**一、基础环境准备**

**VMware创建两台CentOS 7.6虚拟机，内存2G，硬盘40G，双网卡（ens32、ens34）**

**controller：(内网，桥接)ens32：192.168.107.101 （外网，NAT）ens34:192.168.65.128**

**compute： (内网，桥接)ens32：192.168.107.102 （外网，NAT）ens34:192.168.65.129**

| **Passwords**[**¶**](https://docs.openstack.org/ocata/install-guide-rdo/environment-security.html#id1) | |
| --- | --- |
| **Password name** | **Description** |
| ROOT\_DBPASS | Root password for the database 000000 |
| ADMIN\_PASS | Password of user 000000 |
| CINDER\_DBPASS | Database password for the Block Storage service 000000 |
| CINDER\_PASS | Password of Block Storage service user 000000 |
| DASH\_DBPASS | Database password for the Dashboard 000000 |
| DEMO\_PASS | Password of user 000000 |
| GLANCE\_DBPASS | Database password for Image service 000000 |
| GLANCE\_PASS | Password of Image service user 000000 |
| KEYSTONE\_DBPASS | Database password of Identity service 000000 |
| METADATA\_SECRET | Secret for the metadata proxy 000000 |
| NEUTRON\_DBPASS | Database password for the Networking service 000000 |
| NEUTRON\_PASS | Password of Networking service user 000000 |
| NOVA\_DBPASS | Database password for Compute service 000000 |
| NOVA\_PASS | Password of Compute service user 000000 |
| PLACEMENT\_PASS | Password of the Placement service user 000000 |
| RABBIT\_PASS | Password of user openstack of RabbitMQ 000000 |

