**虚拟机配置及搭建：**<https://blog.csdn.net/fly1574/article/details/103228836>

外部网卡需要配置DNS,保证可以连接外网

# 搭建本地yum源

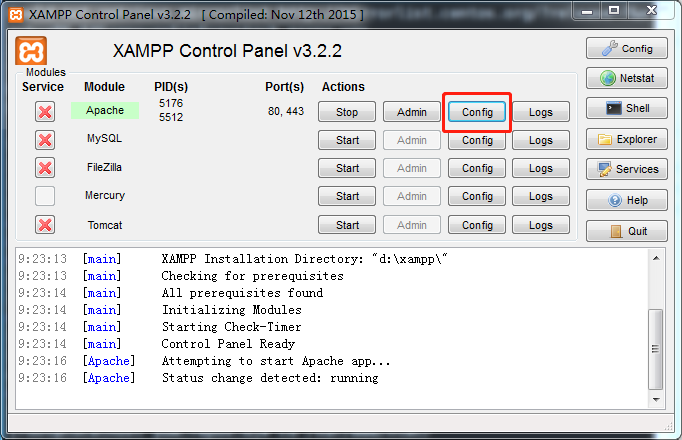
**centos系统**

# yum install httpd

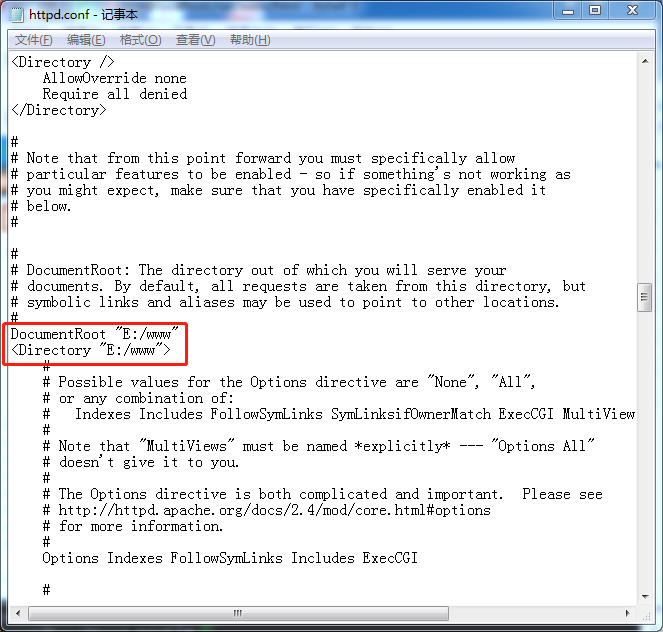
然后将文件复制到目录 /var/www/html中

**windows系统**

安装xampp软件，打开Apache服务，将文件放在某个磁盘下的www文件夹下，



点击config下的Apache（http.conf）



改成存放文件的目录，打开Apache服务

关闭防火墙

# systemctl disable firewalld

# systemctl stop firewalld

在要装的服务器的/etc/yum.repos.d/下添加文件(IP:192.168.1.18)

把IP改为存有源文件的主机IP

yum7.repo

[centos]

name=centos7.2

baseurl= http://192.168.1.211/centos7.2/7.2/os/x86\_64/

enabled=1

gpgcheck=0

[epel]

name=epel7

baseurl= http://192.168.1.211/epel/7/x86\_64/

enabled=1

gpgcheck=0

mitaka.repo

[openstack]

name=mitaka

baseurl= http://192.168.1.211/openstack-mitaka/

enabled=1

gpgcheck=0

mariadb.repo

[mariadb]

