

# openstack手动分布式部署

## 一、环境准备

参考: [https://docs.openstack.org/zh\\_CN/install-guide/](https://docs.openstack.org/zh_CN/install-guide/)

192.168.122.0/24为管理网络  
192.168.100.0/24为外部网络(上外网用)

<b>控制节点 controller</b> CPU: 4+ MEM: 6G+ disk: 50G+  NIC: 192.168.122.11 192.168.100.11 gateway:192.168.100.1 dns:114.114.114.114	<b>计算节点 compute</b> CPU: 4+ MEM: 6G+ disk: 50G+  NIC: 192.168.122.12 192.168.100.12 gateway:192.168.100.1 dns:114.114.114.114	<b>存储节点 cinder</b> CPU: 2+ MEM: 2G+ disk1: 50G(系统盘) disk2: 50G(存储盘)  NIC: 192.168.122.13
--	---	--

1, 静态IP(NetworkManager服务可以关闭)

2,主机名与绑定

```
192.168.122.11 controller
192.168.122.12 compute
192.168.122.13 cinder
```

3, 关闭防火墙和selinux

4, 时间同步

## 所有节点准备yum源

```
# yum install yum-plugin-priorities -y
```

```
# yum install https://mirrors.aliyun.com/centos-
vault/altarch/7.5.1804/extras/aarch64/Packages/centos-
release-openstack-pike-1-1.el7.x86_64.rpm -y

# vim /etc/yum.repos.d/CentOS-OpenStack-pike.repo
把
baseurl=http://mirror.centos.org/centos/7/cloud/$basear
ch/openstack-pike/
替换成
baseurl=https://mirror.tuna.tsinghua.edu.cn/cc/7/cloud/
x86_64/openstack-pike/

# yum repolist
```

repo id	repo name	status
base/7/x86_64	CentOS-7 - Base	10,070
centos-ceph-jewel/7/x86_64	CentOS-7 - Ceph Jewel	101
centos-openstack-pike pike	CentOS-7 - OpenStack	3,426+2
centos-qemu-ev/7/x86_64	CentOS-7 - QEMU EV	63
extras/7/x86_64	CentOS-7 - Extras	412
updates/7/x86_64	CentOS-7 - Updates	884
repolist: 14,956		

## 所有节点安装openstack基础工具

---

```
# yum install python-openstackclient openstack-selinux  
openstack-utils -y
```

## 计算节点安装基本软件包

---

```
[root@compute ~]# yum install qemu-kvm libvirt bridge-  
utils -y
```

```
[root@compute ~]# ln -sv /usr/libexec/qemu-kvm  
/usr/bin/  
'/usr/bin/qemu-kvm' -> '/usr/libexec/qemu-kvm'
```

## 二、安装支撑性服务

### 数据库部署

---

在**控制节点**安装mariadb(也可以安装单独的节点,甚至安装数据库集群)

参考: [https://docs.openstack.org/zh\\_CN/install-guide/environment-sql-database-rdo.html](https://docs.openstack.org/zh_CN/install-guide/environment-sql-database-rdo.html)

```
[root@controller ~]# yum install mariadb mariadb-server  
python2-PyMySQL -y
```

增加子配置文件

```
[root@controller ~]# vim /etc/my.cnf.d/openstack.cnf
[mysqld]
bind-address = 192.168.122.11          # ip为控制节点管理网段IP

default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

## 启动服务

```
[root@controller ~]# systemctl restart mariadb
[root@controller ~]# systemctl enable mariadb
```

## 安装初始化

密码请自行记住,或者全部统一密码

```
[root@controller ~]# mysql_secure_installation
```

# rabbitmq部署

---

消息队列rabbitmq的目的:

- 组件之间相互通讯的工具
- 异步方式信息同步

1, 在**控制节点**安装rabbitmq

```
[root@controller ~]# yum install erlang socat rabbitmq-server -y
```

2, 启动服务并验证端口

```
[root@controller ~]# systemctl restart rabbitmq-server
[root@controller ~]# systemctl enable rabbitmq-server
```

```
[root@controller ~]# netstat -ntlp |grep 5672
tcp        0      0 0.0.0.0:25672          0.0.0.0:*
            LISTEN      26806/beam.smp
tcp6       0      0 :::5672              :::*
            LISTEN      26806/beam.smp
```

### 3, 增加openstack用户,并授予权限

列表用户

```
[root@controller ~]# rabbitmqctl list_users
Listing users ...
guest    [administrator]
```

增加openstack用户,密码我这里还是统一为daniel.com

```
[root@controller ~]# rabbitmqctl add_user openstack
daniel.com
Creating user "openstack" ...
```

标记为administrator

```
[root@controller ~]# rabbitmqctl set_user_tags
openstack administrator
Setting tags for user "openstack" to [administrator]
...
```

给openstack对所有资源有配置,读,写权限

```
[root@controller ~]# rabbitmqctl set_permissions
openstack ".*" ".*" ".*"
Setting permissions for user "openstack" in vhost "/"
...
```

查看验证

```
[root@controller ~]# rabbitmqctl list_users
Listing users ...
```

```
openstack      [administrator]
guest         [administrator]
```

#### 4, 开启rabbitmq的web管理监控插件

rabbitmq有很多插件,使用下面命令查看

```
[root@controller ~]# rabbitmq-plugins list
Configured: E = explicitly enabled; e = implicitly
enabled
| Status:   * = running on rabbit@controller
|/
[ ] amqp_client          3.6.5
[ ] cowboy              1.0.3
[ ] cowlib              1.0.1
[ ] mochiweb            2.13.1
[ ] rabbitmq_amqp1_0    3.6.5
[ ] rabbitmq_auth_backend_ldap 3.6.5
[ ] rabbitmq_auth_mechanism_ssl 3.6.5
[ ] rabbitmq_consistent_hash_exchange 3.6.5
[ ] rabbitmq_event_exchange 3.6.5
[ ] rabbitmq_federation 3.6.5
[ ] rabbitmq_federation_management 3.6.5
[ ] rabbitmq_jms_topic_exchange 3.6.5
[ ] rabbitmq_management 3.6.5
[ ] rabbitmq_management_agent 3.6.5
[ ] rabbitmq_management_visualiser 3.6.5
[ ] rabbitmq_mqtt       3.6.5
[ ] rabbitmq_recent_history_exchange 1.2.1
[ ] rabbitmq_sharding   0.1.0
[ ] rabbitmq_shovel     3.6.5
[ ] rabbitmq_shovel_management 3.6.5
[ ] rabbitmq_stomp      3.6.5
[ ] rabbitmq_top        3.6.5
[ ] rabbitmq_tracing    3.6.5
[ ] rabbitmq_trust_store 3.6.5
[ ] rabbitmq_web_dispatch 3.6.5
```

[ ]	rabbitmq_web_stomp	3.6.5
[ ]	rabbitmq_web_stomp_examples	3.6.5
[ ]	sockjs	0.3.4
[ ]	webmachine	1.10.3

说明：

E代表开启插件

e被依赖开启插件

\*代表运行中插件

## 5, 开启rabbitmq\_management插件

```
[root@controller ~]# rabbitmq-plugins enable
rabbitmq_management
```

The following plugins have been enabled:

```
mochiweb
webmachine
rabbitmq_web_dispatch
amqp_client
rabbitmq_management_agent
rabbitmq_management
```

Applying plugin configuration to rabbit@controller...  
started 6 plugins.

```
[root@controller ~]# rabbitmq-plugins list
```

Configured: E = explicitly enabled; e = implicitly  
enabled

```
| Status:  * = running on rabbit@controller
|/
```

[e*]	amqp_client	3.6.5
[ ]	cowboy	1.0.3
[ ]	cowlib	1.0.1
[e*]	mochiweb	2.13.1
[ ]	rabbitmq_amqp1_0	3.6.5
[ ]	rabbitmq_auth_backend_ldap	3.6.5

```

[ ] rabbitmq_auth_mechanism_ssl          3.6.5
[ ] rabbitmq_consistent_hash_exchange 3.6.5
[ ] rabbitmq_event_exchange              3.6.5
[ ] rabbitmq_federation                  3.6.5
[ ] rabbitmq_federation_management       3.6.5
[ ] rabbitmq_jms_topic_exchange          3.6.5
[E*] rabbitmq_management                  3.6.5
[e*] rabbitmq_management_agent            3.6.5
[ ] rabbitmq_management_visualiser       3.6.5
[ ] rabbitmq_mqtt                         3.6.5
[ ] rabbitmq_recent_history_exchange      1.2.1
[ ] rabbitmq_sharding                     0.1.0
[ ] rabbitmq_shovel                       3.6.5
[ ] rabbitmq_shovel_management            3.6.5
[ ] rabbitmq_stomp                        3.6.5
[ ] rabbitmq_top                          3.6.5
[ ] rabbitmq_tracing                      3.6.5
[ ] rabbitmq_trust_store                  3.6.5
[e*] rabbitmq_web_dispatch                 3.6.5
[ ] rabbitmq_web_stomp                    3.6.5
[ ] rabbitmq_web_stomp_examples           3.6.5
[ ] sockjs                               0.3.4
[e*] webmachine                           1.10.3

```

15672为rabbitmq的web管理界面端口

```

[root@controller ~]# netstat -ntlp |grep 15672
tcp                0          0 0.0.0.0:15672          0.0.0.0:*
                   LISTEN          26806/beam.smp

```

6, 在**宿主机**上使用下面命令访问(ip为控制节点管理网络IP)

```

[root@daniel ~]# firefox 192.168.122.11:15672

```





Username: openstack \*  
Password: ..... \*

Login

使用前面设置的openstack用户  
与daniel.com密码登录

RabbitMQ :

Overview Connections Channels Exchanges Queues Admin

## Users

▼ All users

Filter:  ☐ Regex (?)(?)

2 items, page size up to 100

Name	Tags	Can access virtual hosts	Has password
guest	administrator	/	●
openstack	administrator	/	●

(?)

点进去,确认权限



User: openstack  
Cluster: rabbit@controller (change)  
RabbitMQ 3.6.5, Erlang 18.3.4.4

Log out

Overview Connections Channels Exchanges Queues Admin

## User: openstack

▼ Overview

Tags administrator

Can log in with password ●

▼ Permissions

Current permissions

Virtual host	Configure regexp	Write regexp	Read regexp
/	.*	.*	.*

Clear

Set permission

确认权限OK

## memcache部署

memcache作用: memcached缓存openstack各类服务的验证的token令牌。

## 1, 在控制节点安装相关软件包

```
[root@controller ~]# yum install memcached python-memcached -y
```

## 2,配置memcached监听

```
[root@controller ~]# vim /etc/sysconfig/memcached
PORT="11211"
USER="memcached"
MAXCONN="1024"
CACHESIZE="64"
OPTIONS="-l 192.168.122.11,::1"
```

将127.0.0.1改为控制节点的管理网络IP,以便其它节点组件也可以访问memcache

## 启动服务并验证端口

```
[root@controller ~]# systemctl restart memcached
[root@controller ~]# systemctl enable memcached
```

```
[root@controller ~]# netstat -ntlp | grep :11211
tcp        0      0 192.168.122.11:11211  0.0.0.0:*
             LISTEN      30586/memcached
tcp6       0      0 :::11211              :::*
             LISTEN      30586/memcached
udp        0      0 192.168.122.11:11211  0.0.0.0:*
             30586/memcached
udp6       0      0 :::11211              :::*
             30586/memcached
```

# 三、认证服务keystone

参考: <https://docs.openstack.org/keystone/pike/install/>

认证功能介绍:

keystone主要有两个功能:

- 用户管理
- 服务目录(catalog)

用户管理包括:

- 认证 token令牌,账号密码,证书,密钥
- 授权

服务目录: openstack所有可用服务的记录和API endpoint(就是一个url访问地址)

keystone支持3A:

- account
- authentication
- authorization

endpoint(端点)

- public 对外服务
- internal 对内服务
- admin 跟管理相关的服务

术语概念:

- user
- project
- role

给一个User赋予在指定Project中一个资源访问的Role角色

例: 张三(user)是运维学科(project)的讲师(role)

## 安装与配置

---

参考: <https://docs.openstack.org/keystone/pike/install/keystone-install-rdo.html>

### 1, 数据库创建keystone库并授权

```
[root@controller ~]# mysql -pdaniel.com

MariaDB [(none)]> create database keystone;

MariaDB [(none)]> grant all on keystone.* to
'keystone'@'localhost' identified by 'daniel.com';

MariaDB [(none)]> grant all on keystone.* to
'keystone'@'%' identified by 'daniel.com';

MariaDB [(none)]> flush privileges;
```

### 验证授权OK

```
[root@controller ~]# mysql -h controller -u keystone -
pdaniel.com -e 'show databases'

+-----+
| Database                |
+-----+
| information_schema      |
| keystone                 |
+-----+
```

### 2,在控制节点安装keystone相关软件

```
[root@controller ~]# yum install openstack-keystone  
httpd mod_wsgi -y
```

keystone基于httpd启动

httpd需要mod\_wsgi模块才能运行python开发的程序

### 3, 配置keystone

```
[root@controller ~]# cp /etc/keystone/keystone.conf  
/etc/keystone/keystone.conf.bak
```

```
[root@controller ~]# vim /etc/keystone/keystone.conf
```

配置连接rabbitmq

```
405 transport_url =  
rabbit://openstack:daniel.com@controller:5672
```

配置连接keystone

```
661 connection =  
mysql+pymysql://keystone:daniel.com@controller/keystone
```

打开下面这名的注释, fernet为令牌的提供者(也就是令牌的一种方式, fernet方式小巧且加密)

```
2774 provider = fernet
```

```
[root@controller ~]# grep -n '^[a-Z]'
```

```
/etc/keystone/keystone.conf
```

```
405:transport_url =  
rabbit://openstack:daniel.com@controller:5672  
661:connection =  
mysql+pymysql://keystone:daniel.com@controller/keystone  
2774:provider = fernet
```

### 4, 初始化数据库里的数据

```
[root@controller ~]# mysql -h controller -u keystone -  
pdaniel.com -e 'use keystone;show tables;'
```

```
[root@controller ~]# su -s /bin/sh -c "keystone-manage  
db_sync" keystone
```

su -s表示给bash环境,因为keystone默认不是/bin/bash  
su -c keystone表示以keystone用户身份执行命令

```
[root@controller ~]# mysql -h controller -u keystone -  
pdaniel.com -e 'use keystone;show tables;' |wc -l  
39
```

初始化导入了39多张表,表示成功

## 5, 初始化keystone认证信息

```
[root@controller ~]# keystone-manage fernet_setup --  
keystone-user keystone --keystone-group keystone  
[root@controller ~]# keystone-manage credential_setup -  
-keystone-user keystone --keystone-group keystone
```

在/etc/keystone/目录产生以下两个目录表示初始化成功  
credential-keys  
fernet-keys

## 6,初始化openstack管理员账号的api信息

```
[root@controller ~]# keystone-manage bootstrap --  
bootstrap-password daniel.com \  
--bootstrap-admin-url http://controller:35357/v3/ \  
--bootstrap-internal-url http://controller:5000/v3/ \  
--bootstrap-public-url http://controller:5000/v3/ \  
--bootstrap-region-id RegionOne
```

daniel.com为我设置的openstack管理员的密码

## 7, 配置httpd,并启动服务

```
[root@controller ~]# vim /etc/httpd/conf/httpd.conf
95 ServerName controller:80                                修改

[root@controller ~]# ln -s /usr/share/keystone/wsgi-
keystone.conf /etc/httpd/conf.d/

[root@controller ~]# systemctl restart httpd
[root@controller ~]# systemctl enable httpd