**1、配置主机名**

**controller节点 & compute节点**

**（1）修改主机名：**

**[root@basic ~]# hostnamectl set-hostname controller & compute**

**（2）配置地址解析：**

**[root@basic ~]# vi /etc/hosts**

**添加：**

**192.168.107.101 controller**

**192.168.107.102 compute**

**[root@basic ~]# init 6**

**（3）验证（配置完双网卡后才能验证）：**

**[root@controller ~]# ping compute**

**[root@compute ~]# ping controller**

**2、关闭防火墙和禁用安全访问策略**

**controller节点 & compute节点**

**（1）关闭防火墙，并取消其开机启动：**

**[root@controller ~]# systemctl stop firewalld**

**[root@controller ~]# ^stop^disable**

**（2）禁用安全访问策略（两处配置文件的SELINUX都改为禁用）：**

**[root@controller ~]# vi /etc/selinux/config**

**[root@controller ~]# vi /etc/sysconfig/selinux**

**SELINUX=disabled**

**[root@controller ~]# init 6 #重启后生效，但别忘了验证**

**（3）验证：**

**查看防火墙状态：**

**[root@controller ~]# firewall-cmd --state**

**查看是否禁用安全访问策略：**

**[root@controller ~]# getenforce**

**3、双网卡内外网同时访问路由设置**

**controller节点 & compute节点**

**说明：此处只演示controller节点的操作过程，compute节点配置基本一样，**

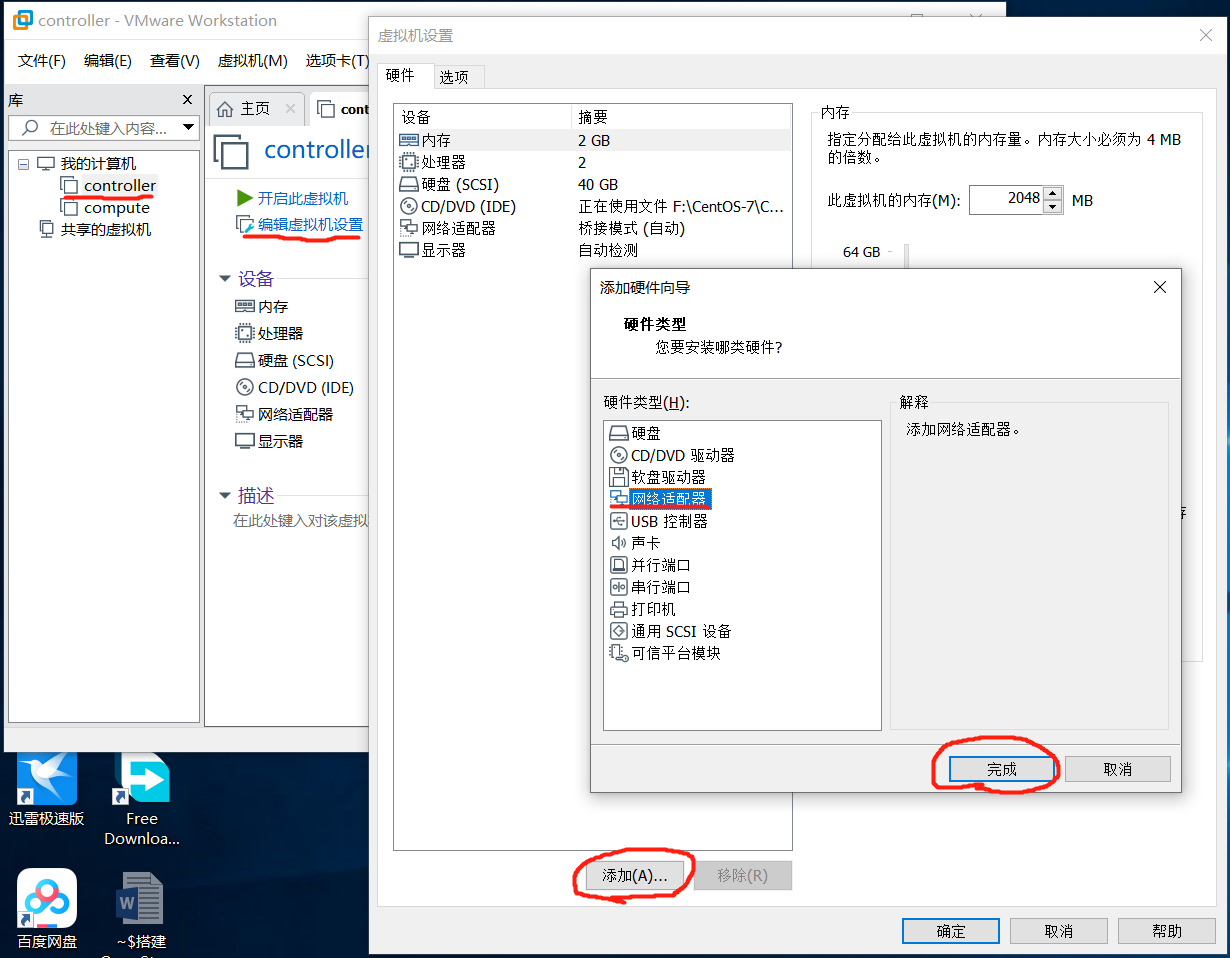
**服务器有两张网卡分别是ens32、ens34，**

**ens32配置内网IP：192.168.107.101，**

**ens34配置外网IP：192.168.65.128；**

**要求192.168.107.0/24网段走网卡ens32，网关是192.168.107.1，其余网段走网卡ens34，网关是192.168.65.2。**

**（1）添加物理网卡（注意顺序，网卡1-桥接，网卡2-NAT）**



**（2）配置内外网网卡IP（静态分配，保持网络稳定）**

**内网：**

**[root@controller ~]# cd /etc/sysconfig/network-scripts/**

**[root@controller ~]# vi ifcfg-ens32 #配置内网**

**修改：**

**BOOTPROTO=static**

**添加：**

**IPADDR=192.168.107.101**

**NETMASK=255.255.255.0**

**#GATEWAY=192.168.107.1 #只有一个默认路由，外网网卡做默认路由，这里注释掉**

**#DNS1=192.168.107.1**

**外网（注：NAT模式下IP段不能瞎搞，配置参考VMware虚拟网络编辑器里的信息）：**

**[root@controller ~]# cd /etc/sysconfig/network-scripts/**

**[root@controller ~]# cp ifcfg-ens32 ifcfg-ens34 #复制ens32的配置文件进行修改**

**[root@controller ~]# vi ifcfg-ens34 #配置外网**

**修改：**

**NAME=ens34**

**DEVICE=ens34**

**IPADDR=192.168.65.128**

**NETMASK=255.255.255.0**

**GATEWAY=192.168.65.2**

**DNS1=192.168.65.2**

**（3）添加内网访问路由**

**[root@controller ~]# vi /etc/sysconfig/network-scripts/route-ens32 #配置内网路由**

**192.168.107.0/24 via 192.168.107.1 dev ens32**

**（4）重启网络服务，并验证上述操作：**

**[root@controller ~]# systemctl restart network.service**

**[root@controller ~]# ping baidu.com #外网**

**[root@controller ~]# ping 192.168.107.102 #内网**

**4、搭建网络Yum源（注：开始使用putty连接工具操作）**

**controller节点 & compute节点**

**（1）删除所有源配置文件，并下载阿里源：**

**[root@controller ~]# rm -rf /etc/yum.repos.d/\***

**阿里源（主用）：**

**[root@controller ~]# curl -o /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo**

**网易源（备用）：**

**[root@controller ~]# curl -o /etc/yum.repos.d/CentOS-Base.repo http://mirrors.163.com/.help/CentOS7-Base-163.repo**

**（2）编辑源配置文件：**

**[root@controller ~]# vi /etc/yum.repos.d/CentOS-Base.repo**

**文件末尾添加如下两个源：**

**[openstack-ocata]**

**name=openstack-ocata**

**baseurl=https://mirrors.aliyun.com/centos/7.6.1810/cloud/x86\_64/openstack-ocata/**

**gpgcheck=0**

**enabled=1**

**[kvm-common]**

**name=kvm-common**

**baseurl=https://mirrors.aliyun.com/centos/7.6.1810/virt/x86\_64/kvm-common/**

**gpgcheck=0**

**enabled=1**

**网易源（备用）：**

**http://mirrors.163.com/centos/7.6.1810/cloud/x86\_64/openstack-ocata/**

**http://mirrors.163.com/centos/7.6.1810/virt/x86\_64/kvm-common/**

**（3）清除并重新生成缓存，且更新系统包，保证系统一致性：**

**[root@controller ~]# yum clean all**

**[root@controller ~]# yum makecache**

**[root@controller ~]# yum upgrade -y #更新系统包，慎用！卡死或中断会导致系统崩溃！**

**[root@controller ~]# init 6**

**（4）验证（通过安装软件来检验源配置是否正确）：**

**安装openstack客户端：**

**[root@controller ~]# yum install -y python-openstackclient openstack-selinux（备用，先不用安装）**

**安装vim，并设置编辑时显示行号：**

**[root@controller ~]# yum install -y vim**

**[root@controller ~]# vim /etc/vimrc**

**空白处添加：set number**

**5、同步时间**

**controller节点 & compute节点**

**（1）安装并配置chrony时间同步工具：**

**[root@controller ~]# yum install -y chrony**

**[root@controller ~]# vim /etc/chrony.conf #两个节点配置不同的地方只有这里！**

**controller节点：**

**注释掉原来的服务器，添加阿里云时间服务器,且允许192.168.107段IP地址访问此服务器：**

**server ntp1.aliyun.com iburst**

**server ntp2.aliyun.com iburst**

**server ntp3.aliyun.com iburst**

**allow 192.168.107.0/24**

**compute节点：**

**注释掉原来的服务器，添加controller节点为上游时间服务器：**

**server controller iburst**

**（2）启动服务并设置开机启动：**

**[root@controller ~]# systemctl start chronyd.service**

**[root@controller ~]# ^start^enable**

**（3）验证（有上游时间服务器状态输出即可）：**

**[root@controller ~]# date #系统时间**

**[root@controller ~]# hwclock #硬件时间**

**[root@controller ~]# hwclock --systohc**

**[root@controller ~]# chronyc sources #上游时间服务器状态**

**[root@controller ~]# ntpdate ntp1.aliyun.com**

**[root@controller ~]# timedatectl status**

**[root@controller ~]# timedatectl set-timezone Asia/Shanghai**

**6、消息队列（RabbitMQ）**

**controller节点**

**（1）安装rabbitmq，启动服务并设置开机启动：**

**[root@controller ~]# yum install -y rabbitmq-server**

**[root@controller ~]# systemctl start rabbitmq-server.service #服务占用端口5672**

**[root@controller ~]# ^start^enable**

**（2）验证（通过监听5672端口来查看rabbitmq服务是否启动）：**

**[root@controller ~]# netstat -lntp | grep 5672**

**（3）在rabbitmq里新增openstack用户，密码RABBIT\_PASS，给此用户授权并修改用户标签：**

**[root@controller ~]# rabbitmqctl add\_user openstack RABBIT\_PASS**

**[root@controller ~]# rabbitmqctl set\_permissions openstack ".\*" ".\*" ".\*"**

**（4）验证（启用rabbitmq的web管理插件来检验openstack用户是否可用）：**

**[root@controller ~]# rabbitmq-plugins list #查看rabbitmq插件**

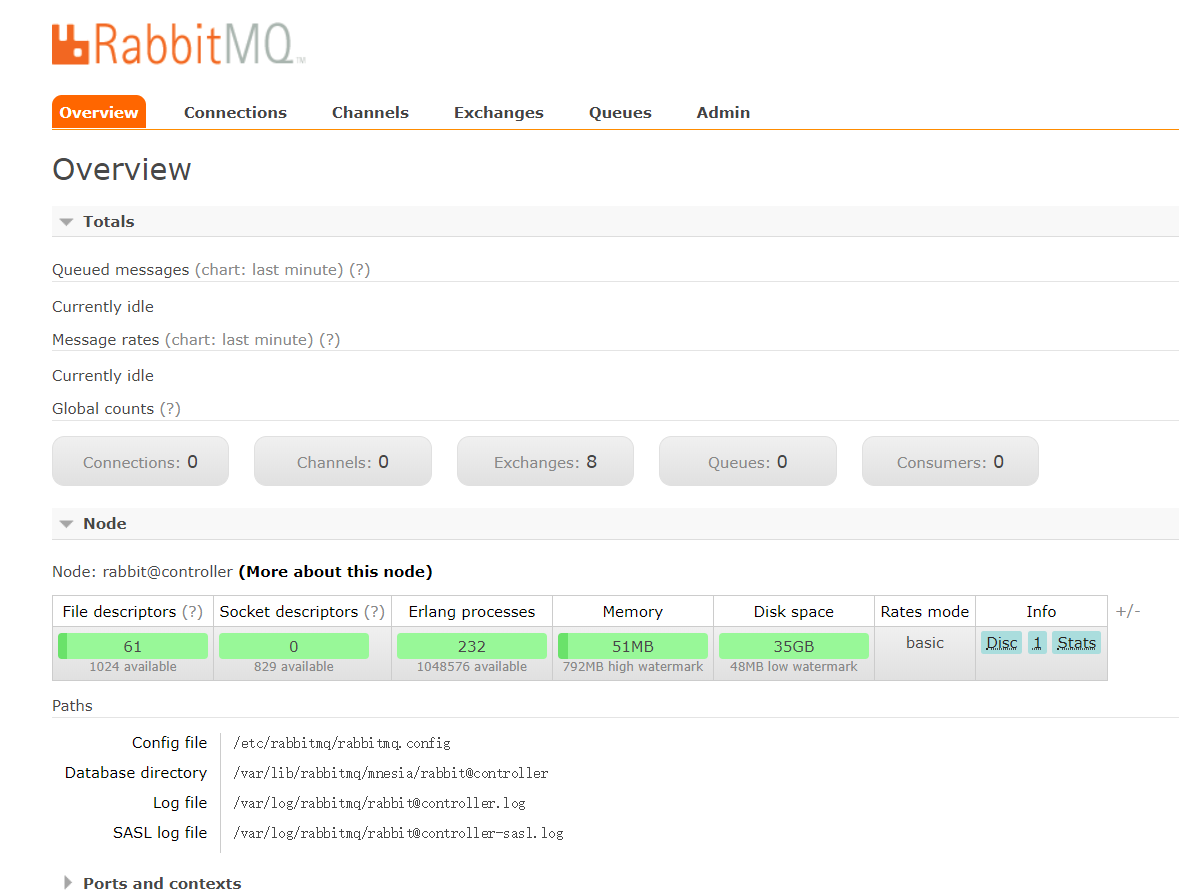
**[root@controller ~]# rabbitmq-plugins enable rabbitmq\_management #启用web管理插件，占用端口15672**

**[root@controller ~]# netstat -lntp | grep 15672**

**[root@controller ~]# systemctl restart rabbitmq-server.service #重启服务才能生效**

**浏览器访问192.168.65.128:15672，默认账号密码guest/guest，自建账号密码openstack/ RABBIT\_PASS，使用openstack用户登录：**





**7、SQL数据库**

**controller节点**

**（1）安装并配置mysql：**

**[root@controller ~]# yum install -y mariadb mariadb-server python2-PyMySQL**

**[root@controller ~]# vim /etc/my.cnf.d/openstack.cnf**

**[mysqld]**

**bind-address=192.168.107.101 #mysql监听地址**

**default-storage-engine = innodb #数据库默认存储引擎**

**innodb\_file\_per\_table = on #独立表空间**

**max\_connections= 4096 #最大连接数**

**collation-server = utf8\_general\_ci #默认校对规则**

**character-set-server = utf8 #数据库建库字符集**

**（2）启动服务并设置开机启动：**

**[root@controller ~]# systemctl start mariadb.service**

**[root@controller ~]# ^start^enable**

**（3）验证（查看mysql服务是否启动）：**

**[root@controller ~]# netstat -lntp | grep 3306**

**（4）mysql初始化设置：**

**[root@controller ~]# mysql\_secure\_installation**

**注意！按照以下步骤进行配置：**

**Enter current password for root (enter for none): #初次运行直接回车**

**Set root password? [Y/n] #是否设置root用户密码，输入Y并回车或直接回车**

**New password: #设置root用户的密码ROOT\_DBPASS**

**Re-enter new password: #再输入一次你设置的密码ROOT\_DBPASS**

**Remove anonymous users? [Y/n] #是否删除匿名用户,生产环境建议删除，所以直接回车**

**Disallow root login remotely? [Y/n] #是否禁止root远程登录，不禁止，输入n**

**Remove test database and access to it? [Y/n] #是否删除test数据库，直接回车**

**Reload privilege tables now? [Y/n] #是否重新加载权限表，直接回车**

**（5）验证新建的root用户是否可用（注：使用putty连接工具进行操作）：**

**[root@controller ~]# mysql -uroot -p #输入密码ROOT\_DBPASS登录，输入exit退出**

**8、Memcached（缓存服务）**

**controller节点**

**（1）安装并配置memcached：**

**[root@controller ~]# yum install -y memcached python-memcached**

**[root@controller ~]# vim /etc/sysconfig/memcached**

**修改：**

**OPTIONS="-l 127.0.0.1,::1,controller"**

**（2）启动服务并设置开机启动：**

**[root@controller ~]# systemctl start memcached.service #服务占用端口11211**

**[root@controller ~]# ^start^enable**

**（3）验证（检验Memcached服务是否成功启动）：**

**[root@controller ~]# systemctl status memcached.service**

**[root@controller ~]# netstat -lntp | grep 11211**

**[root@controller ~]# telnet 192.168.107.101 11211 #没有安装telnet的话yum安装一下**

**连接成功后按ctrl+]，然后输入quit退出。**

**到此处，两个节点的基础环境搭建完成，极少会报错，统统给我拍摄快照！**

**二、Keystone认证服务**

**controller节点**

**1、mysql中创建keystone服务所需的数据库和用户**

**（1）用root用户创建keystone数据库和用户，并给用户授权，密码为KEYSTONE\_DBPASS：**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "create database keystone;"**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "grant all privileges on keystone.\* to keystone@'localhost' identified by 'KEYSTONE\_DBPASS';" #本地访问keystone库的账号**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "grant all privileges on keystone.\* to keystone@'%' identified by 'KEYSTONE\_DBPASS';" #远程访问keystone库的账号**

**（2）验证（查看建好的数据库和用户）：**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "show databases;" |egrep "keystone"**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "select user,host from mysql.user;" |egrep "keystone"**

**（3）排错（还排个鸡儿的错，后续的操作就是在keystone库里建表，往表里写入数据，查表数据的话比较复杂，极其影响游戏体验，所以把删库跑路的精神贯彻到底，一言不合就删库从头再来）：**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "drop database keystone;"**

**2、安装并配置keystone服务**

**（1）安装keystone服务：**

**[root@controller ~]# yum install -y openstack-keystone**

**（2）编辑keystone配置文件：**

**[root@controller ~]# openssl rand -hex 10 #生成一个随机数**

**947493e6fa1690ca78bf**

**[root@controller ~]# vim /etc/keystone/keystone.conf #编辑配置文件**

**提示：vim模式中，/进入查找模式，输入关键字回车，小写n，向下查找，大写N，向上查找；**

**按gg跳到第一行，按shift+g跳到最后一行。**

**[DEFAULT]**

**admin\_token = 947493e6fa1690ca78bf**

**verbose = true #详细输出日志**

**[database]**

**connection = mysql+pymysql://keystone:KEYSTONE\_DBPASS@controller/keystone**

**[memcache]**

**servers = localhost:11211**

**[revoke]**

**driver = sql**

**[token]**

**provider = fernet**

**driver = memcache #token设置存储在memcache**

**查询修改过的所有行：**

**[root@controller ~]# grep -n '^[a-z]' /etc/keystone/keystone.conf**

**17:admin\_token = 947493e6fa1690ca78bf**

**147:verbose = true**

**713:connection = mysql+pymysql://keystone:KEYSTONE\_DBPASS@controller/keystone**

**1511:servers = localhost:11211**

**2489:driver = sql**

**2832:provider = fernet**

**2840:driver = memcache**

**（3）初始化keystone数据库(keystone服务与mysql建立联系)：**

**[root@controller ~]# su -s /bin/sh -c "keystone-manage db\_sync" keystone**

**（4）验证（报错基本就两个情况：mysql有问题，keystone.conf有问题）：**

**查看日志是否报错**

**[root@controller ~]# tail -200f /var/log/keystone/keystone.log**

**登录keystone库检查上述操作是否建表成功**

**[root@controller ~]# mysql -ukeystone -pKEYSTONE\_DBPASS -e "use keystone;show tables;"**

**（5）初始化Fernet keys、credential-keys，创建证书：**

**[root@controller ~]# keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone**

**[root@controller ~]# keystone-manage credential\_setup --keystone-user keystone --keystone-group keystone**

**验证（在/etc/keystone/会多出Fernet keys、credential-keys两个文件）：**

**[root@controller ~]# ls -l /etc/keystone/**

**[root@controller ~]# yum install -y tree**

**[root@controller ~]# tree /etc/keystone/**

**上面的（5）步骤是keystone对自己授权的一个过程，创建了一个keystone用户与一个keystone组。并对这个用户和组授权。因为keystone是对其他组件认证的服务，那么它自己就合格么？所以它先要对自己进行一下认证。**

**3、安装并配置apache整合keystone：**

**（1）安装并配置apache服务器：**

**[root@controller ~]# yum install -y httpd mod\_wsgi**

**[root@controller ~]# vim /etc/httpd/conf/httpd.conf**

**95:ServerName controller**

**（2）使用apache代理keystone（httpd和keystone建立联系）：**

**[root@controller ~]# ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/**

**（3）启动apache服务并设置开机启动（相当于启动keystone服务）：**

**[root@controller ~]# systemctl start httpd.service #公共端口5000，管理端口35357**

**[root@controller ~]# ^start^enable**

**（4）验证：**

**[root@controller ~]# netstat -lntp | grep 5000和35357**

**到这里keystone就部署完了，接下来就是如何要操作它了。**

**4、登录openstack并对keystone进行操作**



**（1）服务目录部分，创建keystone的catalog（认证服务实体）**

**设置环境变量使用admin（超级用户）登录openstack：**

**[root@controller ~]# export OS\_TOKEN=75f44400976856ad66ff**

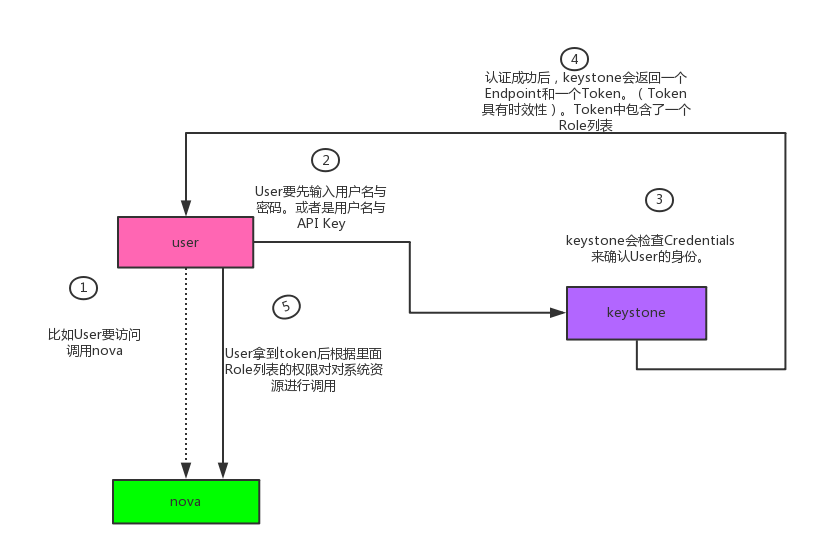
**[root@controller ~]# export OS\_URL=http://controller:35357/v3**

