Twitter: @Longgeek
Blog: http://www.longgeek.com
Weibo: http://weibo.com/longgeek

本文地址：http://longgeek.com/2013/03/11/openstack-grizzly-g3-for-ubuntu-12-04-all-in-one-installation/

前言：

Grizzly 发布日期为：2013.04.04
本文 Grizzly 的版本为：2013.01.g3

本文会安装 Keystone、Glance、Quantum、Cinder、Nova、Horizon，其中 cinder 有点问题，戳这里.

Quantum 采用 GRE 模式, 关于 Quantum 模式详细介绍点击这里，在写这篇文档之前网上没有找到相关 Grizzly 安装的资料，可能本文会有披漏，欢迎大家指正。

目录 [隐藏]

网络环境

GRE 模式最少需要三块网卡，我在这里用了三个网卡。

1.管理网络： eth0 172.16.0.254/16 用来 mysql、AMQP
2.业务网络： eth1 10.0.0.1/8 各计算节点通讯、api
3.外部网络： eth2 192.168.8.20/24 br-ex

网卡设置

eth2 用来做 quantum 的 external 网络，暂时没有把 ip 地址写到配置文件里，在后面配置 ovs 时候会在文件增加一个 br-ex 网卡信息.

```
# cat /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 172.16.0.254
    netmask 255.255.0.0

auto eth1
iface eth1 inet static
    address 10.0.0.1
    netmask 255.0.0.0

auto eth2
iface eth2 inet manual
# /etc/init.d/networking restart
# ifconfig eth2 192.168.8.20/24 up
# route add default gw 192.168.8.1 dev eth2
# echo '8.8.8.8' > /etc/resolv.conf
```

添加 Grizzly 源, 并更新软件包

```
# echo 'deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-
updates/grizzly main' > /etc/apt/sources.list.d/grizzly.list
# apt-get install ubuntu-cloud-keyring
# apt-get update
# apt-get upgrade
```

安装 mysql

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

使用 sed 编辑 /etc/mysql/my.cnf 文件的更改绑定地址（0.0.0.0）从本地主机（127.0.0.1）
禁止 mysql 做域名解析，防止 连接 mysql 出现错误和远程连接 mysql 慢的现象。
然后重新启动 mysql 服务.

```
# sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
# sed -i '44 i skip-name-resolve' /etc/mysql/my.cnf
# /etc/init.d/mysql restart
```

安装 RabbitMQ

安装休息队列服务器，RabbitMQ，或者你也可以安装 Apache Qpid。

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

安装和配置 Keystone

```
# apt-get install keystone
```

删除默认 keystone 的 sqlite db 文件

```
# rm -f /var/lib/keystone/keystone.db
```

创建 keystone 数据库

在 mysql 里创建 keystone 库，并授权 keystone 用户访问:

```
# mysql -uroot -pmysql
mysql> create database keystone;
mysql> grand all on keystone.* to 'keystone'@'%' identified by 'keystone';
mysql> flush privileges; quit;
```

改 keystone.conf

修改 /etc/keystone/keystone.conf:

```
admin_token = www.longgeek.com
debug = True
verbose = True
connection = mysql://keystone:keystone@172.16.0.254/keystone
token_format = UUID
```

启动 keystone 服务:

```
/etc/init.d/keystone restart
```

同步 keystone 表数据到 db 中:

```
keystone-manage db_sync
```

用脚本导入数据

创建 user、role、tenant、service、endpoint:

下载脚本:

```
# wget http://download.longgeek.com/openstack/grizzly/keystone.sh
```

自定义脚本内容:

```
ADMIN_PASSWORD=${ADMIN_PASSWORD:-password}     #租户 admin 的密码
SERVICE_PASSWORD=${SERVICE_PASSWORD:-password}
#nova,glance,cinder,quantum,swift 的密码
export SERVICE_TOKEN="www.longgeek.com"    # token
export SERVICE_ENDPOINT="http://172.16.0.254:35357/v2.0"
SERVICE_TENANT_NAME=${SERVICE_TENANT_NAME:-service}      #租户 service, 包含了
nova,glance,ciner,quantum,swift 等服务
KEYSTONE_REGION=RegionOne
KEYSTONE_IP="172.16.0.254"
#KEYSTONE_WLAN_IP="172.16.0.254"
SWIFT_IP="172.16.0.254"
#SWIFT_WLAN_IP="172.16.0.254"
COMPUTE_IP=$KEYSTONE_IP
EC2_IP=$KEYSTONE_IP
GLANCE_IP=$KEYSTONE_IP
VOLUME_IP=$KEYSTONE_IP
QUANTUM_IP=$KEYSTONE_IP
```

执行脚本:

```
# sh keystone.sh
```

设置环境变量

这里变量对于 keystone.sh 里的设置:

```
# cat > /root/export.sh << _GEEK_
export OS_TENANT_NAME=admin      #这里如果设置为 service 其它服务会无法验证.
export OS_USERNAME=admin
export OS_PASSWORD=password
export OS_AUTH_URL=http://172.16.0.254:5000/v2.0/
export OS_REGION_NAME=RegionOne
export SERVICE_TOKEN=www.longgeek.com
export SERVICE_ENDPOINT=http://172.16.0.254:35357/v2.0/
_GEEK_
# echo 'source /root/export.sh' >> /root/.bashrc
# source /root/export.sh
```

验证 keystone

```
keystone user-list
keystone role-list
keystone tenant-list
keystone endpoint-list
```

Troubleshooting Keystone

1. 查看 5000 和 35357 端口是否在监听
2. 查看 /var/log/keystone/keystone.log 报错信息
3. keystone.sh 脚本执行错误解决：(检查脚本内容变量设置)

```
# mysql -uroot -pmysql
mysql> drop database keystone;
mysql> create database keystone; quit;
# keystone-manage db_sync
# sh keystone.sh
```

4. 步骤 6.5 出现错误，先去查看 log，在检查 6.4 环境变量是否设置正确

安装和配置 Glance

安装 glance

```
# apt-get install glance
```

删除 glance sqlite 文件：

```
# rm -f /var/lib/glance/glance.sqlite
```

创建 glance 数据库

```
# mysql -uroot -pmysql
mysql> create database glance;
mysql> grant all on glance.* to 'glance'@'%' identified by 'glance';
mysql> flush privileges;
```

修改 glance 配置文件

修改 glance-api.conf

修改下面的选项，其它默认。

```
verbose = True
debug = True
sql_connection = mysql://glance:glance@172.16.0.254/glance
workers = 4
registry_host = 172.16.0.254
notifier_strategy = rabbit
rabbit_host = 172.16.0.254
rabbit_userid = guest
rabbit_password = guest
[keystone_authtoken]
auth_host = 172.16.0.254
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = password
```

```
[paste_deploy]
config_file = /etc/glance/glance-api-paste.ini
flavor = keystone
```

修改 glance-registry.conf

修改下面的选项，其它默认。

```
verbose = True
debug = True
sql_connection = mysql://glance:glance@172.16.0.254/glance
[keystone_authtoken]
auth_host = 172.16.0.254
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = password
[paste_deploy]
config_file = /etc/glance/glance-registry-paste.ini
flavor = keystone
```

启动 glance 服务：

```
# /etc/init.d/glance-api restart
# /etc/init.d/glance-registry restart
```

同步到 db

```
# glance-manage version_control 0
# glance-manage db_sync
```

检查 glance

```
# glance image-list
```

上传镜像文件

下载 Cirros img 作为测试使用，只有 10M：

```
# wget https://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-
disk.img
# glance image-create --name='cirros' --public --container-format=ovf --disk-
format=qcow2 < ./cirros-0.3.0-x86_64-disk.img
Added new image with ID: f61ee640-82a7-4d6c-8816-608bb91dab7d
```

Cirros img 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:)

Troubleshooting Glance

1. 确保配置文件正确，9191 9292 端口存在
2. /var/log/glance/ 两个 log 文件

3. 确保环境变量中的 OS_TENANT_NAME=admin, 否则会报 401 错误
4. 上传镜像的格式对应命令中指定的格式

安装 Openvswitch

```
# apt-get install openvswitch-controller openvswitch-brcompat openvswitch-switch
```

设置 ovs-brcompatd 启动：

```
# sed -i 's/no/yes/g' /etc/default/openvswitch-switch
```

启动 openvswitch-switch:

```
# /etc/init.d/openvswitch-switch restart
 * ovs-brcompatd is not running            #brcompatd 没有启动
 * ovs-vswitchd is not running
 * ovsdb-server is not running
 * Inserting openvswitch module
 * /etc/openvswitch/conf.db does not exist
 * Creating empty database /etc/openvswitch/conf.db
 * Starting ovsdb-server
 * Configuring Open vSwitch system IDs
 * Starting ovs-vswitchd
 * Enabling gre with iptables
```

再次启动,直到 ovs-brcompatd、ovs-vswitchd、ovsdb-server 等服务都启动：

```
# /etc/init.d/openvswitch-switch restart
```

添加网桥

添加 External 网络网桥 br-ex

用 openvswitch 添加网桥 br-ex 并把网卡 eth2 加入 br-ex：

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

做完上面操作后，eth2 这个网卡是没有工作的，手工设置 ip：

```
# ifconfig eth2 0
# ifconfig br-ex 192.168.8.20/24
# route add default gw 192.168.8.1 dev br-ex
# echo '8.8.8.8' > /etc/resolv.conf
```

在写到网卡配置文件：

```
# cat /etc/network/interfaces
auto lo
iface lo inet loopback
```

```
auto eth0
iface eth0 inet static
    address 172.16.0.254
    netmask 255.255.0.0

auto eth1
iface eth1 inet static
    address 10.0.0.1
    netmask 255.0.0.0

auto eth2
iface eth2 inet manual
    up ifconfig $IFACE 0.0.0.0 up
    down ifconfig $IFACE down

auto br-ex
iface br-ex inet static
    address 192.168.8.20
    netmask 255.255.255.0
    gateway 192.168.8.1
    dns-nameservers 8.8.8.8
```

