# （一）安装Keystone认证服务

## 1.创建keystone数据库并授权

登录root用户

mysql -uroot

创建数据库keystone

CREATE DATABASE keystone;

给keystone数据库的所有表赋予权限，keystone用户在本地命令行或远程登录模式下有所有权限，注意密码是‘KEYSTONE\_DBPASS’

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY 'KEYSTONE\_DBPASS';

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY 'KEYSTONE\_DBPASS';

## 2. 安装keystone软件包

keystone软件包名是openstack-keystone，安装httpd和mod\_wsgi的原因是，社区主推的组合是apache+keystone。

openstack-keystone本质就是一款基于wsgi协议的web app,而httpd本质就是一个兼容wsgi协议的web server，所以我们需要为httpd安装mod\_wsgi模块

[root@controller ~]# yum install openstack-keystone httpd mod\_wsgi

## 3. 修改配置文件

复制keystone的配置文件到.bak文件中

cp -a /etc/keystone/keystone.conf{,.bak}

从.bak文件中搜索出相应的配置信息到conf文件中，精简conf配置文件

grep -Ev "^$|#" /etc/keystone/keystone.conf.bak > /etc/keystone/keystone.conf

安装openstack-utils工具

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

让openstack-keystone能够知道如何连接到后端的数据库keystone；

mysql+pymysql：pymysql是一个python库，使用python可以操作mysql原生sql；

在keystone配置文件中设置keystone数据库连接的密码（具体可以查询keystone.conf配置文件），其实就是修改keystone.conf配置文件

格式： 用户名:密码@mysql地址/哪个库

注意在mysql数据库中创建密码的时候加上了引号“123”而在配置文件中不要加上引号，不然会保错

注意，地址写IP地址，不要写controller

[root@controller ~]# openstack-config --set /etc/keystone/keystone.conf database connection mysql+pymysql://keystone:KEYSTONE\_DBPASS@192.168.112.146/keystone

在keystone配置文件中设置token令牌

[root@controller ~]# openstack-config --set /etc/keystone/keystone.conf token provider fernet

 可能有的时候大家在部署的时候会遇到报错，怎么办，各位遇到了不要慌，在log日志中查看报错的问题是什么。

|  |  |
| --- | --- |
| 1 | tail -f -n 20 /var/log/keystone/keystone.log |

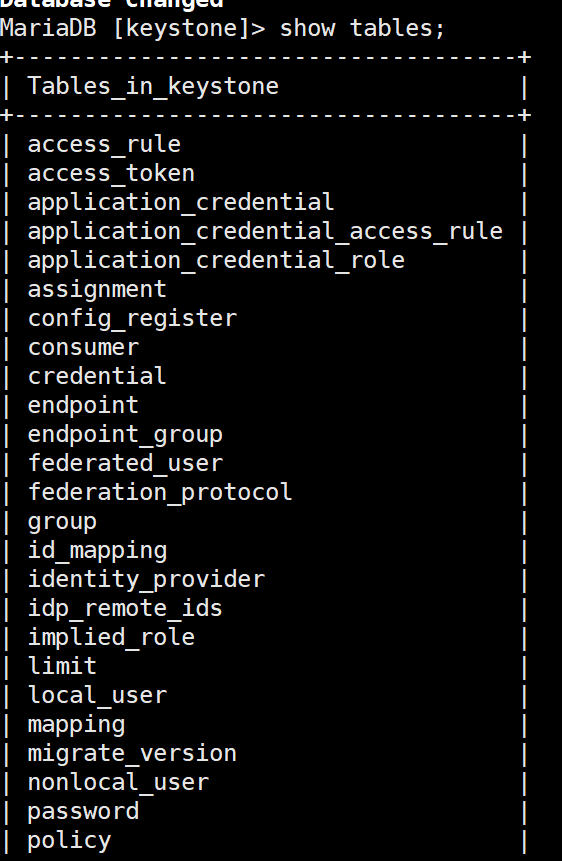
## 4. 填充数据库

su命令即作为root执行命令，-s 表示运行shell，-c表示运行shell命令，最后的keystone表示keystone用户

这条命令的作用就是初始化数据库，之所以要初始化，是因为python的orm对象关系映射，需要初始化来生成数据库表结构。什么是orm（Object Relation Mapping）的缩写叫对象关系映射。

