## .8 Horizon(Dashboard)控制面板（仪表板）服务

### 8.1 Dashboard？

Dashboard(horizon)是一个web接口，使得云平台管理员以及用户可以管理不同的Openstack资源以及服务。

### **8.2 Dashboard（Horizon）服务安装和配置**

web服务器：Apache httpd。

安装节点：控制节点。

特点：无需配置数据库。

1. 安装软件包



1. 修改配置文件

备份：



a.在 controller 节点上配置仪表盘以使用 OpenStack 服务：

OPENSTACK\_HOST = "controller"

b.允许所有主机访问仪表板：

ALLOWED\_HOSTS = ['\*', ]

c.配置 memcached 会话存储服务：

SESSION\_ENGINE = 'django.contrib.sessions.backends.cache'

CACHES = {

'default': {

'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',

'LOCATION': 'controller:11211',

}

}

d.启用第3版认证API:

OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v3" % OPENSTACK\_HOST

e.启用对域的支持

OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True

说明：值为“True”，则用户在登录是需要选择域，如果为“False”，则默认登录为“default”域。

f.配置API版本:

OPENSTACK\_API\_VERSIONS = {

"identity": 3,

"image": 2,

"volume": 2,

}

g.通过仪表盘创建用户时的默认域配置为 default :

OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = "default"

h.通过仪表盘创建的用户默认角色配置为 user ：

OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "user"

i.禁用支持3层网络服务：

OPENSTACK\_NEUTRON\_NETWORK = {

'enable\_router': False,

'enable\_quotas': False,

'enable\_distributed\_router': False,

'enable\_ha\_router': False,

'enable\_lb': False,

'enable\_firewall': False,

'enable\_vpn': False,

'enable\_fip\_topology\_check': False,

}

j.可以选择性地配置时区：

TIME\_ZONE = "Asia/Shanghai"

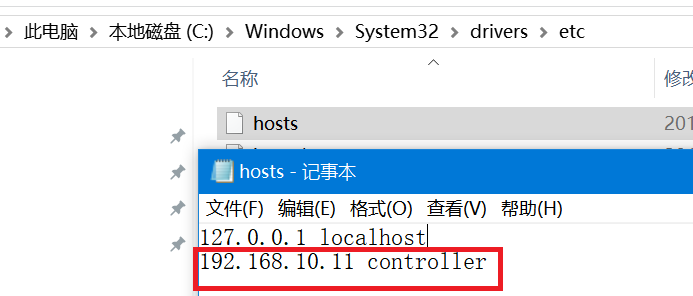
1. 完成安装

重启web服务器以及会话存储服务：

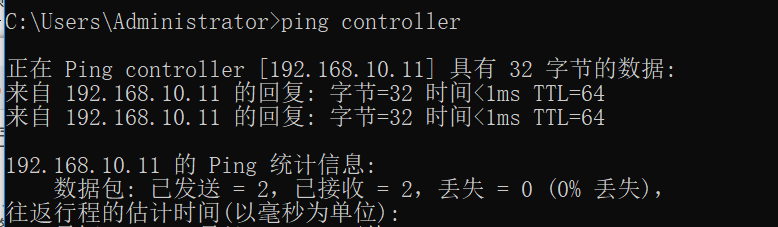
systemctl restart httpd.service memcached.service