**[root@controller ~]# export OS\_IDENTITY\_API\_VERSION=3**

**#1第一步的值等于keystone.conf中admin\_token的值；并且只要在当前终端执行命令，当前终端就是管理员用户。**

**#2因为现在keystone没有提供Endpoint，所以自己手动指定一个Endpoint，以后的请求就往这个url提交；v3代表用的keystone版本为3。**

**#3认证版本为3。**



**为keystone创建catalog（认证服务实体）：**

**[root@controller ~]# openstack service create \**

**--name keystone --description "OpenStack Identity" identity**

**基于建立的服务实体，创建访问该实体的三个api端点：**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

**identity public http://controller:5000/v3**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

**identity internal http://controller:5000/v3**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

**identity admin http://controller:35357/v3**

**上面的步骤操作成功后，这样keystone就可以与其它服务进行交互了。**

**（2）用户认证部分，创建域，项目(租户)，用户，角色，并把四个元素关联在一起**

**在openstack中最大的资源集合就是域->项目(租户)->用户->角色**

**创建域（domain）： #全局唯一，只需要建一次默认的域**

**[root@controller ~]# openstack domain create --description "Default Domain" default**

**创建admin项目（project）在“default”域中：**

**[root@controller ~]# openstack project create --domain default \**

**--description "Admin Project" admin**

**创建admin用户（user）在“default”域中：**

**[root@controller ~]# openstack user create --domain default \**

**--password-prompt admin　　　　　　　　#输入密码ADMIN\_PASS**

**创建admin角色（role）：**

**[root@controller ~]# openstack role create admin**

**创建上述三者的关联：**

**[root@controller ~]# openstack role add --project admin --user admin admin #添加admin角色给admin项目和admin用户**

**（3）使用Bootstrap完成（1）和（2）二者的工作（注：我们使用的是这种配置方法）：**

**[root@controller ~]# keystone-manage bootstrap --bootstrap-password ADMIN\_PASS \**

**--bootstrap-admin-url http://controller:35357/v3/ \**

**--bootstrap-internal-url http://controller:5000/v3/ \**

**--bootstrap-public-url http://controller:5000/v3/ \**

**--bootstrap-region-id RegionOne**

**（4）创建admin登录所需环境变量脚本：**

**[root@controller ~]# vim /etc/keystone/admin-openrc**

**export OS\_PROJECT\_DOMAIN\_NAME=Default**

**export OS\_USER\_DOMAIN\_NAME=Default**

**export OS\_PROJECT\_NAME=admin**

**export OS\_USERNAME=admin**

**export OS\_PASSWORD=ADMIN\_PASS**

**export OS\_AUTH\_URL=http://controller:35357/v3**

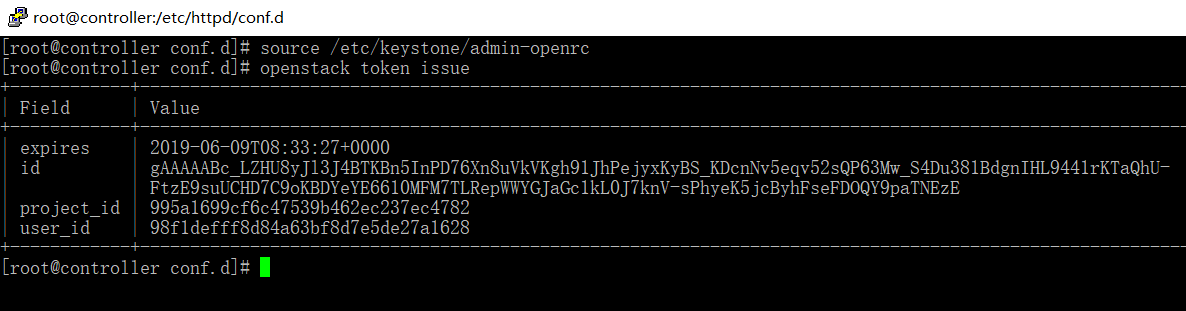
**export OS\_IDENTITY\_API\_VERSION=3**

**export OS\_IMAGE\_API\_VERSION=2**

**验证（使用脚本填写admin认证信息登录openstack，并查询基本信息）：**

**[root@controller ~]# source /etc/keystone/admin-openrc**

**[root@controller ~]# openstack token issue #获取令牌**

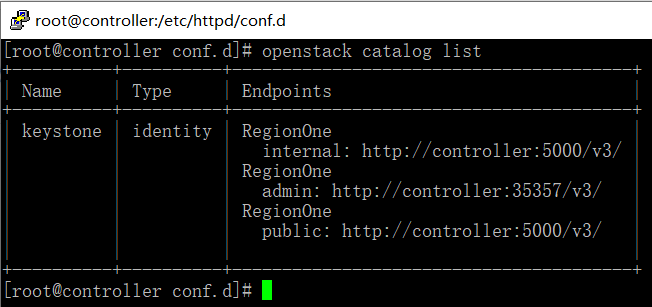


**服务目录部分：**

**[root@controller ~]# openstack service list #服务列表**

**[root@controller ~]# openstack endpoint list #端点列表**

**[root@controller ~]# openstack catalog list #服务实体列表**

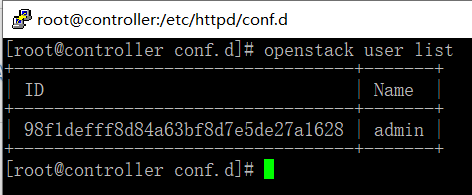


**用户认证部分：**

**[root@controller ~]# openstack domain list #域列表**

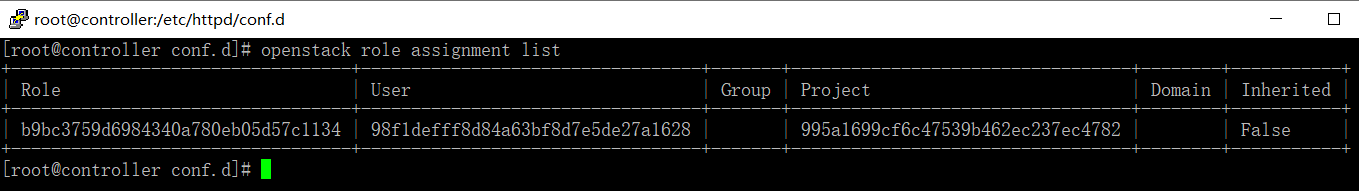
**[root@controller ~]# openstack project list #项目列表**

**[root@controller ~]# openstack user list #用户列表**



**[root@controller ~]# openstack role list #角色列表**

**[root@controller ~]# openstack role assignment list #上述几个的关联列表**



**（5）创建后期测试用的demo**

**创建项目名为demo：**

**[root@controller ~]# openstack project create --domain default \**

**--description "Demo Project" demo**

**创建普通用户为demo并设置密码：**

**[root@controller ~]# openstack user create --domain default \**

**--password-prompt demo #设置DEMO\_PASS密码**

**创建普通用户的角色即user：**

**[root@controller ~]# openstack role create user**

**添加user角色给demo项目和demo用户：**

**[root@controller ~]# openstack role add --project demo --user demo user**

**创建demo登录所需环境变量脚本：**

**[root@controller ~]# vim /etc/keystone/demo-openrc**

**export OS\_PROJECT\_DOMAIN\_NAME=Default**

**export OS\_USER\_DOMAIN\_NAME=Default**

**export OS\_PROJECT\_NAME=demo**

**export OS\_USERNAME=demo**

**export OS\_PASSWORD=DEMO\_PASS**

**export OS\_AUTH\_URL=http://controller:5000/v3**

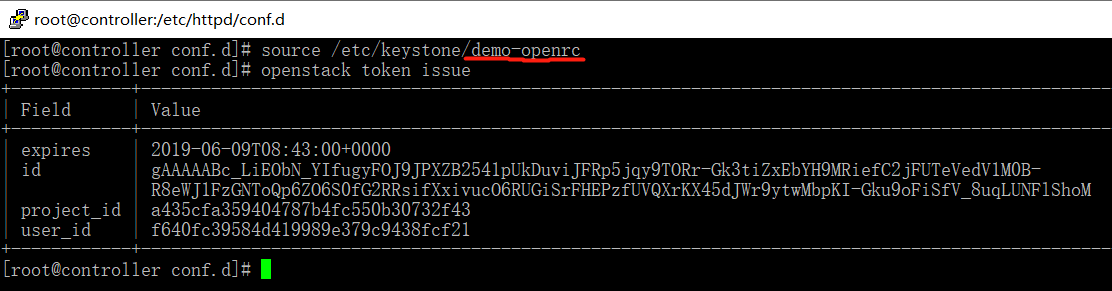
**export OS\_IDENTITY\_API\_VERSION=3**

**export OS\_IMAGE\_API\_VERSION=2**

**验证（用demo用户登录openstack，获取令牌）：**

**[root@controller ~]# source /etc/keystone/demo-openrc**

**[root@controller ~]# openstack token issue #获取令牌**



**（6）为后续的服务创建统一的项目叫service**

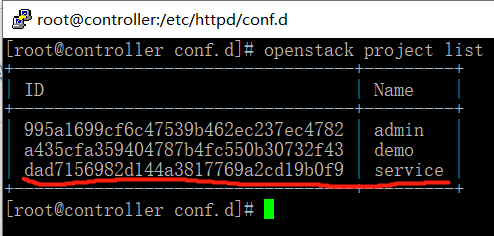
**#后面所有的服务nova、glace等等，公用一个项目service，都是管理员角色admin （组件之间的通信，角色都是admin）**

**[root@controller ~]# source /etc/keystone/admin-openrc #切换admin去创建**

**[root@controller ~]# openstack project create --domain default \**

**--description "Service Project" service**

**[root@controller ~]# openstack project list #查看项目列表，有没有叫service的项目**



**到此处，keystone服务部署完成，关机拍摄快照！**

**三、Glance镜像服务**

**controller节点**

**1、mysql中创建glance服务所需的数据库和用户**

**（1）用root用户创建glance数据库和用户，并给用户授权，密码为GLANCE\_DBPASS：**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "create database glance;"**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "grant all privileges on glance.\* to glance@'localhost' identified by 'GLANCE\_DBPASS';"**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "grant all privileges on glance.\* to glance@'%' identified by 'GLANCE\_DBPASS';"**

**（2）验证（查看建好的数据库和用户）：**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "show databases;" |egrep "glance"**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "select user,host from mysql.user;" |egrep "glance"**

