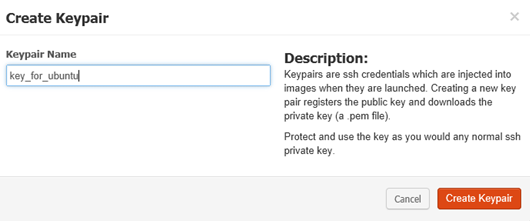
默认是default规则，直接修改default就可以。默认default，是阻止所有的访问。



 2）创建密钥：

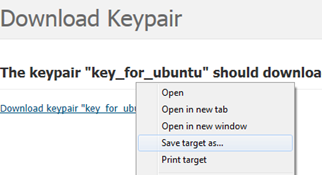
为了安全考虑，ubuntu提供的模板，都是采用密钥登录。简单点说，你创建一对公钥和私钥。私钥下载到本地保存，公钥存放在服务器上，创建虚拟机的时候，会把公钥注入到虚拟机里。

这个时候，你如果有私钥，就可以直接登录。



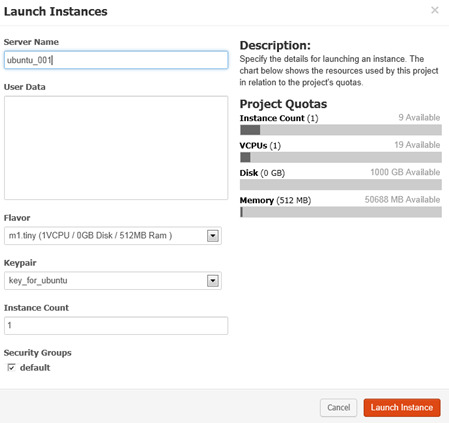
下载私钥：

点击create keypair，就会提示你下载私钥。私钥是pem后缀。

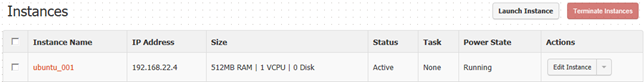


3） 创建vm

创建vm的时候，你会选择keypair，安全组。

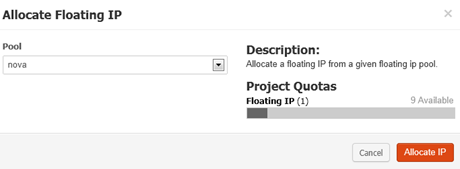


创建完成后，得到其实是一个内网规定IP:

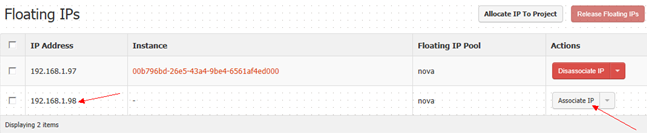


4）floating IP

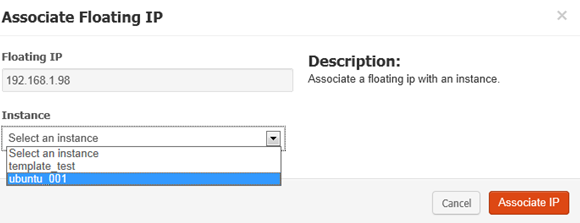
给虚拟机分配一个 floating IP，可以理解成是公网的IP



分配完成：



关联VM：



关联后的效果：



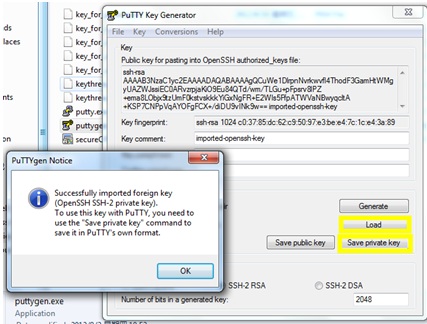
5）ssh 访问vm

这里我们只介绍一下利用putty登录的方法：

对于putty，需要对密钥进行转换。http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

到官方网站下载：PuTTYgen这个工具

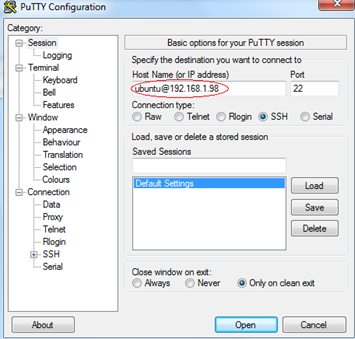
PUTTYGEN，通过load，找到刚才下载的私钥，注意，要选择全部文件(all files)，才可以找到。通过转换，选择保存私钥就可以

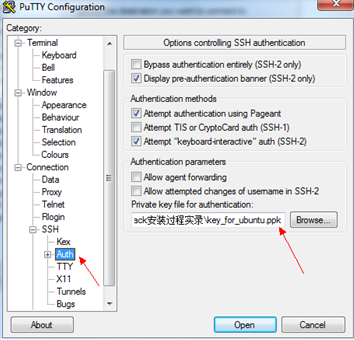


存放到本地:

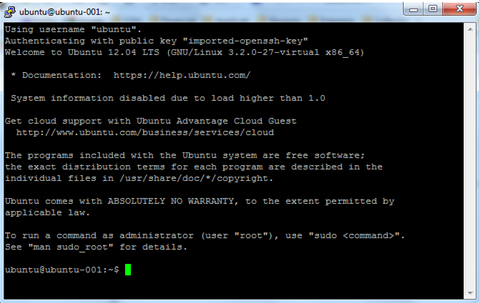
http://img.my.csdn.net/uploads/201208/13/1344829810_3686.png

设置putty使用密钥:





出现这个界面就表示登录成功了！



经历了一个月左右的摸索，我们大致清楚了openstack平台搭建的基本步骤，并对一些常见的错误进行了收集，写出了这篇较为详细的安装文档。之后如果发现新的或者更好的方法，我会对这篇文档进行简化更新。