openstack\_nova\_ubuntu12.04

安装文档

**安装系统**

安装Ubuntu12.04的系统

apt-get update

apt-get upgrade

apt-get install bridge-utils

**配置网络：**

vi /etc/network/interfaces

auto eth0

iface eth0 inet static

address 10.1.1.77

netmask 255.255.255.0

network 10.1.1.0

broadcast 10.1.1.255

gateway 10.1.1.1

auto eth1

iface eth1 inet static

address 192.168.11.77

netmask 255.255.255.0

gateway 192.168.11.1

/etc/init.d/networking restart

**安装mysql：**

apt-get install mysql-server python-mysqldb

bind-address = 0.0.0.0

restart mysql

创建数据库以及数据库用户，设置密码为passwd

Mysql –uroot -ppasswd

create database nova;

create database glance;

create database keystone;

grant all on nova.\* to nova@localhost identified by 'passwd';

grant all on glance.\* to glance@localhost identified by 'passwd';

grant all on keystone.\* to keystone@localhost identified by 'passwd';

**安装keystone**

apt-get install keystone python-keystone python-keystoneclient

vi /etc/keystone/keystone.conf

public\_port = 5000

admin\_port = 35357

admin\_token = admin

compute\_port = 8774

verbose = True

debug = True

[sql]

#connection = sqlite:////var/lib/keystone/keystone.db

connection = mysql://keystone:passwd@localhost/keystone

service keystone restart

keystone-manage db\_sync

**设置环境变量**

vi .bashrc

export SERVICE\_TOKEN=admin

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=admin

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

export SERVICE\_ENDPOINT=”http://localhost:35357/v2.0”

source .bashrc

**创建组间**

keystone tenant-create --name admin

keystone tenant-create --name service

**创建用户**

keystone user-create --name admin --pass admin --email admin@foobar.com

keystone user-create --name nova --pass nova --email nova@foobar.com

keystone user-create --name glance --pass glance --email glance@foobar.com

keystone user-create --name swift --pass swift --email [swift@foobar.com](mailto:swift@foobar.com)

**创建角色**

keystone role-create --name admin

keystone role-create --name Member

**查看组间、用户、角色**

keystone tenant-list

keystone user-list

keystone role-list

**为特定组间中的用户绑定角色(将ID变量换成相应的ID)**

为“admin”租间中的“admin”用户绑定“admin”角色：

keystone user-role-add --user $admin \_ID --role $admin\_ID --tenant\_id $admin\_ID

为“service”租间中的“nova”、“glance”、“swift”用户绑定“admin”角色：

keystone user-role-add --user $nova\_ID --role $admin\_ID --tenant\_id $service\_ID

keystone user-role-add --user $glance\_ID --role $admin\_ID --tenant\_id $service\_ID

keystone user-role-add --user $swift\_ID --role $admin\_ID --tenant\_id $service\_ID

Horizon和Swift只“Member”角色即可（使用admin用户ID和admin组间ID），所以相应地执行如下命令：

keystone user-role-add --user $admin\_ID --role $Member\_ID --tenant\_id $admin\_ID

**创建服务**

创建授权用户可以享用的服务

keystone service-create --name nova --type compute --description 'OpenStack Compute Service'

keystone service-create --name volume --type volume --description 'OpenStack Volume Service'

keystone service-create --name glance --type image --description 'OpenStack Image Service'

keystone service-create --name swift --type object-store --description 'OpenStack Storage Service'

keystone service-create --name keystone --type identity --description 'OpenStack Identity Service'

keystone service-create --name ec2 --type ec2 --description 'EC2 Service'

查看创建的服务的ID，这些ID将被用于所属服务的入口

keystone service-list

**创建服务入口(将ID变量换成相应的ID)**

创建nova-computer服务入口

keystone endpoint-create --region myregion --service\_id $service\_id --publicurl 'http://10.1.1.77:8774/v2/$(tenant\_id)s' --adminurl 'http://10.1.1.77:8774/v2/$(tenant\_id)s' --internalurl 'http://10.1.1.77:8774/v2/$(tenant\_id)s'