重启网卡可能会出现：

```
RTNETLINK answers: File exists
Failed to bring up br-ex.
```

br-ex 可能有 ip 地址，但没有网关和 DNS，需要手工配置一下，或者重启机器. 重启机器后就正常了。

创建 internal 网络 br-int

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

查看网络

```
# ovs-vsctl list-br
br-ex
br-int
# ovs-vsctl show
1a8d2081-4ba4-4cad-8020-ccac5772836a
  Bridge br-int
    Port br-int
      Interface br-int
        type: internal
  Bridge br-ex
    Port br-ex
      Interface br-ex
        type: internal
    Port "eth2"
      Interface "eth2"
```

```
  ovs_version: "1.4.0+build0"
```

安装 quantum

安装 Quantum 服务器和 Client API：

```
apt-get install quantum-server python-cliff python-pyparsing python-quantumclient
```

安装 openvswitch 插件来支持 OVS:

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

创建 Quantum DB

```
# mysql -uroot -pmysql
mysql> create database quantum;
mysql> grant all on quantum.* to 'quantum'@'%' identified by 'quantum';
mysql> flush privileges; quit;
```

配置 quantum.conf

```
# cat /etc/quantum/quantum.conf | grep -v ^$ | grep -v ^#
[DEFAULT]
debug = True
verbose = True
state_path = /var/lib/quantum
bind_host = 0.0.0.0
bind_port = 9696
core_plugin = quantum.plugins.openvswitch.ovs_quantum_plugin.OVSQuantumPluginV2
api_paste_config = /etc/quantum/api-paste.ini
control_exchange = quantum
rabbit_host = 172.16.0.254
rabbit_password = guest
rabbit_port = 5672
rabbit_userid = guest
notification_driver = quantum.openstack.common.notifier.rpc_notifier
default_notification_level = INFO
notification_topics = notifications
[QUOTAS]
[DEFAULT_SERVICETYPE]
[SECURITYGROUP]
[AGENT]
root_helper = sudo quantum-rootwrap /etc/quantum/rootwrap.conf
[keystone_authtoken]
auth_host = 172.16.0.254
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = quantum
admin_password = password
signing_dir = /var/lib/quantum/keystone-signing
```

配置 Open vSwitch Plugin

```
# cat /etc/quantum/plugins/openvswitch/ovs_quantum_plugin.ini | grep -v ^$ | grep
-v ^#
[DATABASE]
sql_connection = mysql://quantum:quantum@172.16.0.254/quantum
reconnect_interval = 2
[OVS]
enable_tunneling = True
tenant_network_type = gre
tunnel_id_ranges = 1:1000
local_ip = 10.0.0.1
integration_bridge = br-int
tunnel_bridge = br-tun
[AGENT]
polling_interval = 2
[SECURITYGROUP]
```

启动 quantum 服务

```
# /etc/init.d/quantum-server restart
```

安装 OVS agent

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

启动 ovs-agent 时候确保 ovs_quantum_plugin.ini 里有 local_ip 存在. 确保 br-int 网桥已创建.

```
# /etc/init.d/quantum-plugin-openvswitch-agent restart
```

启动 ovs-agent 后会根据配置文件自动创建一个 br-tun 网桥:

```
# ovs-vsctl list-br
br-ex
br-int
br-tun
# ovs-vsctl show
1a8d2081-4ba4-4cad-8020-ccac5772836a
  Bridge br-int
    Port br-int
      Interface br-int
        type: internal
    Port patch-tun
      Interface patch-tun
        type: patch
        options: {peer=patch-int}
  Bridge br-ex
    Port br-ex
      Interface br-ex
        type: internal
    Port "eth2"
```

```
    Interface "eth2"
  Bridge br-tun
    Port br-tun
      Interface br-tun
        type: internal
    Port patch-int
      Interface patch-int
        type: patch
        options: {peer=patch-tun}
  ovs_version: "1.4.0+build0"
```

安装 quantum-dhcp-agent

```
# apt-get install quantum-dhcp-agent
```

配置 quantum-dhcp-agent:

```
# cat /etc/quantum/dhcp_agent.ini | grep -v ^$ | grep -v ^#
[DEFAULT]
debug = True
verbose = True
use_namespaces = True
signing_dir = /var/cache/quantum
admin_tenant_name = service
admin_user = quantum
admin_password = password
auth_url = http://172.16.0.254:35357/v2.0
dhcp_agent_manager = quantum.agent.dhcp_agent.DhcpAgentWithStateReport
root_helper = sudo quantum-rootwrap /etc/quantum/rootwrap.conf
state_path = /var/lib/quantum
interface_driver = quantum.agent.linux.interface.OVSInterfaceDriver
dhcp_driver = quantum.agent.linux.dhcp.Dnsmasq
```

启动服务:

```
# /etc/init.d/quantum-dhcp-agent restart
```

安装 L3 Agent

```
# apt-get install quantum-l3-agent
```

配置 L3 Agent:

```
# cat /etc/quantum/l3_agent.ini | grep -v ^$ | grep -v ^#
[DEFAULT]
debug = True
verbose = True
use_namespaces = True
external_network_bridge = br-ex
signing_dir = /var/cache/quantum
admin_tenant_name = service
admin_user = quantum
```

```
admin_password = password
auth_url = http://172.16.0.254:35357/v2.0
l3_agent_manager = quantum.agent.l3_agent.L3NATAgentWithStateReport
root_helper = sudo quantum-rootwrap /etc/quantum/rootwrap.conf
interface_driver = quantum.agent.linux.interface.OVSInterfaceDriver
```

启动 L3 agent:

```
# /etc/init.d/quantum-l3-agent restart
```

配置 Metadata agent

```
# cat /etc/quantum/metadata_agent.ini | grep -v ^$ | grep -v ^#

[DEFAULT]
debug = True
auth_url = http://172.16.0.254:35357/v2.0
auth_region = RegionOne
admin_tenant_name = service
admin_user = quantum
admin_password = password
state_path = /var/lib/quantum
nova_metadata_ip = 172.16.0.254
nova_metadata_port = 8775
```

启动 Metadata agent:

```
# /etc/init.d/quantum-metadata-agent restart
```

Troubleshooting Quantum

1. 所有配置文件配置正确，9696 端口启动
2. /var/log/quantum/下所有 log 文件
3. br-ex、br-int 提前添加好

在文档末尾会用命令和界面方式结合来理解 Quantum 网络。

安装 Cinder

在 Grizzly 里 Cinder 有一个 Bug, 先配置好再说吧:

```
# apt-get install cinder-api cinder-common cinder-scheduler cinder-volume
```

创建 DB

```
# mysql -uroot -pmysql
mysql> create database cinder;
mysql> grant all on cinder.* to 'cinder'@'%' identified by 'cinder';
mysql> flush privileges; quit;
```

建立一个逻辑卷卷组 cinder-volumes

创建一个普通分区，我这里用的 sdb，创建了一个主分区，大小为所有空间

```
# fdisk /dev/sdb
n
p
1
Enter
Enter
t
8e
w
# partx -a /dev/sdb
# pvcreate /dev/sdb1
# vgcreate cinder-volumes /dev/sdb1
# vgs
 VG             #PV #LV #SN Attr   VSize   VFree
 cinder-volumes   1   0   0 wz--n- 150.00g 150.00g
 localhost        1   2   0 wz--n- 279.12g  12.00m
```

修改配置文件

修改 cinder.conf

```
# cat /etc/cinder/cinder.conf
[DEFAULT]
# LOG/STATE
verbose = True
debug = True
iscsi_helper = tgtadm
auth_strategy = keystone
volume_group = cinder-volumes
volume_name_template = volume-%s
state_path = /var/lib/cinder
volumes_dir = /var/lib/cinder/volumes
rootwrap_config = /etc/cinder/rootwrap.conf
api_paste_confg = /etc/cinder/api-paste.ini
# RPC
rabbit_host = 172.16.0.254
rabbit_password = guest
rpc_backend = cinder.openstack.common.rpc.impl_kombu
# DATABASE
sql_connection = mysql://cinder:cinder@172.16.0.254/cinder
# API
osapi_volume_extension = cinder.api.contrib.standard_extensions
```

修改 api-paste.ini

修改文件末尾[filter:authtoken]字段：

```
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
service_protocol = http
service_host = 172.16.0.254
service_port = 5000
```

```
auth_host = 172.16.0.254
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = cinder
admin_password = password
signing_dir = /var/lib/cinder
```

同步并启动服务

同步到 db 中:

```
# cinder-manage db sync
2013-03-11 13:41:57.885 30326 DEBUG cinder.utils [-] backend <module
'cinder.db.sqlalchemy.migration' from '/usr/lib/python2.7/dist-
packages/cinder/db/sqlalchemy/migration.pyc'> __get_backend
/usr/lib/python2.7/dist-packages/cinder/utils.py:561
```

启动服务:

```
# for serv in api scheduler volume
do
  /etc/init.d/cinder-$serv restart
done
# /etc/init.d/tgt restart
```

检查

```
# cinder list
```

Troubleshooting Cinder

1. 服务和 8776 端口启动
2. /var/log/cinder 中日志文件
3. 依赖配置文件指定的 volume_group = cinder-volumes, 卷组存在
4. tgt 服务正常.