[root@controller ~]# netstat -ntlp |grep http
tcp6      0      0 :::5000          :::*
           LISTEN      387/httpd
tcp6      0      0 :::80            :::*
           LISTEN      387/httpd
tcp6      0      0 :::35357         :::*
           LISTEN      387/httpd
```

## 创建domain,project,user和role

参考: <https://docs.openstack.org/keystone/pike/install/keystone-users-rdo.html>

配置用户变量信息

1,创建admin用户的变量脚本

```
[root@controller ~]# vim admin-openstack.sh
export OS_USERNAME=admin
export OS_PASSWORD=daniel.com
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_AUTH_URL=http://controller:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

## 2,创建project

需要将上面的变量脚本source生效(相当于使用admin用户登录),才能操作

```
[root@controller ~]# source admin-openstack.sh

[root@controller ~]# openstack project list
```

ID	Name
4fa10f2089d149eca374af9497730535	admin

## 3,创建service项目



```
[root@controller ~]# openstack project create --domain
default --description "Service Project" service
```

Field	Value
description	Service Project
domain_id	default
enabled	True
id	cdc645fc266e4f35bfc23f36ecc223f3
is_domain	False
name	service
parent_id	default

#### 4,创建demo项目

```
[root@controller ~]# openstack project create --domain
default --description "Demo Project" demo
```

Field	Value
description	Demo Project
domain_id	default
enabled	True
id	5abe51bdb68c453c935a2179b5ed06a1
is_domain	False
name	demo
parent_id	default

```
[root@controller ~]# openstack project list
```

ID	Name
4fa10f2089d149eca374af9497730535	admin
5abe51bdb68c453c935a2179b5ed06a1	demo
cdc645fc266e4f35bfc23f36ecc223f3	service

## 5,创建demo用户

```
[root@controller ~]# openstack user list
```

ID	Name
528911ce70634cc296d69ef463d9e3fb	admin

```
[root@controller ~]# openstack user create --domain
default --password daniel.com demo
```

Field	Value
domain_id	default
enabled	True
id	a1fa2787411c432096d4961ddb4e1a03
name	demo
options	{}

```
| password_expires_at | None
```

```
|
```

```
+-----+-----+
```

```
--+
```

```
[root@controller ~]# openstack user list
```

```
+-----+-----+
```

```
| ID | Name |
```

```
+-----+-----+
```

```
| 528911ce70634cc296d69ef463d9e3fb | admin |
```

```
| a1fa2787411c432096d4961ddb4e1a03 | demo |
```

```
+-----+-----+
```

## 6,创建role

```
[root@controller ~]# openstack role list
```

```
+-----+-----+
```

```
| ID | Name |
```

```
+-----+-----+
```

```
| 92065899c45e469abed725db3e232a3 | admin |
```

```
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_ |
```

内置角色,不用管

```
+-----+-----+
```

```
[root@controller ~]# openstack role create user
```

```
+-----+-----+
```

```
| Field | Value |
```

```
+-----+-----+
```

```
| domain_id | None |
```

```
| id | 9bc0e93e91714972937a699e0e4dd06e |
```

```
| name | user |
```

```
+-----+-----+
```

```
[root@controller ~]# openstack role list
```

ID	Name
92065899c45e469abeed725db3e232a3	admin
9bc0e93e91714972937a699e0e4dd06e	user
9fe2ff9ee4384b1894a90878d3e92bab	_member_

7, 把demo用户加入到user角色中

```
[root@controller ~]# openstack role add --project demo
--user demo user
```

## 验证

参考: <https://docs.openstack.org/keystone/pike/install/keystone-verify-rdo.html>

1, 取消前面source过的admin用户环境变量

```
[root@controller ~]# unset OS_AUTH_URL OS_PASSWORD

[root@controller ~]# openstack user list
Missing value auth-url required for auth plugin
password
```

2, 使用admin用户验证

```
[root@controller ~]# openstack --os-auth-url
http://controller:35357/v3 --os-project-domain-name
Default --os-user-domain-name Default --os-project-name
admin --os-username admin token issue
Password: 输入admin的密码
```

3, 使用demo用户验证

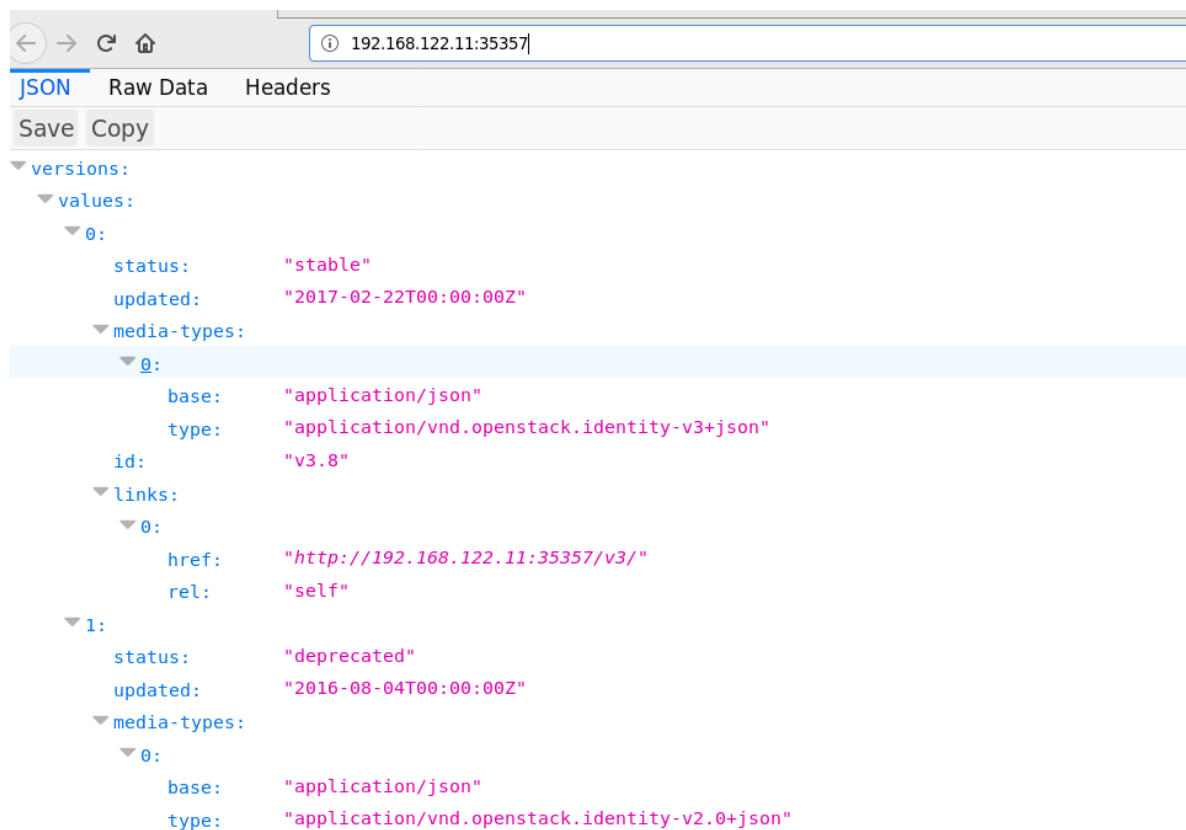
```
[root@controller ~]# openstack --os-auth-url
http://controller:5000/v3 --os-project-domain-name
Default --os-user-domain-name Default --os-project-name
demo --os-username demo token issue
Password: 输入demo的密码
```

4, 在**宿主机**上使用下面命令访问(ip为控制节点管理网络IP)

```
[root@daniel ~]# firefox 192.168.122.11:35357

[root@daniel ~]# firefox 192.168.122.11:5000
```

得到如下访问信息,这些是给程序员访问使用的



## 用户环境变量脚本

参考: <https://docs.openstack.org/keystone/pike/install/keystone-op-enrc-rdo.html>

前面创建过admin用户环境变量脚本,这里再把demo用户环境变量写好,后面方便使用脚本切换用户身份

```
[root@controller ~]# vim demo-openstack.sh
export OS_USERNAME=demo
export OS_PASSWORD=daniel.com
export OS_PROJECT_NAME=demo
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_AUTH_URL=http://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

source不同用户环境变量脚本,查看不同的token信息来验证环境变量脚本OK

```
[root@controller ~]# source admin-openstack.sh
[root@controller ~]# openstack token issue

[root@controller ~]# source demo-openstack.sh
[root@controller ~]# openstack token issue
```

## 四、镜像服务glance

参考: <https://docs.openstack.org/glance/pike/install/>

## 数据库配置

---

### 1,数据建库和授权

```
[root@controller ~]# mysql -pdaniel.com

MariaDB [(none)]> create database glance;

MariaDB [(none)]> grant all on glance.* to
'glance'@'localhost' identified by 'daniel.com';

MariaDB [(none)]> grant all on glance.* to 'glance'@'%'
identified by 'daniel.com';

MariaDB [(none)]> flush privileges;

MariaDB [(none)]> quit
```

## 2,连接验证

```
[root@controller ~]# mysql -h controller -u glance -
pdaniel.com -e 'show databases'
+-----+
| Database                |
+-----+
| glance                   |
| information_schema       |
+-----+
```

# 权限配置

---

## 1,创建用户

```
[root@controller ~]# source admin-openstack.sh

[root@controller ~]# openstack user create --domain
default --password daniel.com glance

[root@controller ~]# openstack user list
```

ID	Name
528911ce70634cc296d69ef463d9e3fb	admin
693998862e8b4261828cc0a356df1234	glance
a1fa2787411c432096d4961ddb4e1a03	demo

## 2, 把glance用户加入到Service项目的admin角色组

```
[root@controller ~]# openstack role add --project
service --user glance admin
```

## 3, 创建 glance 服务

```
[root@controller ~]# openstack service create --name
glance --description "OpenStack Image" image

[root@controller ~]# openstack service list
```

ID	Name	Type
2da4060802bf4e4bbf9328fb68b819b6	keystone	
59c3f3f50fc4466f8f3bbb72ca9a9e70	glance	image



#### 4,创建glance服务的API的endpoint(url访问)

```
[root@controller ~]# openstack endpoint create --region  
RegionOne image public http://controller:9292
```

```
[root@controller ~]# openstack endpoint create --region  
RegionOne image internal http://controller:9292
```

```
[root@controller ~]# openstack endpoint create --region  
RegionOne image admin http://controller:9292
```

验证

```
[root@controller ~]# openstack endpoint list
```

```
+-----+-----+-----+
-----+-----+-----+-----+
-----+
| ID                                     | Region   |
Service Name | Service Type | Enabled | Interface | URL
|
+-----+-----+-----+-----+
-----+-----+-----+-----+
-----+
| 4bbe9d5c517a4262bb9ce799215aabdc | RegionOne | glance
| image          | True      | internal   |
http://controller:9292          |
| 8c31c5a8060c4412b67b9acfad7f3071 | RegionOne |
keystone      | identity   | True       | admin       |
http://controller:35357/v3/    |
| 92244b7d5091491a997eecfa1cbff2fb | RegionOne |
keystone      | identity   | True       | internal    |
http://controller:5000/v3/     |
| 961a300c801246f2890e3168b55b2076 | RegionOne | glance
| image          | True      | public     |
http://controller:9292          |
| c05adadbc74541a2a5cf014466d82473 | RegionOne | glance
| image          | True      | admin      |
http://controller:9292          |
| c2481e7a89a34c0d8b85e50b9162bc01 | RegionOne |
keystone      | identity   | True       | public      |
http://controller:5000/v3/     |
+-----+-----+-----+
-----+-----+-----+-----+
-----+

```

# glance安装与配置

---

## 1,在控制节点安装

```
[root@controller ~]# yum install openstack-glance -y
```

## 2,备份配置文件

```
[root@controller ~]# cp /etc/glance/glance-api.conf  
/etc/glance/glance-api.conf.bak
```

```
[root@controller ~]# cp /etc/glance/glance-  
registry.conf /etc/glance/glance-registry.conf.bak
```

## 3, 修改glance-api.conf配置文件

```
[root@controller ~]# vim /etc/glance/glance-api.conf  
[database]  
1823 connection =  
mysql+pymysql://glance:daniel.com@controller/glance  
  
[glance_store]  
1943 stores = file,http  
  
1975 default_store = file  
  
2294 filesystem_store_datadir = /var/lib/glance/images  
  
3283 [keystone_authtoken]                                注意:这句不用改,  
下面的3284-3292行加在此参数组后面  
3284 auth_uri = http://controller:5000  
3285 auth_url = http://controller:35357  
3286 memcached_servers = controller:11211  
3287 auth_type = password  
3288 project_domain_name = default  
3289 user_domain_name = default  
3290 project_name = service
```

```
3291 username = glance
3292 password = daniel.com

[paste_deploy]
4235 flavor = keystone
```

最终配置效果如下

```
[root@controller ~]# grep -Ev '#|^$'
/etc/glance/glance-api.conf
[DEFAULT]
[cors]
[database]
connection =
mysql+pymysql://glance:daniel.com@controller/glance
[glance_store]
stores = file,http
default_store = file
filesystem_store_datadir = /var/lib/glance/images
[image_format]
[keystone_auth_token]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = glance
password = daniel.com
[matchmaker_redis]
[oslo_concurrency]
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
```

```
[oslo_middleware]
[oslo_policy]
[paste_deploy]
flavor = keystone
[profiler]
[store_type_location_strategy]
[task]
[taskflow_executor]
```

#### 4,配置glance-registry.conf配置文件

```
[root@controller ~]# vim /etc/glance/glance-registry.conf
```

```
1141 connection =
mysql+pymysql://glance:daniel.com@controller/glance
```

```
1234 [keystone_auth_token] 注意:这句不用改,下面的
```

```
1235-1243行加在此参数组后面
```

```
1235 auth_uri = http://controller:5000
1236 auth_url = http://controller:35357
1237 memcached_servers = controller:11211
1238 auth_type = password
1239 project_domain_name = default
1240 user_domain_name = default
1241 project_name = service
1242 username = glance
1243 password = daniel.com
```

```
2158 flavor = keystone
```

```
[root@controller ~]# grep -Ev '#|^$'
/etc/glance/glance-registry.conf
[DEFAULT]
[database]
```

```
connection =
mysql+pymysql://glance:daniel.com@controller/glance
[keystone_auth_token]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = glance
password = daniel.com
[matchmaker_redis]
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
[oslo_policy]
[paste_deploy]
flavor = keystone
[profiler]
```

## 导入数据到glance数据库

---

```
[root@controller ~]# su -s /bin/sh -c "glance-manage
db_sync" glance
/usr/lib/python2.7/site-
packages/oslo_db/sqlalchemy/enginefacade.py:1330:
OsloDBDeprecationWarning: EngineFacade is deprecated;
please use oslo_db.sqlalchemy.enginefacade
    expire_on_commit=expire_on_commit, _conf=conf)
INFO [alembic.runtime.migration] Context impl
MySQLImpl.
INFO [alembic.runtime.migration] Will assume non-
transactional DDL.
INFO [alembic.runtime.migration] Running upgrade ->
liberty, liberty initial
INFO [alembic.runtime.migration] Running upgrade
liberty -> mitaka01, add index on crea
ted_at and updated_at columns of
'images' table
INFO [alembic.runtime.migration] Running upgrade
mitaka01 -> mitaka02, update metadef o
s_nova_server
INFO [alembic.runtime.migration] Running upgrade
mitaka02 -> ocata01, add visibility to
and remove is_public from images
INFO [alembic.runtime.migration] Running upgrade
ocata01 -> pike01, drop glare artifact
s tables
INFO [alembic.runtime.migration] Context impl
MySQLImpl.
INFO [alembic.runtime.migration] Will assume non-
transactional DDL.
Upgraded database to: pike01, current revision(s):
pike01
```

验证数据是否导入

