

本节按照以下步骤部署 DevStack 实验环境，包括控制节点和计算节点

**创建虚拟机**

按照物理资源需求创建 devstack-controller 和 devstak-compute 虚拟机

**安装操作系统**

安装 Ubuntu 14.04，并配置 eth0 的 IP devstack-controller  192.168.104.10 devstak-compute    192.168.104.11

**下载代码**

下载 devstack 代码，并切换到 stable/liberty 分支

apt-get install git -y   
git clone https://git.openstack.org/openstack-dev/devstack -b stable/liberty

**配置 stack 用户**

创建 stack 用户

devstack/tools/create-stack-user.sh

为方便起见，将devstack目录放到/opt/stack下，并设置权限

mv devstack /opt/stack   
chown -R stack:stack /opt/stack/devstack

切换到 stack 用户

su – stack   
cd devstack

**编写运行配置文件**

在 /opt/stack/devstack 目录下，创建 local.conf。

**devstack-controller**

[[local|localrc]]

MULTI\_HOST=true  
HOST\_IP=192.168.104.10 # management & api network   
LOGFILE=/opt/stack/logs/stack.sh.log

# Credentials   
ADMIN\_PASSWORD=admin   
MYSQL\_PASSWORD=secret   
RABBIT\_PASSWORD=secret   
SERVICE\_PASSWORD=secret   
SERVICE\_TOKEN=abcdefghijklmnopqrstuvwxyz

# enable neutron-ml2-vlan   
disable\_service n-net   
enable\_service q-svc,q-agt,q-dhcp,q-l3,q-meta,neutron,q-lbaas,q-fwaas,q-vpn   
Q\_AGENT=linuxbridge   
ENABLE\_TENANT\_VLANS=True   
TENANT\_VLAN\_RANGE=3001:4000   
PHYSICAL\_NETWORK=default

LOG\_COLOR=False   
LOGDIR=$DEST/logs   
SCREEN\_LOGDIR=$LOGDIR/screen

**devstack-compute**

[[local|localrc]]

MULTI\_HOST=true   
HOST\_IP=192.168.104.11 # management & api network

# Credentials   
ADMIN\_PASSWORD=admin   
MYSQL\_PASSWORD=secret   
RABBIT\_PASSWORD=secret   
SERVICE\_PASSWORD=secret   
SERVICE\_TOKEN=abcdefghijklmnopqrstuvwxyz

# Service information   
SERVICE\_HOST=192.168.104.10   
MYSQL\_HOST=$SERVICE\_HOST   
RABBIT\_HOST=$SERVICE\_HOST   
GLANCE\_HOSTPORT=$SERVICE\_HOST:9292   
Q\_HOST=$SERVICE\_HOST   
KEYSTONE\_AUTH\_HOST=$SERVICE\_HOST   
KEYSTONE\_SERVICE\_HOST=$SERVICE\_HOST

CEILOMETER\_BACKEND=mongodb   
DATABASE\_TYPE=mysql

ENABLED\_SERVICES=n-cpu,q-agt,neutron   
Q\_AGENT=linuxbridge   
ENABLE\_TENANT\_VLANS=True   
TENANT\_VLAN\_RANGE=3001:4000   
PHYSICAL\_NETWORK=default

# vnc config   
NOVA\_VNC\_ENABLED=True   
NOVNCPROXY\_URL="http://$SERVICE\_HOST:6080/vnc\_auto.html"   
VNCSERVER\_LISTEN=$HOST\_IP   
VNCSERVER\_PROXYCLIENT\_ADDRESS=$VNCSERVER\_LISTEN

LOG\_COLOR=False   
LOGDIR=$DEST/logs   
SCREEN\_LOGDIR=$LOGDIR/screen

另外，为了加快安装速度，还可以加上下面的配置使用国内的 devstack 镜像站点

# use TryStack git mirror   
GIT\_BASE=http://git.trystack.cn   
NOVNC\_REPO=http://git.trystack.cn/kanaka/noVNC.git   
SPICE\_REPO=http://git.trystack.cn/git/spice/spice-html5.git

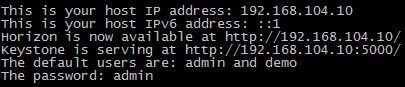
**开始部署**

分别在 devstack-controller 和 devstak-compute 上执行命令

./stack.sh

会输出各项操作的结果。 日志会写到 stack.sh.log 文件。   
整个过程需要连接Internet，网速慢可能会花较长时间，成功后最后会打印出相关信息。

