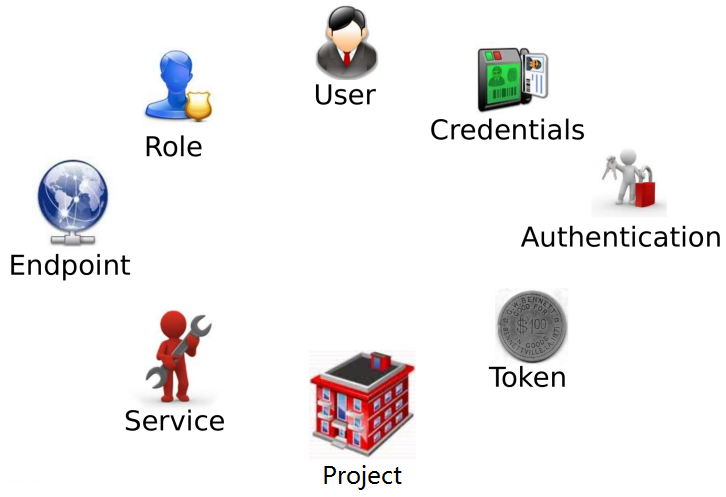


作为 OpenStack 的基础支持服务，Keystone 做下面这几件事情：

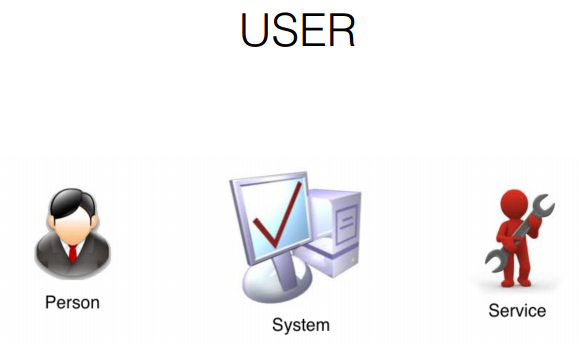
1. 管理用户及其权限
2. 维护 OpenStack Services 的 Endpoint
3. Authentication（认证）和 Authorization（鉴权）

学习 Keystone，得理解下面这些概念：



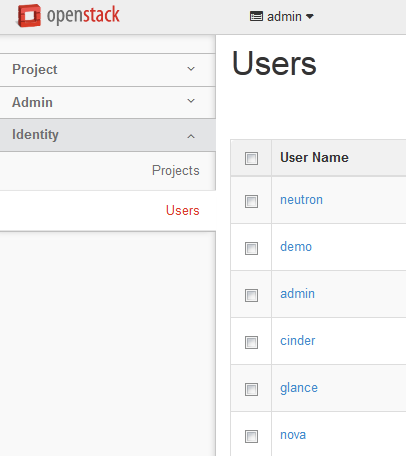
**User**

User 指代任何使用 OpenStack 的实体，可以是真正的用户，其他系统或者服务。

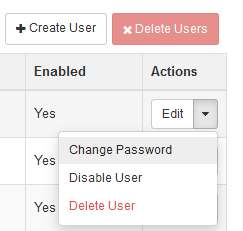


当 User 请求访问 OpenStack 时，Keystone 会对其进行验证。

Horizon 在 Identity->Users 管理 User

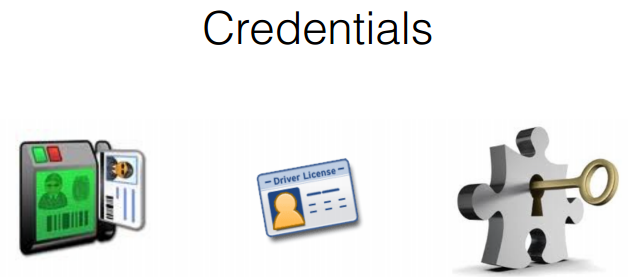


除了 admin 和 demo，OpenStack 也为 nova、cinder、glance、neutron 服务创建了相应的 User。 admin 也可以管理这些 User。