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

在这一步之后，去keystone数据库查看，如果数据库初始化成功会生成很多数据表：



初始化Fernet密钥存储库 这是新版本的OpenStack的新功能，在Train版本下，keystone不再使用简单的字符串作为临时token，而是使用下面创建的fernet的用户来运行keystone。同时，keystone也不再对管理员用户和普通用户的服务端点区分使用不同的端口5000和35357，而是只使用5000端口不再使用35357端口。

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

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

上面的两个步骤是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

## 5.修改apache配置

修改主机名，方便操作

hostnamectl set-hostname controller

配置hosts

192.168.112.146 controller

修改httpd服务器配置

[root@controller ~]# echo "ServerName controller" >> /etc/httpd/conf/httpd.conf

创建wsgi配置文件软链接 ，为mod\_wsgi模块添加配置文件，直接拷贝模块文件或者做软连接都可以，

wsgi：apache关于mod\_wsgi模块的配置文件

keystone：用来连接keystone

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

wsgi-keystone.conf文件中的500端口说明keystone要开放这个端口，apache才能把请求交给keystone

启动和开机自启动apache

[root@controller ~]# systemctl enable httpd.service

[root@controller ~]# systemctl start httpd.service

## 6本质就是在为keystone创建catalog

注意，将controller都修改为IP地址

接下来就是初始化keystone的操作，**官方最新给出了一个框架叫做Bootsrap能够直接完成初始化操作**：

[root@controller ~]# keystone-manage bootstrap --bootstrap-password ADMIN\_PASS \ #设置keystone对自己进行认证的密码

--bootstrap-admin-url http://192.168.112.146:5000/v3/ \ #admin用户访问的端点

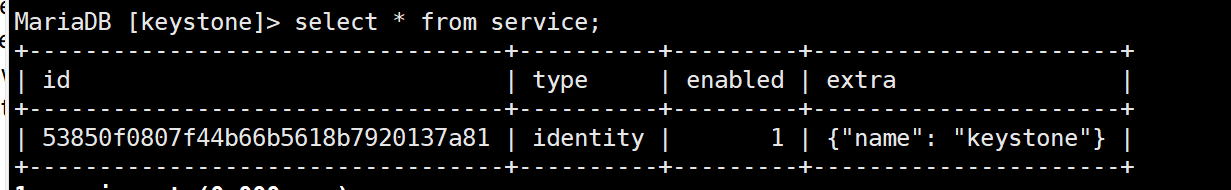
--bootstrap-internal-url http://192.168.112.146:5000/v3/ \ #内部访问的端点

