

## 02.Keystone 认证服务组件

参考任务时间： 80min

(因 CentOS、Openstack 版本更新的缘故，教材中部分配置不再适用)

系统环境准备 CentOS7 + OpenStack Rocky

官方文档：

<https://docs.openstack.org/keystone/rocky/install/keystone-install-rdo.html>

<https://docs.openstack.org/keystone/rocky/install/keystone-users-rdo.html>

<https://docs.openstack.org/keystone/rocky/install/keystone-verify-rdo.html>

<https://docs.openstack.org/keystone/rocky/install/keystone-openrc-rdo.html>

### 2.0. keystone 认证服务

#### 2.1.在控制节点创建 keystone 相关数据库

1) 创建 keystone 数据库并授权

#### 2.2.在控制节点安装 keystone 相关软件包

1) 安装 keystone 相关软件包

2) 快速修改 keystone 配置

#### 2.3.初始化同步 keystone 数据库

1) 同步 keystone 数据库

2) 同步完成进行连接测试

#### 2.4.初始化 Fernet 令牌库

#### 2.5.配置启动 Apache (httpd)

1) 修改 httpd 主配置文件

2) 配置虚拟主机

3) 启动 httpd 并配置开机自启动

#### 2.6.初始化 keystone 认证服务

- 1) 创建 **keystone** 用户,初始化的服务实体和 **API** 端点
- 2) 临时配置管理员账户的相关变量进行管理

## 2.7.创建 **keystone** 的一般实例

- 1) 创建一个名为 **example** 的 **keystone** 域
- 2) 为 **keystone** 系统环境创建名为 **service** 的项目提供服务
- 3) 创建 **myproject** 项目和对应的用户及角色
- 4) 在默认域创建 **myuser** 用户
- 5) 在 **role** 表创建 **myrole** 角色
- 6) 将 **myrole** 角色添加到 **myproject** 项目中和 **myuser** 用户组中

## 2.8.验证操作 **keystone** 是否安装成功

- 1) 去除环境变量
- 2) 作为管理员用户去请求一个认证的 **token**
- 3) 使用普通用户获取认证 **token**

## 2.9.创建 **OpenStack** 客户端环境脚本

- 1) 创建 **admin** 用户的环境管理脚本
- 2) 创建普通用户 **myuser** 的客户端环境变量脚本
- 3) 测试环境管理脚本
- 4) 请求认证令牌

## 2.0.keystone 认证服务基本概念

### 1) 用户与认证:

User	用户
Tenant	租户
Token	令牌
Role	角色

### 2) 服务目录: 提供服务目录, 包括所有服务项与相关 API 的端点

Service	服务
Endpoint	端点

## 2.1.在控制节点创建 keystone 相关数据库

### 1) 创建 keystone 数据库并授权

```
mysql -p123456
```

在数据库 MariaDB [(none)]> 提示符后执行如下命令:

-----

```
CREATE DATABASE keystone;
```

```
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY 'keystone';
```

```
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'keystone';
```

```
flush privileges;
```

```
show databases;
```

```
select user,host from mysql.user;
```

```
exit
```

-----

## 2.2.在控制节点安装 keystone 相关软件包

### 1) 安装 keystone 软件包

配置 Apache 服务, 使用带有“mod\_wsgi”的 HTTP 服务器来相应认证服务请求, 端口为 5000。

```
yum install openstack-keystone httpd mod_wsgi -y
```

```
yum install openstack-keystone python-keystoneclient openstack-utils -y
```

## 2) 快速修改 keystone 配置

下面使用的快速配置方法需要安装 Openstack-utils

```
openstack-config --set /etc/keystone/keystone.conf database connection
```

```
mysql+pymysql://keystone:keystone@controller/keystone
```

```
openstack-config --set /etc/keystone/keystone.conf token provider fernet
```

其他方式查看生效配置

```
grep '^[\a-z]' /etc/keystone/keystone.conf
```

例:

```
[root@openstack01 tools]# grep '^[\a-z]' /etc/keystone/keystone.conf
```

```
connection = mysql+pymysql://keystone:keystone@controller/keystone
```

```
provider = fernet
```