**（3）排错（删库跑路，从头再来）：**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "drop database glance;"**

**2、安装并配置glance服务**

**（1）安装glance服务**

**[root@controller ~]# yum install -y openstack-glance**

**（2）编辑配置文件（有两个：glance-api.conf和glance-registry.conf）**

**[root@controller ~]# vim /etc/glance/glance-api.conf**

**[database]**

**connection = mysql+pymysql://glance:GLANCE\_DBPASS@controller/glance**

**[glance\_store]**

**stores = file,http**

**default\_store = file**

**filesystem\_store\_datadir = /var/lib/glance/images/**

**[keystone\_authtoken]**

**auth\_uri = http://controller:5000**

**auth\_url = http://controller:35357**

**memcached\_servers = controller:11211**

**auth\_type = password**

**project\_domain\_name = default**

**user\_domain\_name = default**

**project\_name = service**

**username = glance**

**password = GLANCE\_PASS**

**[paste\_deploy]**

**flavor = keystone**

**查询修改过的所有行：**

**[root@controller ~]# grep -n '^[a-z]' /etc/glance/glance-api.conf**

**1825:connection = mysql+pymysql://glance:000000@controller/glance**

**1941:stores = file,http**

**1973:default\_store = file**

**2292:filesystem\_store\_datadir = /var/lib/glance/images**

**3294:auth\_uri = http://controller:5000**

**3295:auth\_url = http://controller:35357**

**3296:memcached\_servers = controller:11211**

**3297:auth\_type = password**

**3298:project\_domain\_name = default**

**3299:user\_domain\_name = default**

**3300:project\_name = service**

**3301:username = glance**

**3302:password = 000000**

**4266:flavor = keystone**

**[root@controller ~]# vim /etc/glance/glance-registry.conf**

**[database]**

**connection = mysql+pymysql://glance:GLANCE\_DBPASS@controller/glance**

**[keystone\_authtoken]**

**auth\_uri = http://controller:5000**

**auth\_url = http://controller:35357**

**memcached\_servers = controller:11211**

**auth\_type = password**

**project\_domain\_name = default**

**user\_domain\_name = default**

**project\_name = service**

**username = glance**

**password = GLANCE\_PASS**

**[paste\_deploy]**

**flavor = keystone**

**查询修改过的所有行：**

**[root@controller ~]# grep -n '^[a-z]' /etc/glance/glance-registry.conf**

**1115:connection = mysql+pymysql://glance:000000@controller/glance**

**1217:auth\_uri = http://controller:5000**

**1218:auth\_url = http://controller:35357**

**1219:memcached\_servers = controller:11211**

**1220:auth\_type = password**

**1221:project\_domain\_name = default**

**1222:user\_domain\_name = default**

**1223:project\_name = service**

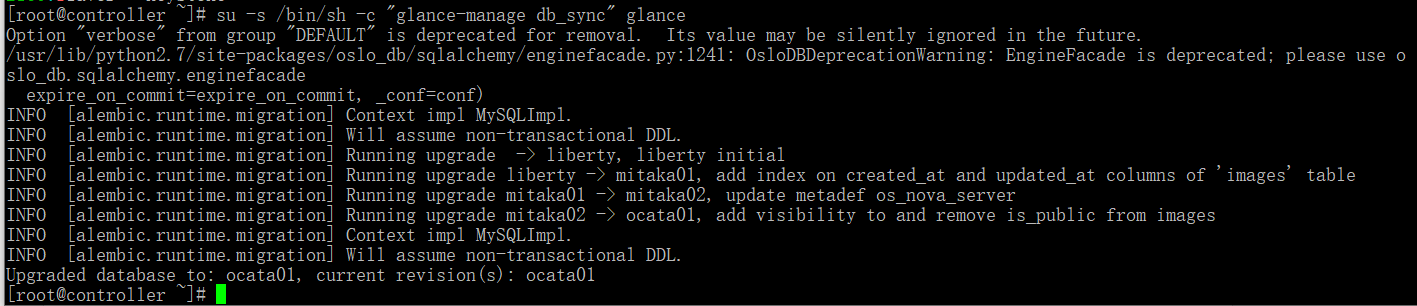
**1224:username = glance**

**1225:password = 000000**

**2160:flavor = keystone**

**3、初始化glance数据库(glance服务与mysql建立联系)**

**[root@controller ~]# su -s /bin/sh -c "glance-manage db\_sync" glance**



**验证（登录glance库检查上述操作是否建表成功）：**

**[root@controller ~]# mysql -uglance -pGLANCE\_DBPASS -e "use glance;show tables;"**

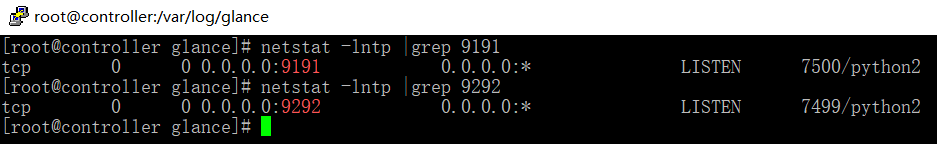
**4、启动glance服务并设置开机启动**

**[root@controller ~]# systemctl start openstack-glance-api.service openstack-glance-registry.service**

**[root@controller ~]# ^start^enable**

**验证（registry监听9191端口，glance-api监听9292端口）：**

**[root@linux-node1 ~]# netstat -lntp |grep 9191 和 9292**



**5、登录keystone给做glance服务注册**

**（1）admin用户登录openstack：**

**[root@controller ~]# source /etc/keystone/admin-openrc**

**（2）用户认证部分：**

**创建glance用户（user）在“default”域中：**

**[root@controller ~]# openstack user create --domain default \**

**--password-prompt glance #设置GLANCE\_PASS密码**

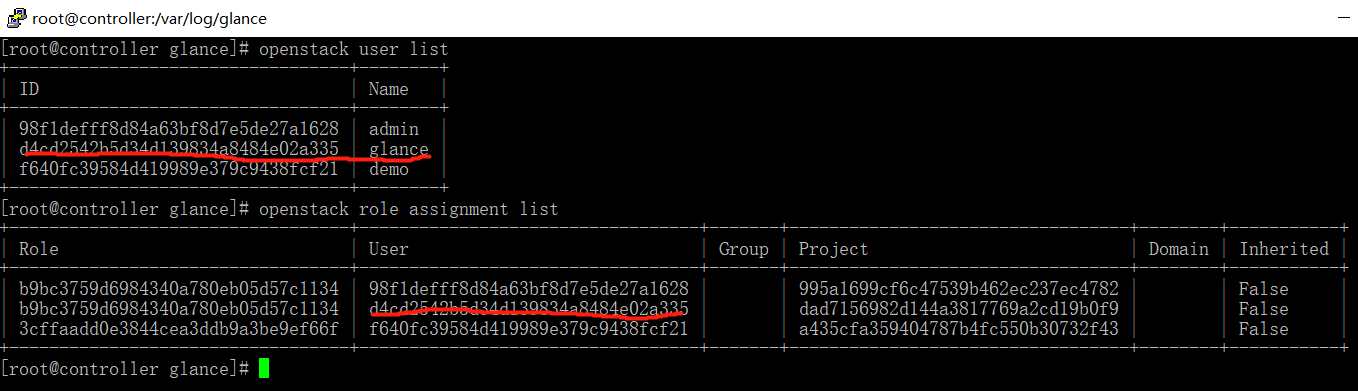
**将glance用户添加到admin角色，和service项目：**

**[root@controller ~]# openstack role add --project service --user glance admin**

**验证：**

**[root@controller ~]# openstack user list #用户列表**

**[root@controller ~]# openstack role assignment list #关联列表**



**（3）服务目录部分（建立访问glance的catalog）：**

**[root@controller ~]# openstack service create --name glance \**

**--description "OpenStack Image" image**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

**image public http://controller:9292**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

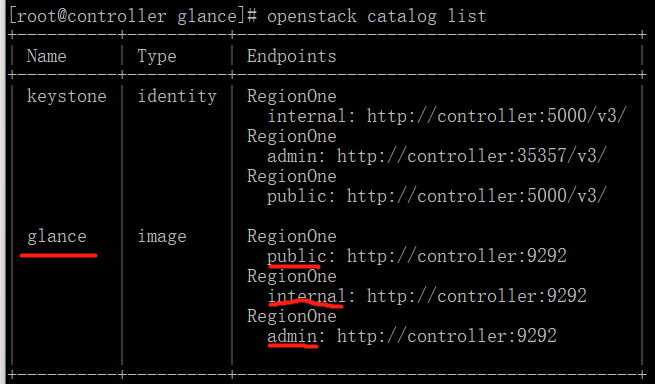
**image internal http://controller:9292**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

