OpenStack Essex版本

VLAN 部署方案范例

（文档版本：R1.0 Rev.4）

目录

[第一章 说明 1](#_Toc353441270)

[1.1 架构描述 1](#_Toc353441271)

[1.2 软件版本 2](#_Toc353441272)

[1.3 硬件需求 2](#_Toc353441273)

[1.4 网络配置 2](#_Toc353441274)

[第二章 准备系统 3](#_Toc353441275)

[2.1 安装ubuntu 12.04 3](#_Toc353441276)

[2.2 设置网络 3](#_Toc353441277)

[2.2.1 修改hosts文件 3](#_Toc353441278)

[2.2.2 修改interfaces 文件 3](#_Toc353441279)

[第三章 安装OpenStack组件并配置数据库 5](#_Toc353441280)

[3.1 安装节点1 5](#_Toc353441281)

[3.1.1 安装RabbitMQ（消息服务器）： 5](#_Toc353441282)

[3.1.2 安装MySQL（数据库服务器）： 5](#_Toc353441283)

[3.1.3 安装Nova（运算服务）： 5](#_Toc353441284)

[3.1.4 安装Glance（镜像服务）： 6](#_Toc353441285)

[3.1.5 安装Keystone（身份识别服务）： 6](#_Toc353441286)

[3.2 安装节点2 6](#_Toc353441287)

[3.3 在节点1上配置 6](#_Toc353441288)

[3.3.1 修改Nova的配置文件 6](#_Toc353441289)

[3.3.2 修改Glance的配置文件 8](#_Toc353441290)

[3.3.3 修改keystone的配置文件 8](#_Toc353441291)

[3.3.4 配置Nova和Glance使用Keystone 8](#_Toc353441292)

[3.3.5 创建各个数据库中的表 9](#_Toc353441293)

[3.4 在节点2上配置 9](#_Toc353441294)

[3.4.1 修改nova.conf文件 9](#_Toc353441295)

[3.5 检查所有服务的运行状态 9](#_Toc353441296)

[3.6 配置nova-network 10](#_Toc353441297)

[3.7 配置nova-volume 16](#_Toc353441298)

[3.8 安装Dashboard 16](#_Toc353441299)

[3.9 配置Keystone 17](#_Toc353441300)

[3.10 配置用户与租客（Keystone） 18](#_Toc353441301)

[3.11 Flavors（实例类型）配置 21](#_Toc353441302)

[第四章 NAT方案 22](#_Toc353441303)

# 说明

## 架构描述

本文档将在两台服务器中部署OpenStack（见下图，网络拓扑）。

节点1用作：

* 云平台的**控制节点**（运行nova-api、nova-scheduler、nova-network、glance、horizon、MySQL及RabbitMQ）。
* 一个**计算节点**（运行nova-compute和KVM）。
* 一个**存储节点**（运行nova-volume和iSCSI）。

节点2用作：

* 一个**计算节点**（运行nova-compute和KVM）。



## 软件版本

* 操作系统：Ubuntu Linux Server 12.04, 64-bit
* 云计算平台：OpenStack v2012.1 (Essex) 包含 Navo、Glance、Keystone及Horizon

## 硬件需求

需要的磁盘及网络适配器数量：

|  |  |  |
| --- | --- | --- |
|  | 磁盘 | 网络适配器 |
| 节点1 | 2 (sda 和 sdb) | 2 (eth0 和 eth1) |
| 节点2 | 1 (sda) | 1 (eth1) |

## 网络配置

网络类型：VLAN

**eth0(外部网络)：**

* 网络1：192.168.1.0/24

**eth1(内部网络)：**

* 网络2：172.16.1.0/24（OpenStack 管理和存储网络）
* 网络3：10.0.0.0/8（OpenStack 服务网络）

**管理网络**是用作OpenStack模块、MySQL数据库及RabbitMQ信息服务器之间的通讯。

**存储网络**用于运算节点与存储节点的iSCSI卷的通讯。

**服务网络**用于实例的IP分配，并为不同租户创建VLAN和子网。完全由OpenStack控制。

外部及内部网络IP地址：

|  |  |  |
| --- | --- | --- |
|  | eth0 | eth1 |
| 节点1 | 192.168.1.50 | 172.16.1.101 |
| 节点2 | 无 | 172.16.1.102 |
| 客户1 | 192.168.1.5 |  |