## 2.3.初始化同步 keystone 数据库

### 1) 同步 keystone 数据库

```
su -s /bin/sh -c "keystone-manage db_sync" keystone
```

### 2) 同步后进行测试

确定需要的表已经建立

```
mysql -h127.0.0.1 -ukeystone -pkeystone -e "use keystone;show tables;"
```

例:

```
[root@openstack01 ~]# mysql -h127.0.0.1 -ukeystone -pkeystone -e "use keystone;show tables;"
```

```
+-----+
```

```
| Tables_in_keystone |
```

```
+-----+
```

```
| access_token |
```

```
| application_credential |
```

application_credential_role	
assignment	
config_register	
consumer	
credential	
endpoint	
endpoint_group	
federated_user	
federation_protocol	
group	
id_mapping	
identity_provider	
idp_remote_ids	
implied_role	
limit	
local_user	
mapping	
migrate_version	
nonlocal_user	
password	
policy	
policy_association	
project	
project_endpoint	
project_endpoint_group	
project_tag	
region	
registered_limit	
request_token	
revocation_event	

role	
sensitive_config	
service	
service_provider	
system_assignment	
token	
trust	
trust_role	
user	
user_group_membership	
user_option	
whitelisted_config	
+-----+	

```
[root@openstack01 ~]# mysql -h192.168.56.126 -ukeystone -pkeystone -e "use
keystone;show tables;" | wc -l
```

45

## 2.4.初始化 Fernet 令牌库

注：以下命令执行后没有反馈信息

```
keystone-manage fernet_setup --keystone-user keystone --keystone-group
keystone
```

```
keystone-manage credential_setup --keystone-user keystone
--keystone-group keystone
```

## 2.5.配置启动 Apache

1) 修改 httpd 配置文件

```
vi /etc/httpd/conf/httpd.conf
```

-----

ServerName controller

---

注：controller 是上一部分实验设定的 hostname

## 2) 配置虚拟主机

创建 keystone 虚拟主机配置文件的快捷方式

```
ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/
```

或者可以手动编辑创建该文件

```
cat /usr/share/keystone/wsgi-keystone.conf
```

-----

```
[root@openstack01 ~]# cat /usr/share/keystone/wsgi-keystone.conf
```

```
Listen 5000
```

```
<VirtualHost *:5000>
```

```
    WSGIDaemonProcess keystone-public processes=5 threads=1 user=keystone
    group=keystone display-name=%{GROUP}
```

```
    WSGIProcessGroup keystone-public
```

```
    WSGIScriptAlias / /usr/bin/keystone-wsgi-public
```

```
    WSGIApplicationGroup %{GLOBAL}
```

```
    WSGIPassAuthorization On
```

```
    LimitRequestBody 114688
```

```
<IfVersion >= 2.4>
```

```
    ErrorLogFormat "%{cu}t %M"
```

```
</IfVersion>
```

```
    ErrorLog /var/log/httpd/keystone.log
```

```
    CustomLog /var/log/httpd/keystone_access.log combined
```

```
<Directory /usr/bin>
```

```
    <IfVersion >= 2.4>
```

```
        Require all granted
```

```
    </IfVersion>
```

```
    <IfVersion < 2.4>
```

```
        Order allow,deny
```

```

        Allow from all
    </IfVersion>

    </Directory>
</VirtualHost>

Alias /identity /usr/bin/keystone-wsgi-public
<Location /identity>
    SetHandler wsgi-script
    Options +ExecCGI

    WSGIProcessGroup keystone-public
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On
</Location>

```

-----

3) 启动 httpd 并配置开机自启动

```

systemctl start httpd.service
systemctl status httpd.service
netstat -anptl|grep httpd
systemctl enable httpd.service
systemctl list-unit-files |grep httpd.service