1. 验证

a.在windows主机中配置域名映射。



映射验证：



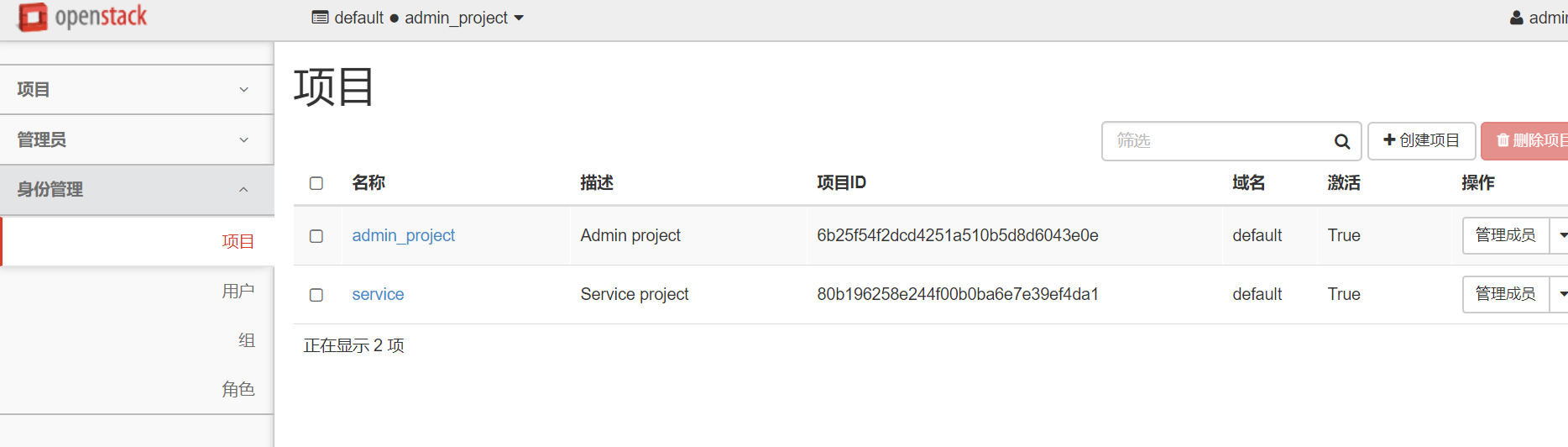
以上说明可通过hostname访问。

b.在浏览器中输入 http://controller/dashboard访问仪表盘。

验证使用 admin 或者``demo``用户凭证和``default``域凭证。



点击“连接”，可以看到：



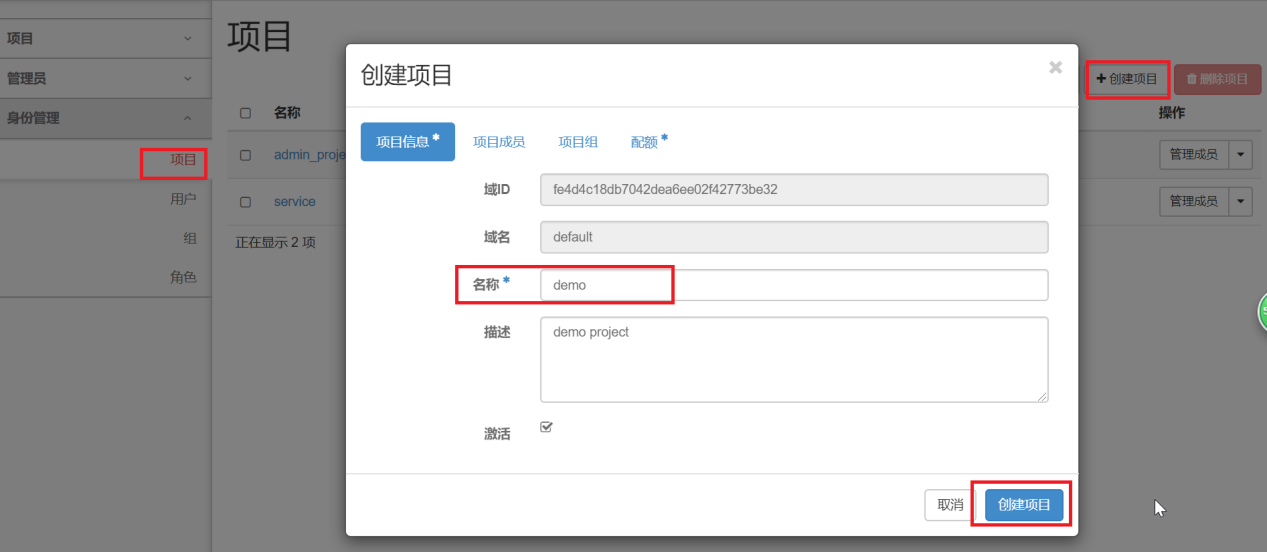
### **8.3 web方式创建一个普通用户并登录**

用户：demo