devstack-controlle 上的输出

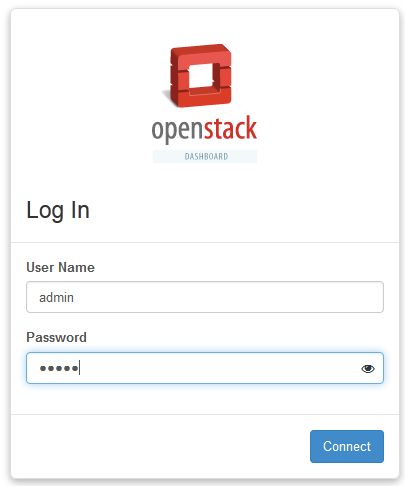


devstack-compute 上的输出

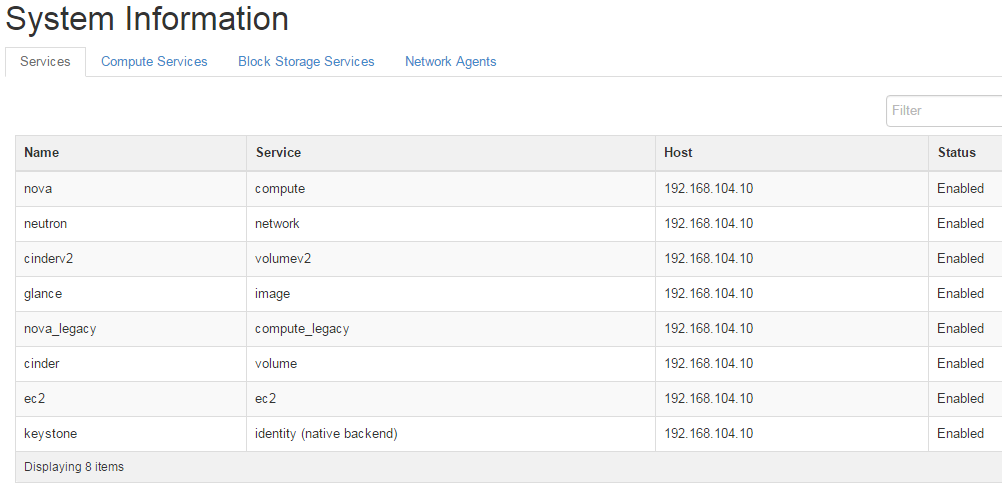
http://7xo6kd.com1.z0.glb.clouddn.com/upload-ueditor-image-20160405-1459867476764031618.jpg

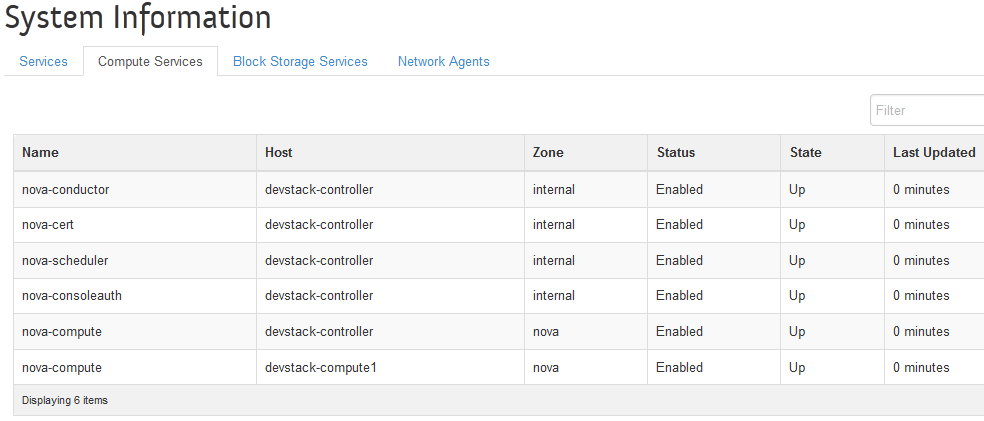
**验证 OpenStack**

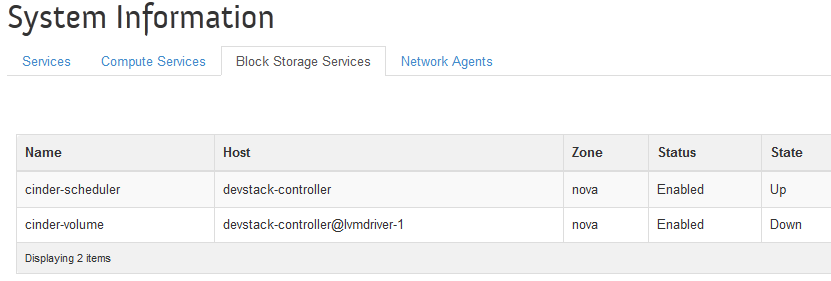
下面验证 OpenStack 已经正常运行 浏览器访问 http://192.168.104.10/（devstack-controller 的 IP） admin/admin 登录