**image admin http://controller:9292**

**验证和排错：**

**[root@controller ~]# openstack catalog list #服务实体列表**



**[root@controller ~]# openstack endpoint --help #查看帮助文档，endpoint建错时可以删除**

**（4）查看并上传镜像**

**[root@controller ~]# openstack image list #镜像列表**

**[root@controller ~]# yum install -y wget #安装wget**

**[root@controller ~]# cd /opt #切换到opt目录，下载的东西统一放这里**

**[root@controller ~]# wget http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86\_64-disk.img #下载cirros磁盘镜像**

**上传镜像，使用 QCOW2 磁盘格式：**

**[root@controller ~]# openstack image create "cirros" \**

**--file cirros-0.4.0-x86\_64-disk.img \**

**--disk-format qcow2 --container-format bare \**

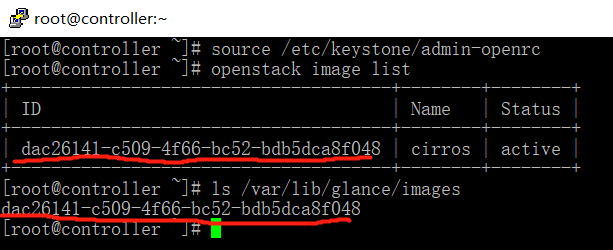
**--public**

**验证：**

**[root@controller ~]# source /etc/keystone/admin-openrc**

**[root@controller ~]# openstack image list #镜像列表**

**[root@controller ~]# ls /var/lib/glance/images #openstack里磁盘镜像的默认存放路径**



**四、Nova计算服务**

**controller节点 & compute节点**

**controller节点上的配置：**

**1、数据库操作**

**（1）创建nova数据库（3个，nova、nova\_api、nova\_cell0）：**

**[root@controller ~]# mysql -uroot -p #输入ROOT\_DBPASS登录数据库**

**MariaDB [(none)]> CREATE DATABASE nova\_api;**

**MariaDB [(none)]> CREATE DATABASE nova;**

**MariaDB [(none)]> CREATE DATABASE nova\_cell0;**

**（2）创建数据库用户并赋予权限（2个nova账户，本地访问，远程访问，管理nova\_api，nova，nova\_cell0三个数据库）：**

**MariaDB [(none)]> grant all privileges on nova\_api.\* to nova@'localhost' identified by 'NOVA\_DBPASS';**

**MariaDB [(none)]> grant all privileges on nova\_api.\* to nova@'%' identified by 'NOVA\_DBPASS';**

**MariaDB [(none)]> grant all privileges on nova.\* to nova@'localhost' identified by 'NOVA\_DBPASS';**

**MariaDB [(none)]> grant all privileges on nova.\* to nova@'%' identified by 'NOVA\_DBPASS';**

**MariaDB [(none)]> grant all privileges on nova\_cell0.\* to nova@'localhost' identified by 'NOVA\_DBPASS';**

**MariaDB [(none)]> grant all privileges on nova\_cell0.\* to nova@'%' identified by 'NOVA\_DBPASS';**

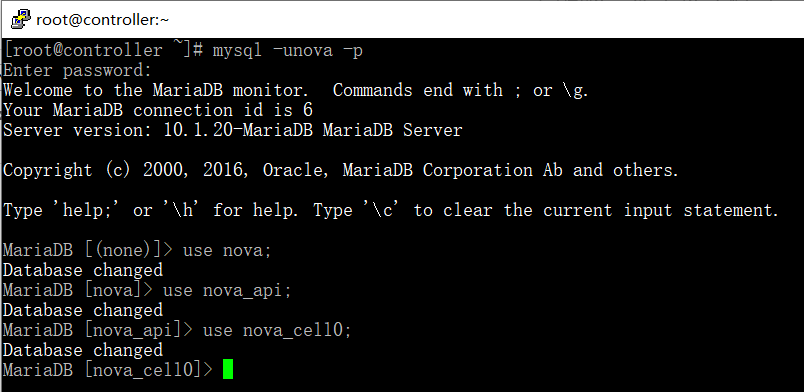
**MariaDB [(none)]> exit #退出**

**（3）验证（查看建好的数据库和用户）：**

**[root@controller ~]# mysql -uroot -pROOT\_DBPASS -e "show databases;" |egrep "nova|nova\_api|nova\_cell0"**

**[root@controller ~]# mysql -unova -p #输入NOVA\_DBPASS登录**

**MariaDB [(none)]> use nova; #nova，nova\_api，nova\_cell0三个数据库切换一下**



**2、登录keystone给做nova服务注册**

**（1）admin用户登录openstack：**

**[root@controller ~]# source /etc/keystone/admin-openrc**

**（2）用户认证部分：**

**创建nova用户（user）在“default”域中：**

**[root@controller ~]# openstack user create --domain default \**

**--password-prompt nova #设置NOVA\_PASS密码**

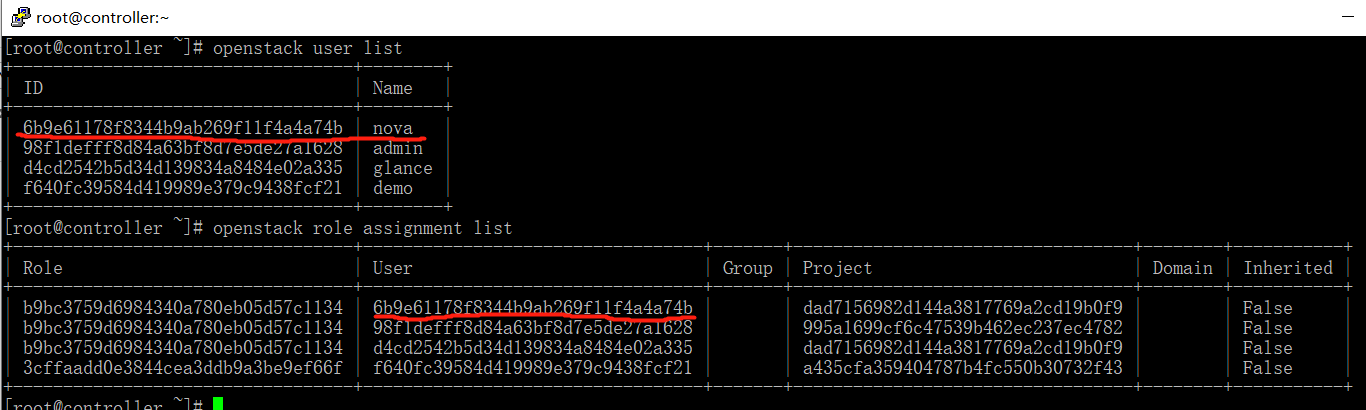
**将nova用户添加到admin角色，和service项目：**

**[root@controller ~]# openstack role add --project service --user nova admin**

**验证：**

**[root@controller ~]# openstack user list #用户列表**

**[root@controller ~]# openstack role assignment list #关联列表**



**（3）服务目录部分（创建nova服务实体，1个服务带3个端点）：**

**[root@controller ~]# openstack service create --name nova \**

**--description "OpenStack Compute" compute**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

**compute public http://controller:8774/v2.1**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

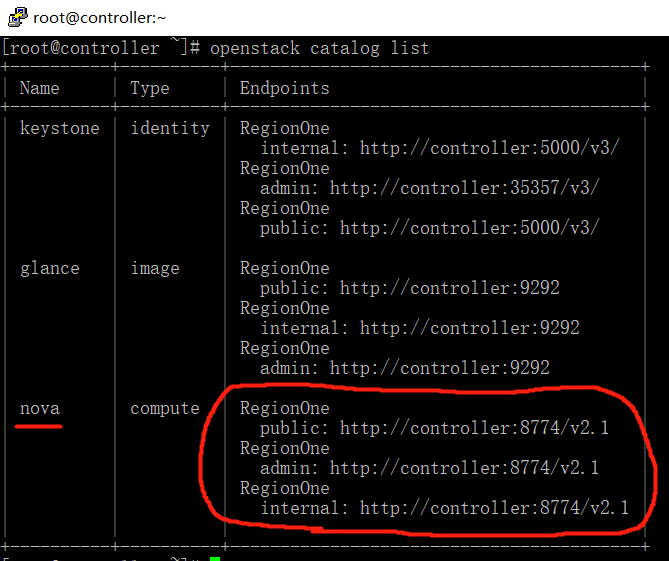
**compute internal http://controller:8774/v2.1**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

**compute admin http://controller:8774/v2.1**

**验证和排错：**

**[root@controller ~]# openstack catalog list #服务实体列表**



**[root@controller ~]# openstack endpoint --help #查看帮助文档，endpoint建错时可以删除**

**3、登录keystone给做placement服务注册**

**（1）admin用户登录openstack：**

**[root@controller ~]# source /etc/keystone/admin-openrc**

**（2）用户认证部分：**

**[root@controller ~]# openstack user create --domain default \**

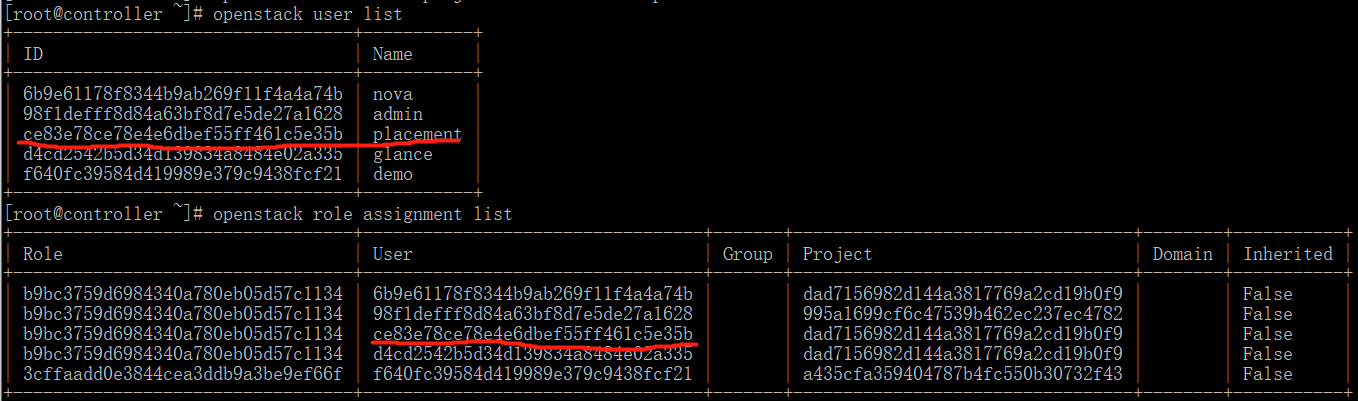
**--password-prompt placement #设置PLACEMENT\_PASS密码**

**[root@controller ~]# openstack role add --project service --user placement admin**

**验证：**

**[root@controller ~]# openstack user list #用户列表**

**[root@controller ~]# openstack role assignment list #关联列表**



**（3）服务目录部分（创建placement服务实体，1个服务带3个端点）：**

**[root@controller ~]# openstack service create --name placement --description "Placement API" placement**

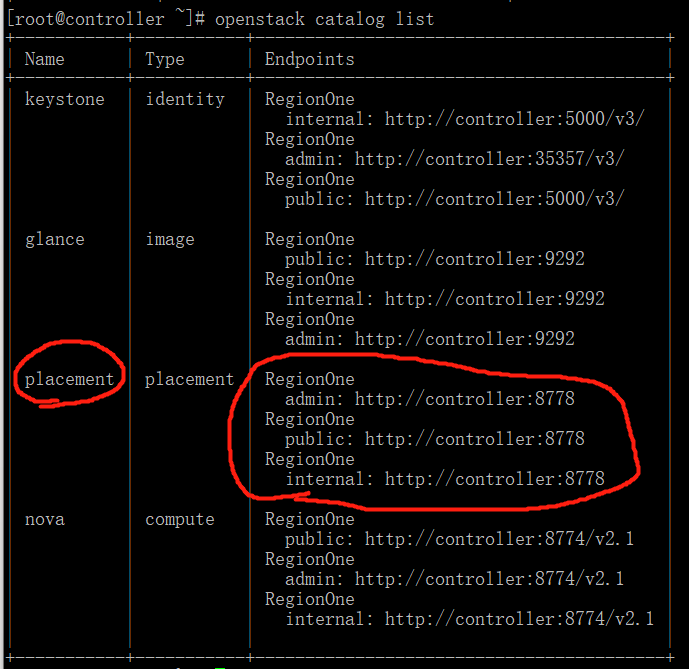
**[root@controller ~]# openstack endpoint create --region RegionOne placement public http://controller:8778**