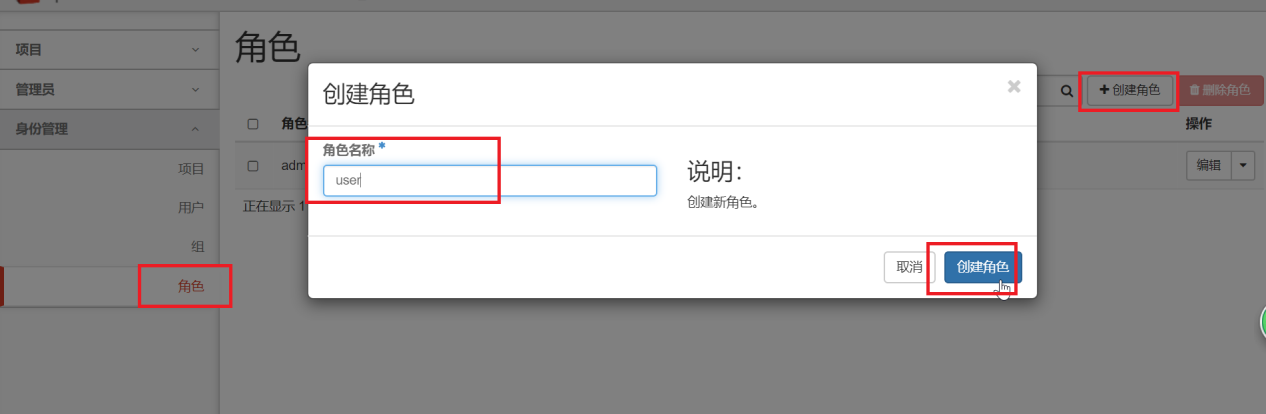
用户所属项目：demo

用户角色：user

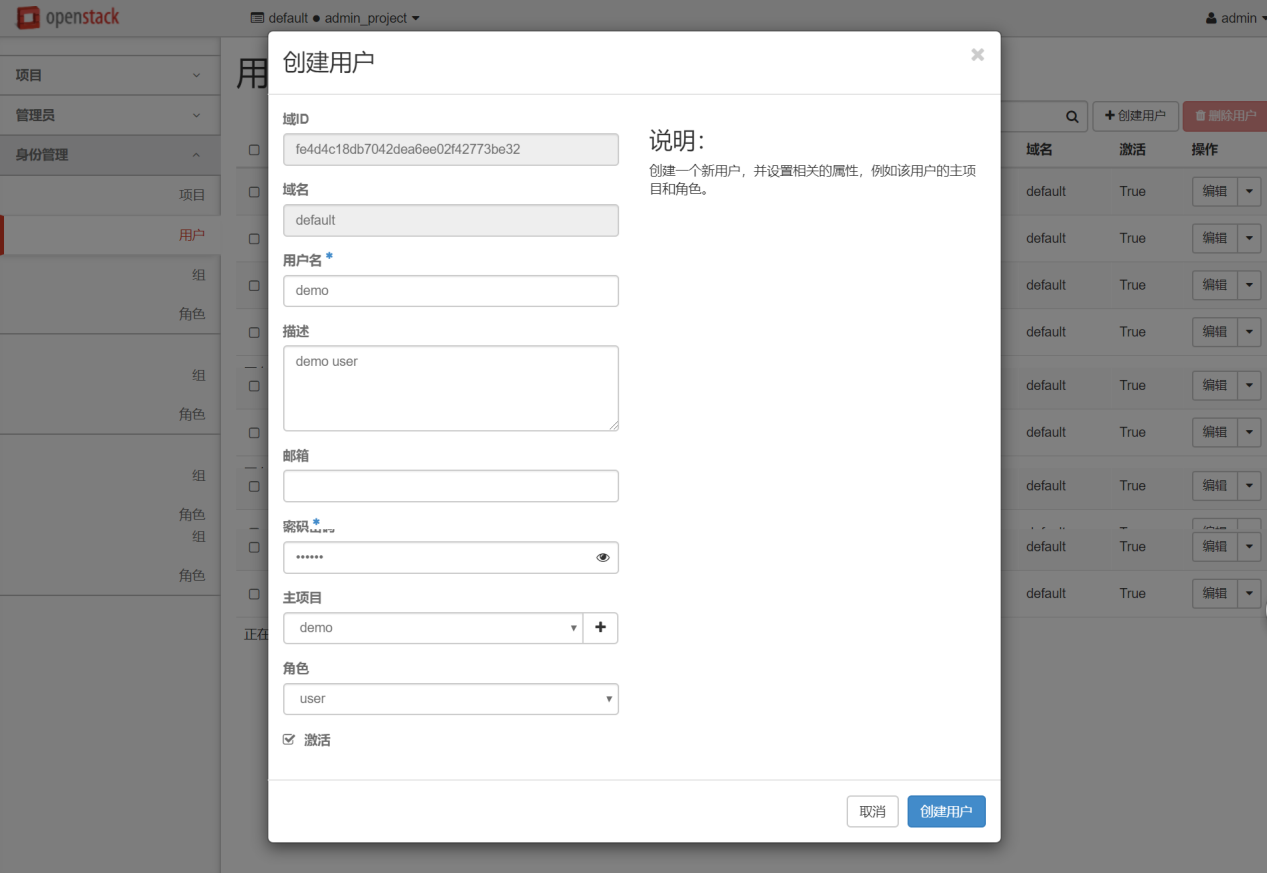
1. 创建项目



1. 创建角色



1. 创建用户



1. 用户登录

在浏览器选项中，选择打开新的无痕窗口，在界面中输入demo的相关信息并登录。



### **8.4 启动一个实例（云主机）**

两种方式：CLI方式和web方式。

#### 8.4.1 CLI方式启动一个云主机

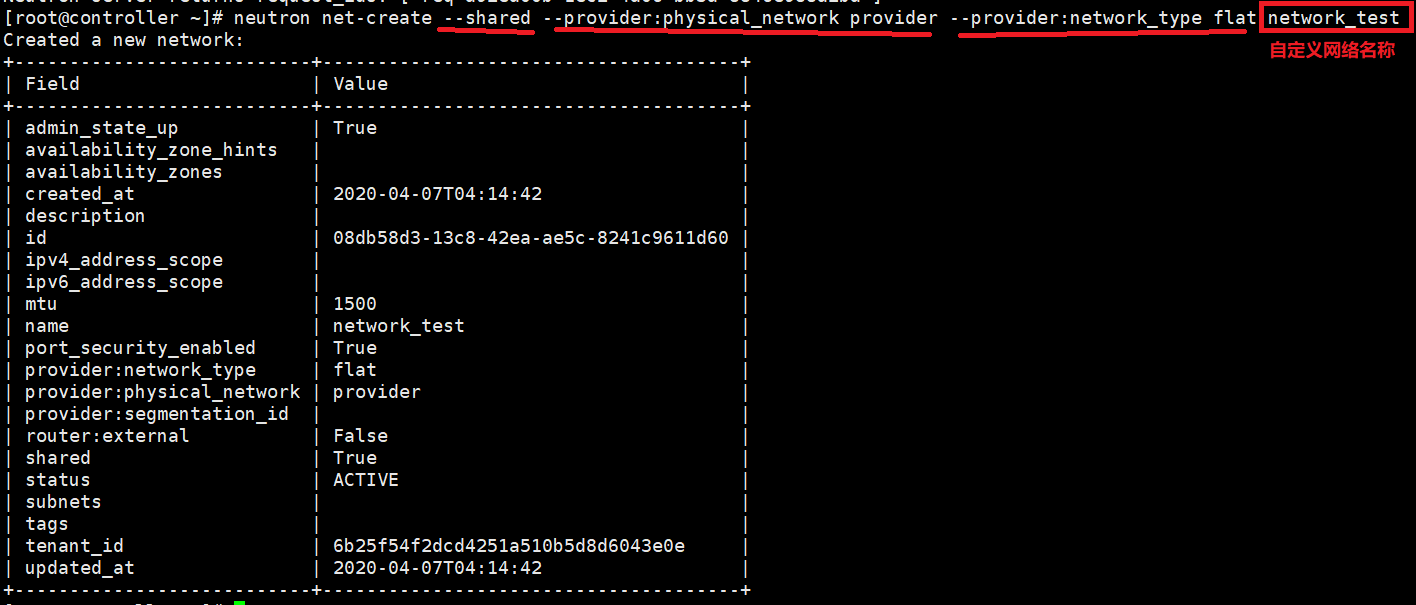
##### 创建虚拟网络

基于第7章，neutron中创建的是provider 网络，则本次需要创建一个公有网络。实例使用该网络，并通过L2（桥/交换机）设备连接到物理网络。这个网络包括为实例提供IP地址的DHCP服务器。

1. 在控制节点上，加载 admin 凭证来获取管理员能执行的命令访问权限.

source /root/admin-openrc

1. 创建网络



参数说明： --shared 表示当前创建的网络可以共享

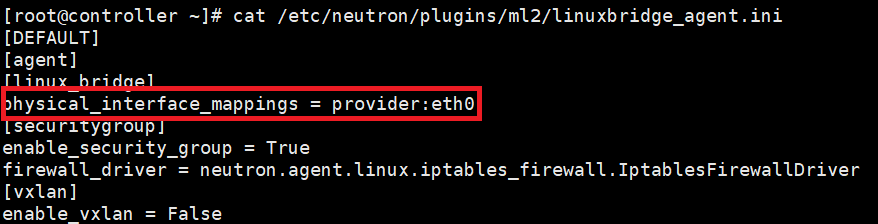
--provider:physical\_network provider表示物理网络名称，要和控制节点中