--bootstrap-public-url http://192.168.112.146:5000/v3/ \ #外部访问的端点

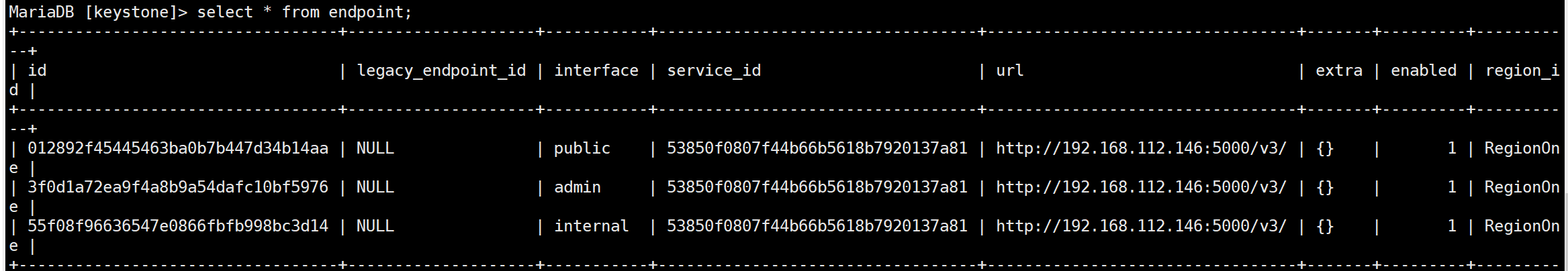
--bootstrap-region-id RegionOne

为keystone创建catalog，基于前两步给的权限，创建认证服务实体，基于建立的服务实体，创建访问该实体的三个api端点：public, internal, admin三个端点，用于外部，内部以及admin用户访问。

可以从keystone数据库中查看当前keystone的各种信息，如当前的Keystone service服务ID：



也可以查询出当前keystone的所有endpoint端点：



可以看见上面三个端点的service\_id一样，且和之前我们查询出来的一样，则说明他们是属于一个服务的Endpoint 。

## 7.初始化环境变量

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

创建一个默认的域“default” ；

创建admin项目 在“default”域中；

创建admin用户 在“default”域中；

创建admin角色；

创建上述三者的关联，不关联用户无法使用

同样，要将controller转换为IP地址

配置当前的环境变量：

[root@controller ~]# cat >> ~/.bashrc << EOF

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS #就是keystone-manage中设定的--bootstrap-password

export OS\_PROJECT\_NAME=admin #项目名　project <-- 用户 <--角色 若想让用户获取权限必须要指定用户所在的项目是哪个

