

# 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

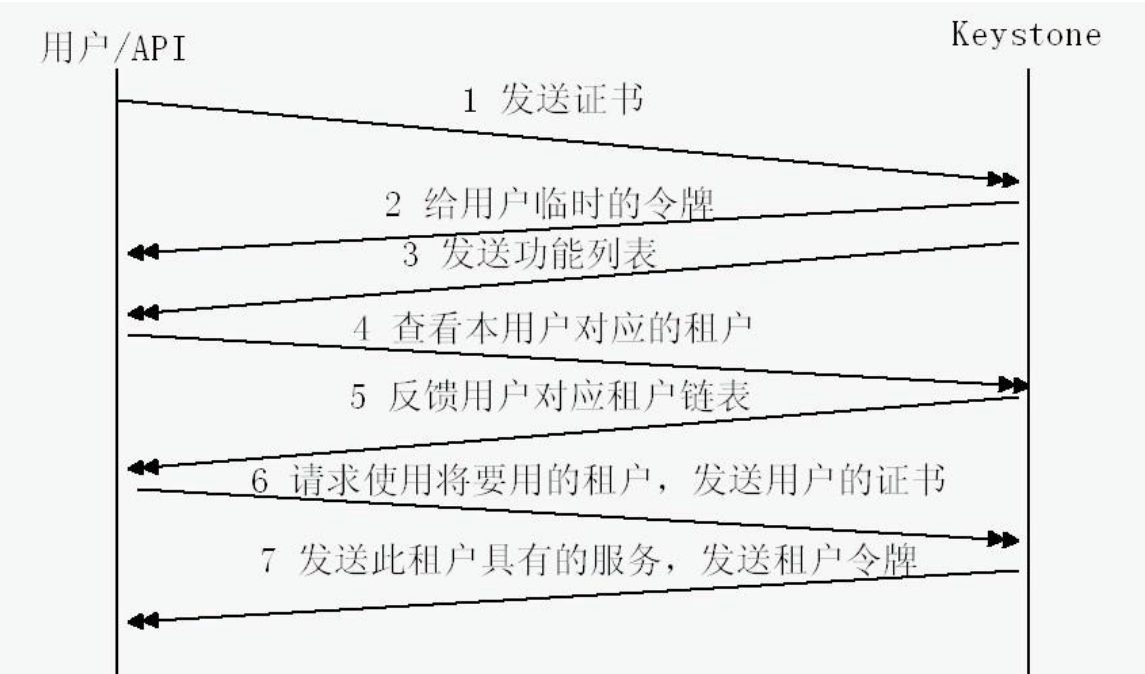


图 2 用户与 keystone 交互图

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

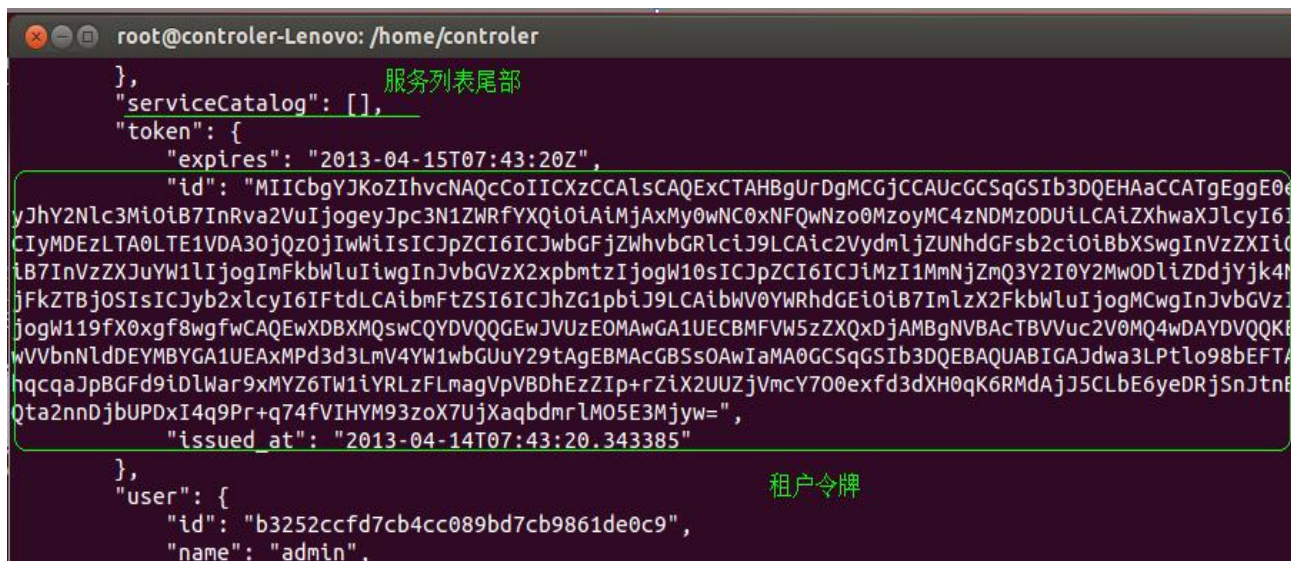






图 8 用户获得服务列表

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。

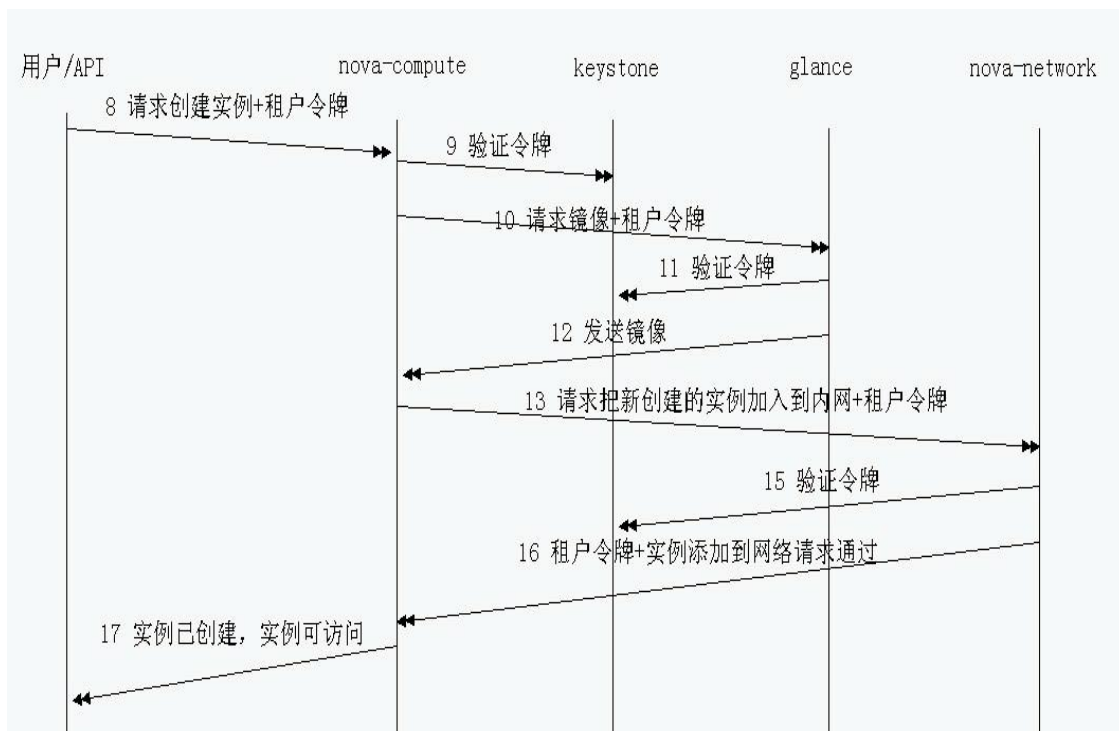


```
root@controler-Lenovo: /home/controler
},
"serviceCatalog": [],
"token": {
  "expires": "2013-04-15T07:43:20Z",
  "id": "MIICBgYJKoZIhvcNAQcCoIICXzCCAIsCAQExCTAHBgUrDgMCGjCCAUCGCSqGSIB3DQEHAaCCATgEggE0e
yJhY2Nlc3MiOiB7InRva2VuIjogeyJpc3N1ZWRFYXQiOiAiMjAxMy0wNC0xNFQwNzo0MzoyMC4zNDMzODUiLCAiZXhwaXJlcyI6I
CIyMDEzLTA0LTEiVDA3OjQzOjIwWiIsICJpZCI6ICJwbGFjZWhvbGRlcj9LCAic2VydmJjZUNhdGFsb2ciOiBbXSsgInVzZXIiOi