# 准备系统

## 安装ubuntu 12.04

最小化安装，只需要安装ssh server就可以。

apt-get update && apt-get -y dist-upgrade

## 设置网络

### 修改hosts文件

在**两个节点**上，修改hosts文件：

root@nodeX:~# vi /etc/hosts

添加以下内容：

|  |
| --- |
| 127.0.0.1 localhost  192.168.1.50 node1pub  172.16.1.101 node1priv node1  172.16.1.102 node2priv node2 |

### 修改interfaces 文件

在**节点1**上，执行

root@node1:~# vi /etc/network/interfaces

修改为以下内容：

|  |
| --- |
| auto lo  iface lo inet loopback  auto eth0  iface eth0 inet static  address 192.168.1.50  netmask 255.255.255.0  network 192.168.1.0  broadcast 192.168.1.255  gateway 192.168.1.1  up iptables -t nat -A POSTROUTING -s 172.16.1.0/24 -o eth0 -j SNAT --to 192.168.1.50  auto eth1  iface eth1 inet static  address 172.16.1.101  netmask 255.255.255.0  network 172.16.1.0  broadcast 172.16.1.255 |

以太网桥不需要手动设置，网桥会按需在网络节点和运算节点间自动生成。（每个租户有一个或以上的网桥）。

iptable规则通知节点1需要将所有来自网络172.16.1.0/24的在eth0出站数据包的源IP地址改为192.168.1.50。

在**节点2**上，执行

root@node2:~# vi /etc/network/interfaces

修改为以下内容：

|  |
| --- |
| auto lo  iface lo inet loopback  auto eth1  iface eth1 inet static  address 172.16.1.102  netmask 255.255.255.0  network 172.16.1.0  broadcast 172.16.1.255  gateway 172.16.1.101 |

节点1是节点2的网关

然后在**节点1**上配置节点1成为一个路由器：

root@node1:~# vi /etc/sysctl.conf

把这句的注释去掉

|  |
| --- |
| net.ipv4.ip\_forward=1 |

重启，使其生效。

# 安装OpenStack组件并配置数据库

## 安装节点1

### 安装RabbitMQ（消息服务器）：

root@node1:~# aptitude install rabbitmq-server

### 安装MySQL（数据库服务器）：

本文中所有的密码均为123456

root@node1:~# aptitude install mysql-server

修改mysql的配置文件使得所有节点都可以访问数据库：

root@node1:~# sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf

root@node1:~# service mysql restart

Nova、Glance和Keystone 将使用MySQL管理数据库，但相互不在同一个数据库：

### 安装Nova（运算服务）：

创建nova数据库和nova用户名

root@node1:~# mysql -uroot -p123456 -e 'CREATE DATABASE nova;'

root@node1:~# mysql -uroot -p123456 -e "GRANT ALL PRIVILEGES ON \*.\* TO 'nova'@'%' WITH GRANT OPTION;"

root@node1:~# mysql -uroot -p123456 -e "SET PASSWORD FOR 'nova'@'%' = PASSWORD('123456');"

安装所有的Nova组件：

控制节点：

root@node1:~# aptitude install nova-api nova-scheduler nova-network

存储节点：

root@node1:~# aptitude install nova-volume

运算节点：

root@node1:~# aptitude install nova-compute

KVM/QEMU及libvirt会在安装nova-compute时候自动安装

### 安装Glance（镜像服务）：

创建glance数据库及glance的用户名：

root@node1:~# mysql -uroot -p123456 -e 'CREATE DATABASE glance;'

root@node1:~# mysql -uroot -p123456 -e "GRANT ALL PRIVILEGES ON \*.\* TO 'glance'@'%' WITH GRANT OPTION;"

root@node1:~# mysql -uroot -p123456 -e "SET PASSWORD FOR 'glance'@'%' = PASSWORD('123456');"