```
[root@controller ~]# mysql -h controller -u glance -
pdaniel.com -e 'use glance; show tables'
```

+-----+	
Tables_in_glance	
+-----+	
alembic_version	
image_locations	
image_members	
image_properties	
image_tags	
images	
metadef_namespace_resource_types	
metadef_namespaces	
metadef_objects	
metadef_properties	
metadef_resource_types	
metadef_tags	
migrate_version	
task_info	
tasks	
+-----+	

## 启动服务

---



```
[root@controller ~]# systemctl restart openstack-  
glance-api  
[root@controller ~]# systemctl enable openstack-glance-  
api  
  
[root@controller ~]# systemctl restart openstack-  
glance-registry  
[root@controller ~]# systemctl enable openstack-glance-  
registry  
  
[root@controller ~]# netstat -ntlp |grep -E  
'9191|9292'  
  
tcp        0      0 0.0.0.0:9191          0.0.0.0:*  
            LISTEN      7417/python2  
tcp        0      0 0.0.0.0:9292          0.0.0.0:*  
            LISTEN      7332/python2  
  
9191是glance-registry端口  
9292是glance-api端口
```

## 验证

---

### 1,下载测试镜像

```
[root@controller ~]# wget http://download.cirros-  
cloud.net/0.3.5/cirros-0.3.5-x86_64-disk.img
```

### 2,上传镜像

```
[root@controller ~]# source admin-openstack.sh
```

```
[root@controller ~]# openstack image create "cirros" --  
file cirros-0.3.5-x86_64-disk.img --disk-format qcow2 --  
container-format bare --public
```

public表示所有项目可用

### 3,验证镜像上传OK

```
[root@controller ~]# openstack image list
```

```
+-----+-----+-----+  
--+  
| ID                                     | Name      |  
Status |  
+-----+-----+-----+  
--+  
| 3aa31299-6102-4eab-ae91-84d204255fe2 | cirros    |  
active |  
+-----+-----+-----+  
--+
```

```
[root@controller ~]# ls /var/lib/glance/images/  
3aa31299-6102-4eab-ae91-84d204255fe2
```

## 五、计算组件nova

参考: <https://docs.openstack.org/nova/pike/install/get-started-compute.html>

## nova控制节点部署

---

# 数据库配置

```
[root@controller ~]# mysql -pdaniel.com
```

```
MariaDB [(none)]> create database nova_api;
```

```
MariaDB [(none)]> create database nova;
```

```
MariaDB [(none)]> create database nova_cell0;
```

```
MariaDB [(none)]> grant all on nova_api.* to  
'nova'@'localhost' identified by 'daniel.com';
```

```
MariaDB [(none)]> grant all on nova_api.* to 'nova'@'%'  
identified by 'daniel.com';
```

```
MariaDB [(none)]> grant all on nova.* to  
'nova'@'localhost' identified by 'daniel.com';
```

```
MariaDB [(none)]> grant all on nova.* to 'nova'@'%'  
identified by 'daniel.com';
```

```
MariaDB [(none)]> grant all on nova_cell0.* to  
'nova'@'localhost' identified by 'daniel.com';
```

```
MariaDB [(none)]> grant all on nova_cell0.* to  
'nova'@'%' identified by 'daniel.com';
```

```
MariaDB [(none)]> flush privileges;
```

```
MariaDB [(none)]> quit
```

```
[root@controller ~]# mysql -h controller -u nova -  
pdaniel.com -e 'show databases'
```

```
+-----+  
| Database          |  
+-----+  
| information_schema |  
| nova              |  
| nova_api          |  
| nova_cell0        |  
+-----+
```

## 权限配置

创建nova用户

```
[root@controller ~]# source admin-openstack.sh
```

```
[root@controller ~]# openstack user create --domain  
default --password daniel.com nova
```

```
[root@controller ~]# openstack user list
```

```
+-----+-----+  
| ID              | Name  |  
+-----+-----+  
| 528911ce70634cc296d69ef463d9e3fb | admin |  
| 648ef5d3f85e4894bbbacc8d45f8ebdb | nova  |  
| 693998862e8b4261828cc0a356df1234 | glance |  
| a1fa2787411c432096d4961ddb4e1a03 | demo  |  
+-----+-----+
```

2,把nova用户加入到Service项目的admin角色组

```
[root@controller ~]# openstack role add --project  
service --user nova admin
```

3, 创建nova服务

```
[root@controller ~]# openstack service create --name
nova --description "OpenStack Compute" compute

[root@controller ~]# openstack service list
```

ID	Name	Type
2da4060802bf4e4bbf9328fb68b819b6	keystone	identity
59c3f3f50fc4466f8f3bbb72ca9a9e70	glance	image
8bfb289223284a939b54f043f786b17f	nova	compute

#### 4,配置nova服务的api地址记录

```
[root@controller ~]# openstack endpoint create --region
RegionOne compute public http://controller:8774/v2.1

[root@controller ~]# openstack endpoint create --region
RegionOne compute internal http://controller:8774/v2.1

[root@controller ~]# openstack endpoint create --region
RegionOne compute admin http://controller:8774/v2.1
```

```
[root@controller ~]# openstack endpoint list
```

Region	Service	Interface	URL
RegionOne	compute	public	http://controller:8774/v2.1
RegionOne	compute	internal	http://controller:8774/v2.1
RegionOne	compute	admin	http://controller:8774/v2.1

ID	Region	Service Name	Service Type	Enabled	Interface	URL
12af4c0bd34b4588bb17bd0702066ed5	RegionOne	nova	compute	True	internal	http://controller:8774/v2.1
4bbe9d5c517a4262bb9ce799215aabdc	RegionOne	glance	image	True	internal	http://controller:9292
513e7612169c4be9aae6af659ea536db	RegionOne	nova	compute	True	admin	http://controller:8774/v2.1
77f2d6b77d224b598cd4334d3980b82f	RegionOne	nova	compute	True	public	http://controller:8774/v2.1
8c31c5a8060c4412b67b9acfad7f3071	RegionOne	keystone	identity	True	admin	http://controller:35357/v3/
92244b7d5091491a997eecfa1cbff2fb	RegionOne	keystone	identity	True	internal	http://controller:5000/v3/
961a300c801246f2890e3168b55b2076	RegionOne	glance	image	True	public	http://controller:9292
c05adadbc74541a2a5cf014466d82473	RegionOne	glance	image	True	admin	http://controller:9292
c2481e7a89a34c0d8b85e50b9162bc01	RegionOne	keystone	identity	True	public	http://controller:5000/v3/

## 5, 创建placement用户, 用于资源的追踪记录

```
[root@controller ~]# openstack user create --domain default --password daniel.com placement
```

```
[root@controller ~]# openstack user list
```

+-----+-----+	
ID	Name
+-----+-----+	
528911ce70634cc296d69ef463d9e3fb	admin
648ef5d3f85e4894bbbacc8d45f8ebdb	nova
693998862e8b4261828cc0a356df1234	glance
6e68e53c047949ce8f72c54c0dd58c34	placement
a1fa2787411c432096d4961ddb4e1a03	demo
+-----+-----+	

## 6, 把placement用户加入到Service项目的admin角色组

```
[root@controller ~]# openstack role add --project service --user placement admin
```

## 7, 创建placement服务

```
[root@controller ~]# openstack service create --name
placement --description "Placement API" placement

[root@controller ~]# openstack service list
```

ID	Name	Type
2da4060802bf4e4bbf9328fb68b819b6	keystone	identity
59c3f3f50fc4466f8f3bbb72ca9a9e70	glance	image
8bfb289223284a939b54f043f786b17f	nova	compute
ebe864d64de14f04b05b67df4dd7b449	placement	placement

## 8,创建placement服务的api地址记录

```
[root@controller ~]# openstack endpoint create --region
RegionOne placement public http://controller:8778

[root@controller ~]# openstack endpoint create --region
RegionOne placement internal http://controller:8778

[root@controller ~]# openstack endpoint create --region
RegionOne placement admin http://controller:8778
```

```
[root@controller ~]# openstack endpoint list
```

RegionOne	placement	public	internal	admin
2da4060802bf4e4bbf9328fb68b819b6	keystone	identity		
59c3f3f50fc4466f8f3bbb72ca9a9e70	glance	image		
8bfb289223284a939b54f043f786b17f	nova	compute		
ebe864d64de14f04b05b67df4dd7b449	placement	placement	placement	placement



ID	Region	Service Name	Service Type	Enabled	Interface	URL
12af4c0bd34b4588bb17bd0702066ed5	RegionOne	nova	compute	True	internal	http://controller:8774/v2.1
4bbe9d5c517a4262bb9ce799215aabdc	RegionOne	glance	image	True	internal	http://controller:9292
513e7612169c4be9aae6af659ea536db	RegionOne	nova	compute	True	admin	http://controller:8774/v2.1
77f2d6b77d224b598cd4334d3980b82f	RegionOne	nova	compute	True	public	http://controller:8774/v2.1
862441d899cb4b8aad4c7463783e3da7	RegionOne	placement	placement	True	admin	http://controller:8778
8c31c5a8060c4412b67b9acfad7f3071	RegionOne	keystone	identity	True	admin	http://controller:35357/v3/
92244b7d5091491a997eecfa1cbff2fb	RegionOne	keystone	identity	True	internal	http://controller:5000/v3/
961a300c801246f2890e3168b55b2076	RegionOne	glance	image	True	public	http://controller:9292
bf8defa2f0b34d8e8a5de3b87ca255e6	RegionOne	placement	placement	True	public	http://controller:8778
c05adadbc74541a2a5cf014466d82473	RegionOne	glance	image	True	admin	http://controller:9292

```
| c2481e7a89a34c0d8b85e50b9162bc01 | RegionOne |
keystone      | identity      | True      | public      |
http://controller:5000/v3/ |
| d1f0416db52a4b9fae5187b29ab138fb | RegionOne |
placement     | placement     | True      | internal    |
http://controller:8778      |

+-----+-----+-----+
-----+-----+-----+-----+
-----+
```

## 软件安装与配置

### 1,在控制节点安装nova相关软件

```
[root@controller ~]# yum install openstack-nova-api
openstack-nova-conductor openstack-nova-console
openstack-nova-novncproxy openstack-nova-scheduler
openstack-nova-placement-api -y
```

### 2,备份配置文件

```
[root@controller ~]# cp /etc/nova/nova.conf
/etc/nova/nova.conf.bak

[root@controller ~]# cp /etc/httpd/conf.d/00-nova-
placement-api.conf /etc/httpd/conf.d/00-nova-placement-
api.conf.bak
```

### 3,修改nova.conf配置文件

```
[root@controller ~]# vim /etc/nova/nova.conf
[DEFAULT]
2753 enabled_apis=osapi_compute,metadata

[api_database]
```

3479

```
connection=mysql+pymysql://nova:daniel.com@controller/nova_api
```

```
[database]
```

4453

```
connection=mysql+pymysql://nova:daniel.com@controller/nova
```

```
[DEFAULT]
```

3130

```
transport_url=rabbit://openstack:daniel.com@controller
```

```
[api]
```

3193 auth\_strategy=keystone

5771 [keystone\_authtoken]            注意:这句不用改, 5772-5780都要加在[keystone\_authtoken]下面

5772 auth\_uri = http://controller:5000

5773 auth\_url = http://controller:35357

5774 memcached\_servers = controller:11211

5775 auth\_type = password

5776 project\_domain\_name = default

5777 user\_domain\_name = default

5778 project\_name = service

5779 username = nova

5780 password = daniel.com

```
[DEFAULT]
```

1817 use\_neutron=true

2479

```
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
```

```
[vnc]
```

9897 enabled=true

9919 vncserver\_listen=192.168.122.11

```
9930 vncserver_proxycient_address=192.168.122.11
```

```
[glance]
```

```
5067 api_servers=http://controller:9292
```

```
[oslo_concurrency]
```

```
7489 lock_path=/var/lib/nova/tmp
```

```
8304 [placement]
```

注意:这句不用改,8305-8312

都要加在[placement]下面

```
8305 os_region_name = RegionOne
```

```
8306 project_domain_name = Default
```

```
8307 project_name = service
```

```
8308 auth_type = password
```

```
8309 user_domain_name = Default
```

```
8310 auth_url = http://controller:35357/v3
```

```
8311 username = placement
```

```
8312 password = daniel.com
```

改的实在太多,可以直接复制下面的配置

```
[root@controller ~]# grep -Ev '^#|^$'
```

```
/etc/nova/nova.conf
```

```
[DEFAULT]
```

```
use_neutron=true
```

```
firewall_driver=nova.virt.libvirt.firewall.IptablesFire  
wallDriver
```

```
enabled_apis=osapi_compute,metadata
```

```
transport_url=rabbit://openstack:daniel.com@controller
```

```
[api]
```

```
auth_strategy=keystone
```

```
[api_database]
```

```
connection=mysql+pymysql://nova:daniel.com@controller/n  
ova_api
```

```
[barbican]
```

```
[cache]
```

```
[cells]
[cinder]
[compute]
[conductor]
[console]
[consoleauth]
[cors]
[crypto]
[database]
connection=mysql+pymysql://nova:daniel.com@controller/nova
[ephemeral_storage_encryption]
[filter_scheduler]
[glance]
api_servers=http://controller:9292
[guestfs]
[healthcheck]
[hyperv]
[ironic]
[key_manager]
[keystone]
[keystone_auth_token]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = daniel.com
[libvirt]
[matchmaker_redis]
[metrics]
[mks]
[neutron]
[notifications]
```

```
[osapi_v21]
[oslo_concurrency]
lock_path=/var/lib/nova/tmp
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
[oslo_middleware]
[oslo_policy]
[pci]
[placement]
os_region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://controller:35357/v3
username = placement
password = daniel.com
[quota]
[rdp]
[remote_debug]
[scheduler]
[serial_console]
[service_user]
[spice]
[trusted_computing]
[upgrade_levels]
[vendordata_dynamic_auth]
[vmware]
[vnc]
enabled=true
vncserver_listen=192.168.122.11
vncserver_proxycient_address=192.168.122.11
[workarounds]
[wsgi]
```

```
[xenserver]
[xvp]
```

#### 4,配置00-nova-placement-api.conf配置文件

```
[root@controller ~]# vim /etc/httpd/conf.d/00-nova-
placement-api.conf

3 <VirtualHost *:8778>
    .....
    将下面一段加到</VirtualHost>上面
    .....
    <Directory /usr/bin>
        <IfVersion >= 2.4>
            Require all granted
        </IfVersion>
        <IfVersion < 2.4>
            Order allow,deny
            Allow from all
        </IfVersion>
    </Directory>
25 </VirtualHost>
```

#### 5, 重启httpd服务

```
[root@controller ~]# systemctl restart httpd
```

## 导入数据到nova相关数据库

#### 导入数据到nova\_api库

```
[root@controller ~]# su -s /bin/sh -c "nova-manage
api_db sync" nova
```

#### 注册cell0数据库

```
[root@controller ~]# su -s /bin/sh -c "nova-manage  
cell_v2 map_cell0" nova
```

## 创建cell1

```
[root@controller ~]# su -s /bin/sh -c "nova-manage  
cell_v2 create_cell --name=cell1 --verbose" nova  
ce887b87-b321-4bc2-a6c5-96642c6bdc4c
```

再次同步信息到nova库(nova库与nova\_cell0库里有相关的表数据)

```
[root@controller ~]# su -s /bin/sh -c "nova-manage db  
sync" nova
```

忽略警告信息,这一步时间较久(在当前环境需要几分钟),耐心等待

## 验证

