CentOS6.4 OpenStack-Havana 单网卡安装

(上海-Tom 原创)

目录

1. 准备环境

1.1. 操作系统

安装 CentOS6.4，单网卡安装 openstack，eth0 网卡 ip 地址为 192.168.200.12，网络使用 nova-network。
安装 openstack 之前先执行:
编辑/boot/grub/grub.conf
在 kernel 一行末尾加入：

```
ipv6.disable=1 selinux=0
```

然后执行：

```
service iptables stop
chkconfig iptables off
```

1.2. 源

增加 H 版本源和 EPEL 增强资源包：

```
rpm -Uvh http://rdo.fedorapeople.org/openstack-havana/rdo-release-havana.rpm
rpm -Uvh http://mirrors.ustc.edu.cn/fedora/epel/6Server/x86_64/epel-release-6-8.noarch.rpm
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
rpm --import RPM-GPG-KEY-RDO-Havana
yum install -y yum-priorities
yum clean all
yum -y update
```

1.3. NTP 服务

```
yum install -y ntp
service ntpd start
chkconfig ntpd on
```

编辑/etc/cron.daily/ntpdate

```
ntpdate 127.0.0.1 hwclock -w
```

1.4. 相关软件

消息队列和 Openstack 相关软件

```
yum install -y openstack-utils memcached dnsmasq-utils qpid-cpp-server
sed -i -e 's/auth=.*/auth=no/g' /etc/qpidd.conf
service qpidd start
chkconfig qpidd on
```

2. 安装数据库

2.1. 安装

```
yum install -y mysql mysql-server MySQL-python
```

2.2. 数据库说明和设置

编辑/etc/my.cnf
在[mysqld]下添加：

```
default_table_type=InnoDB
character-set-server=utf8
init_connect='SET NAMES utf8'
```

允许远程访问 mysql

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

重启服务

```
service mysqld restart
chkconfig mysqld on
```

设置 mysql：

```
mysql_secure_installation
```

设置 mysql 的 root 密码为 mysql。

3. 部署 Keystone

说明 Openstack 的组件都需要用到 mysql，下面是后面安装过程中，要建立的数据库。

所属租户	用户	密码	权限
admin	admin	openstack admin	可以看做超级管理员
admin	glance	openstack admin	glance 服务
admin	cinder	openstack admin	cinder 服务
admin	nova	openstack admin	nova 服务
admin	dash	openstack admin	horizon-web 后台

3.1. 安装

```
yum install -y openstack-keystone python-keystoneclient
```

3.2. 初始化数据库

```
openstack-db --init --service keystone --password keystone
```

生成 Token，并设置环境变量

```
export SERVICE_TOKEN=admin
export SERVICE_ENDPOINT=http://192.168.200.12:35357/v2.0
echo $SERVICE_TOKEN > /tmp/ks_admin_token
cat /tmp/ks_admin_token
```

3.3. 修改配置文件

```
openstack-config --set /etc/keystone/keystone.conf DEFAULT admin_token $SERVICE_TOKEN
```

```
openstack-config --set /etc/keystone/keystone.conf signing token_format UUID
```

3.4. 启动 keystone 服务

```
service openstack-keystone restart
chkconfig openstack-keystone on
```

查看是否启动正常

```
ps -ef | grep -i keystone-all
grep ERROR /var/log/keystone/keystone.log
```

3.5. 创建 Keystone 服务，并注册 Endpoint

创建 keystone 服务：

```
keystone service-create --name=keystone --type=identity --description="Keystone Identity Service"
```

输出：

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |    Keystone Identity Service     |
|      id     | ce01973d9eed42cda3495036ff2182d5 |
|     name    |             keystone             |
|     type    |             identity             |
+-------------+----------------------------------+
```

注册 Endpoint：