安装Glance组件（api及registry）

root@node1:~# aptitude install glance

### 安装Keystone（身份识别服务）：

创建keystone数据库及keystone的用户名：

root@node1:~# mysql -uroot -p123456 -e 'CREATE DATABASE keystone;'

root@node1:~# mysql -uroot -p123456 -e "GRANT ALL PRIVILEGES ON \*.\* TO 'keystone'@'%' WITH GRANT OPTION;"

root@node1:~# mysql -uroot -p123456 -e "SET PASSWORD FOR 'keystone'@'%' = PASSWORD('123456');"

安装Keystone组件：

root@node1:~# aptitude install keystone

## 安装节点2

安装Nova组件（如果有更多的节点，其他节点的安装同节点2）

运算节点：

root@node2:~# aptitude install nova-compute

## 在节点1上配置

### 修改Nova的配置文件

执行以下命令

root@node1:~# vi /etc/nova/nova.conf

修改配置文件如下（粗体部分是节点1的hostname，根据实际情况修改）：

|  |
| --- |
| [DEFAULT]  ##### RabbitMQ #####  rabbit\_host=**node1**  ##### MySQL #####  sql\_connection=mysql://nova:123456@**node1**/nova  ##### nova-api #####  auth\_strategy=keystone  cc\_host=**node1**  ##### nova-network #####  network\_manager=nova.network.manager.VlanManager  public\_interface=eth0  vlan\_interface=eth1  network\_host=**node1**  fixed\_range=10.0.0.0/8  network\_size=1024  dhcpbridge\_flagfile=/etc/nova/nova.conf  dhcpbridge=/usr/bin/nova-dhcpbridge  force\_dhcp\_release=True  fixed\_ip\_disassociate\_timeout=30  my\_ip=172.16.1.101  routing\_source\_ip=192.168.1.50  ##### nova-compute #####  connection\_type=libvirt  libvirt\_type=kvm  libvirt\_use\_virtio\_for\_bridges=True  use\_cow\_images=True  snapshot\_image\_format=qcow2  ##### nova-volume #####  iscsi\_ip\_prefix=172.16.1.20  num\_targets=100  iscsi\_helper=tgtadm  ##### glance #####  image\_service=nova.image.glance.GlanceImageService  glance\_api\_servers=**node1**:9292  ##### Misc #####  logdir=/var/log/nova  state\_path=/var/lib/nova  lock\_path=/var/lock/nova  root\_helper=sudo nova-rootwrap |

**my\_ip**参数必须指定，否则192.168.1.50（外部IP）会取代172.16.1.101（内部IP），导致节点2在外部网络上无法访问节点1。

**routing\_source\_ip**参数必须指定，否则172.16.1.101（内部IP）会取代192.168.1.50（外部IP），路由器无法为来自服务网络（10.0.0.0/8）的数据包路由。

### 修改Glance的配置文件

修改Glance的配置文件，使用mysql数据库

root@node1:~# vi /etc/glance/glance-registry.conf

修改以下地方（粗体部分是节点1的hostname，根据实际情况修改）：

|  |
| --- |
| sql\_connection = mysql://glance:123456@**node1**/glance |

修改以下两个文件，使得Glance使用keystone，而不是用旧的身份识别服务：

root@node1:~# vi /etc/glance/glance-api.conf

root@node1:~# vi /etc/glance/glance-registry.conf

在这两个文件中添加下面的内容

|  |
| --- |
| [paste\_deploy]  flavor = keystone |

### 修改keystone的配置文件

root@node1:~# vi /etc/keystone/keystone.conf

修改以下地方（粗体部分是节点1的hostname，根据实际情况修改）：

|  |
| --- |
| connection = mysql://keystone:123456@**node1**/keystone  driver = keystone.catalog.backends.templated.TemplatedCatalog  template\_file = /etc/keystone/default\_catalog.templates  admin\_token = 999888777666 |

### 配置Nova和Glance使用Keystone

修改以下文件：

root@node1:~# vi /etc/nova/api-paste.ini

root@node1:~# vi /etc/glance/glance-api-paste.ini

root@node1:~# vi /etc/glance/glance-registry-paste.ini

在这些文件中添加以下参数：

|  |
| --- |
| admin\_tenant\_name = admin  admin\_user = admin  admin\_password = 123456  admin\_token = 999888777666 |

