Openstack_v10(CL-210 课程)_02

2018 年 3 月 7 日
22:03

Keystone（Identtity Service）

提供用户认证，基于角色的授权，策略管理，令牌管理，目录服务。

登陆 workstation 主机执行如下命令：

[student@workstation ~]$ **source admin-rc**
[student@workstation ~(admin-admin)]$ **openstack user list**
[student@workstation ~(admin-admin)]$ **openstack role list**
[student@workstation ~(admin-admin)]$ **openstack role assignment list **
> --user admin --project admin
[student@workstation ~(admin-admin)]$ **openstack endpoint list**
[student@workstation ~(admin-admin)]$ **openstack endpoint show nova**

/etc/keystone/policy.json（说明什么角色可以做什么事情）

keystone 会为访问者提高令牌，以方便用户访问其他 Openstack 组件，但当令牌过期后，默认 Openstack 不会删除过期的令牌，这会让数据库变的很大，并且降低性能。

建议定期清空过期令牌（红帽通过计划任务默认一天清理一次，建议修改为 1 小时一次）：

[root@overcloud-controller-0 cron]# **cat /var/spool/cron/keystone**

```
# Puppet Name: keystone-manage token_flush
PATH=/bin:/usr/bin:/usr/sbin SHELL=/bin/sh
```

1 0 * * * sleep `expr ${RANDOM} \% 3600`; keystone-manage token_flush >>/dev/null 2>&1

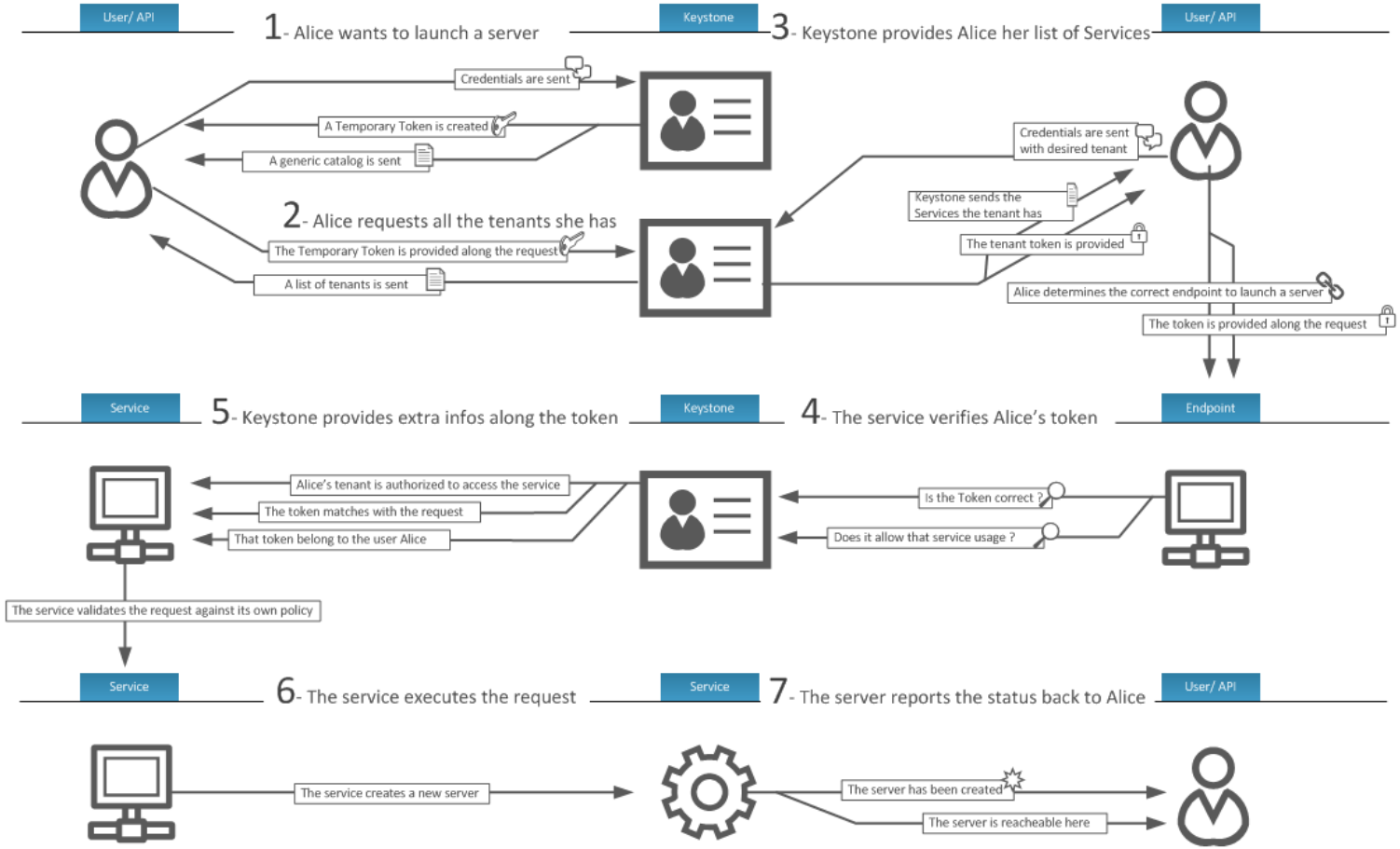
[heat-admin@overcloud-controller-0 ~]$ **openstack endpoint show nova**

```
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| adminurl     | http://172.24.1.50:8774/v2.1     |
| enabled      | True                             |
| id           | 3f08aa5465d74d489c7fae485f7999f0 |
| internalurl  | http://172.24.1.50:8774/v2.1     |
| publicurl    | http://172.25.250.50:8774/v2.1   |
| region       | regionOne                        |
| service_id   | 368e72387b4c4dffa0c14d5b88aa8f4f |
| service_name | nova                             |
| service_type | compute                          |
+--------------+----------------------------------+
```