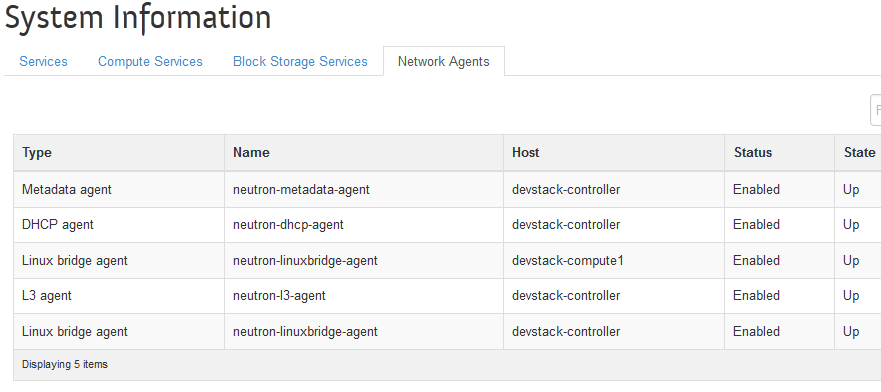
点击 System->System Information，确保各个服务的状态正常







这里 cinder-volume 的状态是 “Down”，这是因为我们还没有配置 cinder，后面会让它 Up 起来。



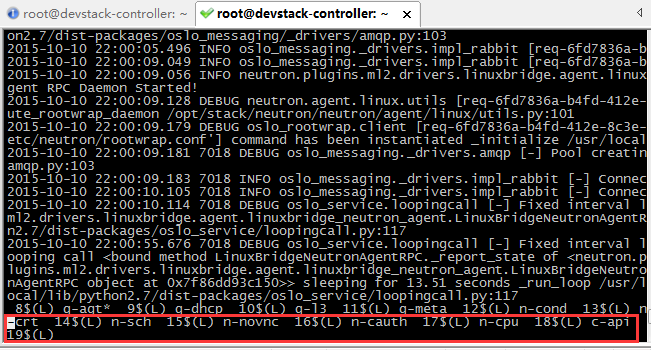
**启动 OpenStack**

如果重启了系统，OpenStack 不会自动启动，可以运行下面命令

cd devstack ./rejoin-stack.sh

运行成功后，OpenStack 的每个服务都在一个 screen 中以进程方式运行。 使用 screen 有几个好处：

1. 可以方便的在不同服务之间切换和查看日志。 因为OpenStack的服务很多，每个服务都有自己的日志文件。 查找日志是一件非常麻烦的事情，screen可以帮我们提高效率。
2. 当我们修改了某个服务的配置文件需要重启服务时，只需在该服务的 screen 窗口 Ctrl+C，然后在命令行中找到上一个命令（Up键）执行就行，这个命令就是启动服务的命令。



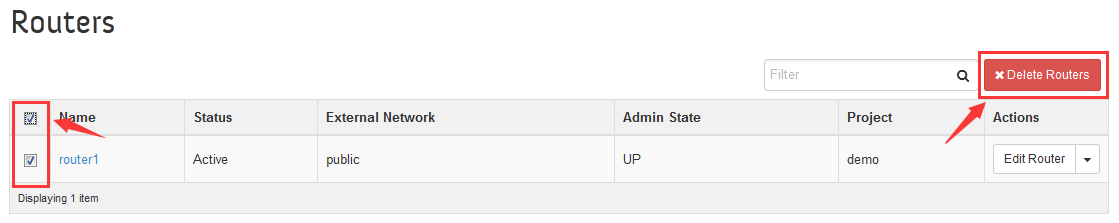
下面是 screen 常用命令   
在 screen 中执行   
Ctrl+a+n 切换到下一个窗口   
Ctrl+a+p 切换到前一个窗口(与 Ctrl+a+n 相对)   
Ctrl+a+0..9 切换到窗口 0..9   
Ctrl+a+d 暂时断开（detach）当前 screen 会话，但不中断 screen 窗口中程序的运行

在 shell 中执行   
screen -ls  列出当前所有的 session   
screen -r stack  回到 devstack 这个 session

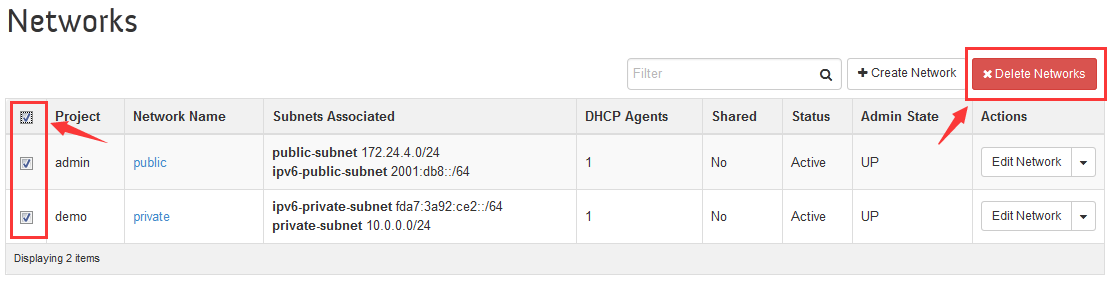
**删除自动创建的网络**

Devstack 在部署时可能会创建几个测试网络，为了得到一个干净的环境需将其删除

删除 Router Admin->System->Routers



删除 Network Admin->System->Networks



至此，我们得到了一个干净的 OpenStack 环境。 后面会在这个环境中实践各个模块。