安装 Nova 控制器

同时安装计算服务，Grizzly 里 nova-compute 依赖 nova-conductor，戳这里

```
# apt-get install nova-api nova-novncproxy novnc nova-ajax-console-proxy nova-cert
nova-consoleauth nova-doc nova-scheduler
# apt-get install nova-compute nova-conductor
```

创建数据库

```
# mysql -uroot -pmysql
mysql> create database nova;
mysql> grant all on nova.* to 'nova'@'%' identified by 'nova';
mysql> flush privileges; quit;
```

配置

配置 nova.conf

```
# cat /etc/nova/nova.conf
[DEFAULT]
# LOGS/STATE
debug = True
verbose = True
logdir = /var/log/nova
state_path = /var/lib/nova
lock_path = /var/lock/nova
rootwrap_config = /etc/nova/rootwrap.conf
dhcpbridge = /usr/bin/nova-dhcpbridge
# SCHEDULER
compute_scheduler_driver = nova.scheduler.filter_scheduler.FilterScheduler
## VOLUMES
volume_api_class = nova.volume.cinder.API
# DATABASE
sql_connection = mysql://nova:nova@172.16.0.254/nova
# COMPUTE
libvirt_type = kvm
compute_driver = libvirt.LibvirtDriver
instance_name_template = instance-%08x
api_paste_config = /etc/nova/api-paste.ini
# COMPUTE/APIS: if you have separate configs for separate services
# this flag is required for both nova-api and nova-compute
allow_resize_to_same_host = True
# APIS
osapi_compute_extension = nova.api.openstack.compute.contrib.standard_extensions
ec2_dmz_host = 172.16.0.254
s3_host = 172.16.0.254
# RABBITMQ
rabbit_host = 172.16.0.254
rabbit_password = guest
# GLANCE
image_service = nova.image.glance.GlanceImageService
glance_api_servers = 172.16.0.254:9292
# NETWORK
network_api_class = nova.network.quantumv2.api.API
quantum_url = http://172.16.0.254:9696
quantum_auth_strategy = keystone
quantum_admin_tenant_name = service
quantum_admin_username = quantum
quantum_admin_password = password
quantum_admin_auth_url = http://172.16.0.254:35357/v2.0
libvirt_vif_driver = nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSInterfaceDriver
firewall_driver = nova.virt.libvirt.firewall.IptablesFirewallDriver
# NOVNC CONSOLE
novncproxy_base_url = http://172.16.0.254:6080/vnc_auto.html
# Change vncserver_proxyclient_address and vncserver_listen to match each compute
host
vncserver_proxyclient_address = 172.16.0.254
vncserver_listen = 172.16.0.254
# AUTHENTICATION
auth_strategy = keystone
```

```
[keystone_authtoken]
auth_host = 172.16.0.254
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = password
signing_dir = /tmp/keystone-signing-nova
```

配置 api-paste.ini

修改 [filter:authtoken]:

```
# vim /etc/nova/api-paste.ini
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = 172.16.0.254
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = password
signing_dir = /tmp/keystone-signing-nova
```

启动服务

```
# for serv in api cert scheduler consoleauth novncproxy conductor compute;
do
  /etc/init.d/nova-$serv restart
done
```

查看服务

出现笑脸表示对应服务正常,如做状态是 XX 的话，注意查看/var/log/nova/下对应服务的 log:

```
# nova-manage service list 2> /dev/null
Binary           Host                                 Zone             Status
State Updated_At
nova-cert        localhost                            internal         enabled
```

```
      2013-03-11 02:56:21
nova-scheduler   localhost                            internal         enabled
```

```
      2013-03-11 02:56:22
nova-consoleauth localhost                            internal         enabled
```

```
      2013-03-11 02:56:22
nova-conductor   localhost                            internal         enabled
```

```
      2013-03-11 02:56:22
nova-compute     localhost                            nova             enabled
```

2013-03-11 02:56:23

组策略

给默认的租策略: default 添加 ping 响应和 ssh 端口:

```
# 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
```

Troubleshooting Nova

1. 配置文件指定的参数是否符合实际环境
2. /var/log/nova/中对应服务的 log
3. 依赖环境变量, 数据库连接，端口启动
4. 硬件是否支持虚拟化等

安装 Horizon

安装 OpenStack Dashboard、Apache 和 WSGI 模块:

```
# apt-get install -y memcached libapache2-mod-wsgi openstack-dashboard
```

配置 Dashboard，修改 Memcache 的监听地址:
去掉 ubuntu 的 主题:

```
# mv /etc/openstack-dashboard/ubuntu_theme.py /etc/openstack-
dashboard/ubuntu_theme.py.bak
# vim /etc/openstack-dashboard/local_settings.py
DEBUG = True
CACHE_BACKEND = 'memcached://172.16.0.254:11211/'
OPENSTACK_HOST = "172.16.0.254"
# sed -i 's/127.0.0.1/172.16.0.254/g' /etc/memcached.conf
```

启动 Memcached 和 Aapache:

```
# /etc/init.d/memcached restart
# /etc/init.d/apache2 restart
```

浏览器访问:

```
http://172.16.0.254/horizon
用户:    admin
密码: password
```

Troubleshooting Horizon

1. 出现无法登录的情况，注意查看 /var/log/apache2/error.log 和 /var/log/keystone/keystone.log
一般会出现 401 的错误，主要和配置文件有关系，quantum cinder nova 配置文件的 keystone
验证信息有误。

2. 登录出现 [Errno 111] Connection refused 错误时候，一般是 cinder-api 和 nova-api 没有启动，

配置 External 网络

介绍

External 就是外部网络，相当于 Float ip，External 网络走的是 br-ex，也就是物理 eth2 网卡，对于 External 网络我们只需要创建一个就够了，而所有的租户都用这一个 External 到外网。
我们用管理员创建一个 External 网络后，剩下的就交给每个租户自己来创建自己的网络了。
Quantum 里的名词理解：
Network：分为 External 和 Internal 两种网络, 也就是一个交换机。
Subnet：这个网络在哪个网段，它的网关和 dns 是多少
Router：一个路由器，可以用来隔离不同租户之间自己创建 的 Internal 网络.
Interface： 路由器上的 WLAN 和 LAN 口
Port：交换机上的端口，这个端口被谁使用，可以知道 IP 地址信息。

对于配置 Quantum 的网络来说，就是自己动手插网线、连路由器的一个过程。例如：比如一个公司是通过 ADSL 拨号上网，出口只有一个，公司内部是一个局域网（External 网络），然而这个公司有多个部门组成（多个租户），A 部门（租户）需要经常测试，IP 地址或 DHCP 服务器会和其他部门（其他租户）冲突，只能在找一个路由器（Router-1）来隔离 A 部门和其它部门的网络， A 部门的网络地址不能设置成和路由器（Router-1）的 WLAN 口在同一网络位，因为路由器的 WLAN 口 IP 和 LAN 口 IP 不能在同一网段，这时候就需要 A 部门自己定义一个私有网段到路由器的 LAN 口，（租户自己创建自己的 Network 、Subnet 以及 Router，并把 Interface 加到 Router 上，设置 Interface 的 WLAN 口 为 External ip， LAN 口为 Subnet 包含的地址）。 A 部门正常可以上外网（Port 通过 Router-1 的 Interface 到 External 上）。同理，现在多个部门都需要隔离网络，那就多个路由器来（Router-2,3,4,5…）隔离。

创建一个 External 网络

注意 router:external=True 参数，它指这是一个 External 网络

```
EXTERNAL_NET_ID=$(quantum net-create external_net1 --router:external=True | awk '/
id / {print $4}')
```

创建一个 Subnet

由于我的 Quantum 版本是 2.0, 而源码包已经更新到了 2.2 了，命令参数以后可能会有些小变化。我这里的 quantum 命令不能直接设置 dns 和 host route。下面这个 192.168.8.0/24 就是我外部网络的网段了，注意网关必须是你指定的这个网络范围里，比如你指定了 cidr 是 192.168.8.32/24，网关是 192.168.8.1, 而 8.1 不再 cidr 的范围里。
创建 Float IP 地址的 Subnet, 这个 Subnet 的 DHCP 服务被禁用：

```
SUBNET_ID=$(quantum subnet-create external_net1 192.168.8.0/24
--name=external_subnet1 --gateway_ip 192.168.8.1 --enable_dhcp=False | awk '/ id /
{print $4}')
```

创建一个 Internal 网络

这里为租户 demo 创建，需要 demo 的 id:

```
# DEMO_ID=$(keystone tenant-list | awk '/ demo / {print $2}')
```

为 demo 租户创建 Internal Network

demo 租户：我给你们部门规划创建了一套网络

```
# INTERNAL_NET_ID=$(quantum net-create demo_net1 --tenant_id $DEMO_ID | awk '/
id / {print $4}')
```

为 demo 租户创建 Subnet

demo 租户：我给你们定义了一个 网段 10.1.1.0/24 , 网关是 10.1.1.1，默认开启了 dhcp 功能

```
# DEMO_SUBNET_ID=$(quantum subnet-create demo_net1 10.1.1.0/24 --name=demo_subnet1
--gateway_ip 10.1.1.1 --tenant_id $DEMO_ID| awk '/ id / {print $4}')
```

为 demo 租户创建一个 Router

又给 demo 租户拿来了一个路由器:

```
# DEMO_ROUTER_ID=$(quantum router-create --tenant_id $DEMO_ID demo_router1 | awk
'/ id / {print $4}')
```

添加 Router 到 Subnet 上

刚才对 demo 说的话, 应用到刚才拿来的路由器上，这个路由器 LAN 口地址为: 10.1.1.1, 网段为 10.1.1.0/24:

```
# quantum router-interface-add  $DEMO_ROUTER_ID $DEMO_SUBNET_ID
```

给 Router 添加 External IP

在给这个路由器的 WLAN 口插上连接外网的网线，并从 External 网络里拿一个 IP 地址设置到 WLAN 口:

```
# quantum router-gateway-set $DEMO_ROUTER_ID $EXTERNAL_NET_ID
```

给 demo 租户创建一个虚拟机

给我们即将要启动的虚拟机创建一个 Port，指定虚拟机用那个 Subnet 和 Network，在指定一个固定的 IP 地址:

```
# quantum net-list
+--------------------------------------+--------------
+--------------------------------------+
| id                                   | name          | subnets
|
+--------------------------------------+--------------
```

```
+--------------------------------------+
| 18ed98d5-9125-4b71-8a37-2c9e3b07b99d | demo_net1     | 75896360-61bb-406e-8c7d-
ab53f0cd5b1b |
| 1d05130a-2b1c-4500-aa97-0857fcb3fa2b | external_net1 | 07ba5095-5fa0-4768-9bee-
7d44d2a493cf |
+--------------------------------------+--------------
+--------------------------------------+
# DEMO_PORT_ID=$(quantum port-create --tenant-id=$DEMO_ID --fixed-ip
subnet_id=$DEMO_SUBNET_ID,ip_address=10.1.1.11 demo_net1 | awk '/ id / {print
$4}')
```

用 demo 启动虚拟机：

```
# glance image-list
+--------------------------------------+--------+-------------+-----------------
+---------+--------+
| ID                                   | Name   | Disk Format | Container Format |
Size    | Status |
+--------------------------------------+--------+-------------+-----------------
+---------+--------+
| f61ee640-82a7-4d6c-8816-608bb91dab7d | cirros | qcow2       | ovf              |
9761280 | active |
+--------------------------------------+--------+-------------+-----------------
+---------+--------+
# nova  --os-tenant-name demo boot --image cirros --flavor 2 --nic port-
id=$DEMO_PORT_ID instance01
```