/etc/neutron/plugins/ml2/ml2\_con.ini中配置的flat\_network=provider参数一致

--provider:network\_type flat 指网络类型也与上面的配置文件一致

network\_test表示创建的网络名称

在linuxbridge\_agent.ini中，定义了flat虚拟网络使用eth0。



1. 在网络上创建子网，给创建的network\_test网络创建子网，并分配IP，网关等参数.

语法：

neutron subnet-create --name subnet1 \

--allocation-pool start=START\_IP\_ADDRESS,end=END\_IP\_ADDRESS \

--dns-nameserver DNS\_RESOLVER --gateway SUBNET1\_GATEWAY \

network1 SUBNET1\_NETWORK\_CIDR

neutron subnet-create --name subnet-hgb07 \

--allocation-pool start=192.168.10.20,end=192.168.10.100 \

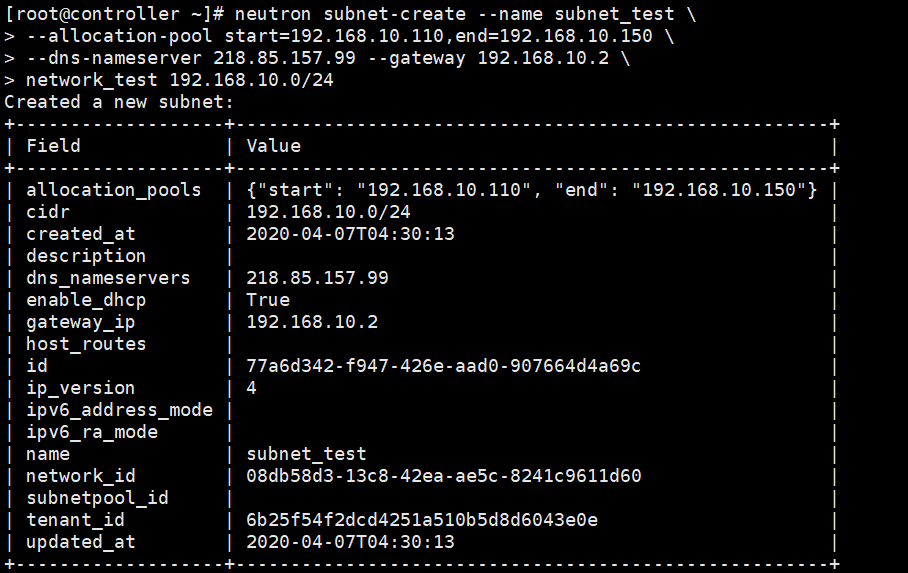
--dns-nameserver 218.85.157.99 --gateway 192.168.10.2 \

network-hgb07 192.168.10.0/24

#参数说明：SUBNET1\_NETWORK\_CIDR 使用创建的subnet1的子网CIDR替换

将``START\_IP\_ADDRESS``和``END\_IP\_ADDRESS``使用想分配给实例的子网网段的第一个和最后一个IP地址。

这个范围不能包括任何已经使用的IP地址。例如：

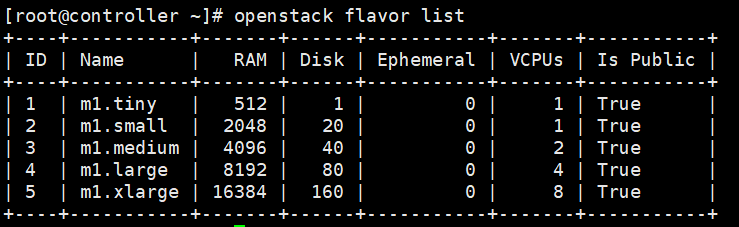


如上例子创建了一个名为network\_test网络192.168.10.0/24的子网，DHCP服务为每个实例分配IP，IP从192.168.10.110 到 192.168.10.150。所有实例的DNS使用218.85.157.99,网关为192.168.10.2。

