Keystone 用户验证原理分析

OpenStack 中,一个租户是一个小组,可以有一个或者多个用户,对应一个 Project (如图 1),共同拥有 Nova 里的虚拟机,或者 Swift 里的容器。



图 1 每个租户对于一个项目

一、获取足够的权限和服务所在地址

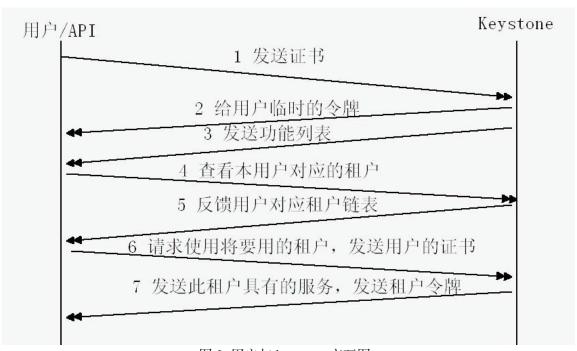


图 2 用户与 keystone 交互图

用户创建实例要有足够的权限,因此首先获得租户令牌。大致流程如图 2。 keystone 默认运行在两个端口:业务端口 5000 和管理端口 35357。在控制节点上用 curl 逐步测试图 2 中的过程。

(1) 第一步, 用户给 keystone 发送证书获取临时 token, 即 http://0.0.0.0:5000/v2.0/tokens。 如图 3。

```
}

证书

root@controler-Lenovo:/home/controler# curl -X POST -d '{"auth":{"passwordCredentials":{"username":'admin", "password":"123"}}}' -H "Content-type:application/json" http://0.0.0.0:5000/v2.0/tokens | python -mjson.tool
```

图 3 用户发送证书到 keystone

② 第二、三步, keystone 返回临时令牌, 见图 4。

图 4 用户获取临时令牌

(3) 第四、五步,用户发送临时 token 到 keystone,获取所属的租户列表(一个用户可以对应 多个租户)。如图 5 所示。

```
"title": "Not Authorized'
                                                                        用户发送令牌获取租户表
.
foot@controler-Lenovo:/home/controler# curl -H "X-Auth-Token:MIICbgYJKoZIhvcNAQcCoIICXzCCAlsCAQExCT
HBgUrDgMCGjCCAUcGCSqGSIb3DQEHAaCCATgEggE0eyJhY2Nlc3Mi0iB7InRva2VuIjogeyJpc3N1ZWRfYXQi0iAiMjAxMy0wNC
xNFQwNjo10To00C4xNTM50TQiLCAiZXhwaXJlcyI6ICIyMDEzLTA0LTE1VDA2OjU50jQ4WiIsICJpZCI6ICJwbGFjZWhvbGRlci
PLCAic2VydmljZUNhdGFsb2ciOiBbXSwgInVzZXIiOiB7InVzZXJuYW1lIjogImFkbWluIiwgInJvbGVzX2xpbmtzIjogW10sIC.
pZCI6ICJiMzI1MmNjZmQ3Y210Y2MwODliZDdjYjk4NjFkZTBjOSIsICJyb2xlcyI6IFtdLCAibmFtZSI6ICJhZG1pbiJ9LCAibW
PYWRhdGEiOiB7ImlzX2FkbWluIjogMCwgInJvbGVzIjogW119fX0xgf8wgfwCAQEwXDBXMQswCQYDVQQGEwJVUzEOMAwGA1UECB
VW5zZXQxDjAMBgNVBAcTBVVuc2V0MQ4wDAYDVQQKEwVVbnNldDEYMBYGA1UEAxMPd3d3LmV4YW1wbGUuY29tAgEBMAcGBss0Aw
MA0GCSqGSIb3DQEBAQUABIGAR9tS+V7W7Le3PY79zh+MbjhgP03GMmTxxEwQnRirvMVJDxJk4fDb2XvtU3VUyhWQJGWUBqTQIlQ
J-fULDPq3DIkQTO-fC+y-aaF1hoc7TF-69-RwboRwlWBjg3t3MswCa8guDqFXw1ZdAMz-KiqtNUuWaMjJBe00Zc2BtvrDxkY=
tp://0.0.0.0:5000/v2.0/tenants | python -mjson.tool
% Total % Received % Xferd Average Speed Time Time Current
                                      Average Speed Time Time Time Current Dload Upload Total Spent Left Speed 17911 0 --:--:-- 19250
100
      231 100
                   231
    "tenants": [
              "description": null,
             "enabled": true,
"id": "3e98c70e11674da3b9f73ebe3d1e582e",
                                                                租户链表
              "name": "admin"
             "description": null,
              "enabled": true,
"id": "8af0456b60bc45f79d693371c9d322e9",
              "name": "demo'
    ],
"tenants_links": []
```