export OS\_USER\_DOMAIN\_NAME=Default #默认域

export OS\_PROJECT\_DOMAIN\_NAME=Default #项目所在的域

export OS\_AUTH\_URL=http://192.168.112.146:5000/v3 #认证url

export OS\_IDENTITY\_API\_VERSION=3 #指定版本信息为3

EOF

source命令用法

source FileName

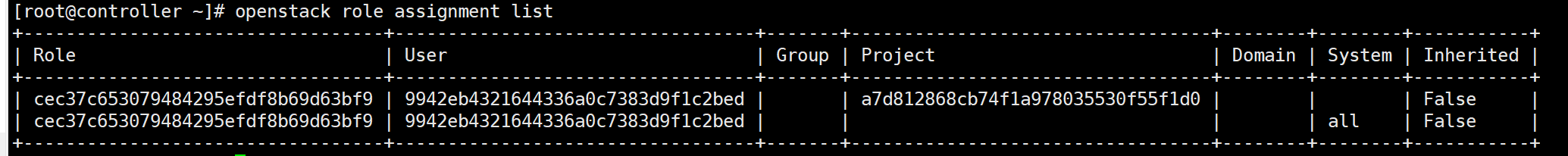
source命令作用

在当前bash环境下读取并执行FileName中的命令。

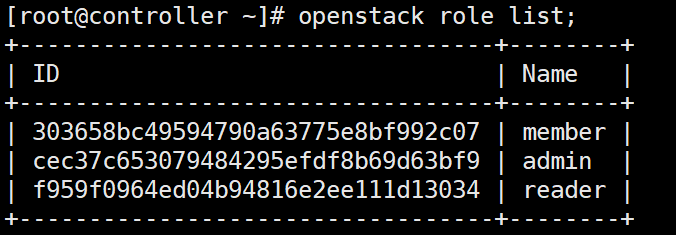
\*注：该命令通常用命令“.”来替代。

[root@controller ~]# source ~/.bashrc #读取刚刚更新的文件

验证初始化操作成功：

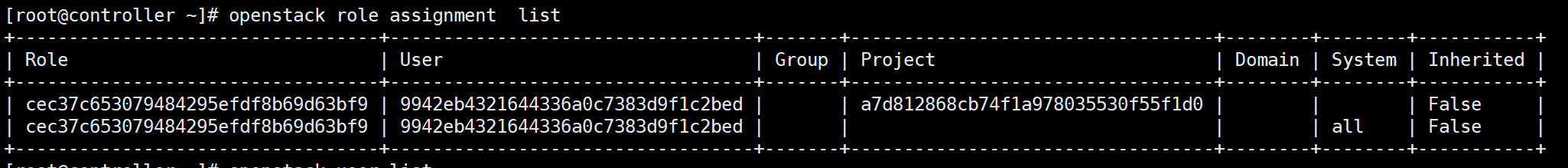


查询所有的角色role：



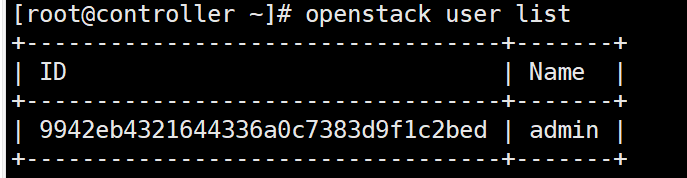
可以看到，当前有三种角色。

查询当前的角色role分配：



可以看到，当前只有admin用户进行了分配，分配给了用户尾号为1c2bed的用户。

查询当前所有的用户user：



可以看到，与上面查询role分配是匹配的，只有一个尾号为1c2bed的用户admin。