然后重启节点1。

root@node1:~# reboot

### 创建各个数据库中的表

root@node1:~# nova-manage db sync

root@node1:~# glance-manage version\_control 0 ; glance-manage db\_sync

root@node1:~# keystone-manage db\_sync

## 在节点2上配置

### 修改nova.conf文件

从节点1中复制nova.conf文件：

root@node2:~# scp root@node1:/etc/nova/nova.conf /etc/nova/

修改文件中的my\_ip参数：

root@node2:~# vi /etc/nova/nova.conf

改成如下所示：

|  |
| --- |
| my\_ip=172.16.1.102 |

然后重启**两个节点**

root@nodeX:~# reboot

## 检查所有服务的运行状态

root@node1:~# nova-manage service list

如果配置无误，执行命令后，可以看到类似下面的输出：

|  |
| --- |
| Binary Host Zone Status State Updated\_At  nova-scheduler node1 nova enabled :-) 2013-04-09 17:32:49  nova-network node1 nova enabled :-) 2013-04-09 17:32:49  nova-volume node1 nova enabled :-) 2013-04-09 17:32:49  nova-compute node1 nova enabled :-) 2013-04-09 17:32:49  nova-compute node2 nova enabled :-) 2013-04-09 17:32:50 |

## 配置nova-network

在VLAN模式中，每个租户都会被分配到一个单独的VLAN或子网。本文档中会配置10个VLAN。

VLAN的IP划分如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VLAN | 网桥 | 子网 | 默认网关 | VPN | 实例 |
| 1 | br1 | 10.0.1.0/24 | 10.0.1.1 | 10.0.1.2 | 10.0.1.3 - 10.0.1.254 |
| 2 | br2 | 10.0.2.0/24 | 10.0.2.1 | 10.0.2.2 | 10.0.2.3 - 10.0.2.254 |
| 3 | br3 | 10.0.3.0/24 | 10.0.3.1 | 10.0.3.2 | 10.0.3.3 - 10.0.3.254 |
| 4 | br4 | 10.0.4.0/24 | 10.0.4.1 | 10.0.4.2 | 10.0.4.3 - 10.0.4.254 |
| 5 | br5 | 10.0.5.0/24 | 10.0.5.1 | 10.0.5.2 | 10.0.5.3 - 10.0.5.254 |
| 6 | br6 | 10.0.6.0/24 | 10.0.6.1 | 10.0.6.2 | 10.0.6.3 - 10.0.6.254 |
| 7 | br7 | 10.0.7.0/24 | 10.0.7.1 | 10.0.7.2 | 10.0.7.3 - 10.0.7.254 |
| 8 | br8 | 10.0.8.0/24 | 10.0.8.1 | 10.0.8.2 | 10.0.8.3 - 10.0.8.254 |
| 9 | br9 | 10.0.9.0/24 | 10.0.9.1 | 10.0.9.2 | 10.0.9.3 - 10.0.9.254 |
| 10 | br10 | 10.0.10.0/24 | 10.0.10.1 | 10.0.10.2 | 10.0.10.3 - 10.0.10.254 |

**默认网关**：自动在网络节点上配置（本文中的节点1）

**Cloudpipe VPN 实例**：通过VPN来访问网络

**分配给实例的IP**：自动从DHCP服务器分配给实例

执行以下命令来创建表中的网络：

root@node1:~# nova-manage network create --label vlan1 --fixed\_range\_v4 10.0.1.0/24 --num\_networks 1 --network\_size 256 --vlan 1

root@node1:~# nova-manage network create --label vlan2 --fixed\_range\_v4 10.0.2.0/24 --num\_networks 1 --network\_size 256 --vlan 2

root@node1:~# nova-manage network create --label vlan3 --fixed\_range\_v4 10.0.3.0/24 --num\_networks 1 --network\_size 256 --vlan 3

root@node1:~# nova-manage network create --label vlan4 --fixed\_range\_v4 10.0.4.0/24 --num\_networks 1 --network\_size 256 --vlan 4

root@node1:~# nova-manage network create --label vlan5 --fixed\_range\_v4 10.0.5.0/24 --num\_networks 1 --network\_size 256 --vlan 5

root@node1:~# nova-manage network create --label vlan6 --fixed\_range\_v4 10.0.6.0/24 --num\_networks 1 --network\_size 256 --vlan 6