iB7InVzZXJ0YyIjogImFkbWluIiwgInJvbGVzX2xpbmtzIjogW10sICJpZCI6ICJlMzI1MmNjZmQ3Y2I0Y2MwODliZDdjYjk4N
jFkZTBjOSIsICJyb2xlcYI6IFtdLCAibmFtZSI6ICJhZG1pbj9LCAibWV0YWRhdGEiOiB7ImIzX2FkbWluIjogMCAwInJvbGVzI
jogW119fX0xgf8wgfwCAQEWXDBXMQswCQYDVQGEwJVUzEOMAwGA1UECBMFV5ZzZXQxZDjAMBgNVBACTBVVuc2V0MQ4wDAYDVQQKE
wVWbnNldDEYMBYGA1UEAxMPd3d3LmV4YW1wbGUuY29tAgEBMACGBSs0AwIaMA0GCSqGSIB3DQEBAQUABIGAJdwa3LPtlo98bEFTA
hqcqaJpBGFd9iDlWar9xMYZ6TW1iYRLzFLmagVpVBDhEzZIp+rZiX2UUZjVmcY700exfd3dXH0qK6RMDAJJ5CLbE6yeDRjSnJtnB
Qta2nnDjbUPDxI4q9Pr+q74fVIHYM93zoX7UjXaqbdmrlM05E3Mjyw=",
  "issued_at": "2013-04-14T07:43:20.343385"
},
"user": {
  "id": "b3252ccfd7cb4cc089bd7cb9861de0c9",
  "name": "admin",
```