图 5 用户用临时令牌获取自己对应的租户集合

这些租户是搭建 Openstack 平台过程中,安装 keystone 组件时创建的,如图 6。

```
updated": "2013-03-06100:00:002"
                                                                          搭建Openstack平台,配置
keystone时所创建的租户。
root@controler-Lenovo:/home/controler# keystone tenant-list
                                                                          可与附图2中用户获取的租户链表
对照。
                 id
                                             name
                                                          | enabled |
 3e98c70e11674da3b9f73ebe3d1e582e
                                            admin
                                                              True
 925e8df3e01d443c9d550a6315ed5350
                                           alt_demo
                                                              True
 8af0456b60bc45f79d693371c9d322e9
                                             demo
                                                              True
 b8548607782546e381e258de5512fd19
                                      invisible_to_admin
                                                               True
 0cfa7e9d9deb4e1d84ca6a1422c10818 |
                                           service
                                                               True
```

图 6 安装 keystone 组件时创建的租户

(4) 第六步, 用户发送想要使用的租户和证书, 如图 7。

图 7 用户发送租户 ID 和证书

(5) 第七步, 用户获取服务列表, 如图 8。

```
"metadata": {
     "is_admin": 0,
     "roles": [
           "67570bceafe04552a146f39bcdd4b465"
},
"serviceCatalog": [
                                       服务列表,篇幅原因不
能全部列出。
           "endpoints": [
                      "adminURL": "http://192.168.1.228:8774/v2/3e98c70e11674da3b9f73ebe
                      "id": "2df2417de31b4192a43cdfcbae608fee",
"internalURL": "http://192.168.1.228:8774/v2/3e98c70e11674da3b9f73ebe"
"publicURL": "http://192.168.1.228:8774/v2/3e98c70e11674da3b9f73ebe
                      "region": "RegionOne'
           ],
"endpoints_links": [],
           "name": "nova",
"type": "compute"
           "endpoints": [
                      "adminURL": "http://192.168.1.228:3333",
                      "id": "a3c56100bc5e4785a665db48cd24c290
                      "internalURL": "http://192.168.1.228:3333",
"publicURL": "http://192.168.1.228:3333",
"region": "RegionOne"
           ],
"endpoints_links": [],
           "name": "s3",
"type": "s3"
           "endpoints": [
                      "adminURL": "http://192.168.1.228:9292",
```

图 8 用户获得服务列表

在服务列表的末尾,还有了与 tenant 对应的租户令牌,此后用户用租户令牌向各服务发送请求,而非临时令牌。租户令牌如图 9 所示。

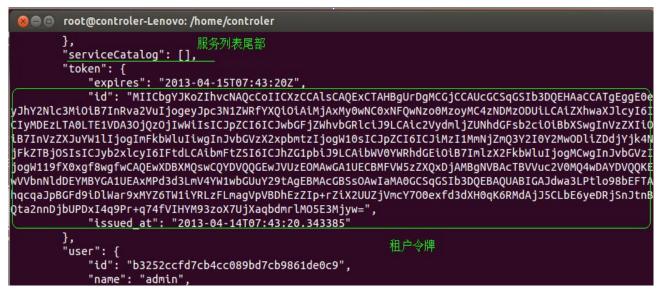


图 9 用户获取租户令牌

二、请求服务

(1)用户获得了租户令牌,有了足够的权限,通过图 8 知道服务列表和 endpoint (即服务所在的 URI)。要创建实例,就把请求和令牌发给组件(创建实例由 nova-compute 负责),各组件向 keystone 验证令牌是否有效。大致流程如图 10 所示。

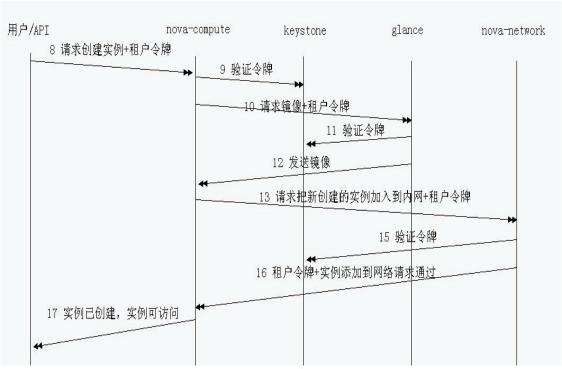


图 10 组件间处理用户请求

(2) Nova-compute、nova-network 和 glance 如何向 keystone 验证令牌? 仍然使用 curl 来测试。 组件向 keystone 发送 token 来验证 token,如图 11 所示。



图 11 组件向 keystone 验证 token

验证结果如图 12 所示。

图 12 组件得到 keystone 发回的验证结果

(3) token 存储在 mysql 数据库中,DB 名称为 keystone,表名称为 token,如图 13 所示。

```
mysql> show tables
 Tables_in_keystone
 credential
 domain
  ec2_credential
 endpoint
 group
 group_domain_metadata
 group_project_metadata
 migrate_version
  policy
 project
 role
                  用来存储token
 service
                  的表
 token
  trust
 trust_role
 user
 user_domain_metadata
 user_group_membership
 user_project_metadata
19 rows in set (0.00 sec)
mysql>
```

图 13 keystone 数据库的 token 表