name=mariadb10.1

baseurl= http://192.168.1.211/mariadb10.1/

enabled=1

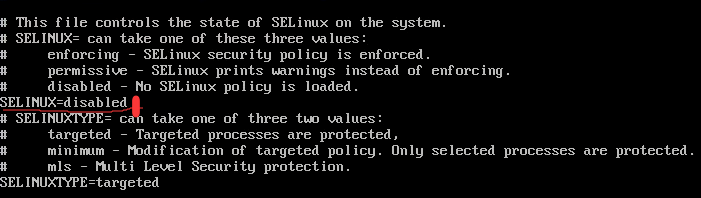
gpgcheck=0

# yum clean all

# yum makecache

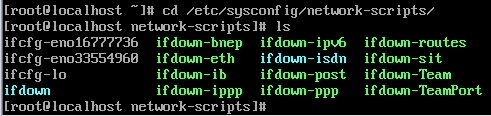
关闭selinux

# vi /etc/sysconfig/selinux

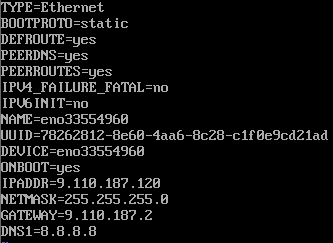


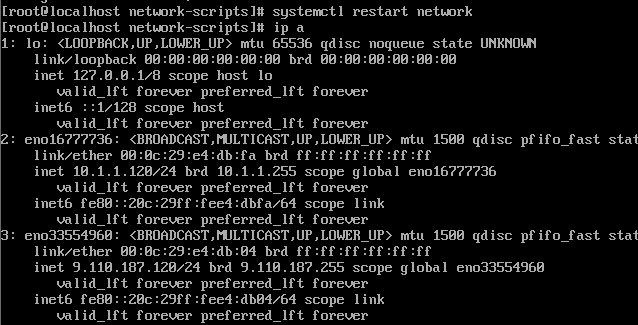
# 配置网络

# cd /etc/sysconfig/network-scripts/









# 更改主机名

# vi /etc/hostname

为配置的主机命名,暂命名为controller，保存后通过reboot重启后生效

配置域名解析，在所有节点的，/etc/hosts文件下添加以下内容（注意更改IP地址）

# controller

10.0.0.11 controller

# compute1

10.0.0.31 compute1

# block1

10.0.0.41 block1

# object1

10.0.0.51 object1

# object2

10.0.0.52 object2

验证连通性

在控制节点ping compute1

# ping -c 4 compute1

PING openstack.org (174.143.194.225) 56(84) bytes of data.

64 bytes from 174.143.194.225: icmp\_seq=1 ttl=54 time=18.3 ms

64 bytes from 174.143.194.225: icmp\_seq=2 ttl=54 time=17.5 ms

64 bytes from 174.143.194.225: icmp\_seq=3 ttl=54 time=17.5 ms

64 bytes from 174.143.194.225: icmp\_seq=4 ttl=54 time=17.4 ms

--- openstack.org ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3022ms

rtt min/avg/max/mdev = 17.489/17.715/18.346/0.364 ms

在其他节点ping controller，（多个节点互相ping一下）

# 安装准备

## 时间同步（所有节点）

控制节点

安装软件包

# yum install chrony

编辑/etc/chrony.conf 文件

server cn.pool.ntp.org iburst（注释剩余server语句）（不用管）

allow 192.168.0.0/16 去掉注释，允许其他节点网段同步时间，请配置为对应网段

启动 NTP 服务并将其配置为随系统启动

# systemctl enable chronyd.service

# systemctl start chronyd.service

其余节点

安装软件包：

# yum install chrony

编辑``/etc/chrony.conf`` 文件并注释除``server`` 值外的所有内容。修改它引用控制节点：（除了改server，其他不用管）

server controller（服务器IP地址） iburst

启动 NTP 服务并将：其配置为随系统启动：

# systemctl enable chronyd.service

# systemctl start chronyd.service

验证操作

在控制节点上执行这个命令：

# chronyc sources

210 Number of sources = 2

MS Name/IP address Stratum Poll Reach LastRx Last sample

===============================================================================

^- 192.0.2.11 2 7 12 137 -2814us[-3000us] +/- 43ms

^\* 192.0.2.12 2 6 177 46 +17us[ -23us] +/- 68ms

出现星号表示同步成功（联网情况下才会出现）

运行timedatectl:查看当前时间是否准确，其中NTP synchronized: yes说明同步成功

在其他节点上运行

# chronyc sources

210 Number of sources = 1

MS Name/IP address Stratum Poll Reach LastRx Last sample

===============================================================================

^\* controller 3 9 377 421 +15us[ -87us] +/- 15ms

## 安装OpenStack客户端（所有节点）

# yum install python-openstackclient

# yum install openstack-selinux

## SQL数据库（控制节点）

安装软件包：

# yum install mariadb mariadb-server python2-PyMySQL

创建并编辑 /etc/my.cnf.d/openstack.cnf，然后完成如下动作：

在 [mysqld] 部分，设置 [``](https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/environment-sql-database.html#id1)bind-address``值为控制节点的管理网络IP地址以使得其它节点可以通过管理网络访问数据库：