root@node1:~# nova-manage network create --label vlan7 --fixed\_range\_v4 10.0.7.0/24 --num\_networks 1 --network\_size 256 --vlan 7

root@node1:~# nova-manage network create --label vlan8 --fixed\_range\_v4 10.0.8.0/24 --num\_networks 1 --network\_size 256 --vlan 8

root@node1:~# nova-manage network create --label vlan9 --fixed\_range\_v4 10.0.9.0/24 --num\_networks 1 --network\_size 256 --vlan 9

root@node1:~# nova-manage network create --label vlan10 --fixed\_range\_v4 10.0.10.0/24 --num\_networks 1 --network\_size 256 --vlan 10

在首次运行云中的实例，如租户1，系统会把VLAN1分配给租户1，并完全属于租户1，从这时起，VLAN1总是给租户1使用。

如果其他租户运行一个新的实例，系统会把首个未分配的VLAN分配给这个租户。

此时，在节点1中运行ip addr（显示IP地址）和brctl show（显示网桥）命令会有如下显示（部分输出内容以删减）：

命令：

root@node1:~# ip addr

输出：

|  |
| --- |
| 1: lo  inet 127.0.0.1/8  inet 169.254.169.254/32 **(1)**  **(1)元数据服务**  2: eth0 **(2)**  ether 90:b1:1c:25:fc:16  inet 192.168.1.50/24 **(3)**  **(2)连接外部网络的接口**  **(3)节点1的外部IP**  3: eth1 **(4)**  ether 90:b1:1c:25:fc:17  inet 172.16.1.101/24 **(5)**  **(4)连接内部网络的接口**  **(5)节点1的内部IP**  4: virbr0 **(6)**  ether ae:4e:3d:1f:97:3b  inet 192.168.122.1/24  **(6)由libvirt API 自动配置的网桥（未使用）** |

命令：

root@node1:~# brctl show

输出：

|  |
| --- |
| virbr0 **(1)**  **(1)由libvirt API 自动配置的网桥（未使用）** |

当不同的租户使用过之后，将会有以下的输出。假设使用情况如下图：



命令：

root@node1:~# ip addr

输出：

|  |
| --- |
| 1: lo  inet 127.0.0.1/8  inet 169.254.169.254/32 **(1)**  **(1)元数据服务**  2: eth0 **(2)**  ether 90:b1:1c:25:fc:16  inet 192.168.1.50/24 **(3)**  inet 192.168.1.129/32 **(4)**  inet 192.168.1.130/32 **(5)**  **(2)连接外部网络的接口**  **(3)节点1的外部IP**  **(4)浮动IP 1（与一个实例相关联）**  **(5)浮动IP 2（与一个实例相关联）**  3: eth1 **(6)**  ether 90:b1:1c:25:fc:17  inet 172.16.1.201/24 **(7)**  (6)**连接内部网络的接口**  (7)**节点1的内部IP**  4: virbr0 **(8)**  ether ae:4e:3d:1f:97:3b  inet 192.168.122.1/24  **(8)由libvirt API 自动配置的网桥（未使用）**  5: vlan1@eth1 **(9)**  ether 90:b1:1c:25:fc:17  **(9) eth1打了VLAN1的标记**  6: br1 **(10)**  ether 90:b1:1c:25:fc:17  inet 10.0.1.1/24 **(11)**  **(10)连接到eth1上的VLAN1的网桥**  **(11)VLAN1的默认网关**  7: vlan2@eth1 **(12)**  ether 90:b1:1c:25:fc:17  **(12)eth1打了VLAN2的标记**  8: br2 **(13)**  ether 90:b1:1c:25:fc:17  inet 10.0.2.1/24 **(14)**  **(13)连接到eth1上的VLAN2的网桥**  **(14)VLAN2的默认网关**  9: vlan3@eth1 **(15)**  ether 90:b1:1c:25:fc:17  **(15)eth1打了VLAN3的标记**  10: br3 **(16)**  ether 90:b1:1c:25:fc:17  inet 10.0.3.1/24 **(17)**  **(16)连接到eth1上的VLAN3的网桥**  **(17)VLAN3的默认网关**  11: vnet0 **(18)**  ether fe:16:3e:2a:a3:f1  **(18)节点1的第1个实例生成的虚拟网络接口**  12: vnet1 **(19)**  ether fe:16:3e:46:07:6b  **(19)节点1的第2个实例生成的虚拟网络接口**  13: vnet2 **(20)**  ether fe:16:3e:34:53:06  **(20)节点1的第3个实例生成的虚拟网络接口** |