这些数据都是从keystone数据库中查询出来的，对应着keystone数据库中的多个表，分别是role表，user表，project表等。

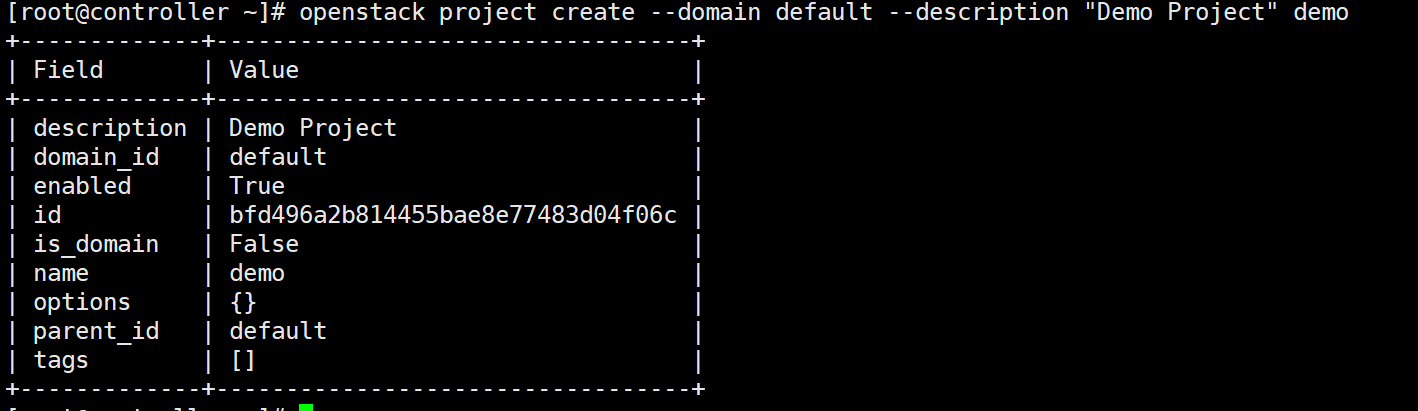
## 8.创建测试用项目，角色，用户

创建用于后期测试用的项目，用户，租户，建立关联

**创建项目名为demo**

openstack project create --domain default \

--description "Demo Project" demo

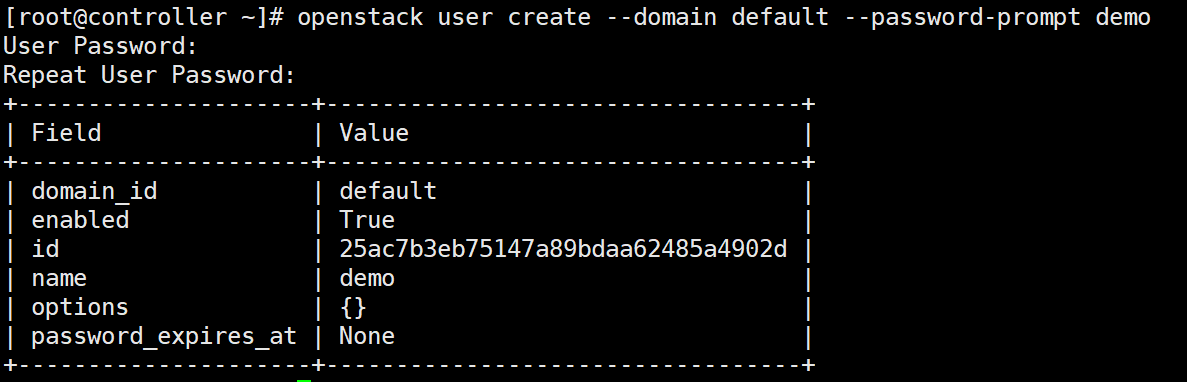


**创建普通用户为demo 设置密码为123**

openstack user create --domain default \

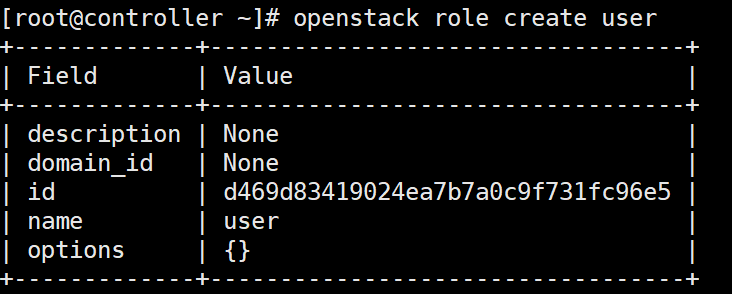
--password-prompt demo

**设置密码为123**

****

**创建普通用户的角色user**

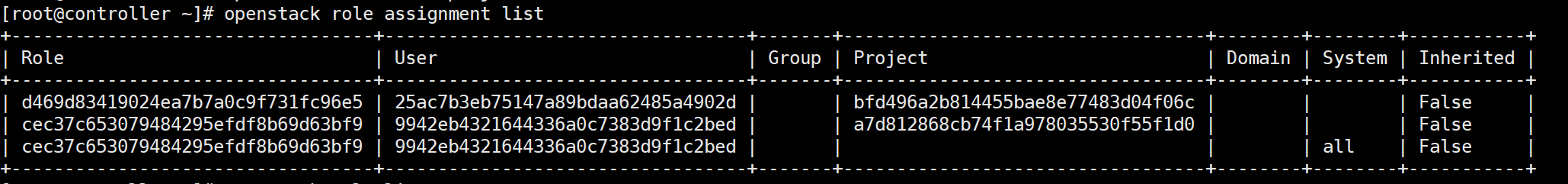
**openstack role create user**

****

**建立 项目demo，用户demo，角色user的关联：**

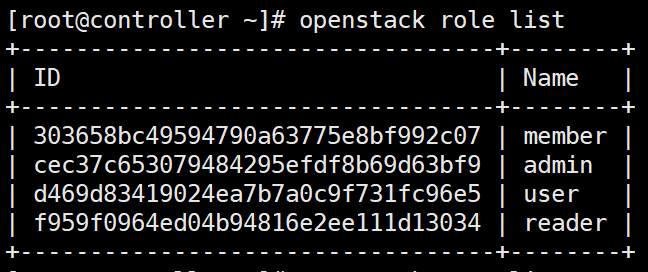
**openstack role add --project demo --user demo user**