**[root@controller ~]# openstack endpoint create --region RegionOne placement internal http://controller:8778**

**[root@controller ~]# openstack endpoint create --region RegionOne placement admin http://controller:8778**

**验证和排错：**

**[root@controller ~]# openstack catalog list #服务实体列表**



**[root@controller ~]# openstack endpoint --help #查看帮助文档，endpoint建错时可以删除**

**4、安装并配置nova组件**

**（1）安装nova服务相关包组：**

**[root@controller ~]# yum install -y openstack-nova-api openstack-nova-conductor \**

**openstack-nova-console openstack-nova-novncproxy \**

**openstack-nova-scheduler openstack-nova-placement-api**

**（2）编辑配置文件（有两个：nova.conf，00-nova-placement-api.conf）：**

**[root@controller ~]# vim /etc/nova/nova.conf**

**[DEFAULT]**

**my\_ip = 192.168.107.101**

**use\_neutron = True**

**firewall\_driver = nova.virt.firewall.NoopFirewallDriver**

**enabled\_apis = osapi\_compute,metadata**

**transport\_url = rabbit://openstack:RABBIT\_PASS@controller**

**[api]**

**auth\_strategy = keystone**

**[api\_database]**

**connection = mysql+pymysql://nova:NOVA\_DBPASS@controller/nova\_api**

**[database]**

**connection = mysql+pymysql://nova:NOVA\_DBPASS@controller/nova**

**[glance]**

**api\_servers = http://controller:9292**

**[keystone\_authtoken]**

**auth\_uri = http://controller:5000**

**auth\_url = http://controller:35357**

**memcached\_servers = controller:11211**

**auth\_type = password**

**project\_domain\_name = default**

**user\_domain\_name = default**

**project\_name = service**

**username = nova**

**password = NOVA\_DBPASS**

**[oslo\_concurrency]**

**lock\_path = /var/lib/nova/tmp**

**[placement]**

**os\_region\_name = RegionOne**

**project\_domain\_name = Default**

**project\_name = service**

**auth\_type = password**

**user\_domain\_name = Default**

**auth\_url = http://controller:35357/v3**

**username = placement**

**password = PLACEMENT\_PASS**

**[vnc]**

**enabled = true**

**vncserver\_listen = $my\_ip**

**vncserver\_proxyclient\_address = $my\_ip**

**查询修改过的所有行：**

**[root@controller ~]# grep -n '^[a-z|[]' /etc/nova/nova.conf**

**1:[DEFAULT]**

**1481:my\_ip=192.168.107.101**

**2306:use\_neutron=true**

**2465:firewall\_driver = nova.virt.firewall.NoopFirewallDriver**

**2629:enabled\_apis=osapi\_compute,metadata**

**3021:transport\_url=rabbit://openstack:000000@controller**

**3069:[api]**

**3085:auth\_strategy=keystone**

**3367:[api\_database]**

**3379:connection = mysql+pymysql://nova:000000@controller/nova\_api**

**……**

**4369:[database]**

**4396:connection = mysql+pymysql://nova:000000@controller/nova**

**……**

**4937:[glance]**

**4955:api\_servers = http://controller:9292**

**……**

**5596:[keystone\_authtoken]**

**5609:auth\_uri = http://controller:5000**

**5610:auth\_url = http://controller:35357**

**5611:memcached\_servers = controller:11211**

**5612:auth\_type = password**

**5613:project\_domain\_name = Default**

**5614:user\_domain\_name = Default**

**5615:project\_name = service**

**5616:username = nova**

**5617:password = 000000**

**……**

**7296:[oslo\_concurrency]**

**7311:lock\_path=/var/lib/nova/tmp**

**……**

**8142:[placement]**

**8156:os\_region\_name = RegionOne**

**8157:project\_domain\_name = Default**

**8158:project\_name = service**

**8159:auth\_type = password**

**8160:user\_domain\_name = Default**

**8161:auth\_url = http://controller:35357/v3**

**8162:username = placement**

**8163:password = 000000**

**……**

**9729:[vnc]**

**9745:enabled=true**

**9768:vncserver\_listen = $my\_ip**

**9780:vncserver\_proxyclient\_address = $my\_ip**

**……**

**[scheduler]**

**discover\_hosts\_in\_cells\_interval = 300**

**由于数据包bug，必须启用访问位置API通过添加以下配置/etc/httpd/conf.d/00-nova-placement-api.conf：**

**[root@controller ~]# vim /etc/httpd/conf.d/00-nova-placement-api.conf**

**添加：**

**<Directory /usr/bin>**

**<IfVersion >= 2.4>**

**Require all granted**

**</IfVersion>**

**<IfVersion < 2.4>**

**Order allow,deny**

**Allow from all**

**</IfVersion>**

**</Directory>**

**重启apache：**

**[root@controller ~]# systemctl restart httpd**

**查看是否apache重启成功：**

**[root@controller ~]# netstat -lntp |grep 8778**

**5、同步数据库操作**

**（1）填充nova-api数据库（同步compute数据库）：**

**[root@controller ~]# su -s /bin/sh -c "nova-manage api\_db sync" nova**

**（2）注册cell0数据库：**

**[root@controller ~]# su -s /bin/sh -c "nova-manage cell\_v2 map\_cell0" nova**

**（3）创建cell1单元格：**

**[root@controller ~]# su -s /bin/sh -c "nova-manage cell\_v2 create\_cell --name=cell1 --verbose" nova**

**（4）填充nova数据库：**

**[root@controller ~]# su -s /bin/sh -c "nova-manage db sync" nova**

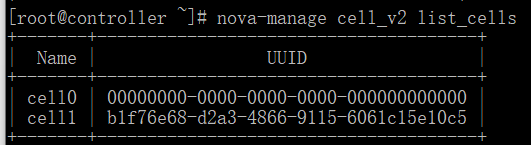
**（5）验证：**

**查看nova库所有表：**

**[root@controller ~]# mysql -unova -pNOVA\_DBPASS -e "use nova;show tables;"**

**核查nova cell0和cell1 是否注册成功：**

**[root@controller ~]# nova-manage cell\_v2 list\_cells**



**日志：**

**[root@controller ~]# tail -200f /var/log/nova/nova-manage.log**

**6、启动nova相关服务并设置开机启动：**

**（1）启动服务并设置开机启动**

**[root@controller ~]# systemctl start openstack-nova-api.service \**

**openstack-nova-consoleauth.service openstack-nova-scheduler.service \**

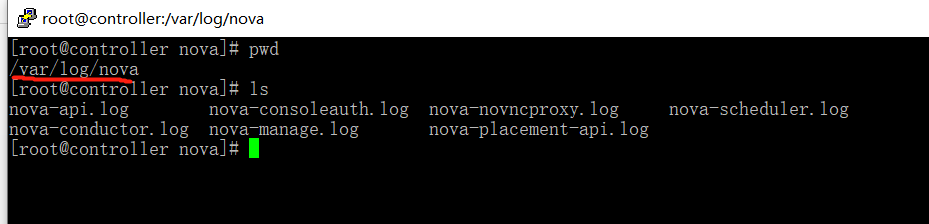
**openstack-nova-conductor.service openstack-nova-novncproxy.service**

**[root@controller ~]# ^start^enable**

**（2）查看日志**

**[root@controller ~]# tail -200f /var/log/nova/具体的日志文件名**

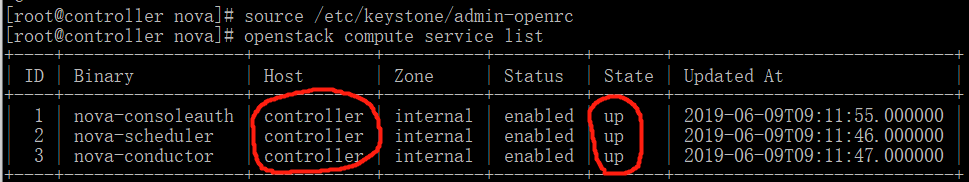
**#此处是日志目录，哪个服务启动不了看哪个的日志**



**（3）验证效果**

**[root@controller ~]# source /etc/keystone/admin-openrc**

**[root@controller ~]# openstack compute service list**



**[root@controller ~]# openstack catalog list**

**[root@controller ~]# openstack image list**

**compute节点上的配置：**

**1、安装相关包组：**

**[root@compute ~]# yum install -y openstack-nova-compute**

**2、编辑/etc/nova/nova.conf文件**

**[root@compute ~]# vim /etc/nova/nova.conf #跟controller节点的配置一样，改如下几行**

**[DEFAULT]**

**修改my\_ip = 192.168.107.102**

**[api\_database]**

**注释connection = mysql+pymysql://nova:NOVA\_DBPASS@controller/nova\_api**

**[database]**

**注释connection = mysql+pymysql://nova:NOVA\_DBPASS@controller/nova**

**[vnc]**

**修改vncserver\_listen = 0.0.0.0**

**增加novncproxy\_base\_url = http://controller:6080/vnc\_auto.html**

**3、启动计算服务包括依赖项和设置开机启动**

**（1）确定计算节点支持虚拟机的硬件加速:**

**[root@compute ~]# egrep -c '(vmx|svm)' /proc/cpuinfo**

**#返回值>=1,那么你的计算节点支持硬件加速且不需要额外的配置，否则在/etc/nova/nova.conf 文件的 [libvirt] 区域做出如下的编辑：**

**[libvirt]**

**virt\_type = qemu**

**（2）启动计算服务包括依赖项和开机自动启动：**

**[root@compute ~]# systemctl enable libvirtd.service openstack-nova-compute.service**

**[root@compute ~]# ^enable^start**

**4、查看日志（启动服务不成功时通过日志来排查错误）**

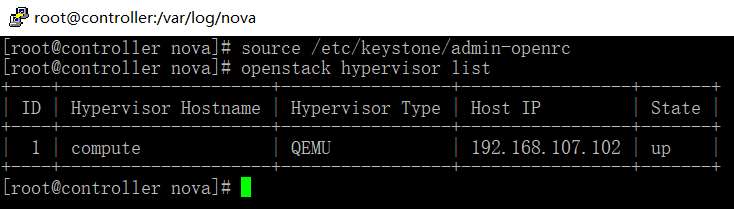
**[root@compute ~]# tail -200f /var/log/nova/nova-compute.log**

**controller节点上的配置：**

**1、admin登录openstack来获取访问权限，并确认1个计算节点是在主机列表中：**

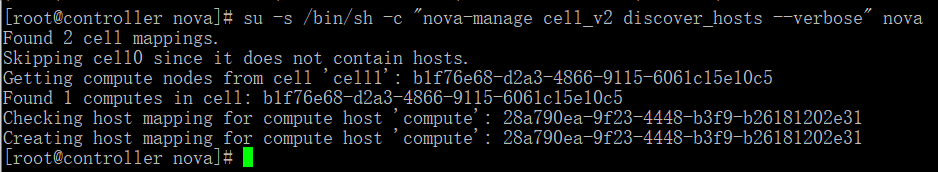
**[root@controller ~]# source /etc/keystone/admin-openrc**

**[root@controller ~]# openstack hypervisor list**



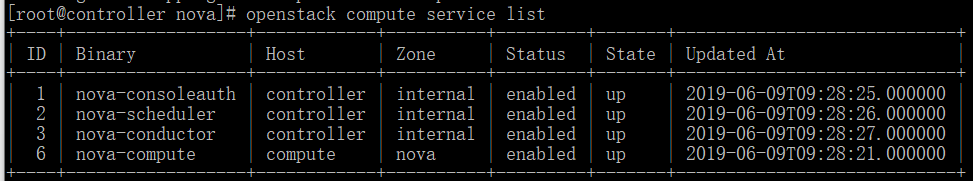
**2、发现compute主机列表（添加新的计算节点时，必须在控制器节点上注册新计算节点）：**

**[root@controller ~]# su -s /bin/sh -c "nova-manage cell\_v2 discover\_hosts --verbose" nova**

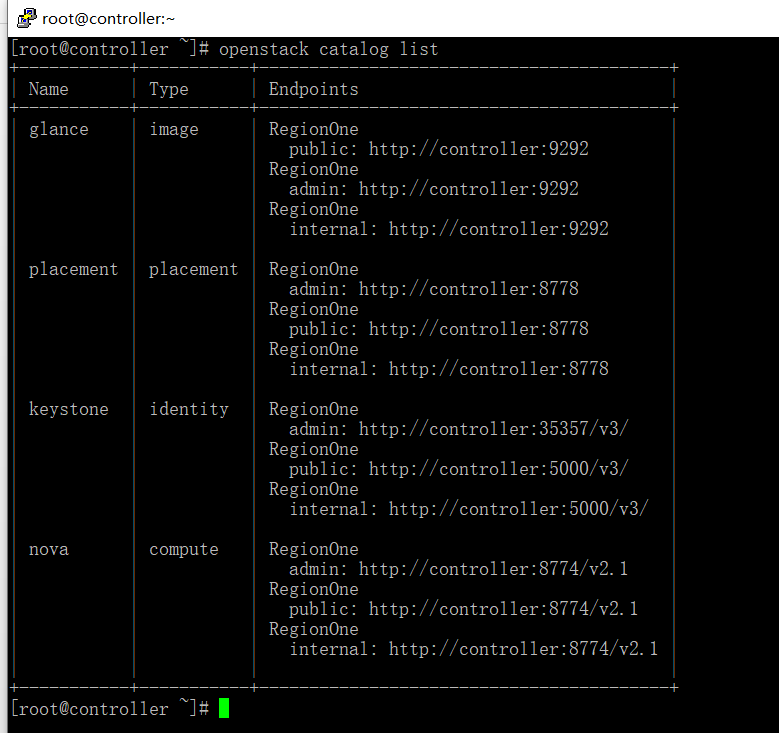


**3、验证：**

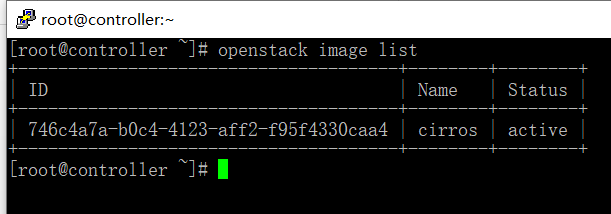
**[root@controller ~]# openstack compute service list**



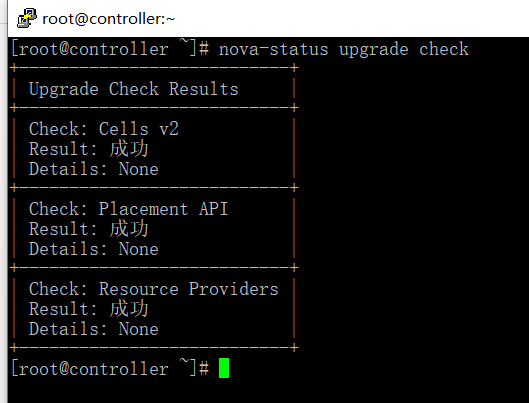
**[root@controller ~]# openstack catalog list**