```
[root@controller ~]# nova-manage cell_v2 list_cells  
+-----+-----+-----+-----+  
-----+  
-----+  
| Name | UUID |  
| Transport URL | Database  
Connection |  
+-----+-----+-----+-----+  
-----+  
| cell0 | 00000000-0000-0000-0000-000000000000 |  
| none:/ |  
mysql+pymysql://nova:****@controller/nova_cell0 |  
| cell1 | ce887b87-b321-4bc2-a6c5-96642c6bdc4c |  
| rabbit://openstack:****@controller |  
| mysql+pymysql://nova:****@controller/nova |  
+-----+-----+-----+-----+  
-----+  
-----+
```



```
[root@controller ~]# mysql -h controller -u nova -
pdaniel.com -e 'use nova;show tables;' |wc -l
111

[root@controller ~]# mysql -h controller -u nova -
pdaniel.com -e 'use nova_api;show tables;' |wc -l
33

[root@controller ~]# mysql -h controller -u nova -
pdaniel.com -e 'use nova_cell0;show tables;' |wc -l
111
```

## 启动服务

```
[root@controller ~]# systemctl start openstack-nova-
api.service openstack-nova-consoleauth.service
openstack-nova-scheduler.service openstack-nova-
conductor.service openstack-nova-novncproxy.service

[root@controller ~]# systemctl enable openstack-nova-
api.service openstack-nova-consoleauth.service
openstack-nova-scheduler.service openstack-nova-
conductor.service openstack-nova-novncproxy.service
```

## 验证访问地址记录

```
[root@controller ~]# openstack catalog list
+-----+-----+-----+
-----+
| Name      | Type      | Endpoints
|          |          |
+-----+-----+-----+
-----+
| keystone  | identity  | RegionOne
|          |          |
|          |          | admin:
http://controller:35357/v3/ |
```

		RegionOne
		internal:
http://controller:5000/v3/		
		RegionOne
		public:
http://controller:5000/v3/		
glance	image	RegionOne
		internal:
http://controller:9292		
		RegionOne
		public:
http://controller:9292		
		RegionOne
		admin:
http://controller:9292		
nova	compute	RegionOne
		internal:
http://controller:8774/v2.1		
		RegionOne
		admin:
http://controller:8774/v2.1		
		RegionOne
		public:
http://controller:8774/v2.1		

```

|           |           |
|           |           |
| placement | placement | RegionOne
|           |           |
|           |           | admin:
http://controller:8778 |
|           |           | RegionOne
|           |           |
|           |           | public:
http://controller:8778 |
|           |           | RegionOne
|           |           |
|           |           | internal:
http://controller:8778 |
|           |           |
|           |           |
+-----+-----+-----+
-----+

```

验证日志文件

```

[root@controller ~]# ls /var/log/nova/
nova-api.log          nova-consoleauth.log  nova-
novncproxy.log        nova-scheduler.log
nova-conductor.log    nova-manage.log        nova-
placement-api.log

```

## nova计算节点部署

参考:<https://docs.openstack.org/nova/pike/install/compute-install.html>

以下操作都在compute节点做

## 安装与配置

## 1,安装软件

```
[root@compute ~]# yum install openstack-nova-compute  
sysfsutils -y
```

## 2,备份配置文件

```
[root@compute ~]# cp /etc/nova/nova.conf  
/etc/nova/nova.conf.bak
```

## 3,修改配置文件(可以直接复制控制节点的nova配置文件过来修改)

```
[root@compute ~]# cat /etc/nova/nova.conf  
[DEFAULT]  
use_neutron=true  
firewall_driver=nova.virt.libvirt.firewall.IptablesFire  
wallDriver  
enabled_apis=osapi_compute,metadata  
transport_url=rabbit://openstack:daniel.com@controller  
[api]  
auth_strategy=keystone  
[api_database]  
connection=mysql+pymysql://nova:daniel.com@controller/n  
ova_api  
[barbican]  
[cache]  
[cells]  
[cinder]  
[compute]  
[conductor]  
[console]  
[consoleauth]  
[cors]  
[crypto]  
[database]  
connection=mysql+pymysql://nova:daniel.com@controller/n  
ova
```

```
[ephemeral_storage_encryption]
[filter_scheduler]
[glance]
api_servers=http://controller:9292
[guestfs]
[healthcheck]
[hyperv]
[ironic]
[key_manager]
[keystone]
[keystone_authtoken]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = daniel.com
[libvirt]
virt_type=qemu
[matchmaker_redis]
[metrics]
[mks]
[neutron]
[notifications]
[osapi_v21]
[oslo_concurrency]
lock_path=/var/lib/nova/tmp
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
[oslo_middleware]
[oslo_policy]
```

```
[pci]
[placement]
os_region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://controller:35357/v3
username = placement
password = daniel.com
[quota]
[rdp]
[remote_debug]
[scheduler]
[serial_console]
[service_user]
[spice]
[trusted_computing]
[upgrade_levels]
[vendordata_dynamic_auth]
[vmware]
[vnc]
enabled = True
vncserver_listen = 0.0.0.0
vncserver_proxycient_address = 192.168.122.12
novncproxy_base_url =
http://192.168.122.11:6080/vnc_auto.html
[workarounds]
[wsgi]
[xenserver]
[xvp]
```

注意:与控制节点nova.conf不同的地方

1,[vnc]下的几个参数有所不同

vncserver\_proxycient\_address接的IP为compute节点管理网络IP

2, [libvirt]参数组下面加上virt\_type=qemu  
不能使用kvm, 因为我们本来就在kvm里面搭建的云平台, cat  
/proc/cpuinfo | egrep 'vmx|svm' 是查不出来的  
但如果是生产环境用物理服务器搭建就应该为virt\_type=kvm

## 启动服务

```
[root@compute ~]# systemctl start libvirtd.service  
openstack-nova-compute.service  
[root@compute ~]# systemctl enable libvirtd.service  
openstack-nova-compute.service
```

## 控制节点上添加计算节点

### 1, 查看服务

```
[root@controller ~]# openstack compute service list
```

ID	Binary	Host	Zone	Status	State
1	nova-scheduler	controller	internal	enabled	up
2	nova-conductor	controller	internal	enabled	up
3	nova-consoleauth	controller	internal	enabled	up
6	nova-compute	compute	nova	enabled	up

计算节点启动服务后会连接  
状态就会变为up.  
如果连接不上查看nova日志和检查配置文件

### 2, 新增计算节点记录, 增加到nova数据库中

```
[root@controller ~]# su -s /bin/sh -c "nova-manage
cell_v2 discover_hosts --verbose" nova
Found 2 cell mappings.
Skipping cell0 since it does not contain hosts.
Getting computes from cell 'cell1': ce887b87-b321-4bc2-
a6c5-96642c6bdc4c
Checking host mapping for compute host 'compute':
ee3f5d57-22be-489b-af2c-35e369c5aff9
Creating host mapping for compute host 'compute':
ee3f5d57-22be-489b-af2c-35e369c5aff9
Found 1 unmapped computes in cell: ce887b87-b321-4bc2-
a6c5-96642c6bdc4c
```

### 3,验证所有API是否正常

```
[root@controller ~]# nova-status upgrade check
+-----+
| Upgrade Check Results |
+-----+
| Check: Cells v2 |
| Result: Success |
| Details: None |
+-----+
| Check: Placement API |
| Result: Success |
| Details: None |
+-----+
| Check: Resource Providers |
| Result: Success |
| Details: None |
+-----+
```

## 六、网络组件neutron



参考: <https://docs.openstack.org/neutron/pike/install/>

# neutron控制节点部署

---

## 数据库配置

```
[root@controller ~]# mysql -pdaniel.com

MariaDB [(none)]> create database neutron;

MariaDB [(none)]> grant all on neutron.* to
'neutron'@'localhost' identified by 'daniel.com';

MariaDB [(none)]> grant all on neutron.* to
'neutron'@'%' identified by 'daniel.com';

MariaDB [(none)]> flush privileges;

MariaDB [(none)]> quit
```

```
[root@controller ~]# mysql -h controller -u neutron -
pdaniel.com -e 'show databases';
+-----+
| Database                |
+-----+
| information_schema      |
| neutron                  |
+-----+
```

## 权限配置

### 1, 创建neutron用户

```
[root@controller ~]# source admin-openstack.sh
```

```
[root@controller ~]# openstack user create --domain
default --password daniel.com neutron
```

```
[root@controller ~]# openstack user list
```

+-----+-----+	
ID	Name
+-----+-----+	
528911ce70634cc296d69ef463d9e3fb	admin
648ef5d3f85e4894bbbacc8d45f8ebdb	nova
693998862e8b4261828cc0a356df1234	glance
6e68e53c047949ce8f72c54c0dd58c34	placement
9f35128a10b84b4fa988aa93b67bf712	neutron
a1fa2787411c432096d4961ddb4e1a03	demo
+-----+-----+	

2, 把neutron用户到Service项目的admin角色组

```
[root@controller ~]# openstack role add --project
service --user neutron admin
```

3, 创建neutron服务

```
[root@controller ~]# openstack service create --name
neutron --description "OpenStack Networking" network
```

```
[root@controller ~]# openstack service list
```

ID	Name	Type
2da4060802bf4e4bbf9328fb68b819b6	keystone	identity
59c3f3f50fc4466f8f3bbb72ca9a9e70	glance	image
8bfb289223284a939b54f043f786b17f	nova	compute
b4cbb4cce6a5446983969e5b6fde51fa	neutron	network
ebe864d64de14f04b05b67df4dd7b449	placement	placement

#### 4, 配置neutron服务的api地址记录

```
[root@controller ~]# openstack endpoint create --region
RegionOne network public http://controller:9696
```

```
[root@controller ~]# openstack endpoint create --region
RegionOne network internal http://controller:9696
```

```
[root@controller ~]# openstack endpoint create --region
RegionOne network admin http://controller:9696
```

```
[root@controller ~]# openstack endpoint list
```

ID	Region	Service Name	Service Type	Enabled	Interface	URL
12af4c0bd34b4588bb17bd0702066ed5	RegionOne	nova	compute	True	internal	<a href="http://controller:8774/v2.1">http://controller:8774/v2.1</a>
4bbe9d5c517a4262bb9ce799215aabd5	RegionOne	glance	image	True	internal	<a href="http://controller:9292">http://controller:9292</a>
513e7612169c4be9aae6af659ea536db	RegionOne	nova	compute	True	admin	<a href="http://controller:8774/v2.1">http://controller:8774/v2.1</a>
7306e765f95c4f3793c00cb56525f5f4	RegionOne	neutron	network	True	public	<a href="http://controller:9696">http://controller:9696</a>
77f2d6b77d224b598cd4334d3980b82f	RegionOne	nova	compute	True	public	<a href="http://controller:8774/v2.1">http://controller:8774/v2.1</a>
862441d899cb4b8aad4c7463783e3da7	RegionOne	placement	placement	True	admin	<a href="http://controller:8778">http://controller:8778</a>
8c31c5a8060c4412b67b9acfad7f3071	RegionOne	keystone	identity	True	admin	<a href="http://controller:3535/v3/">http://controller:3535/v3/</a>
92244b7d5091491a997eefcf1cbff2fb	RegionOne	keystone	identity	True	internal	<a href="http://controller:5000/v3/">http://controller:5000/v3/</a>
961a300c801246f2890e3168b55b2076	RegionOne	glance	image	True	public	<a href="http://controller:9292">http://controller:9292</a>
abb64a112bdb44eb9819c74d78216d2de	RegionOne	neutron	network	True	admin	<a href="http://controller:9696">http://controller:9696</a>
abb95b9156ad4ad3b33303ca026af16d	RegionOne	neutron	network	True	internal	<a href="http://controller:9696">http://controller:9696</a>
bf8defa2f0b34d8e8a5de3b87ca255e6	RegionOne	placement	placement	True	public	<a href="http://controller:8778">http://controller:8778</a>
c05adadb74541a2a5cf014466d82473	RegionOne	glance	image	True	admin	<a href="http://controller:9292">http://controller:9292</a>
c2481e7a89a34cd8b85e50b9162bc01	RegionOne	keystone	identity	True	public	<a href="http://controller:5000/v3/">http://controller:5000/v3/</a>
d1f0416db52a4b9fae5187b29ab138fb	RegionOne	placement	placement	True	internal	<a href="http://controller:8778">http://controller:8778</a>

## 软件安装与配置

我们这里选择第2种网络类型:

<https://docs.openstack.org/neutron/pike/install/controller-install-option2-rdo.html>

1,在控制节点安装neutron相关软件

```
[root@controller ~]# yum install openstack-neutron  
openstack-neutron-ml2 openstack-neutron-linuxbridge  
ebtables -y
```

2, 备份配置文件

```
[root@controller ~]# cp /etc/neutron/neutron.conf  
/etc/neutron/neutron.conf.bak
```

```
[root@controller ~]# cp  
/etc/neutron/plugins/ml2/ml2_conf.ini  
/etc/neutron/plugins/ml2/ml2_conf.ini.bak
```

```
[root@controller ~]# cp  
/etc/neutron/plugins/ml2/linuxbridge_agent.ini  
/etc/neutron/plugins/ml2/linuxbridge_agent.ini.bak
```

3,配置neutron.conf文件

```
[root@controller ~]# vim /etc/neutron/neutron.conf
```

```
[DEFAULT]
```

```
27 auth_strategy = keystone

30 core_plugin = ml2
33 service_plugins = router
85 allow_overlapping_ips = true

98 notify_nova_on_port_status_changes = true
102 notify_nova_on_port_data_changes = true

553 transport_url =
rabbit://openstack:daniel.com@controller
560 rpc_backend = rabbit

[database]
710 connection =
mysql+pymysql://neutron:daniel.com@controller/neutron

794 [keystone_authtoken]                这句不改,795-803都配置
到[keystone_authtoken]下面
795 auth_uri = http://controller:5000
796 auth_url = http://controller:35357
797 memcached_servers = controller:11211
798 auth_type = password
799 project_domain_name = default
800 user_domain_name = default
801 project_name = service
802 username = neutron
803 password = daniel.com

1022 [nova]                            这句不改,1023-1030都配置
到[nova]下面
1023 auth_url = http://controller:35357
1024 auth_type = password
1025 project_domain_name = default
1026 user_domain_name = default
1027 region_name = RegionOne
1028 project_name = service
```

```
1029 username = nova
1030 password = daniel.com

[oslo_concurrency]
1141 lock_path = /var/lib/neutron/tmp
```

## 配置结果

```
[root@controller ~]# grep -Ev '#|^$'
/etc/neutron/neutron.conf
[DEFAULT]
auth_strategy = keystone
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = true
notify_nova_on_port_status_changes = true
notify_nova_on_port_data_changes = true
transport_url =
rabbit://openstack:daniel.com@controller
rpc_backend = rabbit
[agent]
[cors]
[database]
connection =
mysql+pymysql://neutron:daniel.com@controller/neutron
[keystone_auth_token]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = neutron
password = daniel.com
[matchmaker_redis]
[nova]
```

```
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = nova
password = daniel.com
[oslo_concurrency]
lock_path = /var/lib/neutron/tmp
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
[oslo_middleware]
[oslo_policy]
[quotas]
[ssl]
```

#### 4, 配置Modular Layer 2 (ML2)插件 `ml2_conf.ini` 配置文件

```
[root@controller ~]# vim
/etc/neutron/plugins/ml2/ml2_conf.ini
[ml2]
132 type_drivers = flat,vlan,vxlan
137 tenant_network_types = vxlan
141 mechanism_drivers = linuxbridge,l2population
146 extension_drivers = port_security

[ml2_type_flat]
182 flat_networks = provider

[ml2_type_vxlan]
235 vni_ranges = 1:1000
支持1000个隧道网络(注意:在193行也有1个相同参数,不要配错位置
了,否则无法创建自助的私有网络)
```