**查询角色的分派情况：**

****

**可以看到多了一个角色分派，角色尾号为fc96e5，用户尾号为a4902d；**

**继续查询角色list：**

****

**可以看到，user角色的尾号为fc96e5，因此就是user角色的分派。**

注意，每次创建一个用户的过程都是这几个步骤。但是会少了一步创建角色的过程。因为用户就两个，一个是admin，另一个就是user（一个项目里可以有很多admin角色和user角色）注意.角色一定是在项目中。

**9.创建后续组件测试的项目service**

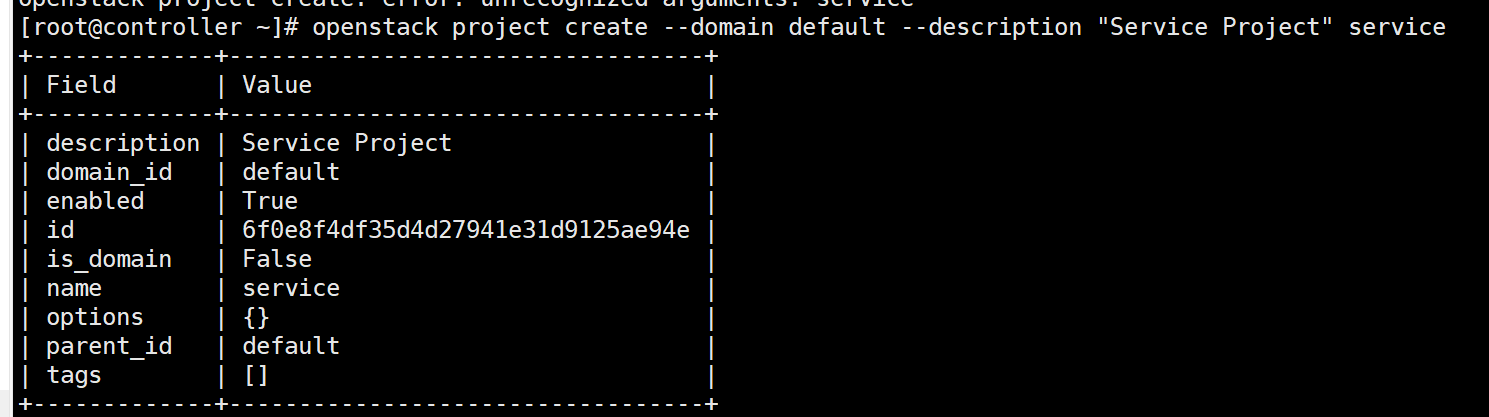
因为keystone是一个独立的授权组件，后面每搭建一个新的服务都需要在keystone中执行四种操作：1.建项目 2.建用户 3.建角色 4.做关联 并且创建catalog

这里将后面所有的服务（nova。glace等等）公用一个项目service，都是管理员角色admin（组件之间的通信都角色都是admin）。

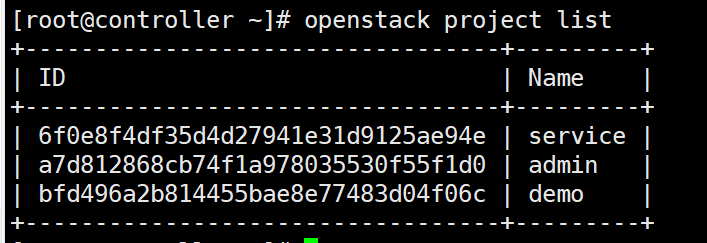
每一个组件，都要有一个访问用户，比如访问glance要有一个glance用户，还有一个角色，还要有一个关联。以后要部署glance，要在keystone中创建一个glance的catalog（service名/Endpoint）。还要有一个访问catalog的用户信息，这样的话还要创建一个域。但是现在已经有了一个默认的域 所以不需要再创建，域有了还要创建一个项目，但是现在项目也已经有了，service项目，所以不用创建了。只需要创建一个叫glance的用户就行了（一般组件叫什么名，用户就叫什么名）。还要创建一个角色，但是现在也不用创建，之前就有一个admin角色，使用admin用户就行。最后就只要做一个关联，以后每个组件都要这么做，就是说学会了keystone的部署，后面的组件也就部署好了