```

若 http 启动报错，可能需关闭 selinux ； 或安装 yum install openstack-selinux;  
或检查 80 端口占用情况等。

例:

```

[root@openstack01 ~]# systemctl start httpd.service
[root@openstack01 ~]# systemctl status httpd.service
httpd.service - The Apache HTTP Server

```



Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)

Active: active (running) since Thu 2019-02-14 12:40:39 CST; 37s ago

Docs: man:httpd(8)

man:apachectl(8)

Main PID: 15410 (httpd)

Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/sec"

CGroup: /system.slice/httpd.service

â15410 /usr/sbin/httpd -DFOREGROUND

â15411 (wsgi:keystone- -DFOREGROUND

â15412 (wsgi:keystone- -DFOREGROUND

â15413 (wsgi:keystone- -DFOREGROUND

â15414 (wsgi:keystone- -DFOREGROUND

â15415 (wsgi:keystone- -DFOREGROUND

â15416 /usr/sbin/httpd -DFOREGROUND

â15417 /usr/sbin/httpd -DFOREGROUND

â15418 /usr/sbin/httpd -DFOREGROUND

â15419 /usr/sbin/httpd -DFOREGROUND

â15420 /usr/sbin/httpd -DFOREGROUND

Feb 14 12:40:39 controller systemd[1]: Starting The Apache HTTP Server...

Feb 14 12:40:39 controller systemd[1]: Started The Apache HTTP Server.

[root@openstack01 ~]# netstat -anptl|grep httpd

tcp	0	0 0.0.0.0:5000	0.0.0.0:*	LISTEN
1978/httpd				

tcp	0	0 0.0.0.0:80	0.0.0.0:*	LISTEN
1978/httpd				

[root@openstack01 ~]# systemctl enable httpd.service

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

```
[root@openstack01 ~]# systemctl list-unit-files |grep httpd.service
```

```
httpd.service                                enabled
```

http 服务配置完

## 2.6.初始化 keystone 认证服务

### 1) 创建 keystone 用户, 初始化的服务实体和 API 端点

创建 keystone 服务实体和身份认证服务, 以下三种类型分别为公共的、内部的、管理的; 需要创建一个密码 ADMIN\_PASS, 作为登陆 openstack 的管理员用户, 这里创建为 123456 命令如下:

```
keystone-manage bootstrap --bootstrap-password 123456 \
--bootstrap-admin-url http://controller:5000/v3/ \
--bootstrap-internal-url http://controller:5000/v3/ \
--bootstrap-public-url http://controller:5000/v3/ \
--bootstrap-region-id RegionOne
```

该命令, 会在 keystone 数据库增加以下内容:

- 1) 在 endpoint 表增加 3 个服务实体的 API 端点
- 2) 在 local\_user 表中创建 admin 用户
- 3) 在 project 表中创建 admin 和 Default 项目 (默认域)
- 4) 在 role 表创建 3 种角色, admin, member 和 reader
- 5) 在 service 表中创建 identity 服务

### 2) 临时配置管理员账户的相关变量进行管理

编写环境变量脚本

```
vi admin-openrc
```

admin-openrc 可根据自己需要进行命名, 名字无需统一。admin-openrc 添加如下内容:

```
-----
export OS_USERNAME=admin
export OS_PASSWORD=123456
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
```

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_AUTH_URL=http://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

查看系统变量

```
env | grep OS_
```

例:

```
[root@openstack01 ~]# env|grep OS_
OS_USER_DOMAIN_NAME=Default
OS_PROJECT_NAME=admin
OS_IDENTITY_API_VERSION=3
OS_PASSWORD=123456
OS_AUTH_URL=http://controller:5000/v3
OS_USERNAME=admin
OS_PROJECT_DOMAIN_NAME=Default
```

查看 keystone 实例信息

```
openstack endpoint list
```

```
openstack project list
```

```
openstack user list
```

实例:

```
[root@openstack01 ~]# openstack endpoint list
```

```
+-----+-----+-----+-----+-----+-----+
+-----+
| ID                | Region | Service Name | Service Type |
| Enabled | Interface | URL          |              |
+-----+-----+-----+-----+-----+-----+
+-----+
|
```

b8dabe6c548e435eb2b1f7efe3b23236	RegionOne	keystone	identity	True
admin	http://controller:5000/v3/			