命令：

root@node1:~# brctl show

输出：

|  |
| --- |
| br1 vlan1  vnet0 **(1)**  vnet1 **(2)**  br2 vlan2  vnet2 **(3)**  br3 vlan3  **(1)节点1的第1个实例生成的虚拟网络接口(VLAN1 / 租户1)**  **(2)节点1的第2个实例生成的虚拟网络接口(VLAN1 / 租户1)**  **(3)节点1的第3个实例生成的虚拟网络接口(VLAN2 / 租户2)** |

配置浮动IP池：

pool1: 192.168.1.128/27 (192.168.1.225 - 192.168.1.254)

pool2: 192.168.1.160/27 (192.168.1.97 - 192.168.1.126)

在**节点1**上执行如下命令

root@node1:~# nova-manage floating create --pool pool1 --ip\_range 192.168.1.128/27

root@node1:~# nova-manage floating create --pool pool2 --ip\_range 192.168.1.160/27

在本文中，iptable被用作浮动IP的配置，同时也用作防火墙。

对于浮动IP的设置，需要使用NAT规则，在节点1上输入：

root@node1:~# iptables -nL -t nat

对于创建防火墙规则，将使用iptable的filter表，在节点1上输入：

root@node1:~# iptables -nL -t filter

## 配置nova-volume

在**节点1**上：

在第二磁盘创建LVM（sdb1）：

root@node1:~# cfdisk /dev/sdb

创建LVM物理卷：

root@node1:~# pvcreate /dev/sdb1

创建LVM组并命名为nova-volumes：

root@node1:~# vgcreate nova-volumes /dev/sdb1

启动nova-volume服务：

root@node1:~# service nova-volume start

本文中iSCSI的流量会在存储网络上跑，这些流量在nova-volume（iSCSI target）组件和nova-compute（iSCSI initiator）之间传输。这里采用iet（iSCSI Enterprise Target）作为iSCSI target软件。

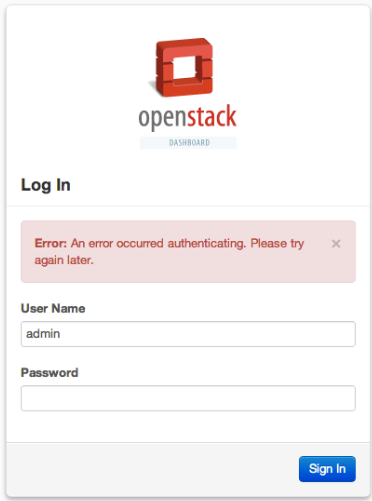
使用iet软件，需要在nova.conf文件添加如下标记：

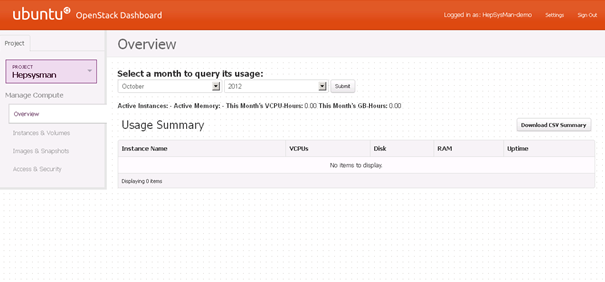
|  |
| --- |
| iscsi\_helper=ietadm |

## 安装Dashboard

root@node1:~# aptitude install openstack-dashboard memcached openstack-dashboard-ubuntu-theme

安装后，在浏览器输入http://192.168.1.50即可见到登陆界面。





## 配置Keystone

在**节点1**上创建如下服务：

root@node1:~# keystone service-create --name nova --type compute --description 'OpenStack Compute Service'

root@node1:~# keystone service-create --name nova-volume --type volume --description 'OpenStack Nova Volume Service'

root@node1:~# keystone service-create --name glance --type image --description 'OpenStack Image Service'

root@node1:~# keystone service-create --name keystone --type identity --description 'OpenStack Identity Service'

root@node1:~# keystone service-create --name quantum --type network --description 'Openstack Network Service'