```
keystone endpoint-create --service_id ce01973d9eed42cda3495036ff2182d5 \
--publicurl 'http://192.168.200.12:5000/v2.0' \
--adminurl 'http://192.168.200.12:35357/v2.0' \
--internalurl 'http://192.168.200.12:5000/v2.0'
```

输出：

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
|   adminurl  | http://192.168.200.12:35357/v2.0 |
|      id     | 25d242cc35fc4f4a84b66d7b43b1f7eb |
| internalurl | http://192.168.200.12:5000/v2.0  |
|  publicurl  | http://192.168.200.12:5000/v2.0  |
|    region   |            regionOne             |
|  service_id | ce01973d9eed42cda3495036ff2182d5 |
+-------------+----------------------------------+
```

3.6. 创建 admin 用户

3.6.1. 创建 admin 用户

```
keystone user-create --name admin --pass admin
```

输出：

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |                                  |
| enabled  |               True               |
|    id    | dd6371a5ebdf454ea296e4b25562a6c5 |
|   name   |              admin               |
+----------+----------------------------------+
```

3.6.2. 创建 admin role

```
keystone role-create --name admin
```

输出：

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|    id    | ad496b04bc3944858a37d5a9ca78239f |
|   name   |              admin               |
+----------+----------------------------------+
```

3.6.3. 创建 admin tenant

```
keystone tenant-create --name admin
```

输出：

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |                                  |
|   enabled   |               True               |
|      id     | 7d62d7c6968e40a893897537f5f10c9a |
|     name    |              admin               |
+-------------+----------------------------------+
```

3.6.4. 将角色和用户关联起来

```
keystone user-role-add --user-id dd6371a5ebdf454ea296e4b25562a6c5 \
--role-id ad496b04bc3944858a37d5a9ca78239f \
--tenant-id 7d62d7c6968e40a893897537f5f10c9a
```

3.6.5. 设置 admin 的环境变量

```
vi keystonerc_admin
```

输入：

```
export OS_USERNAME=admin
export OS_TENANT_NAME=admin
export OS_PASSWORD=admin
export OS_AUTH_URL=http://192.168.200.12:35357/v2.0/
export PS1='[\u@\h \W(keystone_admin)]\$ '
```

3.6.6. 测试是否创建完成

```
unset SERVICE_TOKEN
unset SERVICE_ENDPOINT
source ~/keystonerc_admin
keystone user-list
```

输出：

```
+----------------------------------+-------+---------+-------+
|                id                |  name | enabled | email |
+----------------------------------+-------+---------+-------+
| dd6371a5ebdf454ea296e4b25562a6c5 | admin |   True  |       |
+----------------------------------+-------+---------+-------+
```

4. 部署 Glance

4.1. 安装 glance

```
yum install -y openstack-glance
```

4.2. 设置环境变量

```
source ~/keystonerc_admin
```

4.3. 初始数据库

```
openstack-db --init --service glance --password glance
```

4.4. 修改配置文件

```
openstack-config --set /etc/glance/glance-api.conf paste_deploy flavor keystone
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken admin_tenant_name admin
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken admin_user admin
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken admin_password admin
openstack-config --set /etc/glance/glance-registry.conf paste_deploy flavor keystone
openstack-config --set /etc/glance/glance-registry.conf keystone_authtoken admin_tenant_name
```

```
admin
openstack-config --set /etc/glance/glance-registry.conf keystone_authtoken admin_user admin
openstack-config  --set  /etc/glance/glance-registry.conf  keystone_authtoken  admin_password
admin
```

4.5. 重启服务

```
service openstack-glance-api restart
service openstack-glance-registry restart
chkconfig openstack-glance-api on
chkconfig openstack-glance-registry on
```

4.6. 创建服务

```
keystone service-create --name=glance --type=image --description="Glance Image Service"
```

输出：

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |       Glance Image Service       |
|      id     | 062a304ff36b4da6a05891975255cc3e |
|     name    |              glance              |
|     type    |              image               |
+-------------+----------------------------------+
```

4.7. 注册 Endpoint

```
keystone endpoint-create --service_id 062a304ff36b4da6a05891975255cc3e \
--publicurl http://192.168.200.12:9292 \
--adminurl http://192.168.200.12:9292 \
--internalurl http://192.168.200.12:9292
```