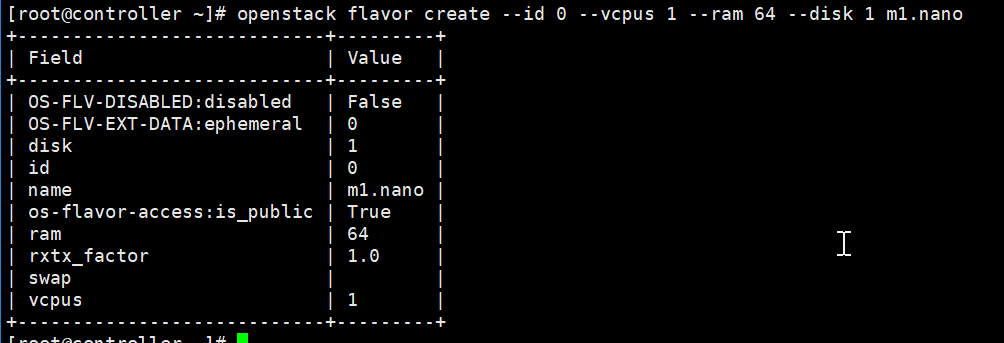
一个网络可以创建多个子网，子网和网络是一对多的关系，各个子网是不同的网段，即subnet不能重叠。属于不同网络的subnet可以重叠，既可以使用其它网络相同的网段。

##### **创建云主机的硬件配置方案**

默认的最小规格的主机需要512 MB内存。对于环境中计算节点内存不足4 GB的，推荐创建只需要64 MB的``m1.nano``规格的主机。默认的主机规格方案如下：



本例使用``m1.nano``规格的主机来加载CirrOS镜像。创建名为m1.nano的方案，cpu数量为1，内存64m，硬盘1G。

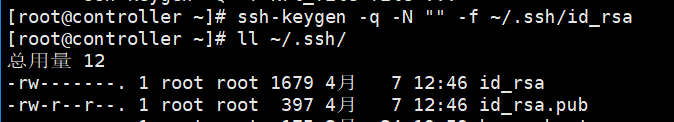


##### **[生成一个键值对](https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/launch-instance.html" \l "generate-a-key-pair)**

大部分云镜像支持公共密钥认证而不是传统的密码认证。在启动实例前，必须添加一个公共密钥到计算服务。

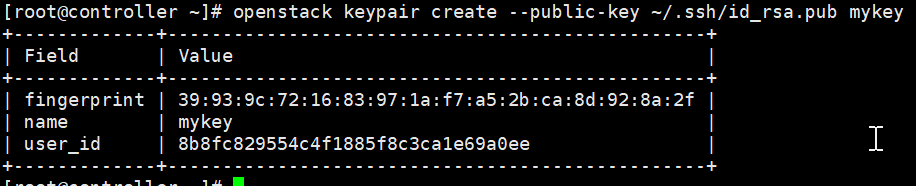
a.一键非交互式生成密钥对

ssh-keygen -q -N "" -f ~/.ssh/id\_rsa

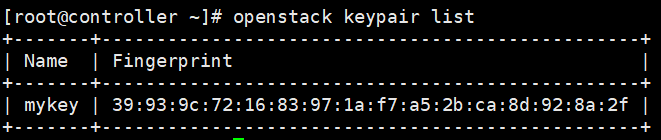


b.上传密钥对到openstack中，名字为mykey

openstack keypair create --public-key ~/.ssh/id\_rsa.pub mykey



1. 验证公钥



##### **创建安全组规则**

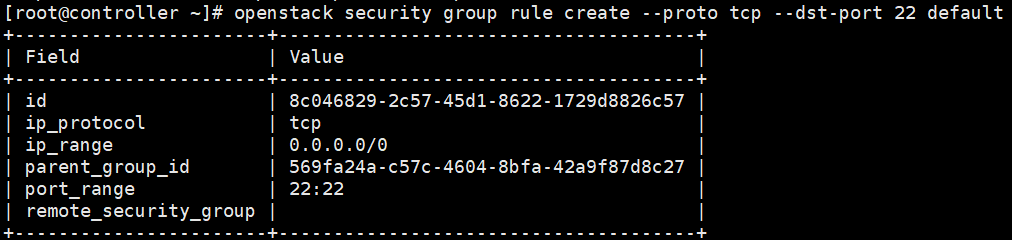
默认情况下， [``](https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/launch-instance.html" \l "id1)default``安全组适用于所有实例并且包括拒绝远程访问实例的防火墙规则。对诸如CirrOS这样的Linux镜像，推荐至少允许ICMP (ping) 和安全shell(SSH)规则。

添加规则到 default 安全组.