**Credentials**

Credentials 是 User 用来证明自己身份的信息，可以是： 1. 用户名/密码 2. Token 3. API Key 4. 其他高级方式



**Authentication**

Authentication 是 Keystone 验证 User 身份的过程。

User 访问 OpenStack 时向 Keystone 提交用户名和密码形式的 Credentials，Keystone 验证通过后会给 User 签发一个 Token 作为后续访问的 Credential。



**Token**

Token 是由数字和字母组成的字符串，User 成功 Authentication 后由 Keystone 分配给 User。

1. Token 用做访问 Service 的 Credential
2. Service 会通过 Keystone 验证 Token 的有效性
3. Token 的有效期默认是 24 小时



**Project**

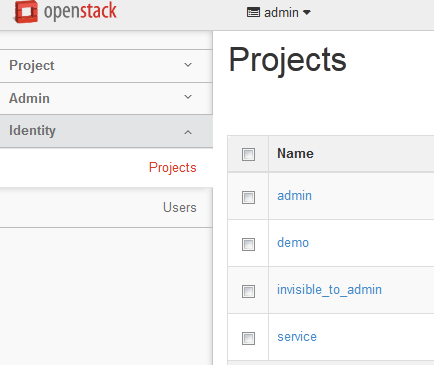
Project 用于将 OpenStack 的资源（计算、存储和网络）进行分组和隔离。 根据 OpenStack 服务的对象不同，Project 可以是一个客户（公有云，也叫租户）、部门或者项目组（私有云）。

这里请注意：

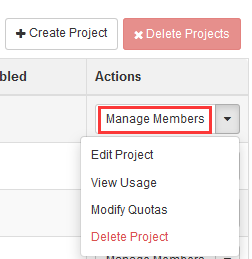
1. 资源的所有权是属于 Project 的，而不是 User。
2. 在 OpenStack 的界面和文档中，Tenant / Project / Account 这几个术语是通用的，但长期看会倾向使用 Project
3. 每个 User（包括 admin）必须挂在 Project 里才能访问该 Project 的资源。 一个User可以属于多个 Project。
4. admin 相当于 root 用户，具有最高权限

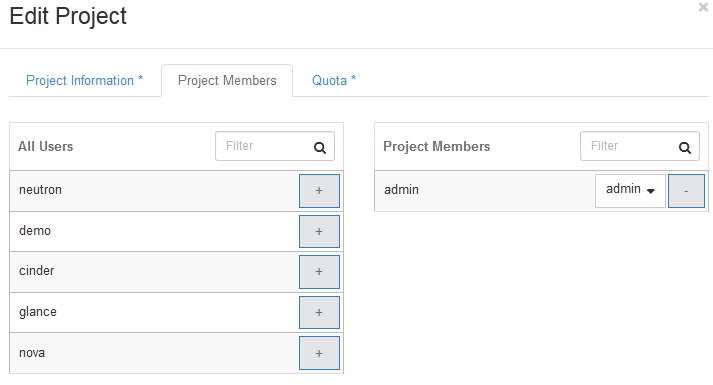


Horizon 在 Identity->Projects 中管理 Project



通过 Manage Members 将 User 添加到 Project 中

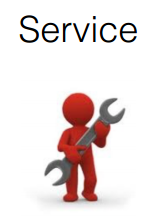




**Service**

OpenStack 的 Service 包括 Compute (Nova)、Block Storage (Cinder)、Object Storage (Swift)、Image Service (Glance) 、Networking Service (Neutron) 等。