```
[securitygroup]
259 enable_ipset = true
组规则效率
```

增强安全

```
[root@controller ~]# grep -Ev '#|^$'
/etc/neutron/plugins/ml2/ml2_conf.ini
[DEFAULT]
[l2pop]
[m12]
type_drivers = flat,vlan,vxlan
tenant_network_types = vxlan
mechanism_drivers = linuxbridge,l2population
extension_drivers = port_security
[m12_type_flat]
flat_networks = provider
[m12_type_geneve]
[m12_type_gre]
[m12_type_vlan]
[m12_type_vxlan]
vni_ranges = 1:1000
[securitygroup]
enable_ipset = true
```

5,配置linuxbridge\_agent.ini文件



```
[root@controller ~]# vim
/etc/neutron/plugins/ml2/linuxbridge_agent.ini
[linux_bridge]
142 physical_interface_mappings = provider:eth1
注意网卡为eth1, 也就是走外网网卡名

[vxlan]
175 enable_vxlan = true
196 local_ip = 192.168.122.11
此IP为管理网卡的IP
220 l2_population = true

[securitygroup]
155 firewall_driver =
neutron.agent.linux.iptables_firewall.IptablesFirewallD
river
160 enable_security_group = true
```

```
[root@controller ~]# grep -Ev '#|^$'
/etc/neutron/plugins/ml2/linuxbridge_agent.ini
[DEFAULT]
[agent]
[linux_bridge]
physical_interface_mappings = provider:eth1
[securitygroup]
firewall_driver =
neutron.agent.linux.iptables_firewall.IptablesFirewallD
river
enable_security_group = true
[vxlan]
enable_vxlan = true
local_ip = 192.168.122.11
l2_population = true
```

## 6, 配置l3\_agent.ini文件

```
[root@controller ~]# vim /etc/neutron/l3_agent.ini
16 interface_driver = linuxbridge
```

```
[root@controller ~]# grep -Ev '#|^$'
/etc/neutron/l3_agent.ini
[DEFAULT]
interface_driver = linuxbridge
[agent]
[ovs]
```

## 7,配置dhcp\_agent.ini文件

```
[root@controller ~]# vim /etc/neutron/dhcp_agent.ini

[DEFAULT]
16 interface_driver = linuxbridge
37 dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
46 enable_isolated_metadata = true
```

```
[root@controller ~]# grep -Ev '#|^$'
/etc/neutron/dhcp_agent.ini
[DEFAULT]
interface_driver = linuxbridge
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
enable_isolated_metadata = true
[agent]
[ovs]
```

## 8,配置metadata\_agent.ini文件

参考:<https://docs.openstack.org/neutron/pike/install/controller-inst-all-rdo.html>

```
[root@controller ~]# vim
/etc/neutron/metadata_agent.ini
[DEFAULT]
23 nova_metadata_host = controller
35 metadata_proxy_shared_secret = metadata_daniel
```

注意:这里的metadata\_daniel仅为一个字符串,需要和nova配置文件里的metadata\_proxy\_shared\_secret对应

9,在nova.conf配置文件中加上下面一段

```
[root@controller ~]# vim /etc/nova/nova.conf

[neutron]                                在[neutron]配置段下添加下面一段
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = daniel.com
service_metadata_proxy = true
metadata_proxy_shared_secret = metadata_daniel
```

```
[root@controller ~]# cat /etc/nova/nova.conf
[DEFAULT]
use_neutron=true
firewall_driver=nova.virt.libvirt.firewall.IptablesFire
wallDriver
enabled_apis=osapi_compute,metadata
transport_url=rabbit://openstack:daniel.com@controller
[api]
auth_strategy=keystone
[api_database]
```

```
connection=mysql+pymysql://nova:daniel.com@controller/nova_api
[barbican]
[cache]
[cells]
[cinder]
[compute]
[conductor]
[console]
[consoleauth]
[cors]
[crypto]
[database]
connection=mysql+pymysql://nova:daniel.com@controller/nova
[ephemeral_storage_encryption]
[filter_scheduler]
[glance]
api_servers=http://controller:9292
[guestfs]
[healthcheck]
[hyperv]
[ironic]
[key_manager]
[keystone]
[keystone_auth_token]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = daniel.com
[libvirt]
[matchmaker_redis]
```

```
[metrics]
[mks]
[neutron]
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = daniel.com
service_metadata_proxy = true
metadata_proxy_shared_secret = metadata_daniel
[notifications]
[osapi_v21]
[oslo_concurrency]
lock_path=/var/lib/nova/tmp
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
[oslo_middleware]
[oslo_policy]
[pci]
[placement]
os_region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://controller:35357/v3
username = placement
password = daniel.com
[quota]
[rdp]
```

```
[remote_debug]
[scheduler]
[serial_console]
[service_user]
[spice]
[trusted_computing]
[upgrade_levels]
[vendordata_dynamic_auth]
[vmware]
[vnc]
enabled=true
vncserver_listen=192.168.122.11
vncserver_proxycient_address=192.168.122.11
[workarounds]
[wsgi]
[xenserver]
[xvp]
```

10, 网络服务初始化脚本需要访问/etc/neutron/plugin.ini来指向ml2\_conf.ini配置文件,所以需要做一个软链接

```
[root@controller ~]# ln -s
/etc/neutron/plugins/ml2/ml2_conf.ini
/etc/neutron/plugin.ini
```

11, 同步数据(时间较长)

```
[root@controller ~]# su -s /bin/sh -c "neutron-db-
manage --config-file /etc/neutron/neutron.conf --
config-file /etc/neutron/plugins/ml2/ml2_conf.ini
upgrade head" neutron
```

## 启动服务

重启nova服务

```
[root@controller ~]# systemctl restart openstack-nova-api.service
```

## 启动neutron服务

```
[root@controller ~]# systemctl start neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service neutron-l3-agent.service
```

```
[root@controller ~]# systemctl enable neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service neutron-l3-agent.service
```

# neutron计算节点部署

---

参考: <https://docs.openstack.org/neutron/pike/install/compute-install-rdo.html>

注意: 下面操作在compute节点操作

## 安装与配置

### 1,安装相关软件

```
[root@compute ~]# yum install openstack-neutron-linuxbridge ebtables ipset -y
```

### 2, 备份配置文件

```
[root@compute ~]# cp /etc/neutron/neutron.conf
/etc/neutron/neutron.conf.bak

[root@compute ~]# cp
/etc/neutron/plugins/ml2/linuxbridge_agent.ini
/etc/neutron/plugins/ml2/linuxbridge_agent.ini.bak
```

### 3, 配置neutron.conf文件

```
[root@compute ~]# vim /etc/neutron/neutron.conf

[DEFAULT]
27 auth_strategy = keystone
553 transport_url =
rabbit://openstack:daniel.com@controller

794 [keystone_authtoken]          在
[keystone_authtoken]下添加下面一段配置
795 auth_uri = http://controller:5000
796 auth_url = http://controller:35357
797 memcached_servers = controller:11211
798 auth_type = password
799 project_domain_name = default
800 user_domain_name = default
801 project_name = service
802 username = neutron
803 password = daniel.com

[oslo_concurrency]
1135 lock_path = /var/lib/neutron/tmp
```

```
[root@compute ~]# grep -Ev '#|^$'
/etc/neutron/neutron.conf
[DEFAULT]
auth_strategy = keystone
transport_url =
rabbit://openstack:daniel.com@controller
```



```
[agent]
[cors]
[database]
[keystone_auth_token]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = neutron
password = daniel.com
[matchmaker_redis]
[nova]
[oslo_concurrency]
lock_path = /var/lib/neutron/tmp
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
[oslo_middleware]
[oslo_policy]
[quotas]
[ssl]
```

4, 仍然是第2类型网络配置

参考:<https://docs.openstack.org/neutron/pike/install/compute-install-option2-rdo.html>

配置linuxbridge\_agent.ini文件

```
[root@compute ~]# vim
/etc/neutron/plugins/ml2/linuxbridge_agent.ini

[linux_bridge]
142 physical_interface_mappings = provider:eth1
为走外部网络网卡名

[vxlan]
175 enable_vxlan = true
196 local_ip = 192.168.122.12
本机管理网络的IP(重点注意)
220 l2_population = true

[securitygroup]
155 firewall_driver =
neutron.agent.linux.iptables_firewall.IptablesFirewallD
river
160 enable_security_group = true
```

```
[root@compute ~]# grep -Ev '#|^$'
/etc/neutron/plugins/ml2/linuxbridge_agent.ini
[DEFAULT]
[agent]
[linux_bridge]
physical_interface_mappings = provider:eth1
[securitygroup]
firewall_driver =
neutron.agent.linux.iptables_firewall.IptablesFirewallD
river
enable_security_group = true
[vxlan]
enable_vxlan = true
local_ip = 192.168.122.12
l2_population = true
```

## 5, 配置nova.conf配置文件

```
[root@compute ~]# vim /etc/nova/nova.conf
```

```
[neutron]
```

在[neutron]下添

加下面一段

```
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = daniel.com
```

```
[root@compute ~]# grep -Ev '#|^$' /etc/nova/nova.conf
```

```
[DEFAULT]
```

```
use_neutron=true
firewall_driver=nova.virt.libvirt.firewall.IptablesFire
wallDriver
enabled_apis=osapi_compute,metadata
transport_url=rabbit://openstack:daniel.com@controller
[api]
auth_strategy=keystone
[api_database]
connection=mysql+pymysql://nova:daniel.com@controller/n
ova_api
[barbican]
[cache]
[cells]
[cinder]
[compute]
[conductor]
[console]
[consoleauth]
```

```
[cors]
[crypto]
[database]
connection=mysql+pymysql://nova:daniel.com@controller/nova
[ephemeral_storage_encryption]
[filter_scheduler]
[glance]
api_servers=http://controller:9292
[guestfs]
[healthcheck]
[hyperv]
[ironic]
[key_manager]
[keystone]
[keystone_auth_token]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = daniel.com
[libvirt]
virt_type=qemu
[matchmaker_redis]
[metrics]
[mks]
[neutron]
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
```

```
project_name = service
username = neutron
password = daniel.com
[notifications]
[osapi_v21]
[oslo_concurrency]
lock_path=/var/lib/nova/tmp
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
[oslo_middleware]
[oslo_policy]
[pci]
[placement]
os_region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://controller:35357/v3
username = placement
password = daniel.com
[quota]
[rdp]
[remote_debug]
[scheduler]
[serial_console]
[service_user]
[spice]
[trusted_computing]
[upgrade_levels]
[vendordata_dynamic_auth]
[vmware]
[vnc]
enabled = True
```

```
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = 192.168.122.12
novncproxy_base_url =
http://controller:6080/vnc_auto.html
[workarounds]
[wsgi]
[xenserver]
[xvp]
```

## 启动服务

1, 在compute节点重启openstack-nova-compute服务

```
[root@compute ~]# systemctl restart openstack-nova-
compute.service
```

2,在compute节点启动neutron-linuxbridge-agent服务

```
[root@compute ~]# systemctl start neutron-linuxbridge-
agent.service
[root@compute ~]# systemctl enable neutron-linuxbridge-
agent.service
```

3,控制节点上验证

```
[root@controller ~]# source admin-openstack.sh
```

```
[root@controller ~]# openstack network agent list
```

ID	Agent Type	Host	Availability Zone	Alive	State	Binary
07e6de23-c49b-4262-bcb8-94b37281ee99	L3 agent	controller	nova	(-)	UP	neutron-l3-agent
336d8499-5aa2-42e1-b8c4-8c0ad59df945	Metadata agent	controller	None	(-)	UP	neutron-metadata-agent
835e672f-1bdf-4339-88fa-a07d499e4f19	Linux bridge agent	controller	None	(-)	UP	neutron-linuxbridge-agent
a3f83692-1dd0-449f-afa9-d64de97f7872	Linux bridge agent	compute	None	(-)	UP	neutron-linuxbridge-agent
ee4dcda2-f211-45b2-8aac-f20fbc375271	DHCP agent	controller	nova	(-)	UP	neutron-dhcp-agent

# 七、dashboard组件horizon

参考: <https://docs.openstack.org/horizon/pike/install/>

## 安装与配置

### 1, 在控制节点安装软件

```
[root@controller ~]# yum install openstack-dashboard -y
```

### 2, 备份配置文件

```
[root@controller ~]# cp /etc/openstack-  
dashboard/local_settings /etc/openstack-  
dashboard/local_settings.bak
```

### 3, 配置local\_settings文件

```
[root@controller ~]# vim /etc/openstack-  
dashboard/local_settings
```

```
38 ALLOWED_HOSTS = ['*',]
```

允许所有,方便测试,生产环境只允许特定IP

```
64 OPENSTACK_API_VERSIONS = {  
66     "identity": 3,  
67     "image": 2,  
68     "volume": 2,  
69     "compute": 2,  
70 }
```

```
75 OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
```

多域支持

```
97 OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = 'Default'
```

默认域名

```
153 SESSION_ENGINE =  
'django.contrib.sessions.backends.cache'           加这  
一句  
154 CACHES = {  
155     'default': {  
156         'BACKEND':  
'django.core.cache.backends.memcached.MemcachedCache',  
157         'LOCATION': 'controller:11211',           表示  
把会话给controller的memcache  
158     },  
159 }
```

```
161 #CACHES = {  
配置了上面一段,则注释这一段  
162 #     'default': {  
163 #         'BACKEND':  
'django.core.cache.backends.locmem.LocMemCache',  
164 #     },  
165 #}
```

```
183 OPENSTACK_HOST = "controller"  
184 OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" %  
OPENSTACK_HOST           改为v3版,而不是v3.0版  
185 OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"  
默认角色
```

```
313 OPENSTACK_NEUTRON_NETWORK = {  
314     'enable_router': True,  
315     'enable_quotas': True,  
316     'enable_ipv6': True,  
317     'enable_distributed_router': True,  
318     'enable_ha_router': True,  
319     'enable_fip_topology_check': True,  
全打开,我们用的是第2种网络类型
```



```
453 TIME_ZONE = "Asia/Shanghai"
```

时区改为亚洲上海

注意: 上面配置文件中所有的True不要写成了true

#### 4, 配置dashborad的httpd子配置文件

```
[root@controller ~]# vim /etc/httpd/conf.d/openstack-  
dashboard.conf
```

```
4 WSGIApplicationGroup %{GLOBAL}
```

第4行加上这一句, 在官方centos文档里没有, 但ubuntu有. 我们这里要加上, 否则后面dashboard访问不了

## 启动服务


---

```
[root@controller ~]# systemctl restart httpd memcached
```

## 登录验证

---

192.168.122.11/dashboard/auth/login/?next=/dashboard/ 120%



# openstack®

Log in

**Domain**

**User Name**

**Password**

或者使用default/demo/daniel.com登录demo用户

Connect

192.168.122.11/dashboard/settings/ 110%

admin

## User Settings

User Settings

**Language \***

简体中文 (zh-cn)

**Timezone \***

UTC +08:00: China (Shanghai) Time

**Items Per Page \***

20

**Log Lines Per Instance \***

35

**Description:**

Modify dashboard settings for your user.

Settings

Help

OpenStack RC File v2

OpenStack RC File v3

Themes:

✓ Default

Material

Sign Out

Save

## 八、块存储组件cinder

参考: <https://docs.openstack.org/cinder/pike/install/>

# cinder控制节点部署

## 数据库配置

参考: <https://docs.openstack.org/cinder/pike/install/cinder-controller-install-rdo.html>

```
[root@controller ~]# mysql -pdaniel.com

MariaDB [(none)]> create database cinder;

MariaDB [(none)]> grant all on cinder.* to
'cinder'@'localhost' identified by 'daniel.com';