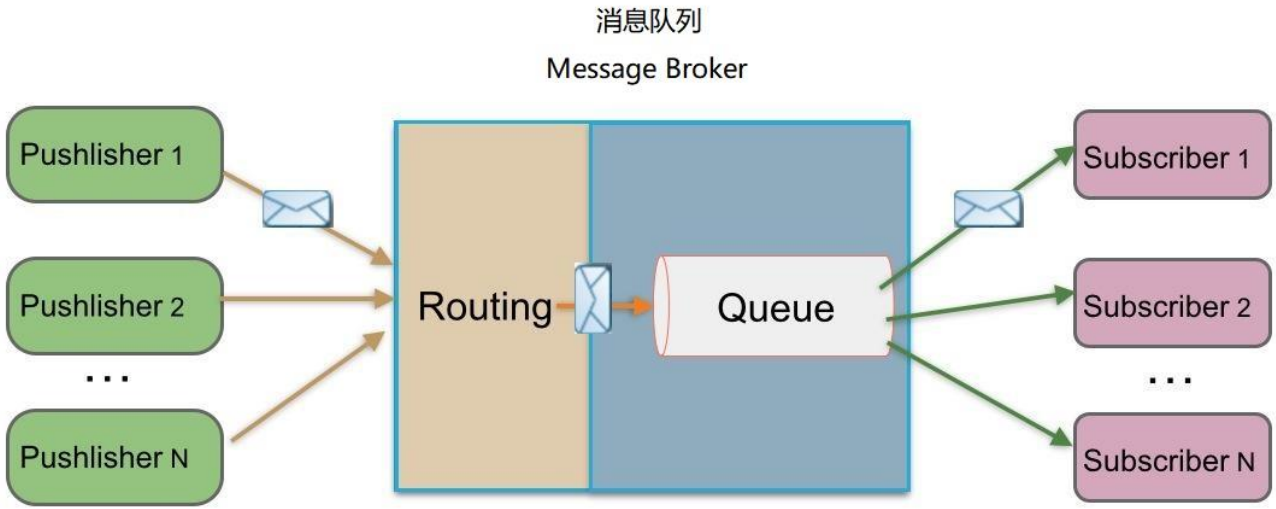
[heat-admin@overcloud-controller-0 ~]$ **openstack catalog show nova**

```
+-----------+-----------------------------------------+
| Field     | Value                                   |
+-----------+-----------------------------------------+
| endpoints | regionOne                               |
|           |   publicURL: http://172.25.250.50:8774/v2.1 |
|           |   internalURL: http://172.24.1.50:8774/v2.1 |
|           |   adminURL: http://172.24.1.50:8774/v2.1    |
|           |                                         |
| name      | nova                                    |
| type      | compute                                 |
+-----------+-----------------------------------------+
```