[mysqld]

...

bind-address = 10.0.0.11

在``[mysqld]`` 部分，设置如下键值来启用一起有用的选项和 UTF-8 字符集：

[mysqld]

...

default-storage-engine = innodb

innodb\_file\_per\_table

max\_connections = 4096

collation-server = utf8\_general\_ci

character-set-server = utf8

完成安装

启动数据库服务，并将其配置为开机自启：

# systemctl enable mariadb.service

# systemctl start mariadb.service

为了保证数据库服务的安全性，运行``mysql\_secure\_installation``脚本。特别需要说明的是，为数据库的root用户设置一个适当的密码。（先回车，再输入新密码）

# mysql\_secure\_installation

## 消息队列（控制节点）

安装包：

# yum install rabbitmq-server

启动消息队列服务并将其配置为随系统启动：

# systemctl enable rabbitmq-server.service

# systemctl start rabbitmq-server.service

添加 openstack 用户：

# rabbitmqctl add\_user openstack RABBIT\_PASS

Creating user "openstack" ...

...done.

用合适的密码替换 RABBIT\_DBPASS。

给``openstack``用户配置写和读权限：

# rabbitmqctl set\_permissions openstack ".\*" ".\*" ".\*"

Setting permissions for user "openstack" in vhost "/" ...

...done.

## Memcached（控制节点）

安装软件包：

# yum install memcached python-memcached

编辑/etc/sysconfig/memcached文件并完成以下操作：

配置服务以使用控制器节点的管理IP地址。这是为了允许其他节点通过管理网络进行访问：

OPTIONS="-l 127.0.0.1,::1,controller" （controller为控制节点管理IP地址）

启动Memcached服务，并且配置它随机启动。

# systemctl enable memcached.service

# systemctl start memcached.service

## Etcd服务（非必要安装）

安装软件包：

# yum install etcd

编辑/etc/etcd/etcd.conf文件，并设置

ETCD\_INITIAL\_CLUSTER，

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS，

ETCD\_ADVERTISE\_CLIENT\_URLS，

ETCD\_LISTEN\_CLIENT\_URLS为控制器节点管理IP地址，以使其他节点经由管理网络进行访问：

*#[Member]*

ETCD\_DATA\_DIR="/var/lib/etcd/default.etcd"

ETCD\_LISTEN\_PEER\_URLS="http://10.0.0.11:2380"

ETCD\_LISTEN\_CLIENT\_URLS="http://10.0.0.11:2379"

ETCD\_NAME="controller"

*#[Clustering]*

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://10.0.0.11:2380"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://10.0.0.11:2379"

ETCD\_INITIAL\_CLUSTER="controller=http://10.0.0.11:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-cluster-01"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

# 认证服务（keystone）（控制节点）

在你配置 OpenStack 身份认证服务前，你必须创建一个数据库和管理员令牌。

完成下面的步骤以创建数据库：

用数据库连接客户端以 root 用户连接到数据库服务器：

$ mysql -u root -p

创建 keystone 数据库：

CREATE DATABASE keystone;

对``keystone``数据库授予恰当的权限：

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' \

IDENTIFIED BY 'KEYSTONE\_DBPASS';

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' \

IDENTIFIED BY 'KEYSTONE\_DBPASS';

用合适的密码替换 KEYSTONE\_DBPASS 。

刷新数据库

FLUSH privileges;

退出数据库客户端。

生成一个随机值作为在初始的配置中的管理员的令牌。

$ openssl rand -hex 10

（本次安装生成结果：5f798af980bf38a5b114）

（安装生成结果：67c7626d0bb3eeb9546b）

安全并配置组件

运行以下命令来安装包。

# yum install openstack-keystone httpd mod\_wsgi

编辑文件 /etc/keystone/keystone.conf 并完成如下动作：

在``[DEFAULT]``部分，定义初始管理令牌的值：

[DEFAULT]

...

admin\_token = ADMIN\_TOKEN

使用前面步骤生成的随机数替换``ADMIN\_TOKEN`` 值。

在 [database] 部分，配置数据库访问：

[database]

...

connection = mysql+pymysql://keystone:KEYSTONE\_DBPASS@controller/keystone

将``KEYSTONE\_DBPASS``替换为你为数据库选择的密码。

在``[token]``部分，配置Fernet UUID令牌的提供者。

[token]