创建nova-volume服务入口

keystone endpoint-create --region myregion --service\_id $service\_id --publicurl 'http://10.1.1.77:8776/v1/$(tenant\_id)s' --adminurl 'http://10.1.1.77:8776/v1/$(tenant\_id)s' --internalurl 'http://10.1.1.77:8776/v1/$(tenant\_id)s'

创建glance服务入口

keystone endpoint-create --region myregion --service\_id $service\_id --publicurl 'http://192.168.1.139:9292/v1' --adminurl 'http://192.168.1.139:9292/v1' --internalurl 'http:// 192.168.1.139:9292/v1'

创建swift 服务入口

keystone endpoint-create --region myregion --service\_id $service\_id --publicurl 'http://192.168.1.139:8080/v1/AUTH\_$(tenant\_id)s' --adminurl 'http:// 192.168.1.139:8080/v1' --internalurl 'http://192.168.1.139:8080/v1/AUTH\_$(tenant\_id)s'

创建keystone 服务入口

keystone endpoint-create --region myregion --service\_id $service\_id --publicurl http:// 192.168.1.139:5000/v2.0 --adminurl http:// 192.168.1.139:35357/v2.0 --internalurl http:// 192.168.1.139:5000/v2.0

创建EC2 服务入口

keystone endpoint-create --region myregion --service\_id $service\_id --publicurl http:// 192.168.1.139:8773/services/Cloud --adminurl http:// 192.168.1.139:8773/services/Admin --internalurl http:// 192.168.1.139:8773/services/Cloud

**安装glance**

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

配置glance服务

修改以下两个配置文件的末尾内容：

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

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

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

admin\_user = %SERVICE\_USER%

admin\_password = %SERVICE\_PASSWORD%

这些值需要用刚才我们安装中的实际值进行替换，admin\_tenant\_name应该是“service”，admin\_user这里是“glance”、admin\_password是“glance”。

编辑完这三行应该是如下的样子：

admin\_tenant\_name = service

admin\_user = glance

admin\_password = glance

修改 /etc/glance/glance-registry.conf配置文件，找到“sql\_connection =”这句作如下修改：

sql\_connection = mysql://glance:passwd@localhost/glance

紧接着还是这份文件，在底部增加如下设置，目的是让glance使用keystone授权：

[paste\_deploy]

flavor = keystone

打开 /etc/glance/glance-api.conf文件，在文件末尾增加如下配置：

[paste\_deploy]

flavor = keystone

在MySQL数据库中创建glance schema，进行同步：

glance-manage version\_control 0

glance-manage db\_sync

上述改动全部完成后，重启glance-api和glance-registry服务：

restart glance-api

restart glance-registry

现在我们可以使用这个命令来测试glance是否正确安装：

glance index

echo $?

如果这条命令没有返回任何结果，则说明glance已经正确安装完毕，并与Keystone正常通信。若是稍微探究一下的话，上述命令其实有返回值，正确安装情况下的返回值是0，可以使用echo $?进行查看。

**上传镜像**

glance add name="cirros-0.3.0-i386-disk" is\_public=true container\_format=ovf disk\_format=qcow2 < /home/cy/cirros-0.3.0-i386-disk.img

**安装nova**

apt-get install nova-api nova-cert nova-compute nova-compute-kvm nova-doc nova-network nova-objectstore nova-scheduler nova-volume rabbitmq-server novnc nova-consoleauth