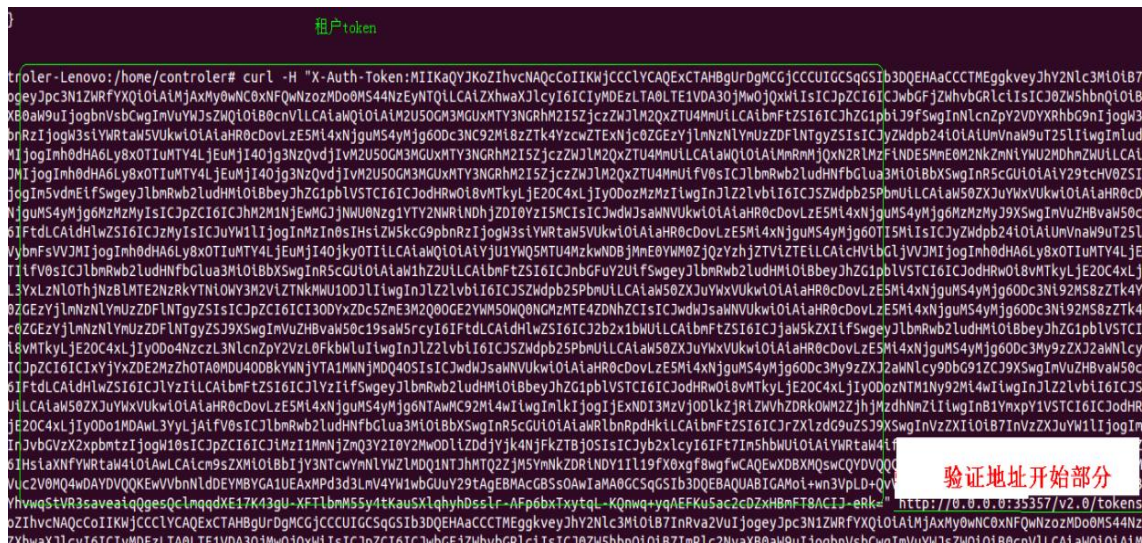
图 9 用户获取租户令牌

## 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。



(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。



验证结果如图 12 所示。

```
type : "identity"
}
],
"token": {
  "expires": "2013-04-15T07:30:41Z",
  "id": "288c16e303dc011af1e965692f70e041",
  "issued_at": "2013-04-14T07:35:38.846725",
  "tenant": {
    "description": null,
    "enabled": true,
    "id": "3e98c70e11674da3b9f73ebe3d1e582e",
    "name": "admin"
  }
},
"user": {
  "id": "b3252ccfd7cb4cc089bd7cb9861de0c9",
  "name": "admin",
  "roles": [
    {
      "id": "67570bceafe04552a146f39bcdd4b465",
      "name": "admin"
    }
  ],
  "roles_links": [],
  "username": "admin"
}
}

ot@controler-Lenovo:/home/controler#
```

验证结果

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                |
| service             |
| token               |
| trust               |
| trust_role          |
| user                |
| user_domain_metadata |
| user_group_membership |
| user_project_metadata |
+-----+
19 rows in set (0.00 sec)

mysql>
```

用来存储token  
的表

图 13 keystone 数据库的 token 表