...

provider = fernet

初始化身份认证服务的数据库：

# su -s /bin/sh -c "keystone-manage db\_sync" keystone

注解

忽略输出中任何不推荐使用的信息。

初始化Fernet keys：

keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone

配置 Apache HTTP 服务器

编辑``/etc/httpd/conf/httpd.conf`` 文件，配置``ServerName`` 选项为控制节点：

ServerName controller

用下面的内容创建文件 /etc/httpd/conf.d/wsgi-keystone.conf。

Listen 5000

Listen 35357

<VirtualHost \*:5000>

WSGIDaemonProcess keystone-public processes=5 threads=1 user=keystone group=keystone display-name=%{GROUP}

WSGIProcessGroup keystone-public

WSGIScriptAlias / /usr/bin/keystone-wsgi-public

WSGIApplicationGroup %{GLOBAL}

WSGIPassAuthorization On

ErrorLogFormat "%{cu}t %M"

ErrorLog /var/log/httpd/keystone-error.log

CustomLog /var/log/httpd/keystone-access.log combined

<Directory /usr/bin>

Require all granted

</Directory>

</VirtualHost>

<VirtualHost \*:35357>

WSGIDaemonProcess keystone-admin processes=5 threads=1 user=keystone group=keystone display-name=%{GROUP}

WSGIProcessGroup keystone-admin

WSGIScriptAlias / /usr/bin/keystone-wsgi-admin

WSGIApplicationGroup %{GLOBAL}

WSGIPassAuthorization On

ErrorLogFormat "%{cu}t %M"

ErrorLog /var/log/httpd/keystone-error.log

CustomLog /var/log/httpd/keystone-access.log combined

<Directory /usr/bin>

Require all granted

</Directory>

</VirtualHost>

完成安装

启动 Apache HTTP 服务并配置其随系统启动：

# systemctl enable httpd.service

# systemctl start httpd.service

创建服务实体和API端点

配置认证令牌：

$ export OS\_TOKEN=ADMIN\_TOKEN

将``ADMIN\_TOKEN``替换为你在 :doc:[`](https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/keystone-services.html#id1)keystone-install`章节中生成的认证令牌。例如：

$ export OS\_TOKEN=5f798af980bf38a5b114

export OS\_TOKEN=67c7626d0bb3eeb9546b

配置端点URL：

$ export OS\_URL=http://controller:35357/v3

配置认证 API 版本：

$ export OS\_IDENTITY\_API\_VERSION=3

创建服务实体和API端点

在你的Openstack环境中，认证服务管理服务目录。服务使用这个目录来决定您的环境中可用的服务。

创建服务实体和身份认证服务：

$ openstack service create \

--name keystone --description "OpenStack Identity" identity

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | OpenStack Identity |

| enabled | True |

| id | 4ddaae90388b4ebc9d252ec2252d8d10 |

| name | keystone |

| type | identity |

+-------------+----------------------------------+

创建认证服务的 API 端点：

$ openstack endpoint create --region RegionOne \

identity public http://controller:5000/v3

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 30fff543e7dc4b7d9a0fb13791b78bf4 |

| interface | public |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 8c8c0927262a45ad9066cfe70d46892c |

| service\_name | keystone |

| service\_type | identity |

| url | http://controller:5000/v3 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne \

identity internal http://controller:5000/v3

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 57cfa543e7dc4b712c0ab137911bc4fe |

| interface | internal |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 6f8de927262ac12f6066cfe70d99ac51 |

| service\_name | keystone |

| service\_type | identity |

| url | http://controller:5000/v3 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne \

identity admin http://controller:35357/v3

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 78c3dfa3e7dc44c98ab1b1379122ecb1 |

| interface | admin |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 34ab3d27262ac449cba6cfe704dbc11f |

| service\_name | keystone |

| service\_type | identity |

| url | http://controller:35357/v3 |

+--------------+----------------------------------+

创建域、项目、用户和角色

 创建域``default``：

$ openstack domain create --description "Default Domain" default

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | Default Domain |

| enabled | True |

| id | e0353a670a9e496da891347c589539e9 |

| name | default |

+-------------+----------------------------------+

 在你的环境中，为进行管理操作，创建管理的项目、用户和角色：

创建 admin 项目：

$ openstack project create --domain default \

--description "Admin Project" admin

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | Admin Project |

| domain\_id | e0353a670a9e496da891347c589539e9 |

| enabled | True |

| id | 343d245e850143a096806dfaefa9afdc |

| is\_domain | False |

| name | admin |

| parent\_id | None |

+-------------+----------------------------------+

 创建 admin 用户：

$ openstack user create --domain default \

--password-prompt admin

User Password:

Repeat User Password:

+-----------+----------------------------------+

| Field | Value |

+-----------+----------------------------------+

| domain\_id | e0353a670a9e496da891347c589539e9 |

| enabled | True |

| id | ac3377633149401296f6c0d92d79dc16 |

| name | admin |

+-----------+----------------------------------+

 创建 admin 角色：

$ openstack role create admin

+-----------+----------------------------------+

| Field | Value |

+-----------+----------------------------------+

| domain\_id | None |

| id | cd2cb9a39e874ea69e5d4b896eb16128 |

| name | admin |

+-----------+----------------------------------+

 添加``admin`` 角色到 admin 项目和用户上：

$ openstack role add --project admin --user admin admin

注解

这个命令执行后没有输出。

本指南使用一个你添加到你的环境中每个服务包含独有用户的service 项目。创建``service``项目：

$ openstack project create --domain default \

--description "Service Project" service

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | Service Project |

| domain\_id | e0353a670a9e496da891347c589539e9 |

| enabled | True |

| id | 894cdfa366d34e9d835d3de01e752262 |

| is\_domain | False |

| name | service |

| parent\_id | None |

+-------------+----------------------------------+

常规（非管理）任务应该使用无特权的项目和用户。作为例子，本指南创建 opt项目和用户。

创建``opt`` 项目：

$ openstack project create --domain default \

--description "Demo Project" opt

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | Demo Project |

| domain\_id | e0353a670a9e496da891347c589539e9 |

| enabled | True |

| id | ed0b60bf607743088218b0a533d5943f |

| is\_domain | False |

| name | demo |

| parent\_id | None |

+-------------+----------------------------------+

在为该项目创建额外用户时，不要重复这一步骤

 创建``opt`` 用户：

$ openstack user create --domain default \

--password-prompt opt

User Password:

Repeat User Password:

+-----------+----------------------------------+

| Field | Value |

+-----------+----------------------------------+

| domain\_id | e0353a670a9e496da891347c589539e9 |

| enabled | True |

| id | 58126687cbcc4888bfa9ab73a2256f27 |

| name | demo |

+-----------+----------------------------------+

 创建 user 角色：

$ openstack role create user

+-----------+----------------------------------+

| Field | Value |

+-----------+----------------------------------+

| domain\_id | None |

| id | 997ce8d05fc143ac97d83fdfb5998552 |

| name | user |

+-----------+----------------------------------+

添加 user``角色到 ``opt项目和用户：

$ openstack role add --project opt --user opt user

注解

这个命令执行后没有输出。

验证操作

因为安全性的原因，关闭临时认证令牌机制：

编辑 /etc/keystone/keystone-paste.ini 文件，从``[pipeline:public\_api]``，[pipeline:admin\_api]``和``[pipeline:api\_v3]``部分删除``admin\_token\_auth 。

重置``OS\_TOKEN``和``OS\_URL`` 环境变量：

$ unset OS\_TOKEN OS\_URL

作为 admin 用户，请求认证令牌：

$ 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:

+------------+-----------------------------------------------------------------+

| Field | Value |

+------------+-----------------------------------------------------------------+

| expires | 2016-02-12T20:14:07.056119Z |

| id | gAAAAABWvi7\_B8kKQD9wdXac8MoZiQldmjEO643d-e\_j-XXq9AmIegIbA7UHGPv |

| | atnN21qtOMjCFWX7BReJEQnVOAj3nclRQgAYRsfSU\_MrsuWb4EDtnjU7HEpoBb4 |

| | o6ozsA\_NmFWEpLeKy0uNn\_WeKbAhYygrsmQGA49dclHVnz-OMVLiyM9ws |

| project\_id | 343d245e850143a096806dfaefa9afdc |

| user\_id | ac3377633149401296f6c0d92d79dc16 |

+------------+-----------------------------------------------------------------+

作为``opt`` 用户，请求认证令牌：

$ openstack --os-auth-url http://controller:5000/v3 \

--os-project-domain-name default --os-user-domain-name default \

--os-project-name opt --os-username opt token issue

Password:

+------------+-----------------------------------------------------------------+

| Field | Value |

+------------+-----------------------------------------------------------------+

| expires | 2016-02-12T20:15:39.014479Z |

| id | gAAAAABWvi9bsh7vkiby5BpCCnc-JkbGhm9wH3fabS\_cY7uabOubesi-Me6IGWW |

| | yQqNegDDZ5jw7grI26vvgy1J5nCVwZ\_zFRqPiz\_qhbq29mgbQLglbkq6FQvzBRQ |

| | JcOzq3uwhzNxszJWmzGC7rJE\_H0A\_a3UFhqv8M4zMRYSbS2YF0MyFmp\_U |

| project\_id | ed0b60bf607743088218b0a533d5943f |

| user\_id | 58126687cbcc4888bfa9ab73a2256f27 |

+------------+-----------------------------------------------------------------+

创建 OpenStack 客户端环境脚本

创建脚本

创建 admin 和 [``](https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/keystone-openrc.html#id1)opt``项目和用户创建客户端环境变量脚本。本指南的接下来的部分会引用这些脚本，为客户端操作加载合适的的凭证。

编辑文件 admin-openrc 并添加如下内容：

export OS\_PROJECT\_DOMAIN\_NAME=default

export OS\_USER\_DOMAIN\_NAME=default

export OS\_PROJECT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://controller:35357/v3

export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

将 ADMIN\_PASS 替换为你在认证服务中为 admin 用户选择的密码。

编辑文件 opt-openrc 并添加如下内容：

export OS\_PROJECT\_DOMAIN\_NAME=default

export OS\_USER\_DOMAIN\_NAME=default

export OS\_PROJECT\_NAME= opt

export OS\_USERNAME= opt

export OS\_PASSWORD=DEMO\_PASS

export OS\_AUTH\_URL=http://controller:5000/v3

export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

将 DEMO\_PASS 替换为你在认证服务中为 opt用户选择的密码。

使用脚本

使用特定租户和用户运行客户端，你可以在运行之前简单地加载相关客户端脚本。例如：

加载``admin-openrc``文件来身份认证服务的环境变量位置和``admin``项目和用户证书：

$ source admin-openrc

请求认证令牌:

$ openstack token issue

+------------+-----------------------------------------------------------------+

| Field | Value |

+------------+-----------------------------------------------------------------+

| expires | 2016-02-12T20:44:35.659723Z |

| id | gAAAAABWvjYj-Zjfg8WXFaQnUd1DMYTBVrKw4h3fIagi5NoEmh21U72SrRv2trl |

| | JWFYhLi2\_uPR31Igf6A8mH2Rw9kv\_bxNo1jbLNPLGzW\_u5FC7InFqx0yYtTwa1e |

| | eq2b0f6-18KZyQhs7F3teAta143kJEWuNEYET-y7u29y0be1\_64KYkM7E |

| project\_id | 343d245e850143a096806dfaefa9afdc |

| user\_id | ac3377633149401296f6c0d92d79dc16 |

+------------+-----------------------------------------------------------------+

# 镜像服务（glance）（控制节点）

安装和配置镜像服务之前，你必须创建创建一个数据库、服务凭证和API端点。

1. 完成下面的步骤以创建数据库：
   * 用数据库连接客户端以 root 用户连接到数据库服务器：
   * $ mysql -u root -p
   * 创建 glance 数据库：

CREATE DATABASE glance;

* + 对``glance``数据库授予恰当的权限：

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' \

IDENTIFIED BY 'GLANCE\_DBPASS';

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' \

IDENTIFIED BY 'GLANCE\_DBPASS';

用一个合适的密码替换 GLANCE\_DBPASS。

* + 退出数据库客户端。

1. 获得 admin 凭证来获取只有管理员能执行的命令的访问权限：
2. $ source admin-openrc
3. 要创建服务证书，完成这些步骤：
   * 创建 glance 用户：

$ openstack user create --domain default --password-prompt glance

User Password:

Repeat User Password:

* + +-----------+----------------------------------+
  + | Field | Value |
  + +-----------+----------------------------------+
  + | domain\_id | e0353a670a9e496da891347c589539e9 |
  + | enabled | True |
  + | id | e38230eeff474607805b596c91fa15d9 |
  + | name | glance |
  + +-----------+----------------------------------+
  + 添加 admin 角色到 glance 用户和 service 项目上。
  + $ openstack role add --project service --user glance admin