  

eb72eb6ea51842feb67ba5849beea48c	RegionOne	keystone	identity	
True	internal	http://controller:5000/v3/		

  

f172f6159ad34fbd8e10e0d42828d8cd	RegionOne	keystone	identity	True
public	http://controller:5000/v3/			

```

+-----+-----+-----+-----+-----+-----+
-----+

```

[root@openstack01 ~]# openstack project list

```

+-----+-----+
| ID                               | Name      |
+-----+-----+
| 3706708374804e2eb4ed056f55d84666 | admin     |
| 84cc7185f2c8461eb19a14968228b272 | myproject |
| b8e318b3c7a844708762169959c34ff8 | service   |
+-----+-----+

```

[root@openstack01 ~]# openstack user list

```

+-----+-----+
| ID                               | Name      |
+-----+-----+
| cbb2b3830a8f44bc837230bca27ae563 | myuser    |
| e5dbfc8b394c41679fd5ce229cdd6ed3 | admin     |
+-----+-----+

```

## 2.7.创建 keystone 的一般实例

### 1) 创建一个名为 example 的 keystone 域

以下命令会在 project 表中创建名为 example 的项目

```
openstack domain create --description "An Example Domain" example
```

例:

```
[root@openstack01 ~]# openstack domain create --description "An Example Domain"
example
```

```

+-----+-----+

```

Field	Value
+-----+	
description	An Example Domain
enabled	True
id	17254ea898de477ca4a1f6f3cbc6c5bc
name	example
tags	[]
+-----+	

2) 为 keystone 系统环境创建名为 **service** 的项目提供服务

以下命令会在 **project** 表中创建名为 **service** 的项目

```
openstack project create --domain default --description "Service Project"
service
```

```
[root@openstack01 ~]# openstack project create --domain default --description "Service
Project" service
```

+-----+	
Field	Value
+-----+	
description	Service Project
domain_id	default
enabled	True
id	b8e318b3c7a844708762169959c34ff8
is_domain	False
name	service
parent_id	default
tags	[]
+-----+	

3) 创建 **myproject** 项目和对应的用户及角色

以下命令会在 **project** 表中创建名为 **myproject** 项目

```
openstack project create --domain default --description "Demo Project"
myproject
```

```
[root@openstack01 ~]# openstack project create --domain default --description "Demo
Project" myproject
```

```
+-----+-----+
| Field      | Value                               |
+-----+-----+
| description | Demo Project                       |
| domain_id   | default                           |
| enabled     | True                              |
| id          | 84cc7185f2c8461eb19a14968228b272 |
| is_domain   | False                             |
| name        | myproject                         |
| parent_id   | default                           |
| tags        | []                                |
+-----+-----+
```

#### 4) 在默认域创建 myuser 用户

使用--password 选项为直接配置密码，使用--password-prompt 选项为交互式输入密码  
直接创建用户和密码

```
openstack user create --domain default --password=myuser myuser
```

例：

```
[root@openstack01 ~]# openstack user create --domain default --password-prompt
myuser
```

User Password:

Repeat User Password:

```
+-----+-----+
| Field      | Value                               |
+-----+-----+
| domain_id   | default                           |
+-----+-----+
```

enabled	True	
id	cbb2b3830a8f44bc837230bca27ae563	
name	myuser	
options	{}	
password_expires_at	None	

+-----+-----+

5) 在 role 表创建 myrole 角色

```
openstack role create myrole
```

例:

```
[root@openstack01 ~]# openstack role create myrole
```

+-----+-----+		
Field	Value	
+-----+-----+		
domain_id	None	
id	75ac33f79cc945afa42a18a3dd0ba0ad	
name	myrole	
+-----+-----+		

6) 将 myrole 角色添加到 myproject 项目中和 myuser 用户组中

```
openstack role add --project myproject --user myuser myrole
```