给 demo 租户的虚拟机添加 Float ip

虚拟机启动后，你发现你无法 ping 通 10.1.1.11, 有路由器在隔离你当然是无法 ping 通, 不过虚拟机可以出外网.（因为 quantum 版本问题，没有 DNS 参数选项，虚拟机的 DNS 有误，自己修改下虚拟机的 resolv.conf）, 如果想 ssh 到虚拟机的话，就加一个 Floating IP 吧：

查看 demo 租户的虚拟机的 id

```
# nova --os_tenant_name=demo list
+--------------------------------------+------------+--------
+---------------------+
| ID                                   | Name       | Status | Networks
|
+--------------------------------------+------------+--------
+---------------------+
| b0b7f0a1-c387-4853-a076-4b7ba2d32ed1 | instance01 | ACTIVE | demo_net1=10.1.1.11
|
+--------------------------------------+------------+--------
+---------------------+
```

获取虚拟机的 port id

```
# quantum port-list -- --device_id b0b7f0a1-c387-4853-a076-4b7ba2d32ed1
+--------------------------------------+------+-------------------
+-------------------------------------------------------------------------------
-+
| id                                   | name | mac_address       | fixed_ips
|
+--------------------------------------+------+-------------------
```

```
+--------------------------------------------------------------------------------
-+
| 95602209-8088-4327-a77b-1a23b51237c2 |      | fa:16:3e:9d:41:df | {"subnet_id":
"75896360-61bb-406e-8c7d-ab53f0cd5b1b", "ip_address": "10.1.1.11"} |
+--------------------------------------+------+-------------------
+--------------------------------------------------------------------------------
-+
```

创建一个 Float ip

注意收集 id:

```
# quantum  --os_tenant_name=demo floatingip-create external_net1
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    |                                      |
| floating_ip_address | 192.168.8.3                          |
| floating_network_id | 1d05130a-2b1c-4500-aa97-0857fcb3fa2b |
| id                  | f3670816-4d76-44e0-8831-5fe601f0cbe0 |
| port_id             |                                      |
| router_id           |                                      |
| tenant_id           | 83792f9193e1449bb90f78400974d533     |
+---------------------+--------------------------------------+
```

关联浮动 IP 到 VM

```
# quantum --os_tenant_name=demo floatingip-associate f3670816-4d76-44e0-8831-
5fe601f0cbe0 95602209-8088-4327-a77b-1a23b51237c2
Associated floatingip f3670816-4d76-44e0-8831-5fe601f0cbe0
```

查看刚才关联的浮动 IP

```
# quantum floatingip-show f3670816-4d76-44e0-8831-5fe601f0cbe0
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    | 10.1.1.11                            |
| floating_ip_address | 192.168.8.3                          |
| floating_network_id | 1d05130a-2b1c-4500-aa97-0857fcb3fa2b |
| id                  | f3670816-4d76-44e0-8831-5fe601f0cbe0 |
| port_id             | 95602209-8088-4327-a77b-1a23b51237c2 |
| router_id           | bf89066b-973d-416a-959a-1c2f9965e6d5 |
| tenant_id           | 83792f9193e1449bb90f78400974d533     |
+---------------------+--------------------------------------+
# ping 192.168.8.3
PING 192.168.8.3 (192.168.8.3) 56(84) bytes of data.
64 bytes from 192.168.8.3: icmp_req=1 ttl=63 time=32.0 ms
64 bytes from 192.168.8.3: icmp_req=2 ttl=63 time=0.340 ms
64 bytes from 192.168.8.3: icmp_req=3 ttl=63 time=0.335 ms
```

租户如何在界面上创建网络？

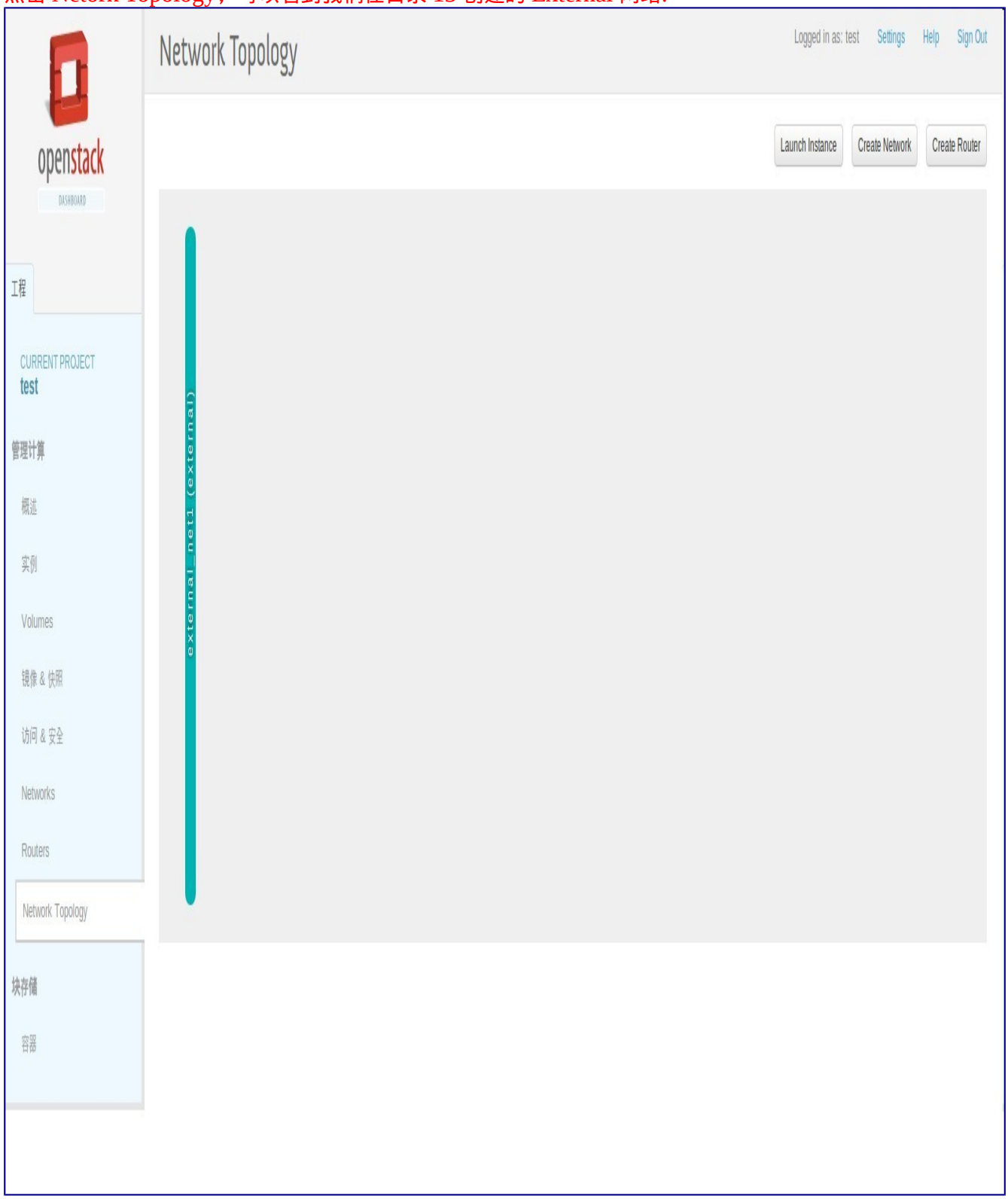
对于浏览器最好用 chrome, 而 firefox 有的按钮点击不了。

创建一个 test 租户，我这里用命令创建:

```
# TEST_TENANT_ID=$(keystone tenant-create --name test | awk '/ id / {print $4}')
# keystone user-create --name test --pass test --tenant-id $TEST_TENANT_ID
```

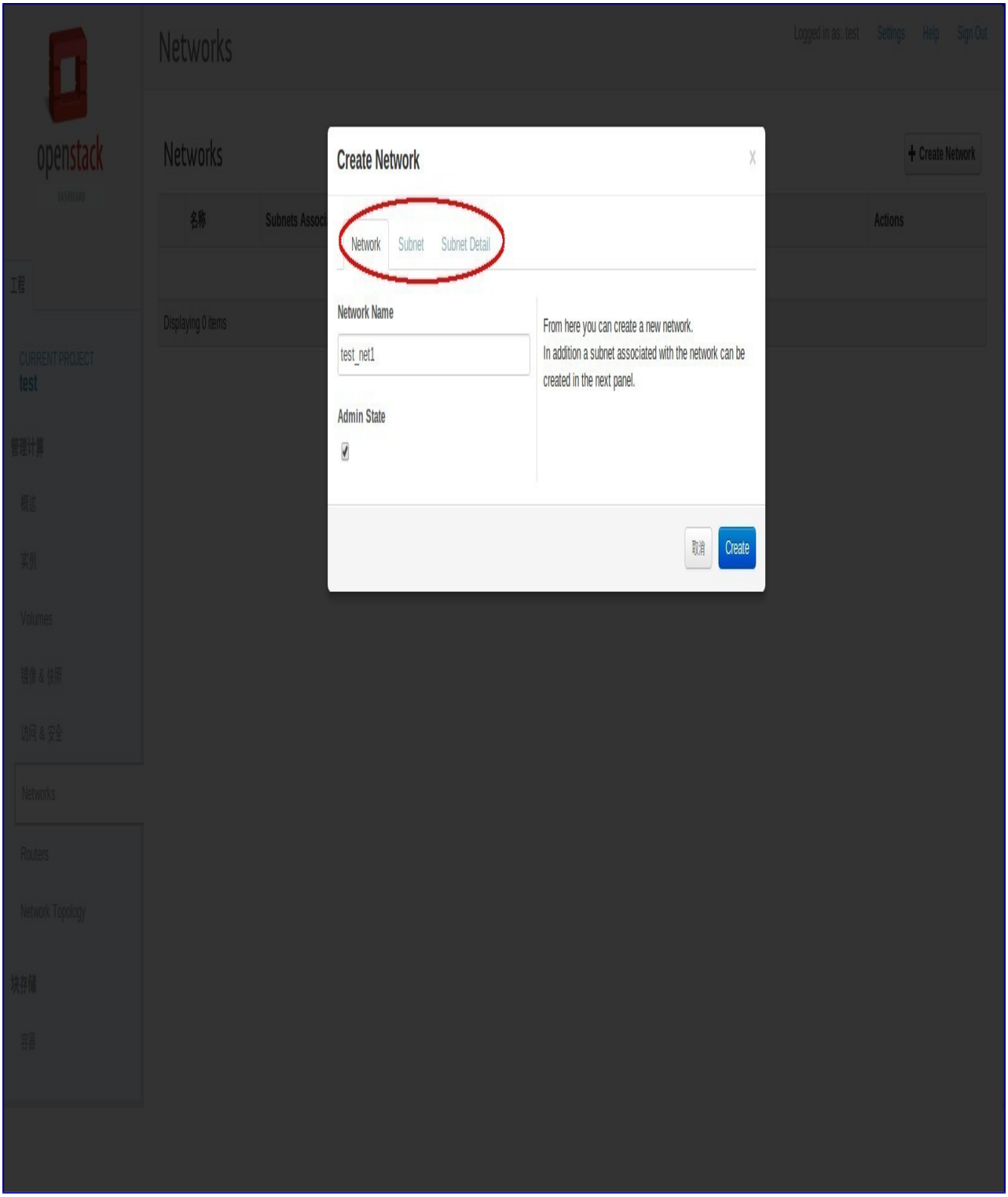
用 test 租户登录界面,并创建自己的网络：

点击 Netork Topology，可以看到我们在目录 13 创建的 External 网络:

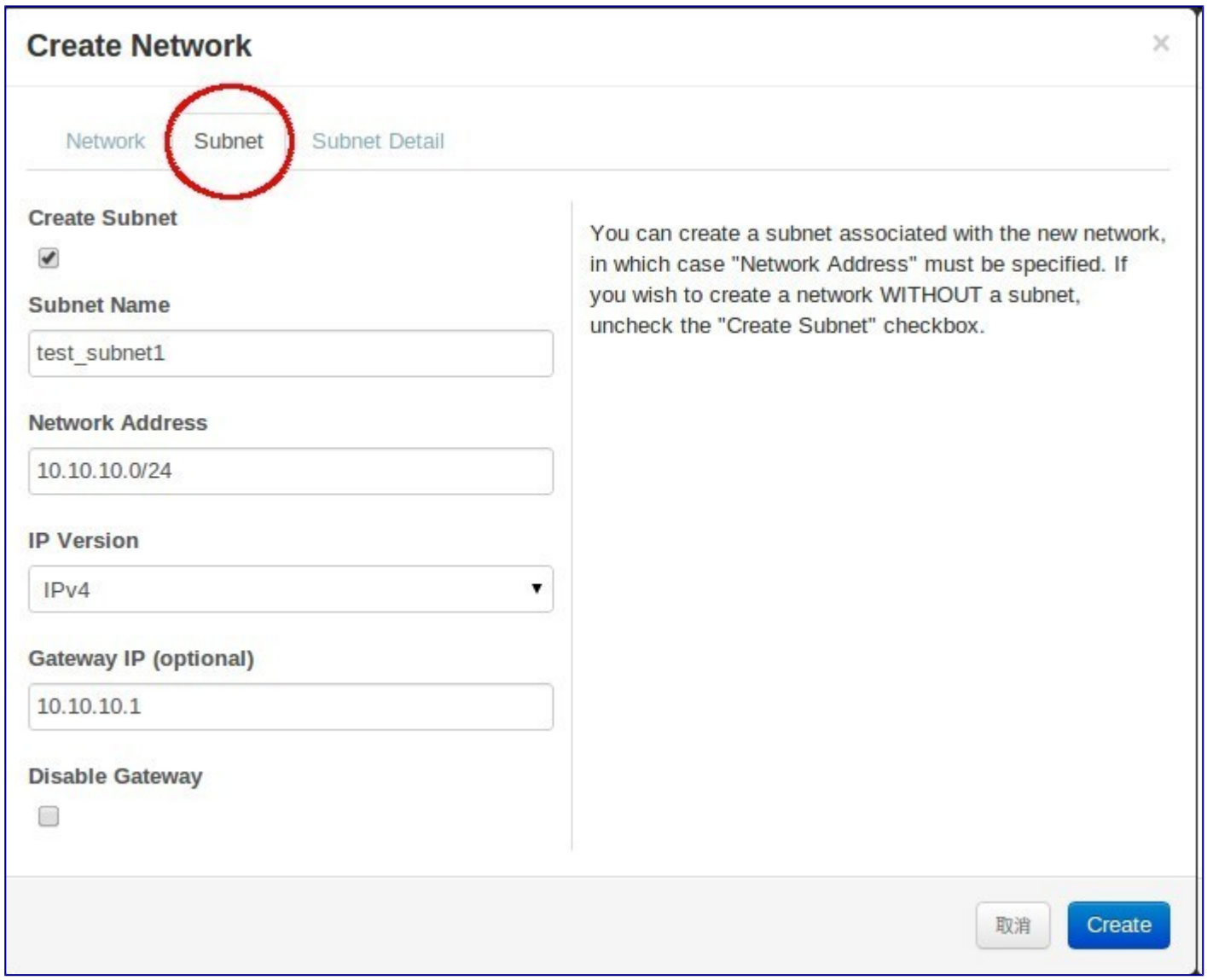


接下来界面的操作对应目录 14 的步骤

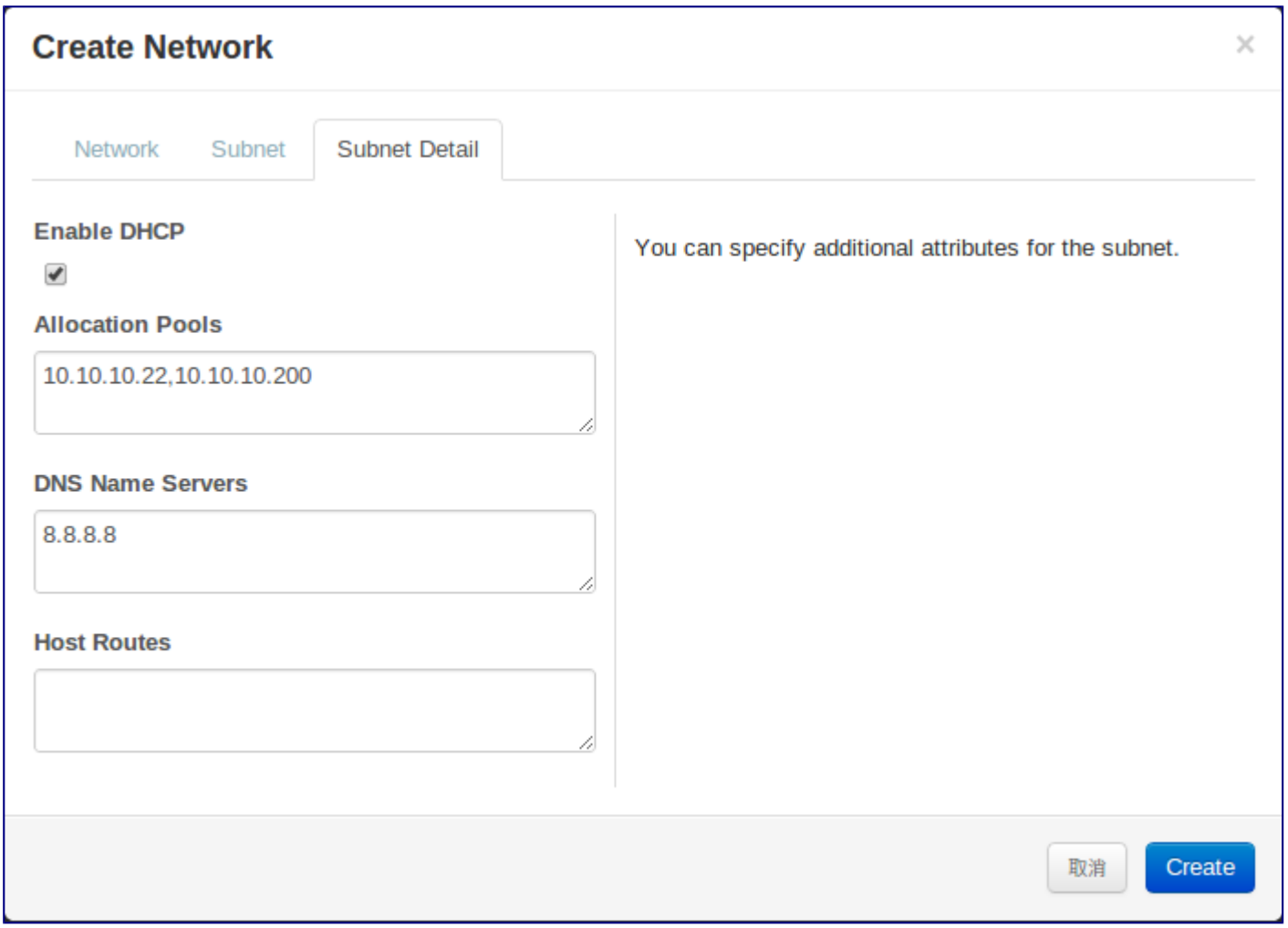
1. 选择 Networks 按钮，在点击 Create Network，输入网络名称：