MariaDB [(none)]> grant all on cinder.* to 'cinder'@'%'
identified by 'daniel.com';

MariaDB [(none)]> flush privileges;

MariaDB [(none)]> quit
```

```
[root@controller ~]# mysql -h controller -u cinder -
pdaniel.com -e 'show databases';
+-----+
| Database                |
+-----+
| cinder                   |
| information_schema      |
+-----+
```

## 权限配置

1, 创建cinder用户

```
[root@controller ~]# source admin-openstack.sh
```

```
[root@controller ~]# openstack user create --domain
default --password daniel.com cinder
```

```
[root@controller ~]# openstack user list
```

```
+-----+-----+
| ID                      | Name      |
+-----+-----+
| 0f92b4526f91451b81b2dc41f187fbf1 | cinder    |
| 528911ce70634cc296d69ef463d9e3fb | admin     |
| 648ef5d3f85e4894bbbacc8d45f8ebdb | nova      |
| 693998862e8b4261828cc0a356df1234 | glance    |
| 6e68e53c047949ce8f72c54c0dd58c34 | placement |
| 9f35128a10b84b4fa988aa93b67bf712 | neutron   |
| a1fa2787411c432096d4961ddb4e1a03 | demo      |
+-----+-----+
```

2,把cinder用户添加到service项目中，并赋予admin角色

```
[root@controller ~]# openstack role add --project
service --user cinder admin
```

3,创建cinderv2和cinderv3服务

```
[root@controller ~]# openstack service create --name
cinderv2 --description "OpenStack Block Storage"
volumev2
```

```
[root@controller ~]# openstack service create --name
cinderv3 --description "OpenStack Block Storage"
volumev3
```

```
[root@controller ~]# openstack service list
```

```
+-----+-----+-----+
----+
| ID                      | Name      | Type  |
|                          |           |      |
```

```

+-----+-----+-----+
----+
| 2bdd5cdb64d1480c96d70ea945c1c529 | cinderv3 |
volumev3 |
| 2da4060802bf4e4bbf9328fb68b819b6 | keystone |
identity |
| 59c3f3f50fc4466f8f3bbb72ca9a9e70 | glance   | image
|
| 8bfb289223284a939b54f043f786b17f | nova     |
compute  |
| b4cbb4cce6a5446983969e5b6fde51fa | neutron  |
network  |
| d7704f00f8fd4b9aa41881852481da06 | cinderv2 |
volumev2 |
| ebe864d64de14f04b05b67df4dd7b449 | placement |
placement |
+-----+-----+-----+
----+

```

#### 4,创建cinder相关endpoint地址记录

```
[root@controller ~]# openstack endpoint create --region  
RegionOne volumev2 public http://controller:8776/v2/%\  
(project_id)s
```

```
[root@controller ~]# openstack endpoint create --region  
RegionOne volumev2 internal  
http://controller:8776/v2/%(project_id)s
```

```
[root@controller ~]# openstack endpoint create --region  
RegionOne volumev2 admin http://controller:8776/v2/%\  
(project_id)s
```

```
[root@controller ~]# openstack endpoint create --region  
RegionOne volumev3 public http://controller:8776/v3/%\  
(project_id)s
```

```
[root@controller ~]# openstack endpoint create --region  
RegionOne volumev3 internal  
http://controller:8776/v3/%(project_id)s
```

```
[root@controller ~]# openstack endpoint create --region  
RegionOne volumev3 admin http://controller:8776/v3/%\  
(project_id)s
```

使用endpoint list列表来验证(结果太长就不贴出来了)

```
[root@controller ~]# openstack endpoint list
```

## 软件安装与配置

### 1,控制节点安装openstack-cinder包

```
[root@controller ~]# yum install openstack-cinder -y
```

## 2,备份配置文件

```
[root@controller ~]# cp /etc/cinder/cinder.conf  
/etc/cinder/cinder.conf.bak
```

## 3,配置cinder.conf配置文件

```
[root@controller ~]# vim /etc/cinder/cinder.conf  
[DEFAULT]  
283 my_ip = 192.168.122.11  
288 glance_api_servers = http://controller:9292      官档  
没有这一句,要加上和glance的连接  
400 auth_strategy = keystone  
1212 transport_url =  
rabbit://openstack:daniel.com@controller  
1219 rpc_backend = rabbit      官档没有这  
一句,将来版本会去掉,现在尽量加上  
  
[database]  
3782 connection =  
mysql+pymysql://cinder:daniel.com@controller/cinder  
  
4009 [keystone_authtoken]      在  
[keystone_authtoken]下面添加这一段  
4010 auth_uri = http://controller:5000  
4011 auth_url = http://controller:35357  
4012 memcached_servers = controller:11211  
4013 auth_type = password  
4014 project_domain_name = default  
4015 user_domain_name = default  
4016 project_name = service  
4017 username = cinder  
4018 password = daniel.com  
  
[oslo_concurrency]  
4298 lock_path = /var/lib/cinder/tmp
```

## 验证

```
[root@controller ~]# grep -Ev '#|^$'
/etc/cinder/cinder.conf
[DEFAULT]
my_ip = 192.168.122.11
glance_api_servers = http://controller:9292
auth_strategy = keystone
transport_url =
rabbit://openstack:daniel.com@controller
rpc_backend = rabbit
[backend]
[backend_defaults]
[barbican]
[brcd_fabric_example]
[cisco_fabric_example]
[coordination]
[cors]
[database]
connection =
mysql+pymysql://cinder:daniel.com@controller/cinder
[fc-zone-manager]
[healthcheck]
[key_manager]
[keystone_auth_token]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = cinder
password = daniel.com
[matchmaker_redis]
[nova]
[oslo_concurrency]
```



```
lock_path = /var/lib/cinder/tmp
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
[oslo_middleware]
[oslo_policy]
[oslo_reports]
[oslo_versionedobjects]
[profiler]
[ssl]
```

#### 4, 配置nova.conf配置文件

```
[root@controller ~]# vim /etc/nova/nova.conf
```

```
[cinder]
os_region_name = RegionOne
```

找到[cinder],在下面加上这一句

#### 5, 重启openstack-nova-api服务

```
[root@controller ~]# systemctl restart openstack-nova-api.service
```

#### 6, 同步数据库

```
[root@controller ~]# su -s /bin/sh -c "cinder-manage db sync" cinder
```

验证数据库表信息

```
[root@controller ~]# mysql -h controller -u cinder -p daniel.com -e 'use cinder;show tables' |wc -l
```

36

# 启动服务

在控制节点启动服务

```
[root@controller ~]# systemctl start openstack-cinder-api.service openstack-cinder-scheduler.service
[root@controller ~]# systemctl enable openstack-cinder-api.service openstack-cinder-scheduler.service
```

验证

```
[root@controller ~]# netstat -ntlp | grep :8776
tcp        0      0 0.0.0.0:8776          0.0.0.0:*
    LISTEN          *
    13719/python2

[root@controller ~]# openstack volume service list
+-----+-----+-----+-----+-----+
--+-----+
| Binary           | Host       | Zone  | Status |
State | Updated At           |
+-----+-----+-----+-----+-----+
--+-----+
| cinder-scheduler | controller | nova  | enabled | up
| 2019-07-01T15:41:32.000000 |
+-----+-----+-----+-----+-----+
--+-----+
```

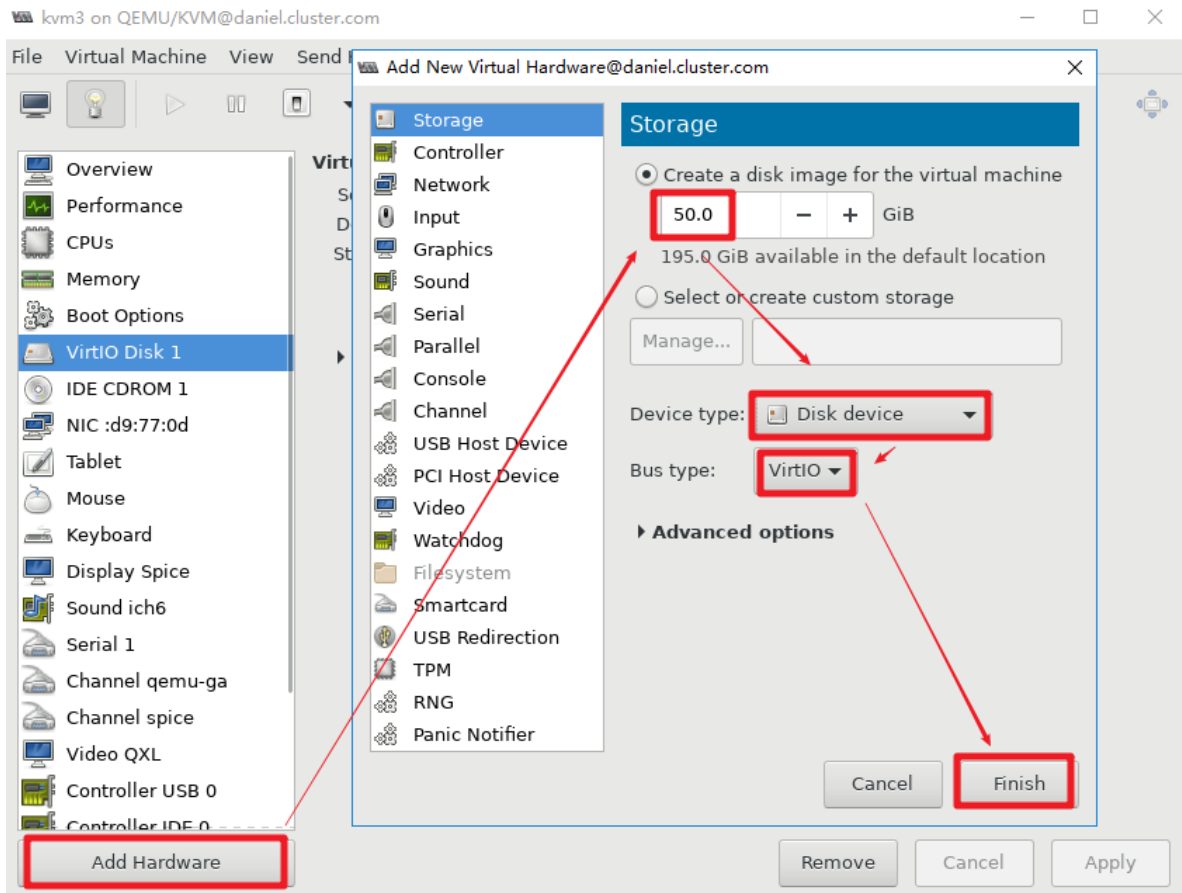
## cinder存储节点部署

参考: <https://docs.openstack.org/cinder/pike/install/cinder-storage-install-rdo.html>

注意: 以下操作在第3台节点(存储节点操作)

## 存储节点添加硬盘

在cinder存储节点添加1个硬盘来模拟存储(如果前面添加过, 这里就可以不再添加)



```
[root@cinder ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0          11:0    1 1024M  0 rom
vda         253:0    0   50G  0 disk
├─vda1      253:1    0  300M  0 part /boot
├─vda2      253:2    0    2G  0 part [SWAP]
└─vda3      253:3    0 47.7G  0 part /
vdb         253:16   0   50G  0 disk
```

确认有vdb这个硬盘了

## 安装与配置

1, 存储节点安装lvm相关软件

```
[root@cinder ~]# yum install lvm2 device-mapper-  
persistent-data -y
```

## 2,启动服务

```
[root@cinder ~]# systemctl start lvm2-lvmetad.service  
[root@cinder ~]# systemctl enable lvm2-lvmetad.service
```

## 3, 创建LVM

```
[root@cinder ~]# pvcreate /dev/vdb  
Physical volume "/dev/vdb" successfully created.  
  
[root@cinder ~]# vgcreate cinder_lvm /dev/vdb  
Volume group "cinder_lvm" successfully created
```

查看pv与vg (注意:如果cinder存储节点安装系统时用的lvm,这里会显示多个,要区分清楚)

```
[root@cinder ~]# pvs  
PV          VG          Fmt  Attr  PSize   PFree  
/dev/vdb    cinder_lvm  lvm2  a--   <50.00g <50.00g  
[root@cinder ~]# vgs  
VG          #PV #LV #SN Attr   VSize   VFree  
cinder_lvm   1   0   0 wz--n- <50.00g <50.00g
```

## 4,配置LVM的过滤

```
[root@cinder ~]# vim /etc/lvm/lvm.conf  
  
142          filter = [ "a/vdb/", "r/.*/" ]
```

增加这句,a代表允许访问accept, r代表拒绝reject

## 5,安装cinder相关软件

```
[root@cinder ~]# yum install openstack-cinder targetcli  
python-keystone -y
```

## 6, 配置cinder.conf配置文件

```
[root@cinder ~]# cp /etc/cinder/cinder.conf  
/etc/cinder/cinder.conf.bak  
  
[root@cinder ~]# vim /etc/cinder/cinder.conf  
  
[DEFAULT]  
283 my_ip = 192.168.122.13           存储节点的管理网络的IP  
288 glance_api_servers = http://controller:9292  
  
400 auth_strategy = keystone  
404 enabled_backends = lvm  
1212 transport_url =  
rabbit://openstack:daniel.com@controller  
1219 rpc_backend = rabbit  
  
[database]  
3782 connection =  
mysql+pymysql://cinder:daniel.com@controller/cinder  
  
4009 [keystone_authtoken]           在  
[keystone_authtoken]下加上一段配置  
4010 auth_uri = http://controller:5000  
4011 auth_url = http://controller:35357  
4012 memcached_servers = controller:11211  
4013 auth_type = password  
4014 project_domain_name = default  
4015 user_domain_name = default  
4016 project_name = service  
4017 username = cinder  
4018 password = daniel.com
```

```
[oslo_concurrency]
4298 lock_path = /var/lib/cinder/tmp

5174 [lvm]                                [lvm]这一段不存在,手动在配置文件最后加上这5行
5175 volume_driver =
cinder.volume.drivers.lvm.LVMVolumeDriver
5176 volume_group = cinder_lvm          一定要和前面创建的vg名一致
5177 iscsi_protocol = iscsi
5178 iscsi_helper = lioadm
```

## 验证配置

```
[root@cinder ~]# grep -Ev '#|^$'
/etc/cinder/cinder.conf
[DEFAULT]
my_ip = 192.168.122.13
glance_api_servers = http://controller:9292
auth_strategy = keystone
enabled_backends = lvm
transport_url =
rabbit://openstack:daniel.com@controller
rpc_backend = rabbit
[backend]
[backend_defaults]
[barbican]
[brcd_fabric_example]
[cisco_fabric_example]
[coordination]
[cors]
[database]
connection =
mysql+pymysql://cinder:daniel.com@controller/cinder
[fc-zone-manager]
[healthcheck]
```

```
[key_manager]
[keystone_authtoken]
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = cinder
password = daniel.com
[matchmaker_redis]
[nova]
[oslo_concurrency]
lock_path = /var/lib/cinder/tmp
[oslo_messaging_amqp]
[oslo_messaging_kafka]
[oslo_messaging_notifications]
[oslo_messaging_rabbit]
[oslo_messaging_zmq]
[oslo_middleware]
[oslo_policy]
[oslo_reports]
[oslo_versionedobjects]
[profiler]
[ssl]
[lvm]
volume_driver =
cinder.volume.drivers.lvm.LVMVolumeDriver
volume_group = cinder_lvm
iscsi_protocol = iscsi
iscsi_helper = lioadm
```

## 启动服务

1,在cinder存储节点启动服务

```
[root@cinder ~]# systemctl start openstack-cinder-  
volume.service target.service  
[root@cinder ~]# systemctl enable openstack-cinder-  
volume.service target.service
```

## 2,在控制节点controller验证