输出：

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
|   adminurl  |    http://192.168.200.12:9292    |
|      id     | 8636cdc6217945999b29b7ddc980e8a8 |
| internalurl |    http://192.168.200.12:9292    |
|  publicurl  |    http://192.168.200.12:9292    |
|    region   |             regionOne            |
|  service_id | 062a304ff36b4da6a05891975255cc3e |
+-------------+----------------------------------+
```

4.8. 重启服务

```
service openstack-glance-registry restart
service openstack-glance-api restart
```

4.9. 修改 glance 默认目录

```
mkdir -p /home/glance/image
cd /home
chown -R glance:glance  glance/image/
openstack-config  --set  /etc/glance/glance-api.conf  DEFAULT  filesystem_store_datadir
/home/glance/image/
service openstack-glance-registry restart
service openstack-glance-api restart
```

4.10. 验证安装

验证 Glance

```
glance image-list
```

应该没任何输出，就表示正确。因为目前还没有上传 image。

上传镜像测试：

```
glance  add  name="Windows  XP"  is_public=true  container_format=ovf  disk_format=qcow2  <
windowsxp.img
```

输出：

```
Added new image with ID: d6ec144d-7d5e-40d6-91bb-9cd565338a65
```

查看镜像：

```
glance image-list
```

输出：

```
+--------------------------------------+------------+-------------+------------------+------
------+--------+
| ID                                   | Name       | Disk Format | Container Format | Size
| Status |
+--------------------------------------+------------+-------------+------------------+------
------+--------+
| d6ec144d-7d5e-40d6-91bb-9cd565338a65 | Windows XP | qcow2       | ovf              |
5368709120 | active |
+--------------------------------------+------------+-------------+------------------+------
------+--------+
```

5. 部署 cinder

5.1. 安装

```
yum install -y openstack-cinder openstack-cinder-doc
```

如果是用网络存储做cinder-volumes，还要安装：

```
yum install -y iscsi-initiator-utils scsi-target-utils
```

5.2. 设置环境变量

```
source ~/keystonerc_admin
```

5.3. 初始化数据库

```
openstack-db --init --service cinder --password cinder
```

5.4. 修改配置文件

```
openstack-config --set /etc/cinder/cinder.conf DEFAULT auth_strategy keystone
openstack-config --set /etc/cinder/cinder.conf keystone_authtoken admin_tenant_name admin
openstack-config --set /etc/cinder/cinder.conf keystone_authtoken admin_user admin
openstack-config --set /etc/cinder/cinder.conf keystone_authtoken admin_password admin
```

5.5. 修改 tgt 配置文件

```
grep -q /etc/cinder/volumes /etc/tgt/targets.conf || sed -i 'liinclude /etc/cinder/volumes/*'
/etc/tgt/targets.conf
```

5.6. 启动 tgt

```
service tgtd start
chkconfig tgtd on
```

5.7. 创建 cinder-volumes

使用一个空闲独立的分区，这里这个分区为/dev/sdb1

```
pvcreate /dev/sdb1
vgcreate cinder-volumes /dev/sdb1
vgdisplay cinder-volumes
```

5.8. 启动服务

```
for srv in api scheduler volume ; do \
sudo service openstack-cinder-$srv start ; \
done
```

5.9. 设置开机启动

```
for srv in api scheduler volume ; do \
sudo chkconfig openstack-cinder-$srv on ; \
```

```
done
```

5.10. 检查是否有报错

```
grep -i ERROR /var/log/cinder/*
grep CRITICAL /var/log/cinder/*
```

5.11. 创建服务

```
keystone service-create --name=cinder --type=volume --description="Cinder Volume Service"
```

输出：

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |      Cinder Volume Service       |
|      id     | 037d94a07d8c409e8823b2a3b1641341 |
|     name    |              cinder              |
|     type    |              volume              |
+-------------+----------------------------------+
```