配置nova

vi /etc/nova/nova.conf

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

--dhcpbridge=/usr/bin/nova-dhcpbridge

--logdir=/var/log/nova

--state\_path=/var/lib/nova

--lock\_path=/var/lock/nova

--force\_dhcp\_release

--iscsi\_helper=tgtadm

--libvirt\_use\_virtio\_for\_bridges

--connection\_type=libvirt

--root\_helper=sudo nova-rootwrap

--verbose

--ec2\_private\_dns\_show\_ip

--allow\_admin\_api=true

--use\_deprecated\_auth=false

--auth\_strategy=keystone

--scheduler\_driver=nova.scheduler.simple.SimpleScheduler

--s3\_host=10.1.1.77

--ec2\_host=10.1.1.77

--rabbit\_host=10.1.1.77

--cc\_host=10.1.1.77

--nova\_url=http://localhost:8774/v1.1/

--routing\_source\_ip=10.1.1.77

--glance\_api\_servers=10.1.1.77:9292

--image\_service=nova.image.glance.GlanceImageService

--iscsi\_ip\_prefix=10.0.0

--sql\_connection=mysql://nova:passwd@localhost/nova

--ec2\_url=http://localhost:8773/services/Cloud

--keystone\_ec2\_url=http://localhost:5000/v2.0/ec2tokens

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

--libvirt\_type=kvm

--libvirt\_use\_virtio\_for\_bridges=true

--start\_guests\_on\_host\_boot=true

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

#novnc

--novnc\_enabled=true

--novncproxy\_base\_url= http://10.1.1.77:6080/vnc\_auto.html

--vncserver\_proxyclient\_address=127.0.0.1

--vncserver\_listen=127.0.0.1

# network specific settings

--network\_manager=nova.network.manager.FlatDHCPManager

--public\_interface=eth0

--flat\_interface=eth1

--flat\_network\_bridge=br100

--fixed\_range=10.0.0.0/24

--floating\_range=10.1.1.77/24

--network\_size=255

--flat\_network\_dhcp\_start=10.0.0.2

--flat\_injected=False

创建一个物理卷：

pvcreate /dev/sda6

创建一个名为“nova-volumnes”的卷组

vgcreate nova-volumes /dev/sda6

修改 /etc/nova文件夹的属主及 /etc/nova/nova.conf文件的访问权限：

chown -R nova:nova /etc/nova

chmod 644 /etc/nova/nova.conf

进入 /etc/nova/api-paste.ini文件，找到末尾三行：

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

admin\_user = %SERVICE\_USER%

admin\_password = %SERVICE\_PASSWORD%

用之前创建的名字进行替换,编辑完毕如下所示：

admin\_tenant\_name = service

admin\_user = nova

admin\_password = nova

仍然在MySQL数据库进行同步：

nova-manage db sync

为实例提供IP池：

nova-manage network create private --fixed\_range\_v4=10.0.0.0/24 --num\_networks=1 --bridge=br100 --bridge\_interface=eth1 --network\_size=255

重启nova服务

创建一个启动脚本start\_nova.sh

# cat start\_nova.sh

/etc/init.d/nova-api restart

/etc/init.d/nova-compute restart

/etc/init.d/nova-network restart

/etc/init.d/nova-scheduler restart

/etc/init.d/nova-cert restart

/etc/init.d/nova-consoleauth restart

/etc/init.d/nova-objectstore restart

/etc/init.d/nova-volume restart

执行下列命令测试nova是否正确安装：

root@ubuntu:~# nova-manage service list

2014-05-26 19:34:37 DEBUG nova.utils [req-74f364ef-2574-46b3-96e6-71b713afbdd7 None None] backend <module 'nova.db.sqlalchemy.api' from '/usr/lib/python2.7/dist-packages/nova/db/sqlalchemy/api.pyc'> from (pid=10200) \_\_get\_backend /usr/lib/python2.7/dist-packages/nova/utils.py:663

Binary Host Zone Status State Updated\_At

nova-consoleauth ubuntu nova enabled :-) 2014-05-26 11:34:36

nova-scheduler ubuntu nova enabled :-) 2014-05-26 11:34:27

nova-cert ubuntu nova enabled :-) 2014-05-26 11:34:27

nova-compute ubuntu nova enabled :-) 2014-05-26 11:34:31

nova-volume ubuntu nova enabled :-) 2014-05-26 11:34:28

nova-network ubuntu nova enabled :-) 2014-05-26 11:34:29

若所有组件都是微笑，说明nova已经正确安装完毕。

**OpenStack管理面板**

执行下列命令安装管理面板：

sudo apt-get install openstack-dashboard

重启Apache：

service apache2 restart

在浏览器里输入：http：//10.1.1.77，就可以看到OpenStack的管理面板登陆界面了。默认用户名和密码都是“admin”。在管理面板中，可以进行创建密匙对，创建、编辑安全群组，新建实例，添加卷等操作。