## 2.8.验证操作 keystone 是否安装成功

1) 作为管理员用户去请求一个认证的 token

测试是否可以使用 admin 账户进行登陆认证, 请求认证令牌

```
openstack --os-auth-url http://controller:5000/v3 \
--os-project-domain-name Default --os-user-domain-name Default \
--os-project-name admin --os-username admin token issue
```

例:

```
[root@controller ~]# openstack --os-auth-url http://controller:5000/v3 \
> --os-project-domain-name Default --os-user-domain-name Default \
> --os-project-name admin --os-username admin token issue

+-----+-----+
+-----+
| Field      | Value |
|            |       |
+-----+-----+
+-----+
|            |       |
| id          |       |
| gAAAAABcZTQtcYET21_J8sdW2McrnTGy9kZ5EeFzgS41eb-ShEM80wK9ERq5U178xE |
| nMlKL3guTqSyeuqipWL_z9avKaIOkdjG6jMARuAs9xi7x3yjckATsvAEQta9oCY09oJwK-V |
| URW3jNwbG9IP4v-eBdWVNy4RVPtBjK0JjOcos2TFspVtg4 |
| project_id | 506f9094723b4860b01ad275c87d83d6 |
|            |       |
| user_id     | b828407ee4f44dc58cd1b57d28c96bc5 |
|            |       |
+-----+-----+
+-----+
```

## 2) 使用普通用户获取认证 token

以下命令使用“myuser”用户的密码和 API 端口 5000,只允许对身份认证服务 API 的常规(非管理)访问。

```
openstack --os-auth-url http://controller:5000/v3 \
--os-project-domain-name Default --os-user-domain-name Default \
--os-project-name myproject --os-username myuser token issue
```

例:

```
[root@openstack01 ~]# openstack --os-auth-url http://controller:5000/v3 \
> --os-project-domain-name Default --os-user-domain-name Default \
> --os-project-name myproject --os-username myuser token issue
```

Password:



Field	Value
expires	2018-10-26T11:49:18+0000
id	gAAAAABb0vEuxOrgkmLfcZJI8vB6dJyrHFtvxBT1m7qLYzuD-WkOV0QUzE9mTGcrKE6CrZbLU57Nc7mv-50-ggH9pf2qrW5uWQu7MRJcUb3rgpmoYn7EVdv8X0IGK3liWEPSF48u1b2y7mEmvYb7TGOF08l87of6L2aaJmdMxp9KgM87_3Mu2-g
project_id	84cc7185f2c8461eb19a14968228b272
user_id	cbb2b3830a8f44bc837230bca27ae563

## 2.9.创建 OpenStack 客户端环境脚本

为了提升客户端操作的效率，OpenStack 支持简单的客户端环境变量脚本即 OpenRC 文件，我这里使用自定义的文件名

### 1) 创建 admin 用户的环境管理脚本

```
vi 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=123456
export OS_AUTH_URL=http://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

```
-----  
env |grep OS_
```

## 2) 测试环境管理脚本

使用脚本加载相关客户端配置，以便快速使用特定租户和用户运行客户端

```
source admin-openrc
```

## 3) 请求认证令牌

```
openstack token issue
```

例：

```
[root@controller ~]# openstack token issue
```

```
+-----+-----+  
-----+  
| Field      | Value  
|  
+-----+-----+  
-----+  
| expires    | 2019-02-14T10:25:14+0000  
|  
| id         |  
gAAAAABcZTP6JPauuyQ5CjURctMZT93QTWkd8w_QJosQYDuEp8bZiQJICK_Qc-7tz9  
vJQzFt0N1gerN4atqc2cM-NEEi_IddO-kDdKxiP2LcRTeMIh1xD8bAK67BdHztlC2SadRwe  
mVHFpZrgc70wVFhL0vJm-b7QQZ11yLs5CChj6SsTpk-u4 |  
| project_id | 506f9094723b4860b01ad275c87d83d6  
|  
| user_id    | b828407ee4f44dc58cd1b57d28c96bc5  
|  
+-----+-----+  
-----+
```

keystone 配置完毕

完成实验