**[root@controller ~]# openstack image list**



**[root@controller ~]# nova-status upgrade check**



**五、Neutron网络服务**

**controller节点 & compute节点**

**controller节点上的配置：**

**1、mysql里建库建用户并授权**

**[root@controller ~]# mysql -uroot -p**

**MariaDB [(none)]> create database neutron;**

**MariaDB [(none)]> grant all privileges on neutron.\* to neutron@'localhost' identified by 'NEUTRON\_DBPASS';**

**MariaDB [(none)]> grant all privileges on neutron.\* to neutron@'%' identified by 'NEUTRON\_DBPASS';**

**2、admin登录keystone给neutron服务注册**

**[root@controller ~]# source /etc/keystone/admin-openrc**

**（1）用户认证部分（建用户、建关联）**

**[root@controller ~]# openstack user create --domain default \**

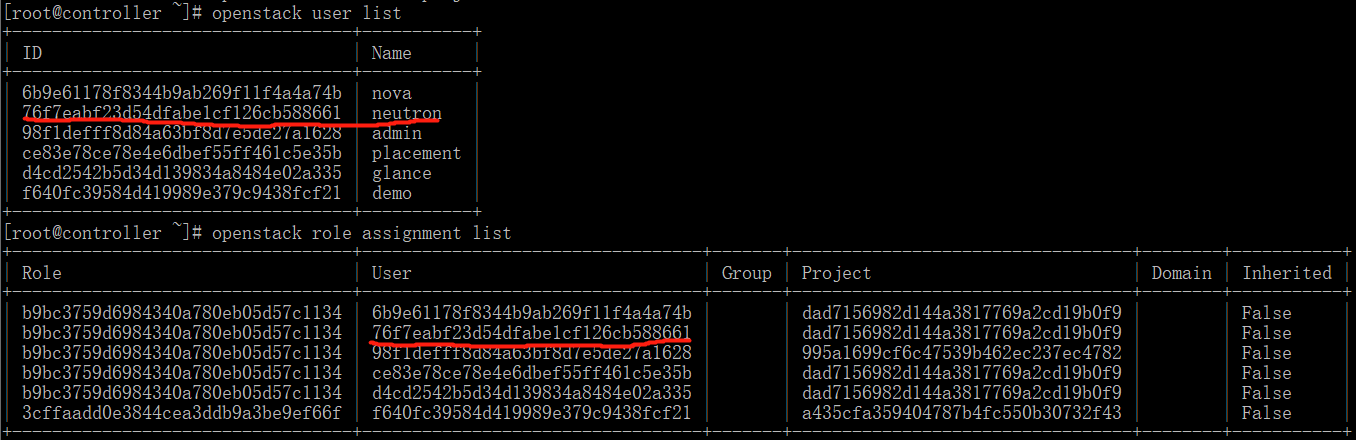
**--password-prompt neutron #设置NEUTRON\_PASS密码**

**[root@controller ~]# openstack role add --project service --user neutron admin**

**验证：**

**[root@controller ~]# openstack user list #用户列表**

**[root@controller ~]# openstack role assignment list #关联列表**



**（2）服务目录部分（建一个服务带三个端点）**

**[root@controller ~]# openstack service create --name neutron \**

**--description "OpenStack Networking" network**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

**network public http://controller:9696**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

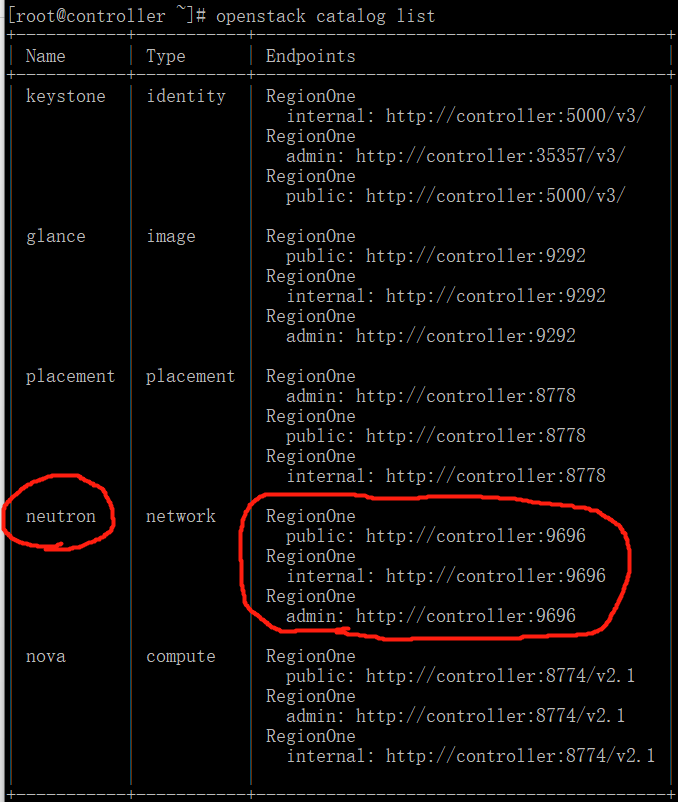
**network internal http://controller:9696**

**[root@controller ~]# openstack endpoint create --region RegionOne \**

**network admin http://controller:9696**

**验证：**

**[root@controller ~]# openstack catalog list #服务实体列表**



**3、安装neutron组件并进行配置**

**（1）安装组件**

**[root@controller ~]# yum install -y openstack-neutron openstack-neutron-ml2 \**

**openstack-neutron-linuxbridge ebtables**

**（2）编辑/etc/neutron/neutron.conf**

**[root@controller ~]# vim /etc/neutron/neutron.conf**

**[DEFAULT]**

**core\_plugin = ml2 #启用ML2插件并禁用其他插件**

**service\_plugins =**

**transport\_url = rabbit://openstack:RABBIT\_PASS@controller**

**auth\_strategy = keystone**

**notify\_nova\_on\_port\_status\_changes = true**

**notify\_nova\_on\_port\_data\_changes = true**

**[database]**

**connection = mysql+pymysql://neutron:NEUTRON\_DBPASS@controller/neutron**

**[keystone\_authtoken]**

**auth\_uri = http://controller:5000**

**auth\_url = http://controller:35357**

**memcached\_servers = controller:11211**

**auth\_type = password**

**project\_domain\_name = default**

**user\_domain\_name = default**

**project\_name = service**

**username = neutron**

**password = NEUTRON\_PASS**

**[nova]**

**auth\_url = http://controller:35357**

**auth\_type = password**

**project\_domain\_name = default**

**user\_domain\_name = default**

**region\_name = RegionOne**

**project\_name = service**

**username = nova**

**password = NOVA\_PASS**

**[oslo\_concurrency]**

**lock\_path = /var/lib/neutron/tmp**

**（3）配置模块化层2（ML2）插件（/etc/neutron/plugins/ml2/ml2\_conf.ini）**

**[root@controller ~]# vim /etc/neutron/plugins/ml2/ml2\_conf.ini**

**[ml2]**

**type\_drivers = flat,vlan**

**tenant\_network\_types =**

**mechanism\_drivers = linuxbridge #启用Linuxbridge机制**

**extension\_drivers = port\_security #启用端口安全扩展驱动**

**[ml2\_type\_flat]**

**flat\_networks = provider**

**[securitygroup]**

**enable\_ipset = true #启用 ipset 增加安全组的方便性**

**（4）配置Linux网桥代理（/etc/neutron/plugins/ml2/linuxbridge\_agent.ini）**

**[root@controller ~]# vim /etc/neutron/plugins/ml2/linuxbridge\_agent.ini**

**[linux\_bridge]**

**physical\_interface\_mappings = provider:ens34**

**[vxlan]**

**enable\_vxlan = false #禁止VXLAN覆盖网络**

**[securitygroup]**

**enable\_security\_group = true**

**firewall\_driver = neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver**

**#启用安全组并配置 Linux 桥接 iptables 防火墙驱动**

**（5）配置DHCP代理（/etc/neutron/dhcp\_agent.ini）**

**[root@controller ~]# vim /etc/neutron/dhcp\_agent.ini**

**[DEFAULT]**

**interface\_driver = linuxbridge**

**dhcp\_driver = neutron.agent.linux.dhcp.Dnsmasq**

**enable\_isolated\_metadata = true**

**（6）配置元数据代理（/etc/neutron/metadata\_agent.ini）**

**[root@controller ~]# vim /etc/neutron/metadata\_agent.ini**

**[DEFAULT]**

**nova\_metadata\_ip = controller**

**metadata\_proxy\_shared\_secret = METADATA\_SECRET**

**（7）配置计算服务以使用网络服务（/etc/nova/nova.conf）**

**[root@controller ~]# vim /etc/nova/nova.conf**

**[neutron]**

**url = http://controller:9696**

**auth\_url = http://controller:35357**

**auth\_type = password**

**project\_domain\_name = default**

**user\_domain\_name = default**

**region\_name = RegionOne**

**project\_name = service**

**username = neutron**

**password = NEUTRON\_PASS**

**service\_metadata\_proxy = true**

**metadata\_proxy\_shared\_secret = METADATA\_SECRET**

**4、完成安装，启动服务**

**创建扩展链接:**

**[root@controller ~]# ln -s /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini**

**同步数据库：**

**[root@controller ~]# su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf \**

**--config-file /etc/neutron/plugins/ml2/ml2\_conf.ini upgrade head" neutron**

**查看neutron库所有表：**

**[root@controller ~]# mysql -uneutron -pNEUTRON\_DBPASS -e "use neutron;show tables;"**

**启动服务并设置开机启动：**

**[root@controller ~]# systemctl restart openstack-nova-api.service**

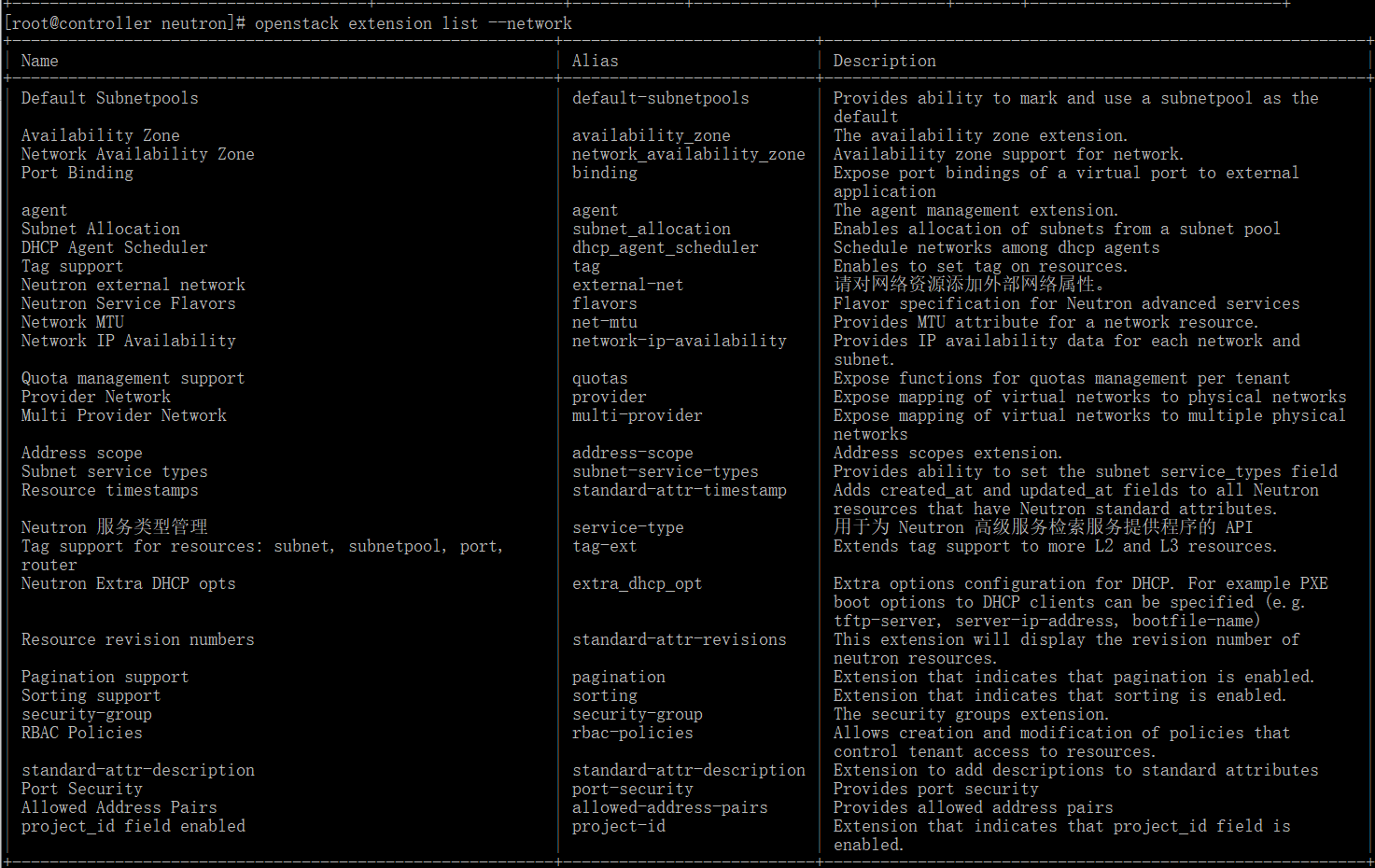
**[root@controller ~]# systemctl start neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service**