The Keystone Identity Manager



消息队列（RabbitMQ）



消息队列模型：

点对点模型（一对一的消息队列）direct

发布订阅模型（1 人生产消息，允许多人订阅消息）topic

概念：

术语	描述
Publisher/Producer	发布消息的应用
Consumer	接受处理消息的应用
Exchange	接受发布者发来的消息，发布到消息队列中去
Queues	存储消息的队列
Binding	连接 Exchange 与 Queues
Routing Key	让 Exchange 决定如何路由消息（哪些消息存储在哪些队列）
Message broker	消息队列服务器（允许生产者和消费者发送和接受消息）的应用程序

登陆 workstation 初始化实验环境

[student@workstation ~]$ **lab communication-msg-brokering setup**

登陆 Director 主机，进行消息队列实验：

[stack@director ~]$ **sudo -i** //切换用户 root

[root@director ~]# **rabbitmqctl report | head** //查看 rabbit 状态

Reporting server status on {{2018,3,7},{22,1,23}}

[root@director ~]# **rabbitmqctl help**

[root@director ~]# **rabbitmqctl add_user rabbitmqauth redhat** //创建账户与密码(redhat)

Creating user "rabbitmqauth" ...

[root@director ~]# **rabbitmqctl help |grep set_permission**

[root@director ~]# **rabbitmqctl set_permissions rabbitmqauth ".*" ".*" ".*"** //为用户配置权限

Setting permissions for user "rabbitmqauth" in vhost "/" ...

//设置 config 配置、write 写、read 读的权限，可以使用通配符允许所有，允许配置所有，写所有，读所有

[root@director ~]# **rabbitmqctl set_user_tags rabbitmqauth administrator** //为账户配置管理员权限，管理后台

Setting tags for user "rabbitmqauth" to [administrator] ...

[root@director ~]# **cat .rabbitmqadmin.conf** //确认 root 家目录有该隐藏文件，内容如下

```
[default]
hostname = 172.25.249.200
port = 15672
username = rabbitmqauth
password = redhat
```

[stack@director ~]$ **rabbitmqadmin help config**

```
[root@director ~]# rabbitmqctl list_users                                    //查看账户列表
Listing users ...
6cddb42684255b8f135c3729b2bdee2ca72943d1      [administrator]
rabbitmqauth      [administrator]

[root@director ~]# rabbitmqadmin help subcommands                            //查看帮助
[root@director ~]# rabbitmqadmin -c .rabbitmqadmin.conf declare exchange name=cl210.topic type=topic
exchange declared
//使用 declare 创建 Exchange，名称为 cl210.topic，类型为 topic（发布订阅模型）
//rabbitmqadmin 是 python 脚本

[root@director ~]# rabbitmqctl list_exchanges |grep cl210
//查看 exchange

[root@director ~]# rabbitmqadmin -c .rabbitmqadmin.conf declare queue name=redhat.queue
queue declared
//创建消息队列 queue
[root@director ~]# rabbitmqctl list_queues |grep redhat

[root@director ~]# rabbitmqadmin -c .rabbitmqadmin.conf  \
> publish routing_key=redhat.queue payload="a message"
Message published
```

//发布消息到 redhat.queue 队列，信息内容为：a message

[root@director ~]# **rabbitmqadmin -c .rabbitmqadmin.conf **
> publish routing_key=redhat.queue payload="another message"
Message published
//发布消息到 redhat.queue 队列，信息内容为：another message

[root@director ~]# **rabbitmqadmin -c .rabbitmqadmin.conf publish routing_key=redhat.queue**
message line1
message line2
message line3
Ctrl+D
Message published
//不使用 payload，程序等待多行输入，输入完成后，Ctrl+D 结束输入

[root@director ~]# **rabbitmqctl list_queues |grep redhat**
redhat.queue 3
//查看消息队列中有 3 条消息

[root@director ~]# **rabbitmqadmin -c .rabbitmqadmin.conf **
> get queue=redhat.queue
//查看 1 条消息
[root@director ~]# **rabbitmqadmin -c .rabbitmqadmin.conf get queue=redhat.queue count=3**

//查看所有 3 条消息

[root@director ~]# **rabbitmqadmin -c .rabbitmqadmin.conf delete queue name=redhat.queue**

登陆 workstation 清除实验环境

[student@workstation ~]$ **lab communication-msg-brokering cleanup**