1. 添加允许ICMP规则



1. 添加允许ssh



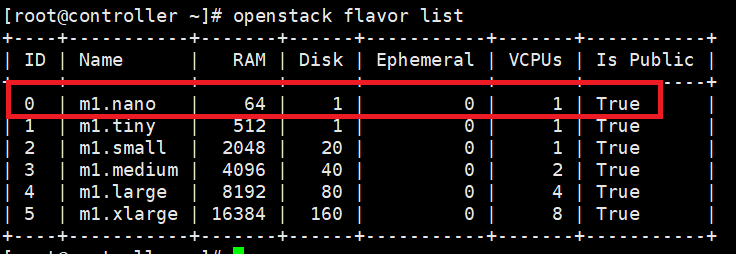
##### **启动一个实例**

* 命令行方式启动一个实例

启动一个实例，必须至少指定一个类型、镜像名称、网络、安全组、密钥和实例名称。

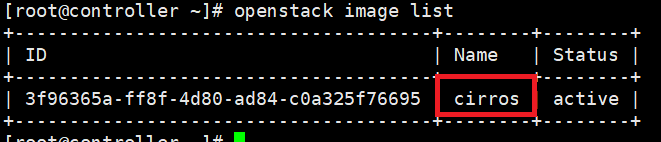
在启动实例前，在控制节点上，需获得 admin 凭证来获取只有管理员能执行的命令的访问权限。即source /root/admin-openrc.

1. 列出可用类型



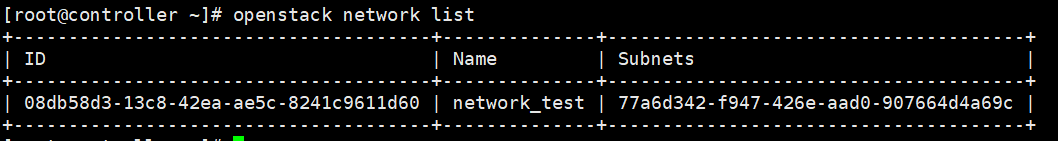
选择m1.nano.