**[root@controller ~]# ^start^enable**

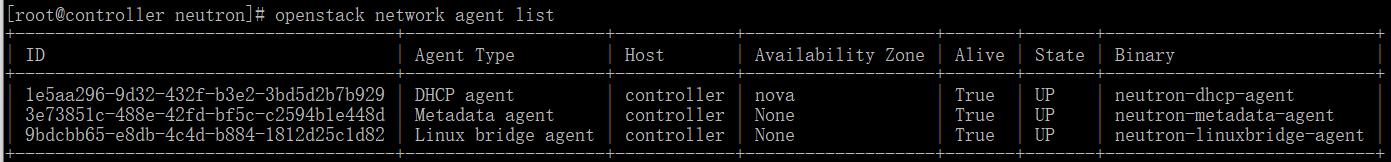
**5、验证**

**[root@controller ~]# source /etc/keystone/admin-openrc**

**[root@controller ~]# openstack extension list --network**



**[root@controller ~]# openstack network agent list**



**compute节点上的配置**

**1、安装neutron组件并进行配置**

**（1）安装组件**

**[root@compute ~]# yum install -y openstack-neutron-linuxbridge ebtables ipset**

**（2）编辑/etc/neutron/neutron.conf**

**[root@compute ~]# vim /etc/neutron/neutron.conf**

**[DEFAULT]**

**transport\_url = rabbit://openstack:RABBIT\_PASS@controller**

**auth\_strategy = keystone**

**[keystone\_authtoken]**

**auth\_uri = http://controller:5000**

**auth\_url = http://controller:35357**

**memcached\_servers = controller:11211**

**auth\_type = password**

**project\_domain\_name = default**

**user\_domain\_name = default**

**project\_name = service**

**username = neutron**

**password = NEUTRON\_PASS**

**[oslo\_concurrency]**

**lock\_path = /var/lib/neutron/tmp**

**（3）配置Linux网桥代理（/etc/neutron/plugins/ml2/linuxbridge\_agent.ini）**

**[root@compute ~]# vim /etc/neutron/plugins/ml2/linuxbridge\_agent.ini**

**[linux\_bridge]**

**physical\_interface\_mappings = provider:ens34**

**[vxlan]**

**enable\_vxlan = false**

**[securitygroup]**

**enable\_security\_group = true**

**firewall\_driver = neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver**

**（4）配置计算服务以使用网络服务（/etc/nova/nova.conf）**

**[root@compute ~]# vim /etc/nova/nova.conf**

**[neutron]**

**url = http://controller:9696**

**auth\_url = http://controller:35357**

**auth\_type = password**

**project\_domain\_name = default**

**user\_domain\_name = default**

**region\_name = RegionOne**

**project\_name = service**

**username = neutron**

**password = NEUTRON\_PASS**

**2、启动服务**

**[root@compute ~]# systemctl restart openstack-nova-compute.service**

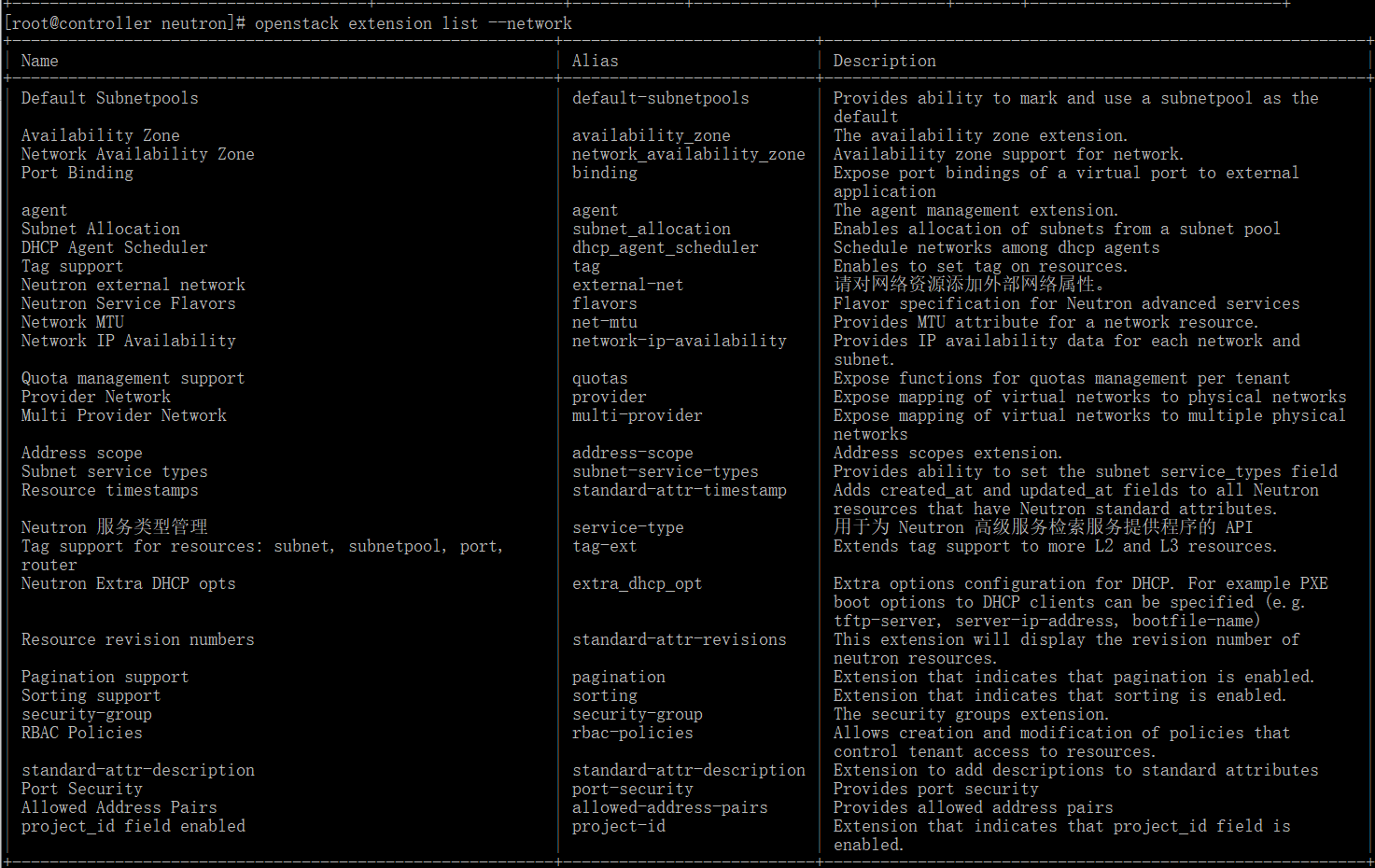
**[root@compute ~]# systemctl start neutron-linuxbridge-agent.service**

**[root@compute ~]# ^start^enable**

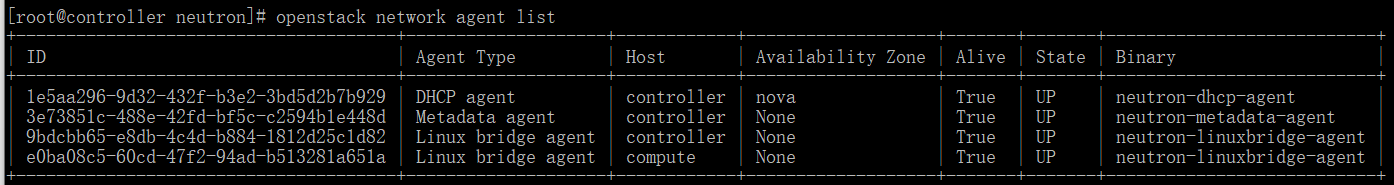
**controller节点**

**验证：**

**[root@controller ~]# openstack extension list --network**



**[root@controller ~]# openstack network agent list**



**六、Dashboard界面服务**

**controller节点**

**[root@controller ~]# yum install -y openstack-dashboard**

**[root@controller ~]# vim /etc/openstack-dashboard/local\_settings**

**OPENSTACK\_HOST = "controller" #keystone服务IP**

**OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v3" % OPENSTACK\_HOST #启用第3版认证API#**

**OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "user" #通过仪表盘创建的用户默认角色user**

**OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = 'Default' #创建普通用户是默认域**

**ALLOWED\_HOSTS = ['\*',]**

**SESSION\_ENGINE = 'django.contrib.sessions.backends.cache' #配置 memcached 会话存储服务**

**CACHES = {**

**'default': {**

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

**'LOCATION': 'controller:11211',**

**},**

**}**

**OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True #启用对域的支持#**

**OPENSTACK\_API\_VERSIONS = { #配置API版本#**

**"identity": 3,**

**"image": 2,**

**"volume": 2,**

**}**

**OPENSTACK\_NEUTRON\_NETWORK = { #禁用支持3层网络服务#**

**'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,**

**}**

**TIME\_ZONE = "Asia/Shanghai" #配置时区#**

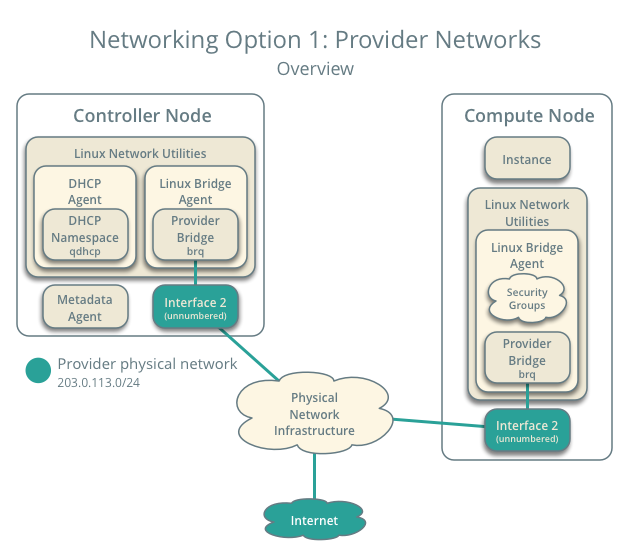
**[root@controller ~]# systemctl restart httpd.service memcached.service**

**浏览器输入**

**http://192.168.65.128/dashboard**



**七、启动实例（期末考核）**



**只提供网络搭建的解决方案：**

**1、首先部署neutron网络服务的时候，我们采用的是最简单的方式，提供商网络（Provider network），所以看文档操作时注意选择正确方案。**

**2、特别注意的是，创建实例时，controller节点、compute节点使用的是外网IP和创建出来的实例虚拟机互相通信，意思是两节点外网IP需要互ping，需NAT模式改桥接模式，IP设置建议瞎设一个比较稀有的网段，如20.5.5.0/24；改完后如何使用连接工具，Windows的IP地址设置跟它们一个段即可。**

**3、按照官方教程搭建出来的实例虚拟机网络，叫做扁平网络（flat）,特点就是和controller、compute节点处于同一局域网，即IP段相同，所以配置实例网络时IP信息怎么配，自己注意。**

**4、具体操作自己参考教程，尝试并验证，估计还会有那么点小bug，自行百度解决。**

**官方部署文档链接：**

[https://docs.openstack.org/ocata/install-guide-rdo/launch-instance.html#](https://docs.openstack.org/ocata/install-guide-rdo/launch-instance.html)

**验证实例是否启动成功：**