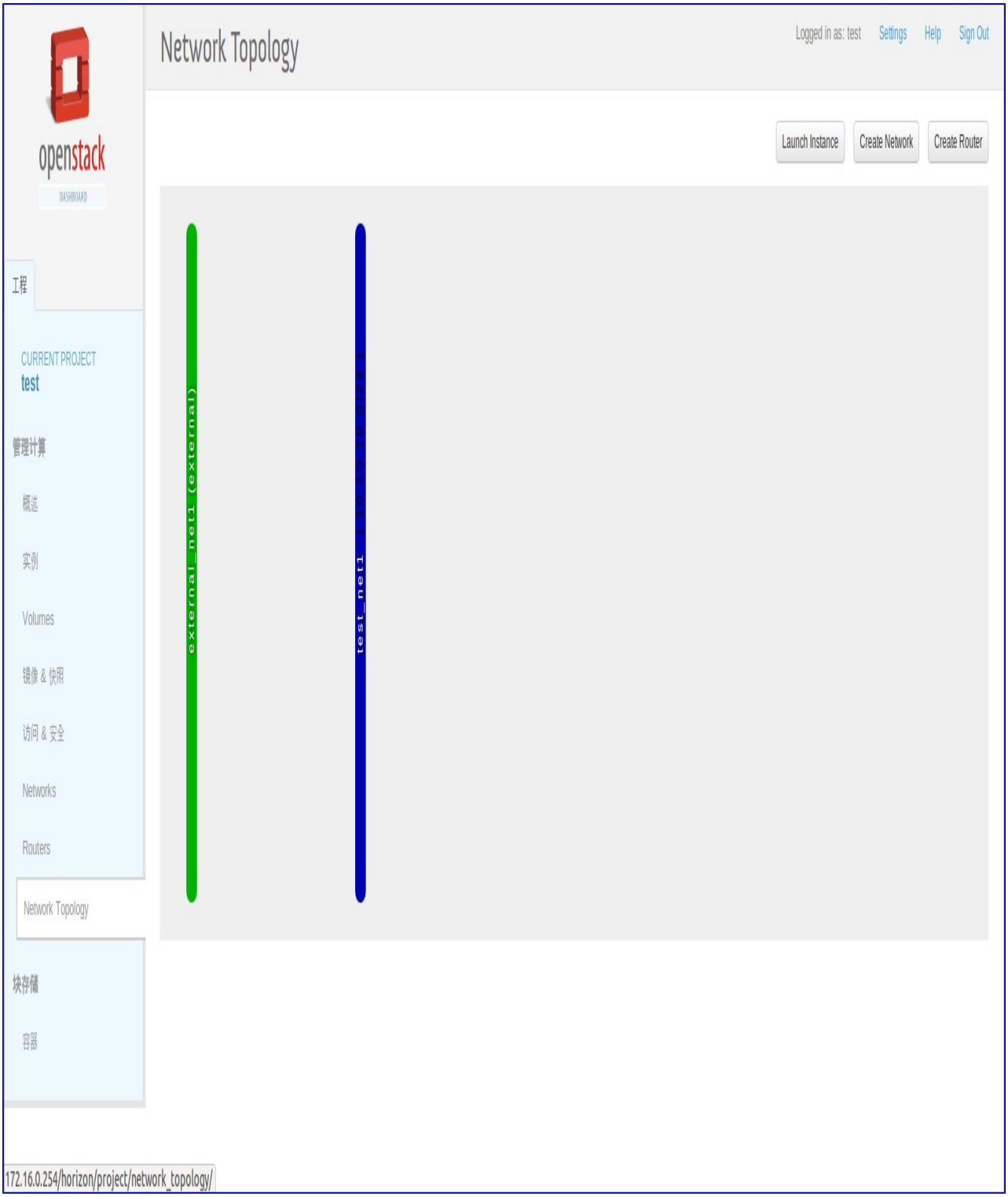
选择 Subnet，输入名称，网络地址和网关：



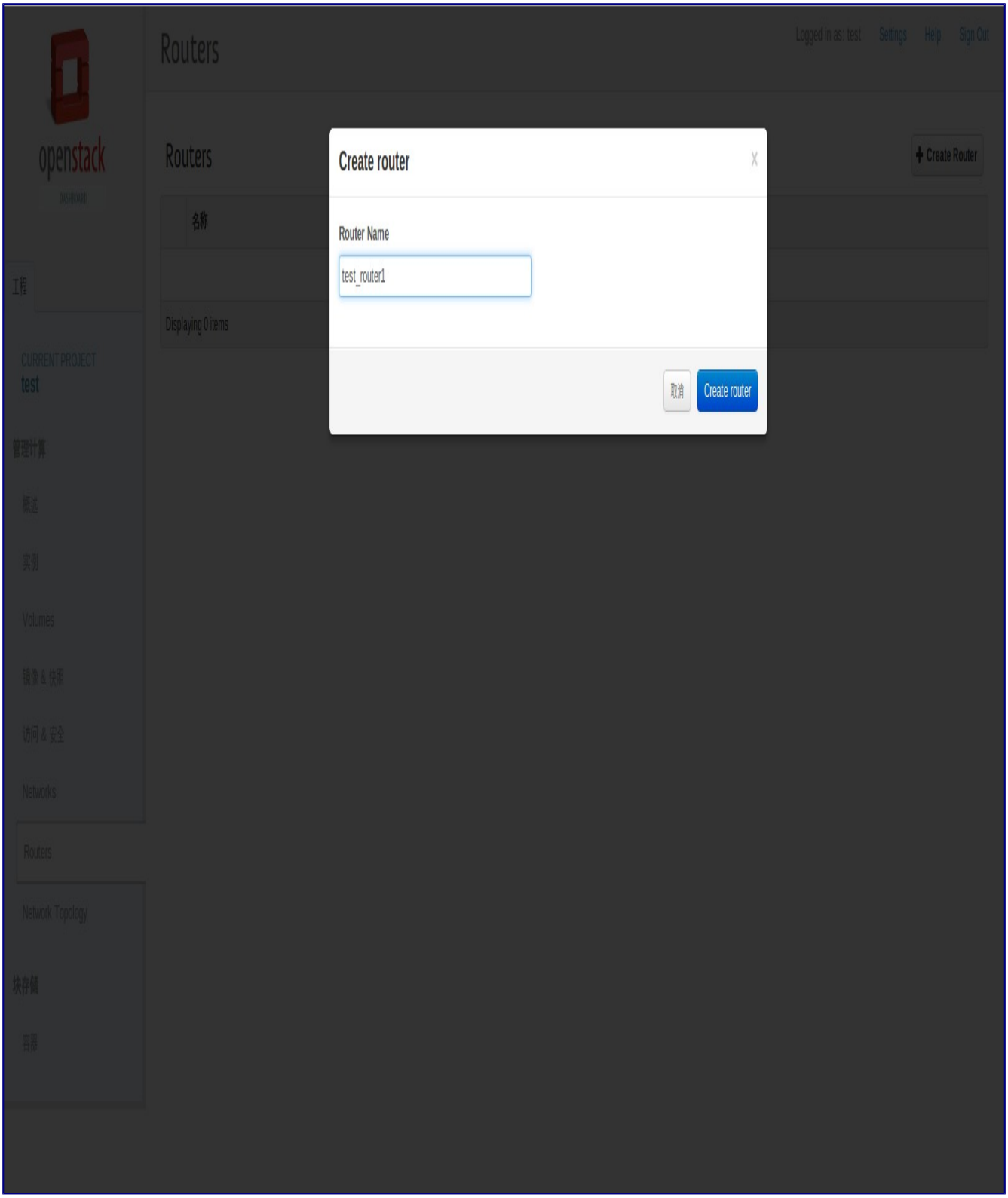
选择 Subnet Detail, 输入 dhcp 范围，输入 DNS 地址，也可以添加一个静态路由，静态路由可以到别的网络：