5.12. 注册 endpoint

```
keystone endpoint-create --service_id 037d94a07d8c409e8823b2a3b1641341 \
--publicurl "http://192.168.200.12:8776/v1/\$(tenant_id)s" \
--adminurl "http:// 192.168.200.12:8776/v1/\$(tenant_id)s" \
--internalurl "http:// 192.168.200.12:8776/v1/\$(tenant_id)s"
```

输出：

```
+-------------+-----------------------------------------+
|   Property  |                  Value                  |
+-------------+-----------------------------------------+
|   adminurl  | http:// 192.168.200.12:8776/v1/$(tenant_id)s |
|      id     |       ff72010e46db42fe8f200330e18d7376       |
| internalurl | http:// 192.168.200.12:8776/v1/$(tenant_id)s |
|  publicurl  | http://192.168.200.12:8776/v1/$(tenant_id)s  |
|    region   |                  regionOne                  |
|  service_id |       037d94a07d8c409e8823b2a3b1641341       |
+-------------+-----------------------------------------+
```

5.13. 配置 iscsi 服务（有网络存储配置此项）

```
sed -i 's/false/true/g' /etc/default/iscsitarget
```

5.14. 重启 cinder 服务

```
for srv in api scheduler volume ; do \
```

```
sudo service openstack-cinder-$srv restart ; \
done
```

5.15. 测试是否正常

```
cinder list
```

应该没任何输出，就表示正确。因为目前还没有创建卷。

查看 cinder 服务状态

```
for srv in api scheduler volume ; do \
sudo service openstack-cinder-$srv status ; \
done
```

6. 部署 Nova

6.1. 网络配置

```
ip link set eth0 promisc on
```

6.2. 配置网桥

编辑/etc/sysconfig/network-scripts/ifcfg-br100

输入：

```
DEVICE=br100
TYPE=Bridge
ONBOOT=yes
DELAY=0
BOOTPROTO=static
IPADDR=192.168.32.1
NETMASK=255.255.255.0
```

添加网桥br100

```
yum install -y bridge-utils
brctl addbr br100
```

重启网络

```
service network restart
```

6.3. 安装

```
yum install -y openstack-nova python-cinderclient
```

6.4. 初始化数据库

```
openstack-db --init --service nova --password nova
```

6.5. 修改配置文件

```
openstack-config --set /etc/nova/nova.conf DEFAULT auth_strategy keystone
openstack-config --set /etc/nova/api-paste.ini filter:authtoken admin_token admin
```

编辑 nova.conf

```
[DEFAULT]
notification_driver=ceilometer.compute.nova_notifier
notification_driver=nova.openstack.common.notifier.rpc_notifier
state_path=/var/lib/nova
enabled_apis=ec2,osapi_compute,metadata
ec2_listen=0.0.0.0
osapi_compute_listen=0.0.0.0
osapi_compute_workers=4
metadata_listen=0.0.0.0
network_manager=nova.network.manager.FlatDHCPManager
service_down_time=60
instance_usage_audit_period=hour
rootwrap_config=/etc/nova/rootwrap.conf
api_paste_config=/etc/nova/api-paste.ini
auth_strategy=keystone
use_forwarded_for=False
service_neutron_metadata_proxy=False
novncproxy_host=0.0.0.0
novncproxy_port=6080
instance_usage_audit=True
glance_api_servers=192.168.200.12:9292
default_floating_pool=nova
auto_assign_floating_ip=False
dhcpbridge_flagfile=/etc/nova/nova.conf
public_interface=eth0
dhcpbridge=/usr/bin/nova-dhcpbridge
flat_network_bridge=br100
flat_injected=False
flat_interface=lo
force_dhcp_release=False
dhcp_domain=novalocal
lock_path=/var/lib/nova/tmp
debug=True
verbose=True
rpc_backend=nova.openstack.common.rpc.impl_qpid
qpid_hostname=192.168.200.12
qpid_port=5672
qpid_username=guest
qpid_password=guest
qpid_heartbeat=60
qpid_tcp_nodelay=True
cpu_allocation_ratio=16.0
```

```
ram_allocation_ratio=1.5
scheduler_default_filters=RetryFilter,AvailabilityZoneFilter,RamFilter,ComputeFilter,Compute
CapabilitiesFilter,ImagePropertiesFilter,CoreFilter
compute_driver=libvirt.LibvirtDriver
libvirt_type=qemu
libvirt_inject_partition=-1
libvirt_cpu_mode=none
novncproxy_base_url=http://192.168.200.12:6080/vnc_auto.html
vncserver_listen=192.168.200.12
vncserver_proxyclient_address=192.168.200.12
vnc_enabled=True
volume_api_class=nova.volume.cinder.API
qpid_reconnect_interval=0
qpid_reconnect_interval_min=0
qpid_reconnect=True
sql_connection=mysql://nova:nova@192.168.200.12/nova
qpid_reconnect_timeout=0
image_service=nova.image.glance.GlanceImageService
logdir=/var/log/nova
qpid_reconnect_interval_max=0
qpid_reconnect_limit=0
osapi_volume_listen=0.0.0.0
fixed_range=192.168.32.0/24
floating_range=192.168.200.128/25
connection_type=libvirt


[hyperv]
#


[zookeeper]

#


[osapi_v3]

#

[conductor]

#
```