每个 Service 都会提供若干个 Endpoint，User 通过 Endpoint 访问资源和执行操作。



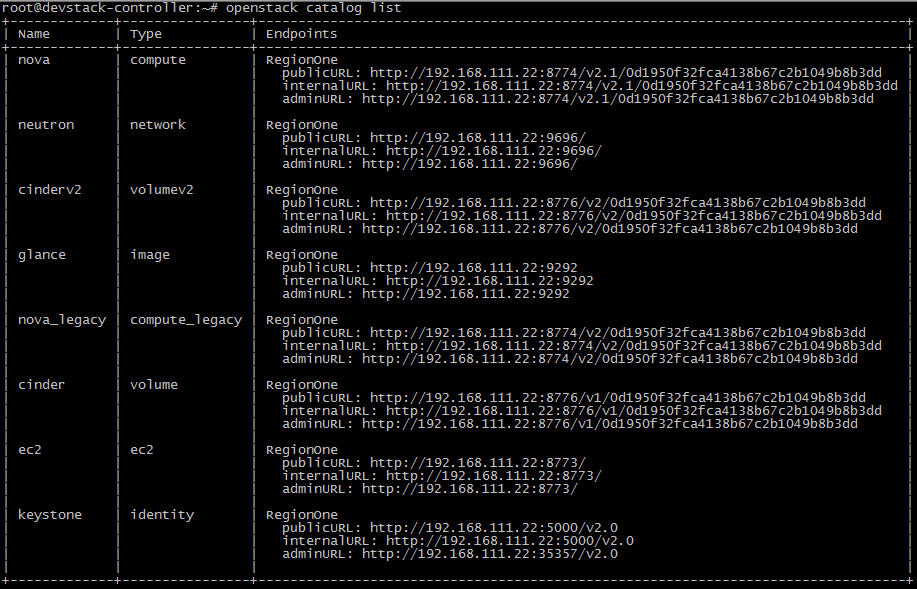
**Endpoint**

Endpoint 是一个网络上可访问的地址，通常是一个 URL。 Service 通过 Endpoint 暴露自己的 API。 Keystone 负责管理和维护每个 Service 的 Endpoint。



可以使用下面的命令来查看 Endpoint。

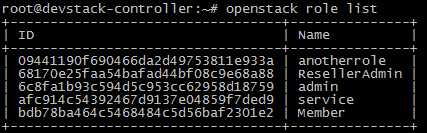
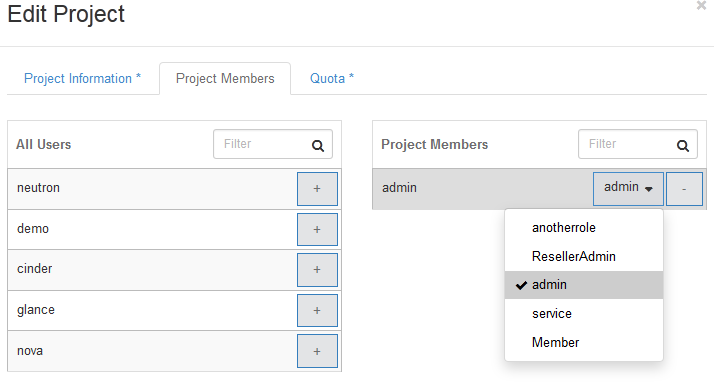
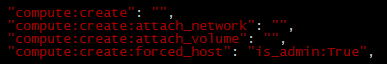
root@devstack-controller:~# source devstack/openrc admin admin   
root@devstack-controller:~# openstack catalog list



**Role**

安全包含两部分：Authentication（认证）和 Authorization（鉴权） Authentication 解决的是“你是谁？”的问题 Authorization 解决的是“你能干什么？”的问题

Keystone 是借助 Role 来实现 Authorization 的：

1. Keystone定义Role  
   
2. 可以为 User 分配一个或多个 Role Horizon 的菜单为 Identity->Project->Manage Members
3. Service 决定每个 Role 能做什么事情 Service 通过各自的 policy.json 文件对 Role 进行访问控制。 下面是 Nova 服务 /etc/nova/policy.json 中的示例

上面配置的含义是：对于 create、attach\_network 和 attach\_volume 操作，任何Role的 User 都可以执行； 但只有 admin 这个 Role 的 User 才能执行 forced\_host 操作。

OpenStack 默认配置只区分 admin 和非 admin Role。 如果需要对特定的 Role 进行授权，可以修改 policy.json。

下一节我们将通过例子加深对这些概念的理解。