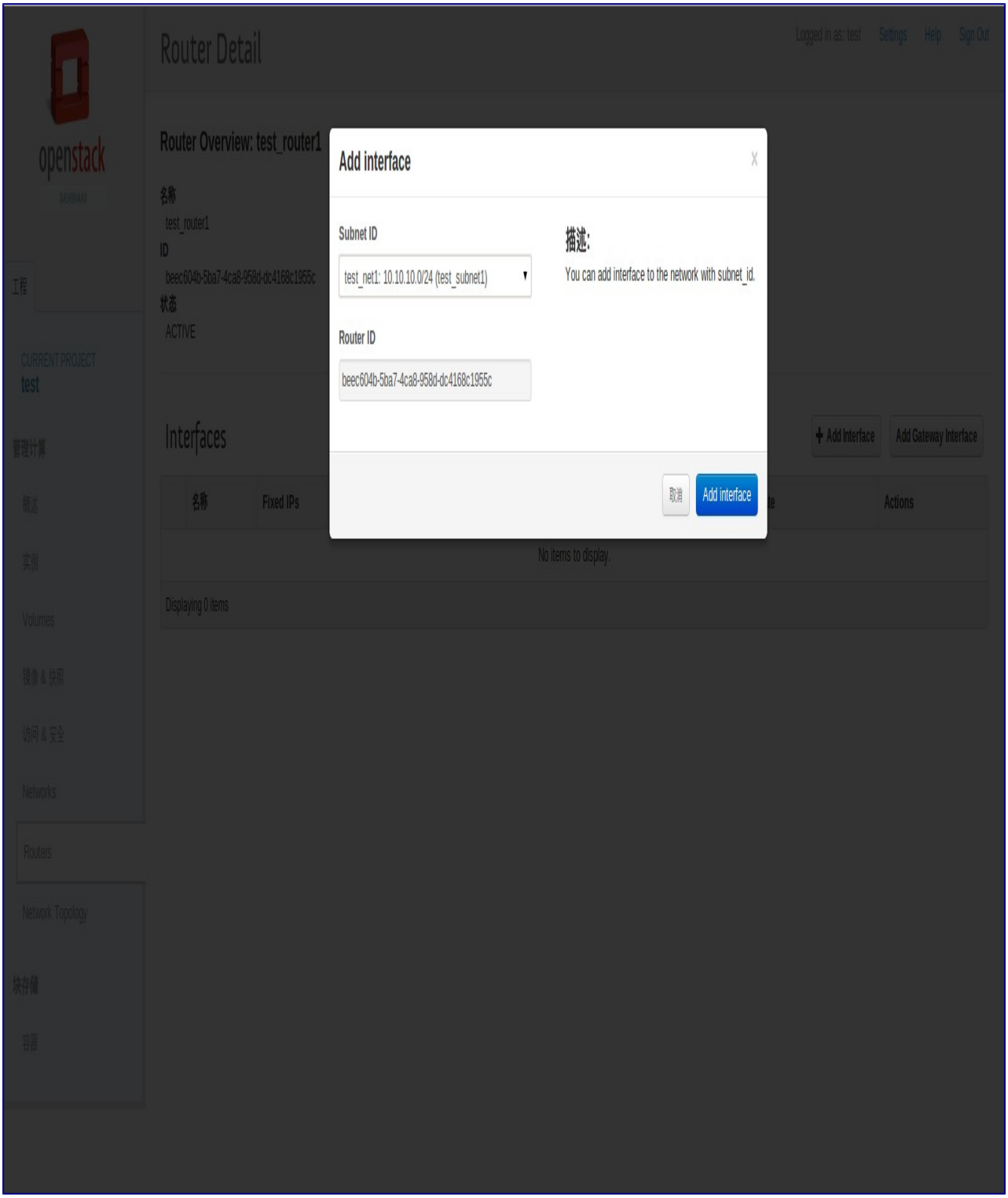
这时候就可以在 Network Topology 里看到刚才创建的网络了：



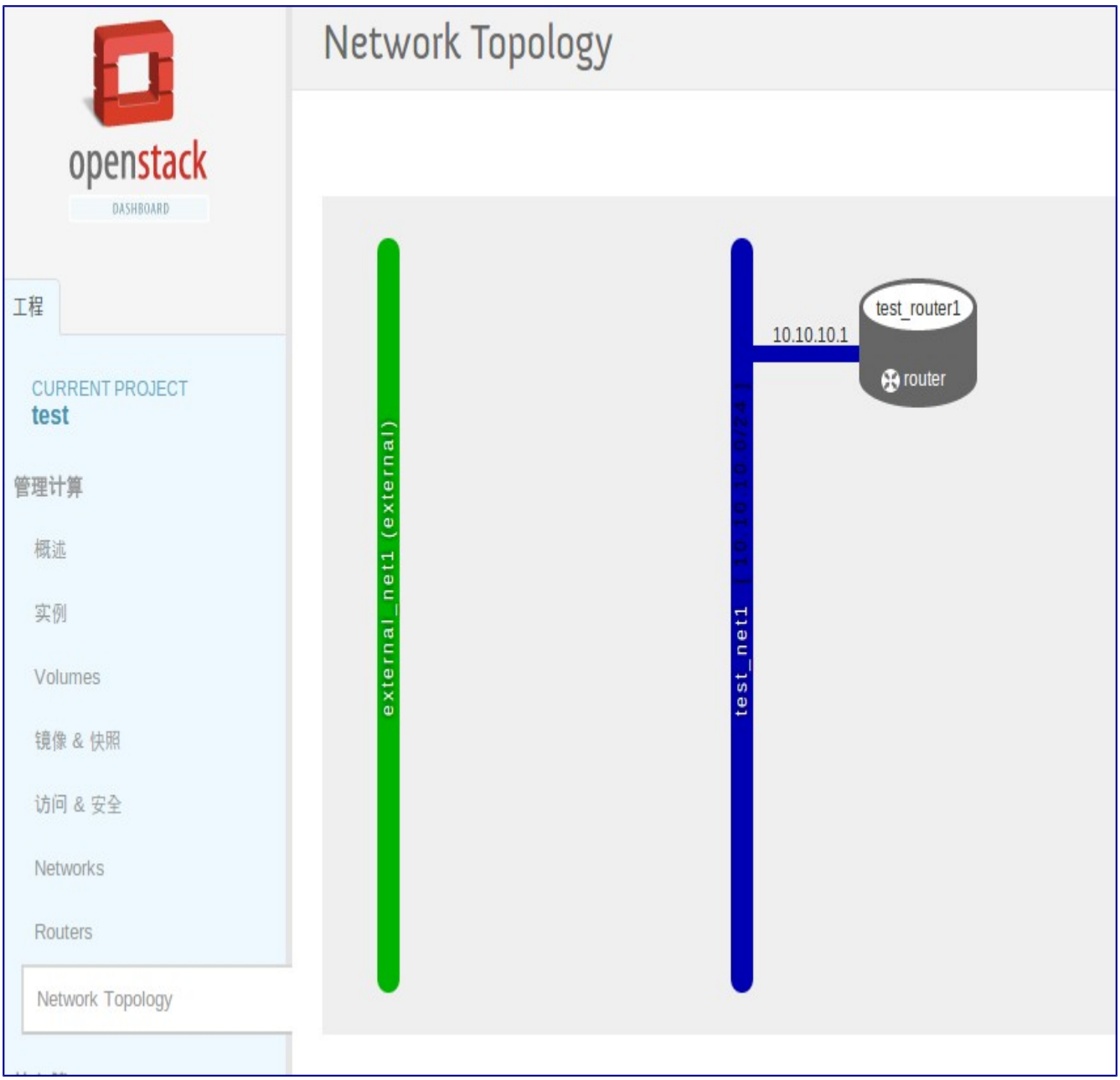
2. 选择 Routers，点击 Create Router, 输入名称：



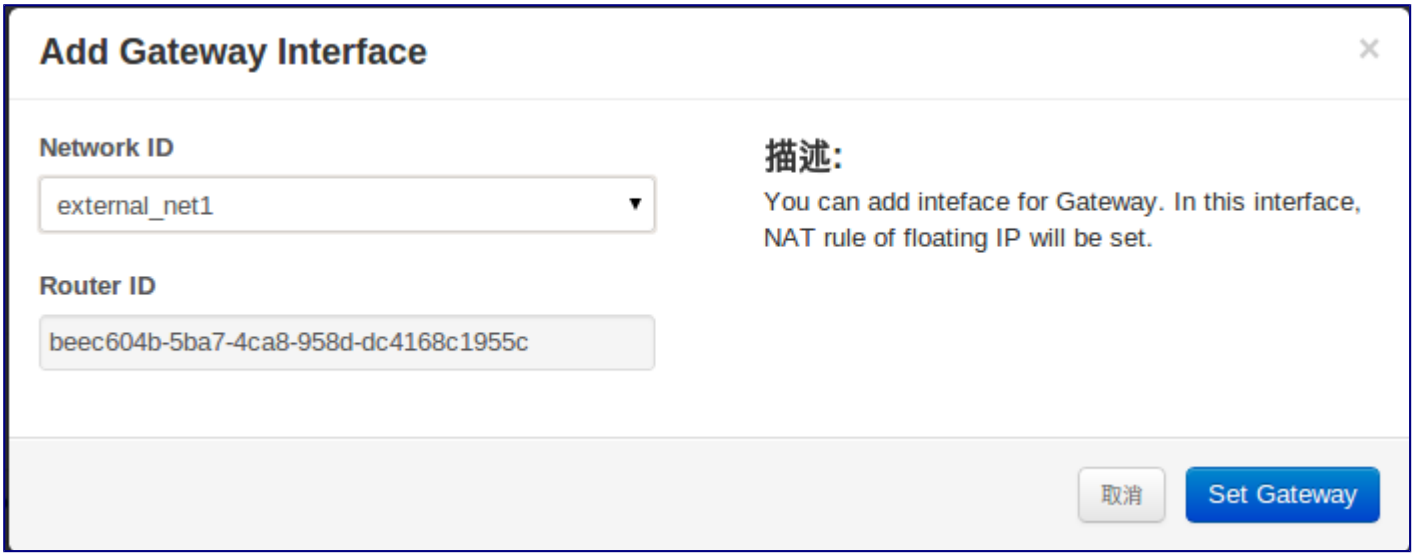
登录路由器，点击刚才创建的 test_router1 名字，进入到 Interface 界面，点击 Add Interface（LAN 口），选择刚才创建的网络 test_subnet：


Router Detail
Logged in as: test
Settings
Help
Sign Out
openstack
Router Overview: test_router1
名称
test_router1
ID
beec604b-5ba7-4ca8-958d-dc4168c1955c
状态
ACTIVE
Add interface
Subnet ID
test_net1: 10.10.10.0/24 (test_subnet1)
Router ID
beec604b-5ba7-4ca8-958d-dc4168c1955c
描述:
You can add interface to the network with subnet_id.
取消
Add interface
Interfaces
+ Add Interface
Add Gateway Interface
名称
Fixed IPs
Actions
No items to display.
Displaying 0 items
工程
CURRENT PROJECT
test
管理计算
概况
实例
Volumes
镜像 & 快照
访问 & 安全
Networks
Routers
Network Topology
块存储
容器