[keymgr]

#

[cells]

#

[database]

#

[rpc_notifier2]

#

[matchmaker_redis]

#

[ssl]

#

[trusted_computing]

#

[upgrade_levels]

#

[matchmaker_ring]

#

[vmware]

#

[spice]

#

[keystone_authtoken]

#

6.6. 同步数据库

```
nova-manage db sync
```

6.7. 设置环境变量

```
source keystonerc_admin
```

6.8. 创建服务

```
keystone service-create --name=nova --type=compute --description="Nova Compute Service"
```

输出：

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |       Nova Compute Service       |
|      id     | a24baec28b7445c8b52abf8fbed452ec |
|     name    |               nova               |
|     type    |             compute              |
+-------------+----------------------------------+
```

6.9. 注册 endpoint

```
keystone endpoint-create --service_id a24baec28b7445c8b52abf8fbed452ec \
--publicurl "http://192.168.200.12:8774/v1.1/\$(tenant_id)s" \
--adminurl "http://192.168.200.12:8774/v1.1/\$(tenant_id)s" \
--internalurl "http://192.168.200.12:8774/v1.1/\$(tenant_id)s"
```

输出：

```
+-------------+-----------------------------------------+
|   Property  |                  Value                  |
+-------------+-----------------------------------------+
|   adminurl  | http://192.168.200.12:8774/v1.1/$(tenant_id)s |
|      id     |        ebf0199b35e64506b586d6ca414cf34a       |
| internalurl | http://192.168.200.12:8774/v1.1/$(tenant_id)s |
|  publicurl  | http://192.168.200.12:8774/v1.1/$(tenant_id)s |
|    region   |                  regionOne                  |
|  service_id |        a24baec28b7445c8b52abf8fbed452ec       |
```

```
+------------+-------------------------------------------+
```

6.10. 启动 nova 相关服务

```
service messagebus start
service libvirtd start
for svc in api cells cert compute conductor console consoleauth network novncproxy objectstore
scheduler xvpvncproxy;do sudo service openstack-nova-$svc start; done
```

6.11. 设置开机启动

```
chkconfig messagebus on
chkconfig libvirtd on
for svc in api cells cert compute conductor console consoleauth network novncproxy objectstore
scheduler xvpvncproxy;do sudo chkconfig openstack-nova-$svc on; done
```

6.12. 验证安装

查看 nova 服务

```
nova-manage service list
```

7. 部署 Dashboard

7.1. 安装

```
yum install -y memcached python-memcached mod_wsgi openstack-dashboard
yum install  -y python-pip
pip install pbr
```

编辑/etc/openstack-dashboard/ local_settings