1. 列出可用镜像

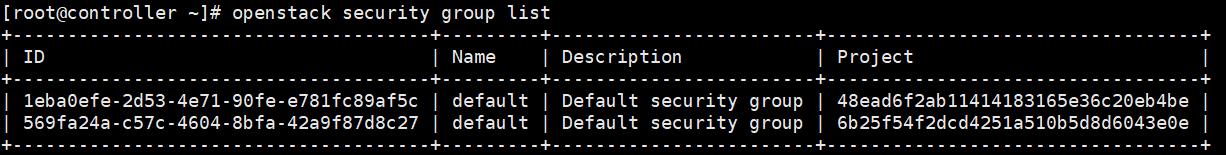


目前只有cirros可用。

1. 列出可用网络

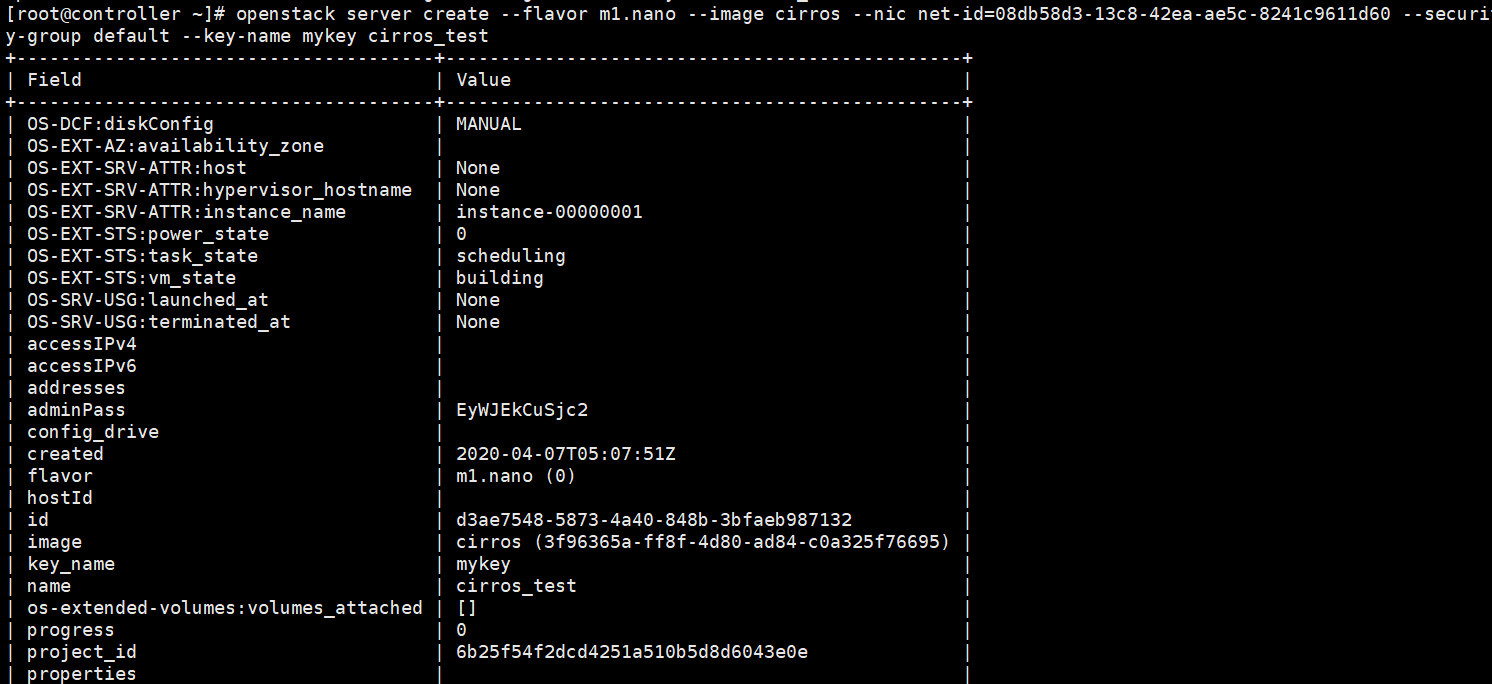


1. 列出可用安全组



选择default安全组。

1. 启动一个实例



语法：

openstack server create --flavor m1.nano --image cirros \

--nic net-id=XXXXXXXXX --security-group default \

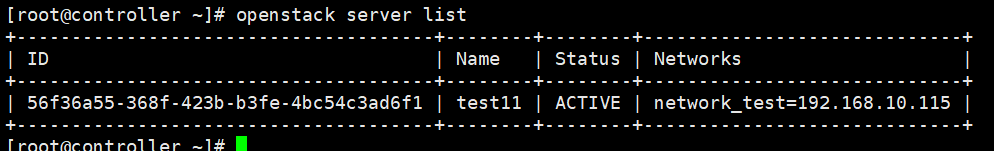
--key-name mykey cirros\_test

#参数说明： net-id=XXXXXXXXX 可通过neutron net-list查看获得。

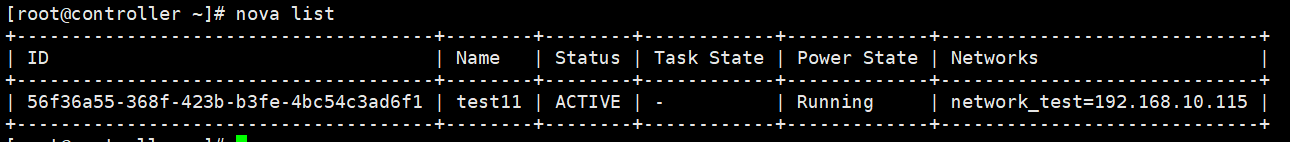
cirros\_test为实例的名称。

1. 查看实例

openstack server list



nova list



可以在计算节点上查看实例文件

compute1# cd /var/lib/nova下 进入对象的实例ID文件夹查看相关文件

错误1：虚机创建后，状态为error，在compute1节点看查看日志/var/log/nova/nova-compute.log发现，如下提示：

libvirtError: 内部错误：qemu unexpectedly closed the monitor:GLib-WARNING \*\*: 14:08:07.021: gmem.c:489: custom memory allocation vtable not supported。

解决方法：安装库文件：

    yum -y install  gperftools-libs

若compute节点没有安装qemu，则通过yum -y install qemu。

若远程连接发现，虚机一直处在booting from hard disk grub

解决方法：

在计算节点的nova配置文件/etc/nova/nova.conf添加下列行

[libvirt]

virt\_type = qemu

cpu\_mode = none

再重启nova-compute服务即可。

systemctl restart openstack-nova-compute.service

#### **8.4.2 Web方式启动一个云主机**