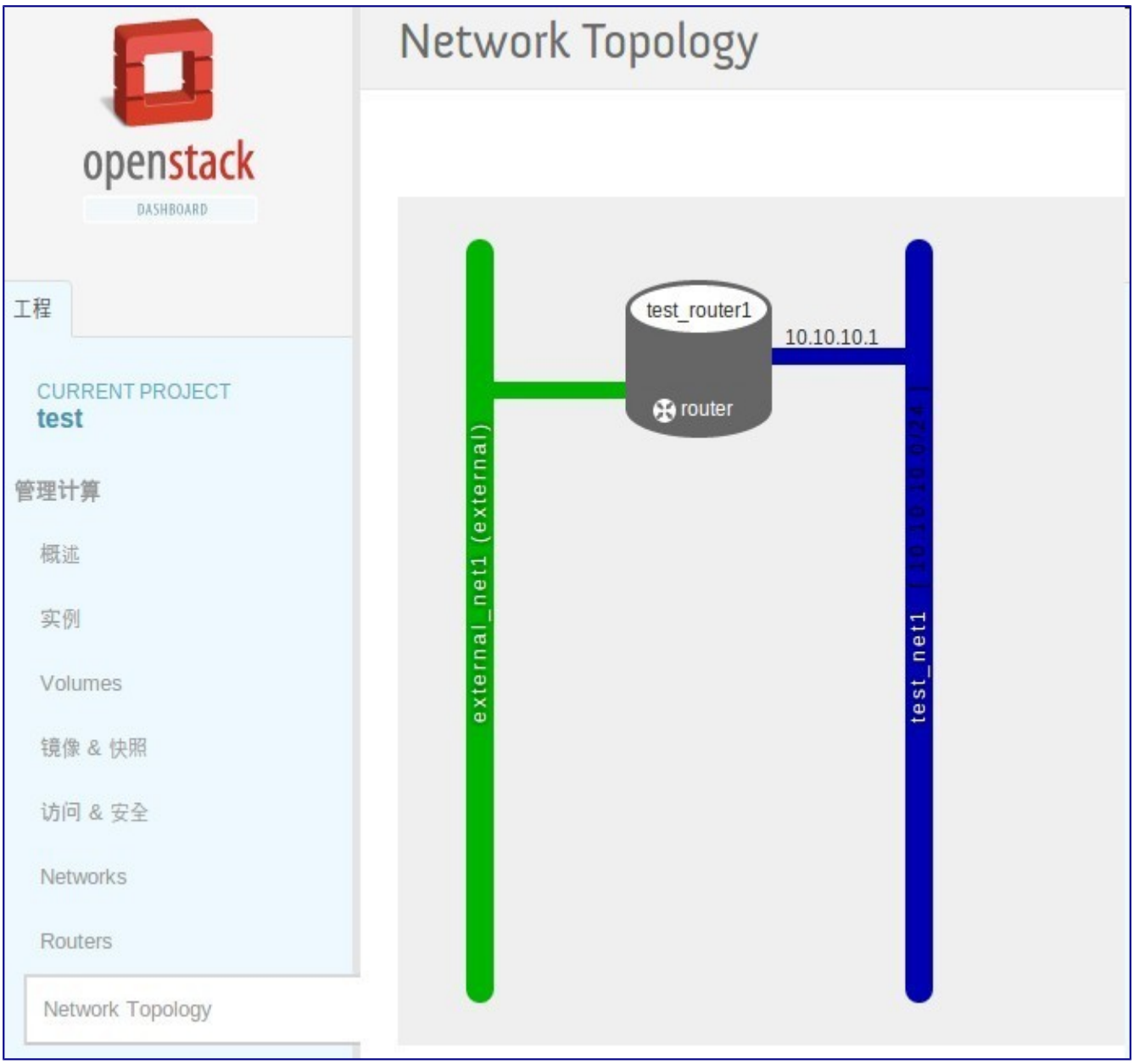
在来看看拓扑图：



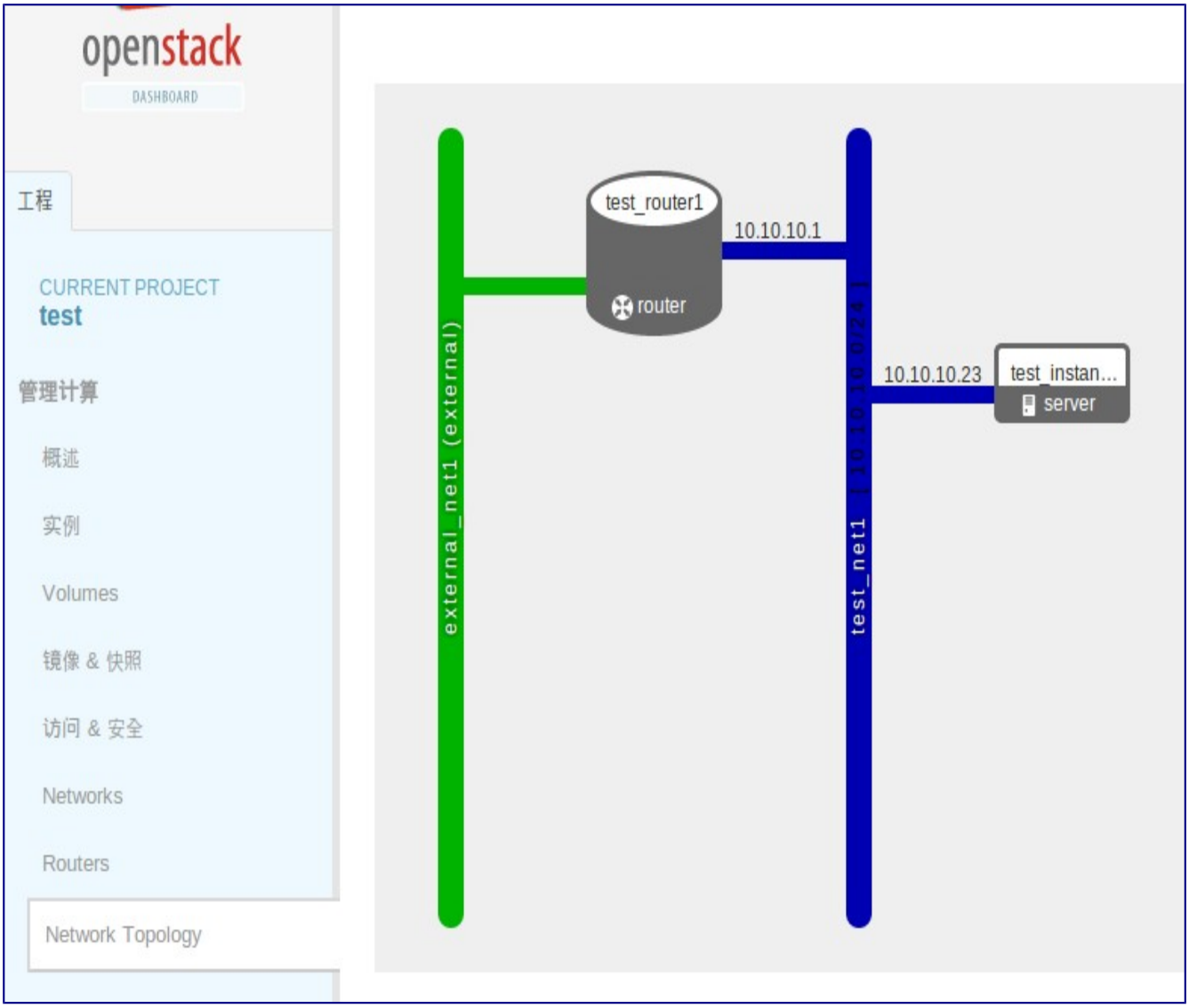
回到 Interface 界面, 在给这个路由器的 WLAN 口设置一个 IP， IP 地址从 External 网络拿一个, 选择 Add Gateway Interface：


Add Gateway Interface
Network ID
external_net1
Router ID
beec604b-5ba7-4ca8-958d-dc4168c1955c
描述:
You can add inteface for Gateway. In this interface, NAT rule of floating IP will be set.
取消
Set Gateway

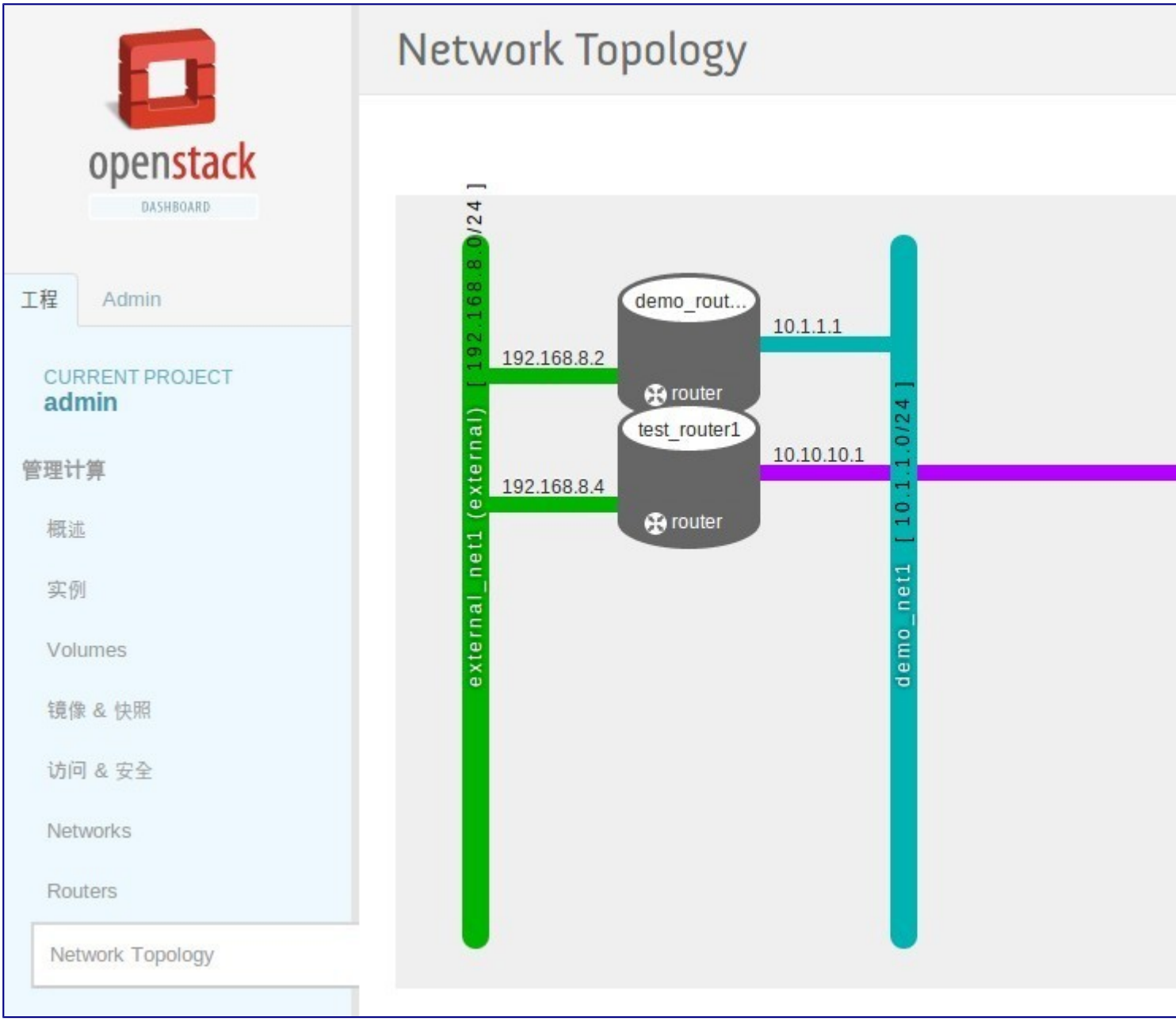
继续看图说话：



用 test 租户创建一个虚拟机后的网络拓扑图：



用 admin 管理员用户登录查看网络拓扑图, 可以看到 External 网络、demo 和 test 租户的网络：



其实 Quantum 的网络一点都不复杂，只要对应结合到实际生活中就会很好理解.

参考资料

http://www.longgeek.com/2012/07/30/rhel-6-2-openstack-essex-install-only-one-node/
http://www.chenshake.com/openstack-folsom-guide-for-ubuntu-12-04/#i-21
http://liangbo.me/index.php/2012/10/07/openstack-folsom-quantum-openvswitch/
http://www.ibm.com/developerworks/cn/cloud/library/1209_zhanghua_openstacknetwork/
http://docs.openstack.org/folsom/openstack-network/admin/content/index.html
http://docs.openstack.org/trunk/openstack-network/admin/content/index.html
http://docs.openstack.org/trunk/openstack-compute/install/apt/content/index.html