```
import os

from django.utils.translation import ugettext_lazy as _

from openstack_dashboard import exceptions

DEBUG = False
TEMPLATE_DEBUG = DEBUG

# Set SSL proxy settings:
# For Django 1.4+ pass this header from the proxy after terminating the SSL,
# and don't forget to strip it from the client's request.
# For more information see:
# https://docs.djangoproject.com/en/1.4/ref/settings/#secure-proxy-ssl-header
# SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTOCOL', 'https')
```

```
# If Horizon is being served through SSL, then uncomment the following two
# settings to better secure the cookies from security exploits
#CSRF_COOKIE_SECURE = True
#SESSION_COOKIE_SECURE = True

# Default OpenStack Dashboard configuration.
HORIZON_CONFIG = {
    'dashboards': ('project', 'admin', 'settings',),
    'default_dashboard': 'project',
    'user_home': 'openstack_dashboard.views.get_user_home',
    'ajax_queue_limit': 10,
    'auto_fade_alerts': {
        'delay': 3000,
        'fade_duration': 1500,
        'types': ['alert-success', 'alert-info']
    },
    'help_url': "http://docs.openstack.org",
    'exceptions': {'recoverable': exceptions.RECOVERABLE,
                   'not_found': exceptions.NOT_FOUND,
                   'unauthorized': exceptions.UNAUTHORIZED},
}

# Specify a regular expression to validate user passwords.
# HORIZON_CONFIG["password_validator"] = {
#     "regex": '.*',
#     "help_text": _("Your password does not meet the requirements.")
# }

# Disable simplified floating IP address management for deployments with
# multiple floating IP pools or complex network requirements.
# HORIZON_CONFIG["simple_ip_management"] = False

# Turn off browser autocompletion for the login form if so desired.
# HORIZON_CONFIG["password_autocomplete"] = "off"

LOCAL_PATH = os.path.dirname(os.path.abspath(__file__))

# Set custom secret key:
# You can either set it to a specific value or you can let horizion generate a
# default secret key that is unique on this machine, e.i. regardless of the
# amount of Python WSGI workers (if used behind Apache+mod_wsgi): However, there
# may be situations where you would want to set this explicitly, e.g. when
# multiple dashboard instances are distributed on different machines (usually
```

```
# behind a load-balancer). Either you have to make sure that a session gets all
# requests routed to the same dashboard instance or you set the same SECRET_KEY
# for all of them.
# from horizon.utils import secret_key
#  SECRET_KEY  =  secret_key.generate_or_read_from_file(os.path.join(LOCAL_PATH,
'.secret_key_store'))
SECRET_KEY = '38e2ac93c24f4baea6cf4f063c37e204'

# We recommend you use memcached for development; otherwise after every reload
# of the django development server, you will have to login again. To use
# memcached set CACHES to something like
# CACHES = {
#    'default': {
#        'BACKEND' : 'django.core.cache.backends.memcached.MemcachedCache',
#        'LOCATION' : '127.0.0.1:11211',
#    }
#}

CACHES = {
    'default': {

        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:11211',

    }
}

# Send email to the console by default
EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
# Or send them to /dev/null
#EMAIL_BACKEND = 'django.core.mail.backends.dummy.EmailBackend'

# Configure these for your outgoing email host
# EMAIL_HOST = 'smtp.my-company.com'
# EMAIL_PORT = 25
# EMAIL_HOST_USER = 'djangomail'
# EMAIL_HOST_PASSWORD = 'top-secret!'