```
[root@controller ~]# openstack volume service list
```

Binary	Host	Zone	Status	State	Updated_at	Disabled Reason
cinder-scheduler	controller	nova	enabled	up	2019-07-02T15:28:24.000000	-
cinder-volume	cinder@lvm	nova	enabled	up	2019-07-02T15:22:20.000000	-

## 3,dashboard上验证

在做cinder前没有"卷"这个选项





退出重新登录就有"卷"这个选项了



## 九、云平台简单使用

参考: [https://docs.openstack.org/zh\\_CN/install-guide/launch-instance.html](https://docs.openstack.org/zh_CN/install-guide/launch-instance.html)

### 创建网络

```
[root@controller ~]# openstack network list
```

```
[root@controller ~]# openstack network create --share --external --provider-physical-network provider --provider-network-type flat provider
```

验证

```
[root@controller ~]# openstack network list
```

```
+-----+-----+-----+
| ID                                     | Name |
Subnets |
+-----+-----+-----+
| 78723928-3bde-4b83-8fb0-4b04096c8f3e | provider |
|                                     |         |
+-----+-----+-----+
```

## 为网络添加子网

创建的网段对应我们eth1网卡的网络

```
[root@controller ~]# openstack subnet create --network provider --allocation-pool start=192.168.100.100,end=192.168.100.250 --dns-nameserver 114.114.114.114 --gateway 192.168.100.1 --subnet-range 192.168.100.0/24 provider
```

验证

```
[root@controller ~]# openstack network list
```

```
+-----+-----+
| ID                                     | Name |
Subnets                               |      |
+-----+-----+
| 78723928-3bde-4b83-8fb0-4b04096c8f3e | provider |
36a0388b-b692-4546-9b50-b6184d8fced7 |
+-----+-----+
```

```
[root@controller ~]# openstack subnet list
```

```
+-----+-----+-----+
| ID                                     | Name |
Network                               | Subnet |
|                                     |      |
+-----+-----+-----+
| 36a0388b-b692-4546-9b50-b6184d8fced7 | provider |
78723928-3bde-4b83-8fb0-4b04096c8f3e | 192.168.100.0/24 |
|                                     |      |
+-----+-----+-----+
```

项目 / 网络 / 网络

正在显示 1 项

名称	已连接的子网	共享的	外部	状态	管理状态	动作
provider	provider 192.168.100.0/24	True	True	运行中	UP	编辑网络

正在显示 1 项

# 创建虚拟机规格(flavor)

```
[root@controller ~]# openstack flavor list
```

```
[root@controller ~]# openstack flavor create --id 0 --  
vcpus 1 --ram 512 --disk 1 m1.nano
```

```
[root@controller ~]# openstack flavor list
```

```
+-----+-----+-----+-----+-----+-----+-----+  
-----+  
| ID | Name      | RAM | Disk | Ephemeral | VCPUs | Is  
Public |  
+-----+-----+-----+-----+-----+-----+-----+  
-----+  
|  0  | m1.nano  | 512 |   1  |           |     1 | True  
      |  
+-----+-----+-----+-----+-----+-----+-----+  
-----+
```

## 创建虚拟机实例

dashboard的admin用户创建虚拟机

正常管理虚拟机不应该使用admin用户,我们在这里简单创建测试一下

### 命令创建VM实例

1,查看镜像,规格,网络等信息

```
[root@controller ~]# openstack image list
```

```
+-----+-----+-----+
--+
| ID                               | Name      |
Status |
+-----+-----+-----+
--+
| 3aa31299-6102-4eab-ae91-84d204255fe2 | cirros    |
active |
+-----+-----+-----+
--+
```

```
[root@controller ~]# openstack flavor list
```

```
+-----+-----+-----+-----+-----+-----+-----+
-----+
| ID | Name      | RAM | Disk | Ephemeral | VCPUs | Is
Public |
+-----+-----+-----+-----+-----+-----+-----+
-----+
| 0   | m1.nano   | 512 | 1    | 0          | 1     | True
      |
+-----+-----+-----+-----+-----+-----+-----+
-----+
```

```
[root@controller ~]# openstack network list
```

```
+-----+-----+-----+
-----+
| ID                               | Name      |
Subnets |
+-----+-----+-----+
-----+
| 78723928-3bde-4b83-8fb0-4b04096c8f3e | provider  |
36a0388b-b692-4546-9b50-b6184d8fced7 |
+-----+-----+-----+
-----+
```

## 2, 创建实例

```
[root@controller ~]# openstack server create --flavor m1.nano --image cirros --nic net-id=78723928-3bde-4b83-8fb0-4b04096c8f3e admin_instance1
```

```
[root@controller ~]# openstack server list
```

ID	Name	Status	Networks	Image	Flavor
7bacabeb-4a07-4bc6-8381-3354042083ee	admin_instance1	ACTIVE	provider=192.168.100.118	cirros	m1.nano

### 3, 查看实例访问的URL地址(每次查询都会变化)

```
[root@controller ~]# openstack console url show admin_instance1
```

Field	Value
type	novnc
url	http://192.168.122.11:6080/vnc_auto.html?token=417bf4d2-fd9e-490e-bff2-3ae708105fcf

### 4,在宿主机上使用firefox访问

```
[root@daniel ~]#  
http://192.168.122.11:6080/vnc_auto.html?  
token=417bf4d2-fd9e-490e-bff2-3ae708105fcf
```

### 5,测试完后删除VM实例做法

```
[root@controller ~]# openstack server delete admin_instance1  
[root@controller ~]# openstack server list
```

# 图形创建VM实例

openstack. Default • admin

admin

项目

访问API

计算

实例

镜像

密钥对

卷

网络

管理员

身份管理

项目 / 计算 / 实例

实例

实例 ID 筛选 创建实例

实例名称	镜像名称	IP 地址	实例类型	密钥对	状态	可用域	任务	电源状态	创建后的时间	动作
没有要显示的条目。										

创建实例

详情

源

实例类型

网络

网络接口

安全组

密钥对

配置

服务器组

scheduler hint

元数据

请提供实例的主机名，欲部署的可用区域和数量。增大数量以创建多个同样配置的实例。

实例名称admin\_instance1

可用域nova

数量1

实例总计 (10 Max)  
10%  
0 当前用量  
1 已添加  
9 剩余量

取消

返回 下一项 创建实例

详情

源

实例类型 \*

网络

网络接口

安全组

密钥对

配置

服务器组

scheduler hint

元数据

实例的源是用来创建实例的模板。可以使用一个镜像、一个实例的快照（镜像快照）、一个卷或一个卷快照（如果启用这个功能）。您也可以通过创建一个新卷来选择使用具有持久性的存储。

?

选择源

镜像

创建新卷

是

否

卷大小 (GB) \*

1

删除实例时删除卷

是

否

已分配

名称	已更新	大小	类型	可见性	
> cirros	7/19/19 8:50 PM	12.65 MB	qcow2	公有	↓

▼ 可用 0

选择一个

名称	已更新	大小	类型	可见性
没有可选项				

✕ 取消

&lt; 返回

下一项 &gt;

创建实例

## 创建实例

✕

详情

源

实例类型

网络

网络接口

安全组

密钥对

配置

服务器组

scheduler hint

元数据

类型管理实例的计算、内存和存储容量的大小。

?

已分配

名称	虚拟内核	内存	磁盘总计	根磁盘	临时磁盘	公有	
> m1.nano	1	512 MB	1 GB	1 GB	0 GB	是	↓

▼ 可用 0

选择一个

名称	虚拟内核	内存	磁盘总计	根磁盘	临时磁盘	公有
----	------	----	------	-----	------	----

✕ 取消

&lt; 返回

下一项 &gt;

创建实例

项目

访问API

计算

概况

实例

镜像

密钥对

卷

网络

管理员

项目 / 计算 / 实例

## 实例

示例 ID

筛选

创建实例

删除实例

更多操作

正在显示 1 项

实例名称	镜像名称	IP 地址	实例类型	密钥对	状态	可用域	任务	电源状态	创建后的时间	动作
admin_instance1	-	192.168.100.111	m1.nano	-	运行	nova	无	运行中	0 分钟	创建快照

正在显示 1 项



# demo用户创建VM实例

## demo用户登录

192.168.122.11/dashboard/auth/login/120%



# openstack.

## 登录

域

default

用户名

demo

密码

daniel.com

连接

## 创建密钥对

openstack. Default • demo demo

项目

访问API

计算

概况

实例

镜像

密钥对

卷

网络

身份管理

项目 / 计算 / 密钥对

## 密钥对

筛选

+ 创建密钥对

导入密钥对

密钥对名称	指纹	动作
没有要显示的条目。		

密钥对是您在实例创建后登陆进去的一种方式。选择一个您易于识别的密钥对名字，名字只能由半角的字母、数字、空格、减号组成。

密钥对名称 \*

key1

## 创建密钥对

### 拷贝私钥到剪贴板

密钥对名称\*

私钥