创建角色：

root@node1:~# keystone role-create --name KeystoneServiceAdmin

root@node1:~# keystone role-create --name Admin

root@node1:~# keystone role-create --name Member

root@node1:~# keystone role-create --name sysadmin

root@node1:~# keystone role-create --name netadmin

## 配置用户与租客（Keystone）

只有admin用户与admin租户需要通过keystone的命令行来创建，其他的用户及租户可以通过Dashboard的Web界面进行操作，本文的用户和租户如下图所示。



为方便理解，此处列出UUID与图中的对应关系：

|  |  |  |
| --- | --- | --- |
|  | UUID | 名称 |
| 租户 | c935b4891263423e583fa7fe6567969f | admin |
| 129c187605009f3750bb7b6f4562c92d | tenant1 |
| ed3df1b67bf5c299a6d88bc0a71f8091 | tenant2 |
| 1708c7aa7c71ca07d03ec132b7e694f8 | tenant3 |
| 286168735d47e3c02ebc3841faeff8b9 | tenant4 |
| 客户 | 4ed95167e374a6522bc59434c1ec3a54 | admin |
| 51bf4e00e2df0315765d54b00a27b5a8 | A |
| 2198f650b4280c6cc45af7502f4fcd77 | B |
| 7374f5ff10a608c6b817ee4324e9c62b | C |
| 角色 | 293844c425898b66cb82b4e727601ae0 | Admin |
| 87b6b969ef1f8bec61938b6e15b865a7 | Member |
| 7bbaa04f2d39b5a45f21a288e4ec3426 | sysadmin |
| 6ef53e7c81b66797ec23c00cddec615f | netadmin |
| 293844c425898b66cb82b4e727601ae0 | KeystoneServiceAdmin |

创建租户（项目）：

root@node1:~# keystone tenant-create --name admin

root@node1:~# keystone tenant-create --name tenant1

root@node1:~# keystone tenant-create --name tenant2

root@node1:~# keystone tenant-create --name tenant3

root@node1:~# keystone tenant-create --name tenant4

创建用户：

创建用户需要用到租户的UUID，先执行：

root@node1:~# keystone tenant-list

查看所有的租户，输出如下：

|  |
| --- |
| +----------------------------------+---------+---------+  | id | name | enabled |  +----------------------------------+---------+---------+  | c935b4891263423e583fa7fe6567969f | admin | True |  | 129c187605009f3750bb7b6f4562c92d | tenant1 | True |  | ed3df1b67bf5c299a6d88bc0a71f8091 | tenant2 | True |  | 1708c7aa7c71ca07d03ec132b7e694f8 | tenant3 | True |  | 286168735d47e3c02ebc3841faeff8b9 | tenant4 | True |  +----------------------------------+---------+---------+ |

创建用户：

root@node1:~# keystone user-create --name admin --tenant\_id <租户admin的UUID>--pass 123456 --email admin@test.com --enabled true

root@node1:~# keystone user-create --name A --tenant\_id <租户A的UUID> --pass 123456 --email A@ test.com --enabled true

root@node1:~# keystone user-create --name B --tenant\_id <租户B的UUID> --pass 123456 --email B@ test.com --enabled true

root@node1:~# keystone user-create --name C --tenant\_id <租户C的UUID> --pass 123456 --email C@ test.com --enabled true

为admin用户添加KeystoneServiceAdmin角色：

这里需要用到租户的UUID、admin用户的UUID和角色的UUID

输入命令：

root@node1:~# keystone user-list

得到用户列表：

|  |
| --- |
| +----------------------------------+---------+-------------------------+----------+  | id | enabled | email | name |  +----------------------------------+---------+-------------------------+----------+  | 4ed95167e374a6522bc59434c1ec3a54 | True | admin@test.com | admin |  | 51bf4e00e2df0315765d54b00a27b5a8 | True | A@test.com | A |  | 2198f650b4280c6cc45af7502f4fcd77 | True | B@test.com | B |  | 7374f5ff10a608c6b817ee4324e9c62b | True | C@test.com | C |  +----------------------------------+---------+-------------------------+----------+ |