创建项目service：

openstack project create --domain default --description “Service Project” service



**查看当前的项目：**

****

## 9.为各种用户申请令牌token

管理员用户admin申请token，这里不需要加上admin用户的密码ADMIN\_PASS

**openstack --os-auth-url http://192.168.112.146:5000/v3 \**

**--os-identity-api-version 3 \**

**--os-project-domain-name default \**

**--os-user-domain-name default \**

**--os-project-name admin \**

**--os-username admin \**

**token issue**

****

**项目是admin项目；**

**用户是admin；**

**令牌token的ID就是生成的；**

**为普通用户demo申请token，注意这里我们之前为demo用户配置了密码123，因此需要加上配置项：**

**openstack --os-auth-url http://192.168.112.146:5000/v3 \**

**--os-identity-api-version 3 \**

**--os-project-domain-name default \**

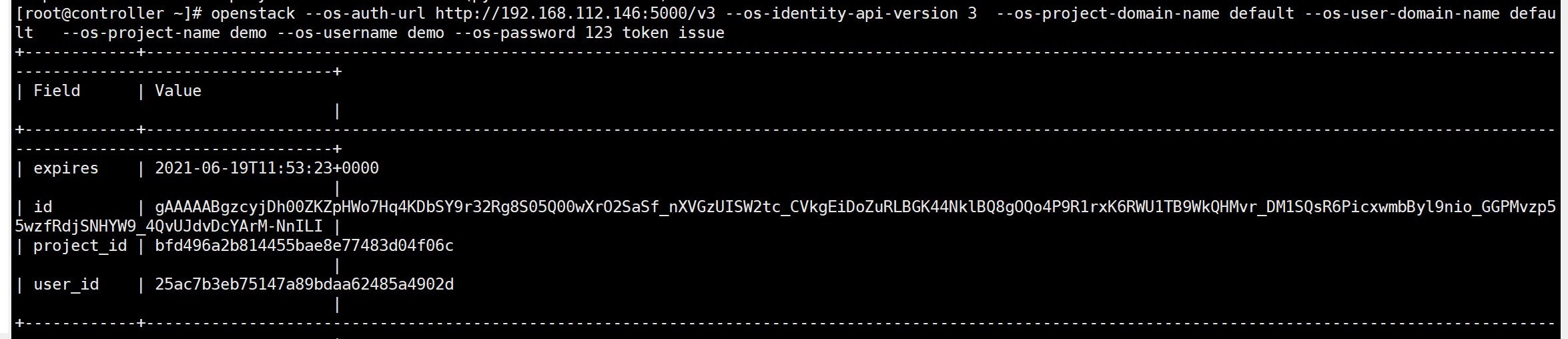
**--os-user-domain-name default \**

**--os-project-name demo \**

**--os-username demo \**

**--os-password 123 \**

**token issue**

****

**10.为不同用户定义用户信息脚本**

Linux export 命令用于**设置或显示环境变量**。

在 shell 中执行程序时，**shell 会提供一组环境变量。export 可新增，修改或删除环境变量，供后续执行的程序使用**。**export 的效力仅限于该次登陆操作**。

**定义好用户信息脚本之后，将对应用户执行对应用户的信息脚本，则该次登录的环境信息就是用户信息脚本中的信息，在执行openstack token issue时就可以直接调用了，不需要传入上面那么多的参数了。**

**对于admin用户，在root用户目录下/root创建admin.sh脚本文件：**

**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://192.168.112.146:5000/v3**

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

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

**对于demo用户，在root用户目录/root中创建demo.sh脚本：**

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

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

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

**export OS\_USERNAME=demo**

**export OS\_PASSWORD=123**

**export OS\_AUTH\_URL=http://192.168.112.146:5000/v3**

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

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

针对不同的业务应该有不同的用户信息，也都应该定义成脚本形式，方便管理