```

nirp4kECyT EAsqpsfNfFbqmq00Qnri0i0i0000aizByA134Eny0zP034Zy20Jic
yYvEYrTOiXTSiPhiz4ZLi9TD3OUxRLv42vjfwvdPEDke9dWww09VF1woECPjd5i2
LVEHh6lyLbNiQJPu1W49G33UHHcZh8c+rxS9pfGiN8RRV0Omww8iGfzUCgYEAyvZT
ImpXYI8572XVmxX4A61QkK4sm1xb0shENnu9p2wJS6QZqlqk/nQulFanRli+fq47A
5CFPE8Erj2ce0zDsnZB4aJVJZuZZJmhw4N8Bv7dF7vDh04R9Md9CeRWJrgfc1fE8
YKqaodXrJozzPp2ddjLW4lkpLit3ErOuFqww4zsCgYAjWho70TepGzPJZ9HHuewK
WRkK4x10JpPLIK2UshIBKJUntoGMU306dwwXWN6w0bqRRd1VhOqHc0QHmTIRNi3
TEZyi11vrfvWz+ZwjFh78j7PmRni/75KfS8pBNdZbUOEN5uUjJoZKrr78X85fg
jH52Ny79563QiCsUDECrjQKBgCS94vP3GGGjB5ieQDTZlvS3DrUzQ5y8cx+pJqY
XATq8ePxmS2Bq8zEi4ofBbjLp0QPW6vwu7f/yDr1mAbzXGJjvQJxxA/0sz7YuVZ8
CH0EWlyg3sDDw0Y7/HsFCYWpttkkESIOwF97sk4QRhiXv+Gd4ninW2UYST4BeQpx
sYfxAoGBAJUF/jKcQ7zS+0x6mBGLBQKuL0IEQ1/DKzOsVmMiFzYHjwos6F5c0W++
1gtQ6rBRHueZkxpXpH7HxUJBBZhrclrv+WQwcuU2lehq1PxXJ6OuMr/Cfk0XjLZ
vBa9ntyOXY2pal0TMjM51s29pkcemYY6yHa8NVQavsHoFCPCZ3
-----END RSA PRIVATE KEY-----

```

可以拷贝到剪贴板,粘贴到客户端,也可以后面再下载

## 创建密钥对

## 拷贝私钥到剪贴板

完成

## 创建安全组

openstack

Default • demo

demo

项目

访问API

计算

卷

网络

网络拓扑

网络

路由

安全组

浮动IP

身份管理

项目 / 网络 / 安全组

安全组

筛选

+ 创建安全组

删除安全组

正在显示 1 项

名称	安全组ID	描述	动作
default	a2f51626-a5e7-48bb-98bf-285aa8ac630f	Default security group	管理规则

正在显示 1 项

## 创建安全组

名称 \*

group1

描述

only allow ping and ssh

### 说明：

安全组是作用于虚拟机网络接口上的一组IP过滤规则。安全组创建完成后，你可以向其中添加规则。

取消

创建安全组

网络拓扑

网络

路由

安全组

浮动IP

正在显示 2 项

筛选

+ 创建安全组

删除安全组

名称	安全组ID	描述	动作
default	a2f51626-a5e7-48bb-98bf-285aa8ac630f	Default security group	管理规则
group1	3a12609a-c301-4b8d-8226-dbe22eb41cb7	only allow ping and ssh	管理规则

正在显示 2 项

点击管理规则进行规则修改与添加

## 添加规则

规则\*

定制ICMP规则

方向

入口

类型?

编码?

远程\*

CIDR

CIDR?

0.0.0.0/0

### 说明：

实例可以关联安全组，组中的规则定义了允许哪些访问到达被关联的实例。安全组由以下三个主要组件组成：

**规则：**您可以指定期望的规则模板或者使用定制规则，选项有定制TCP规则、定制UDP规则或定制ICMP规则。

**打开端口/端口范围：**您选择的TCP和UDP规则可能会打开一个或一组端口。选择“端口范围”，您需要提供开始和结束端口的范围。对于ICMP规则您需要指定ICMP类型和代码。

**远程：**您必须指定允许通过该规则的流量来源。可以通过以下两种方式实现：IP地址块(CIDR)或者来源地址组(安全组)。如果选择一个安全组作为来源地址，则该安全组中的任何实例都被允许使用该规则访问任一其它实例。

取消

添加

+ 添加规则

删除规则

远端安全组

动作

删除规则

删除规则

## 添加规则

规则\*

定制TCP规则

方向

入口

打开端口\*

端口

端口?

22

远程\*

CIDR

CIDR?

0.0.0.0/0

### 说明：

实例可以关联安全组，组中的规则定义了允许哪些访问到达被关联的实例。安全组由以下三个主要组件组成：

**规则：**您可以指定期望的规则模板或者使用定制规则，选项有定制TCP规则、定制UDP规则或定制ICMP规则。

**打开端口/端口范围：**您选择的TCP和UDP规则可能会打开一个或一组端口。选择“端口范围”，您需要提供开始和结束端口的范围。对于ICMP规则您需要指定ICMP类型和代码。

**远程：**您必须指定允许通过该规则的流量来源。可以通过以下两种方式实现：IP地址块(CIDR)或者来源地址组(安全组)。如果选择一个安全组作为来源地址，则该安全组中的任何实例都被允许使用该规则访问任一其它实例。

取消

添加

+ 添加规则

删除规则

远端安全组

动作

删除规则

删除规则

删除规则

管理安全组规则：group1 (3a12609a-c301-4b8d-8226-dbe22eb41cb7)

+ 添加规则

删除规则

正在显示 4 项

<input type="checkbox"/>	方向	以太网类型 (EtherType)	IP协议	端口范围	远端IP前缀	远端安全组	动作
<input type="checkbox"/>	出口	IPv6	任何	任何	:::0	-	<div>删除规则</div>
<input type="checkbox"/>	出口	IPv4	任何	任何	0.0.0.0/0	-	<div>删除规则</div>
<input type="checkbox"/>	入口	IPv4	ICMP	任何	0.0.0.0/0	-	<div>删除规则</div>
<input type="checkbox"/>	入口	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	<div>删除规则</div>

正在显示 4 项

创建自助私有网络

openstack. Default • demo demo

项目

访问API

计算

卷

网络

网络拓扑

路由

安全组

浮动IP

身份管理

项目 / 网络 / 网络

网络

名称 = 筛选 + 创建网络 删除网络

正在显示 1 项

<input type="checkbox"/>	名称	已连接的子网	共享的	外部	状态	管理状态	动作
<input type="checkbox"/>	provider	provider 192.168.100.0/24	True	True	运行中	UP	

正在显示 1 项

创建网络

网络子网子网详情

网络名称demo\_net1

☒ 启用管理员状态

☒ 创建子网

取消

<< 返回

下一步 >>

创建一个新的网络。额外地，网络中的子网可以在向导的下一步中创建。

## 创建网络



网络

子网

子网详情

子网名称

demo\_subnet1

网络地址

192.168.198.0/24

IP版本

IPv4

自定义的一个私有网络

网关IP

192.168.198.1

☐ 禁用网关

创建关联到这个网络的子网。您必须输入有效的“网络地址”和“网关IP”。如果您不输入“网关IP”，将默认使用该网络的第一个IP地址。如果您不想使用网关，请勾选“禁用网关”复选框。点击“子网详情”标签可进行高级配置。

取消

« 返回

下一步 »

网络

子网

子网详情

☒ 激活DHCP

为子网指定扩展属性

分配地址池

192.168.198.100,192.168.198.200

dhcp分配的IP地址范围,中间是逗号点?图标可以查看格式帮助

DNS服务器

114.114.114.114

主机路由

取消

« 返回

已创建

项目 / 网络 / 网络

成功：网络 "demo\_net1" 已创建

访问API

网络

网络

正在显示 2 项

名称 =

筛选

+ 创建网络

删除网络

	名称	已连接的子网	共享的	外部	状态	管理状态	动作
<input type="checkbox"/>	demo_net1	demo_subnet1 192.168.198.0/24	False	False	运行中	UP	编辑网络
<input type="checkbox"/>	provider	provider 192.168.100.0/24	True	True	运行中	UP	

正在显示 2 项

网络

网络拓扑

拓扑 图表

小 正常

路由

安全组

浮动IP

创建实例

+ 创建网络

+ 新建路由

provider

demo\_net1

网络创建好了,但没有连接起来,所以需要创建路由

新建路由

路由名称

demo\_router1

说明：

基于特殊参数创建一路由。

☒ 启用管理员状态

外部网络

provider

取消

新建路由





# 增加接口

子网\*

demo\_net1: 192.168.198.0/24 (demo\_subnet1)

IP地址(可选)

192.168.198.1

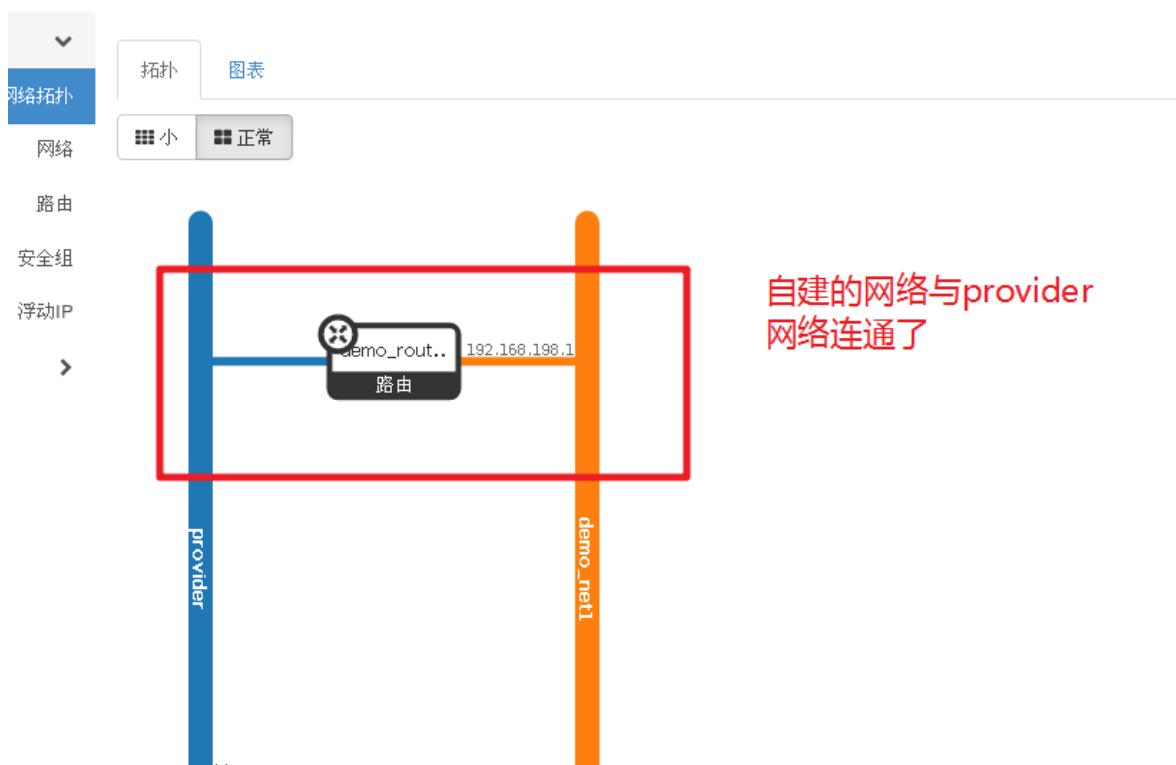
说明:

您可以将一个指定的子网连接到路由器

这里如果你不指定一个IP地址, 则会使用被选定子网的网关地址作为路由器上新建接口的IP地址。如果网关IP地址已经被使用, 你必须使用选定子网的其它地址。

取消

提交



## 创建实例

openstack

Default • demo

demo

项目

访问API

计算

概况

实例

镜像

密钥对

卷

网络

项目 / 计算 / 实例

实例

实例ID

筛选

创建实例

实例名称	镜像名称	IP 地址	实例类型	密钥对	状态	可用域	任务	电源状态	创建后的时间	动作
没有要显示的条目。										

## 创建实例



详情

源

实例类型

网络

网络接口

安全组

密钥对

配置

服务器组

scheduler hint

元数据

请提供实例的主机名，欲部署的可用区域和数量。增大数量以创建多个同样配置的实例。

实例名称 <sup>\*</sup>

demo\_vm1

可用域

nova

数量 <sup>\*</sup>

1

实例总计 (10 Max)

10%

0 当前用量

1 已添加

9 剩余量

取消

返回

下一项 >

创建实例

## 创建实例



详情

源

实例类型

网络

网络接口

安全组

密钥对

配置

服务器组

scheduler hint

元数据

实例的源是用来创建实例的模板。可以使用一个镜像、一个实例的快照（镜像快照）、一个卷或一个卷快照（如果启用这个功能）。您也可以通过创建一个新卷来选择使用具有持久性的存储。

选择源

镜像

创建新卷

是 否

卷大小 (GB) <sup>\*</sup>

1

删除实例时删除卷

是 否

已分配

名称	已更新	大小	类型	可见性	
> cirros	7/19/19 8:50 PM	12.65 MB	qcow2	公有	↓

可用 0 选择一个

Q 点击这里过滤

名称	已更新	大小	类型	可见性
没有可选项				

取消

返回

下一项 >

创建实例

## 创建实例



详情

类型管理实例的计算、内存和存储容量的大小。



源

已分配

实例类型

名称	虚拟内核	内存	磁盘总计	根磁盘	临时磁盘	公有
m1.nano	1	512 MB	1 GB	1 GB	0 GB	是

网络 \*

可用 0

选择一个

网络接口

点击这里过滤

安全组

名称	虚拟内核	内存	磁盘总计	根磁盘	临时磁盘	公有
----	------	----	------	-----	------	----

密钥对

配置

服务器组

scheduler hint

元数据

取消

返回

下一项 >

创建实例

## 创建实例



详情

在云中，网络为实例提供通信通道。



源

已分配 1

从下拉列表选择网络

实例类型

网络	已连接的子网	共享的	管理员状态	状态
demo_net1	demo_subnet1	否	正常	运行中

网络

可用 1

至少选择一个网络

网络接口

点击这里过滤

安全组

网络	已连接的子网	共享的	管理员状态	状态
----	--------	-----	-------	----

密钥对

provider	provider	是	正常	运行中
----------	----------	---	----	-----

配置

服务器组

scheduler hint

元数据

取消

返回

下一项 >

创建实例

使用自建的私有网络

## 创建实例



详情

端口（Ports）为您的实例提供了额外的通信渠道。您可以选择端口而非网络或者二者都选。



源

已分配

选择下面列出的端口。

实例类型

从以下可选项中选择一项

网络

可用 0

选择一个

网络接口

Q 筛选

安全组

名称 IP 管理员状态 状态

密钥对

没有可选项

配置

服务器组

scheduler hint

元数据

取消

返回

下一项

创建实例

## 创建实例



详情

要在其中启动实例的安全组。



源

已分配 1

实例类型

名称 描述  
group1 only allow ping and ssh

网络

可用 1

选择一个或多个

网络接口

Q 点击这里过滤

安全组

名称 描述  
default Default security group

密钥对

配置

服务器组

scheduler hint

元数据

取消

返回

下一项

创建实例

使用自建的安全组

详情

源

实例类型

网络

网络接口

安全组

密钥对

配置

服务器组

scheduler hint

元数据

密钥对允许您SSH到您新创建的实例。您可以选择一个已存在的密钥对、导入一个密钥对或生成一个新的密钥对。

+ 创建密钥对

+ 导入密钥对

已分配

正在显示 1 项

名称	指纹
> key1	42:c1:49:ca:e8:08:47:28:c2:74:0f:17:27:27:65:40

正在显示 1 项

▼ 可用 0

选择一个

Q 点击这里过滤

✕

正在显示 0 项

名称	指纹
没有可显示的条目。	

正在显示 0 项

✕ 取消

< 返回

下一项 >

创建实例

使用创建好的密钥对

实例

概况

实例

镜像

密钥对

正在显示 1 项

示例 ID

筛选

创建实例

删除实例

更多操作

实例名称	镜像名称	IP 地址	实例类型	密钥对	状态	可用域	任务	电源状态	创建后的时间	动作
demo_vm1	-	192.168.198.103	m1.nano	key1	运行	nova	无	运行中	0 分钟	创建快照

正在显示 1 项

## 验证

### 控制台验证

```
3.034797] Freeing unused kernel memory: 1164k freed
further output written to /dev/ttyS0

login as 'cirros' user. default password: 'cubswin:)', use 'sudo' for root.
demo-vm1 login: cirros
Password:
$ ifconfig lhead -2
eth0      Link encap:Ethernet  HWaddr FA:16:3E:90:33:D9
          inet addr:192.168.198.103  Bcast:192.168.198.255  Mask:255.255.255.0
$ ping qq.com
PING qq.com (59.37.96.63): 56 data bytes
64 bytes from 59.37.96.63: seq=0 ttl=54 time=8.314 ms
64 bytes from 59.37.96.63: seq=1 ttl=54 time=11.702 ms

--- qq.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 8.314/10.008/11.702 ms
$
```

确认自助  
网络可以  
上外网

```
$ sudo passwd root
Changing password for root
New password:
Bad password: too short
Retype password:
Password for root changed by root
$
```

可以修改一个简单的root密码

控制节点ssh连接 (现在还无法直接连接自助网络,使用特殊方法,详见如下文档:)

```
[root@controller ~]# openstack network list
```

ID	Name
Subnets	
2d0bc22c-e94b-4efa-ad94-4e5c8efbd606	demo_net1
644b2496-f7cc-4a72-a64f-666783ddd96d	
78723928-3bde-4b83-8fb0-4b04096c8f3e	provider
36a0388b-b692-4546-9b50-b6184d8fced7	

```
[root@controller ~]# source demo-openstack.sh
```

```
[root@controller ~]# openstack server list
```

```
+-----+-----+-----+-----+
| ID                                           | Name       |
Status | Networks                               | Image | Flavor |
+-----+-----+-----+-----+
| 87aa0528-6f5d-498a-b074-c665a3d2032c | demo_vm1 |
ACTIVE | demo_net1=192.168.198.103 |        | m1.nano |
+-----+-----+-----+-----+
```

使用 `ip netns exec qdhcp-网络ID ssh 用户名@IP` 连接

(qdhcp-网络ID也可以通过 `ip netns list` 查询得到,ns是 namespace,用于资源隔离)

```
[root@controller ~]# ip netns exec qdhcp-2d0bc22c-e94b-4efa-ad94-4e5c8efbd606 ssh root@192.168.198.103
```

练习:

1, 创建provider网络的default安全组VM实例

结果: 不能ping通,不能ssh连接(因为default安全组默认拒绝了)

2,创建provider网络的自建安全组(前面创建的允许icmp和ssh)VM实例

结果:在controller节点可以ping通,可以ssh连接,也可以ssh免密连接

```
ssh -i key1 cirros@IP
```

问题: 到底怎么样可以访问前面self-service自助网络的VM实例呢?

# 课后作业

1.让外部可以访问self-service自助网络的VM实例

参考: [https://docs.openstack.org/zh\\_CN/install-guide/launch-instance-selfservice.html](https://docs.openstack.org/zh_CN/install-guide/launch-instance-selfservice.html)

## 实例

示例 ID =

筛选

创建实例

删除实例

更多操作 ▾

正在显示 1 项

<input type="checkbox"/>	实例名称	镜像名称	IP 地址	实例类型	密钥对	状态	可用域	任务	电源状态	创建后的时间	动作
<input type="checkbox"/>	demo-vm1	-	192.168.198.101	m1.nano	keypair1	运行	nova	无	运行中	0 分钟	创建快照 ▾

正在显示 1 项

绑定浮动IP

连接接口

分离接口

编辑实例

连接卷

分离卷

## 管理浮动IP的关联

IP 地址 \*

选择一个IP地址 ▾

+

请为选中的实例或端口选择要绑定的IP地址。

点+号增加一个管理浮动IP

待连接的端口 \*

demo-vm1: 192.168.198.101 ▾

取消

关联



## 分配浮动IP



资源池 \*

provider

说明：

从指定的浮动IP池中分配一个浮动IP。

项目配额

浮动IP

1 已使用，共 50

从provider网络里自动分配一个IP

取消

分配IP

## 管理浮动IP的关联



IP 地址 \*

确认分配的管理浮动IP

192.168.100.106

请为选中的实例或端口选择要绑定的IP地址。

待连接的端口 \*

demo-vm1: 192.168.198.101

取消

关联

正在显示 1 项

<input type="checkbox"/>	实例名称	镜像名称	IP 地址	实例类型	密钥对	状态	可用域	任务	电源状态	创建后的时间	动作
<input type="checkbox"/>	demo-vm1	-	192.168.198.101 浮动IP: 192.168.100.106	m1.nano	keypair1	运行	nova	无	运行中	5 分钟	创建快照

正在显示 1 项

可以使用此IP进行ping和ssh

ping浮动管理IP可以ping通

```
[root@controller ~]# ping -c 2 192.168.100.106
```

```
PING 192.168.100.106 (192.168.100.106) 56(84) bytes of data.
```

```
64 bytes from 192.168.100.106: icmp_seq=1 ttl=63  
time=3.01 ms
```

```
64 bytes from 192.168.100.106: icmp_seq=2 ttl=63  
time=0.967 ms
```

```
--- 192.168.100.106 ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time
999ms
```

```
rtt min/avg/max/mdev = 0.967/1.992/3.017/1.025 ms
```

ssh -i指定密钥对的私钥连接也可以免密登录

```
[root@controller ~]# ssh -i keypair1
```

```
cirros@192.168.100.106
```

```
The authenticity of host '192.168.100.106
(192.168.100.106)' can't be established.
```

```
RSA key fingerprint is
```

```
SHA256:OkdjpTnT5AkhA9m3JN27lV5FQZ02Ql62e9hFU0dSJ3U.
```

```
RSA key fingerprint is
```

```
MD5:94:61:d3:3f:41:30:bb:4c:39:8c:fd:67:00:a2:71:83.
```

```
Are you sure you want to continue connecting (yes/no)?
```

```
yes
```

```
Warning: Permanently added '192.168.100.106' (RSA) to
the list of known hosts.
```

```
$ id
```

```
uid=1000(cirros) gid=1000(cirros) groups=1000(cirros)
```

## 2, 导入自定义的镜像

自行网络查询文档