输入命令：

root@node1:~# keystone role-list

得到角色列表，如下：

|  |
| --- |
| +----------------------------------+----------------------+  | id | name |  +----------------------------------+----------------------+  | 293844c425898b66cb82b4e727601ae0 | Admin |  | 87b6b969ef1f8bec61938b6e15b865a7 | Member |  | 7bbaa04f2d39b5a45f21a288e4ec3426 | sysadmin |  | 6ef53e7c81b66797ec23c00cddec615f | netadmin |  | 293844c425898b66cb82b4e727601ae0 | KeystoneServiceAdmin |  +----------------------------------+----------------------+ |

root@node1:~# keystone user-role-add –-user <用户admin的UUID> –-role <KeystoneServiceAdmin的UUID>

为每个租户添加Admin角色给admin用户

root@node1:~# keystone user-role-add --user <用户admin的UUID> --role <角色Admin的UUID> --tenant\_id <租户admin的UUID>

root@node1:~# keystone user-role-add --user <用户admin的UUID> --role <角色Admin的UUID> --tenant\_id <租户tenant1的UUID>

root@node1:~# keystone user-role-add --user <用户admin的UUID> --role <角色Admin的UUID> --tenant\_id <租户tenant2的UUID>

root@node1:~# keystone user-role-add --user <用户admin的UUID> --role <角色Admin的UUID> --tenant\_id <租户tenant3的UUID>

root@node1:~# keystone user-role-add --user <用户admin的UUID> --role <角色Admin的UUID> --tenant\_id <租户tenant4的UUID>

为用户C分配tenant4：

root@node1:~# keystone user-role-add --user <用户C的UUID> --role <角色Member的UUID> --tenant\_id <租户tenant4的UUID>

## Flavors（实例类型）配置

可以添加一些定制虚拟机的硬件配置模板。

root@node1:~# nova-manage instance\_type create --name <模板名字> --memory <内存容量（MB）> --cpu <处理器数量> --root\_gb <主硬盘容量（GB）> --ephemeral\_gb <第二硬盘容量> --swap <交换分区大小> --flavor <模板编号>

# NAT方案

在路由器上可以配置不同的NAT/PAT方案让外部来访问nova-api、nova-ec2-api、volume-api、keystone-api、以SSH方式访问节点或通过浮动IP以SSH、HTTP等方式访问实例。

以下为内部和外部端口的映射表：

|  |  |  |  |
| --- | --- | --- | --- |
| 规则 | 外部端口 | 内部端口 | IP |
| nova-api | 8774 | 8774 | 192.168.1.50 |
| nova-ec2-api | 8773 | 8773 | 192.168.1.50 |
| volume-api | 8776 | 8776 | 192.168.1.50 |
| glance-api | 9292 | 9292 | 192.168.1.50 |
| keystone-api | 5000 | 5000 | 192.168.1.50 |
| keystone-adm-api | 35357 | 35357 | 192.168.1.50 |
| dashboard | 8001 | 80 | 192.168.1.50 |
| node1-ssh | 2201 | 22 | 192.168.1.50 |
| fip1-ssh | 22 | 22 | 192.168.1.129 |
| fip2-ssh | 2201 | 22 | 192.168.1.130 |
| fip3-ssh | 2202 | 22 | 192.168.1.131 |
| fip4-ssh | 2203 | 22 | 192.168.1.132 |
| fip5-ssh | 2204 | 22 | 192.168.1.133 |
| fip6-ssh | 2205 | 22 | 192.168.1.134 |
| fip7-ssh | 2206 | 22 | 192.168.1.135 |
| fip8-ssh | 2207 | 22 | 192.168.1.136 |
| fip1-http | 80 | 80 | 192.168.1.129 |
| fip2-http | 8041 | 80 | 192.168.1.130 |
| fip3-http | 8042 | 80 | 192.168.1.131 |
| fip4-http | 8043 | 80 | 192.168.1.132 |
| fip5-http | 8044 | 80 | 192.168.1.133 |
| fip6-http | 8045 | 80 | 192.168.1.134 |
| fip7-http | 8046 | 80 | 192.168.1.135 |
| fip8-http | 8047 | 80 | 192.168.1.136 |