# For multiple regions uncomment this configuration, and add (endpoint, title).
# AVAILABLE_REGIONS = [
#     ('http://cluster1.example.com:5000/v2.0', 'cluster1'),
#     ('http://cluster2.example.com:5000/v2.0', 'cluster2'),
# ]
```

```
OPENSTACK_HOST = "192.168.200.12"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v2.0" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "_member_"

# Disable SSL certificate checks (useful for self-signed certificates):
# OPENSTACK_SSL_NO_VERIFY = True

# The OPENSTACK_KEYSTONE_BACKEND settings can be used to identify the
# capabilities of the auth backend for Keystone.
# If Keystone has been configured to use LDAP as the auth backend then set
# can_edit_user to False and name to 'ldap'.
#
# TODO(tres): Remove these once Keystone has an API to identify auth backend.
OPENSTACK_KEYSTONE_BACKEND = {
    'name': 'native',
    'can_edit_user': True,
    'can_edit_project': True
}

OPENSTACK_HYPERVISOR_FEATURES = {
    'can_set_mount_point': True,

    # NOTE: as of Grizzly this is not yet supported in Nova so enabling this
    # setting will not do anything useful
    'can_encrypt_volumes': False
}

# The OPENSTACK_NEUTRON_NETWORK settings can be used to enable optional
# services provided by neutron.  Currently only the load balancer service
# is available.
OPENSTACK_NEUTRON_NETWORK = {
    'enable_lb': False
}

# OPENSTACK_ENDPOINT_TYPE specifies the endpoint type to use for the endpoints
# in the Keystone service catalog. Use this setting when Horizon is running
# external to the OpenStack environment. The default is 'internalURL'.
#OPENSTACK_ENDPOINT_TYPE = "publicURL"

# The number of objects (Swift containers/objects or images) to display
# on a single page before providing a paging element (a "more" link)
# to paginate results.
API_RESULT_LIMIT = 1000
API_RESULT_PAGE_SIZE = 20
```

```
# The timezone of the server. This should correspond with the timezone
# of your entire OpenStack installation, and hopefully be in UTC.
TIME_ZONE = "UTC"

# If you have external monitoring links, eg:


LOGGING = {
    'version': 1,
    # When set to True this will disable all logging except
    # for loggers specified in this configuration dictionary. Note that
    # if nothing is specified here and disable_existing_loggers is True,
    # django.db.backends will still log unless it is disabled explicitly.
    'disable_existing_loggers': False,
    'handlers': {
        'null': {
            'level': 'DEBUG',
            'class': 'django.utils.log.NullHandler',
        },
        'console': {
            # Set the level to "DEBUG" for verbose output logging.
            'level': 'INFO',
            'class': 'logging.StreamHandler',
        },
        'file': {
            'level': 'DEBUG',
            'class': 'logging.FileHandler',
            'filename': '/var/log/horizon/horizon.log',
        },
    },
    'loggers': {
        # Logging from django.db.backends is VERY verbose, send to null
        # by default.
        'django.db.backends': {
            'handlers': ['null'],
            'propagate': False,
        },
        'requests': {
            'handlers': ['null'],
            'propagate': False,
        },
        'horizon': {
            'handlers': ['file'],
```

```
            'propagate': False,
        },
        'openstack_dashboard': {
            'handlers': ['file'],
            'propagate': False,
        },
        'novaclient': {
            'handlers': ['file'],
            'propagate': False,
        },
        'keystoneclient': {
            'handlers': ['file'],
            'propagate': False,
        },
        'glanceclient': {
            'handlers': ['file'],
            'propagate': False,
        },
        'nose.plugins.manager': {
            'handlers': ['file'],
            'propagate': False,
        }
    }
}

LOGIN_URL = '/dashboard/auth/login/'
LOGIN_REDIRECT_URL = '/dashboard'

# The Ubuntu package includes pre-compressed JS and compiled CSS to allow
# offline compression by default.  To enable online compression, install
# the node-less package and enable the following option.
COMPRESS_OFFLINE = True
```

在/var/log/目录下

```
mkdir horizon
```

然后

```
cd horizon
touch horizon.log
cd ..
chown  -R apache:apache horizon
/usr/share/openstack-dashboard/manage.py syncdb
```

7.2. 重启相关服务

```
service httpd restart
```

service memcached restart
chkconfig httpd on
chkconfig memcached on

7.3. 验证安装

这个时候你就可以直接使用 http://192.168.200.12/dashboard 访问
用户名:admin 密码:admin