注解

这个命令执行后没有输出。

创建``glance``服务实体：

$ openstack service create --name glance \

--description "OpenStack Image" image

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | OpenStack Image |

| enabled | True |

| id | 8c2c7f1b9b5049ea9e63757b5533e6d2 |

| name | glance |

| type | image |

+-------------+----------------------------------+

 创建镜像服务的 API 端点：

$ openstack endpoint create --region RegionOne \

image public http://controller:9292

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 340be3625e9b4239a6415d034e98aace |

| interface | public |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 8c2c7f1b9b5049ea9e63757b5533e6d2 |

| service\_name | glance |

| service\_type | image |

| url | http://controller:9292 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne \

image internal http://controller:9292

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | a6e4b153c2ae4c919eccfdbb7dceb5d2 |

| interface | internal |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 8c2c7f1b9b5049ea9e63757b5533e6d2 |

| service\_name | glance |

| service\_type | image |

| url | http://controller:9292 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne \

image admin http://controller:9292

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 0c37ed58103f4300a84ff125a539032d |

| interface | admin |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | 8c2c7f1b9b5049ea9e63757b5533e6d2 |

| service\_name | glance |

| service\_type | image |

| url | http://controller:9292 |

+--------------+----------------------------------+

安装并配置组件

安装软件包：

# yum install openstack-glance

编辑文件 /etc/glance/glance-api.conf 并完成如下动作：

* + 在 [database] 部分，配置数据库访问：
  + [database]
  + ...

connection = mysql+pymysql://glance:GLANCE\_DBPASS@controller/glance

将``GLANCE\_DBPASS`` 替换为你为镜像服务选择的密码。

* + 在 [keystone\_authtoken] 和 [paste\_deploy] 部分，配置认证服务访问：
  + [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 = glance

password = GLANCE\_PASS

* + [paste\_deploy]
  + ...

flavor = keystone

将 GLANCE\_PASS 替换为你为认证服务中你为 glance 用户选择的密码。

注解

在 [keystone\_authtoken] 中注释或者删除其他选项。

在 [glance\_store] 部分，配置本地文件系统存储和镜像文件位置：

[glance\_store]

...

stores = file,http

default\_store = file

filesystem\_store\_datadir = /var/lib/glance/images/

编辑文件 [``](https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/glance-install.html#id1)/etc/glance/glance-registry.conf``并完成如下动作：

* 在 [database] 部分，配置数据库访问：
* [database]
* ...

connection = mysql+pymysql://glance:GLANCE\_DBPASS@controller/glance

将``GLANCE\_DBPASS`` 替换为你为镜像服务选择的密码。

* 在 [keystone\_authtoken] 和 [paste\_deploy] 部分，配置认证服务访问：
* [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 = glance

password = GLANCE\_PASS

* [paste\_deploy]
* ...

flavor = keystone

将 GLANCE\_PASS 替换为你为认证服务中你为 glance 用户选择的密码。

注解

在 [keystone\_authtoken] 中注释或者删除其他选项。

写入镜像服务数据库：

# su -s /bin/sh -c "glance-manage db\_sync" glance

注解

忽略输出中任何不推荐使用的信息。

完成安装

* 启动镜像服务、配置他们随机启动：

# systemctl enable openstack-glance-api.service \

openstack-glance-registry.service

# systemctl start openstack-glance-api.service \

openstack-glance-registry.service

验证操作

 获得 admin 凭证来获取只有管理员能执行的命令的访问权限：

$source admin-openrc

使用 [*QCOW2*](https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/common/glossary.html#term-qemu-copy-on-write-2-qcow2) 磁盘格式， [*bare*](https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/common/glossary.html#term-bare) 容器格式上传镜像到镜像服务并设置公共可见，这样所有的项目都可以访问它：

$ openstack image create "cirros" \

--file cirros-0.3.4-x86\_64-disk.img \

--disk-format qcow2 --container-format bare \

--public

+------------------+------------------------------------------------------+

| Property | Value |

+------------------+------------------------------------------------------+

| checksum | 133eae9fb1c98f45894a4e60d8736619 |

| container\_format | bare |

| created\_at | 2015-03-26T16:52:10Z |

| disk\_format | qcow2 |

| file | /v2/images/cc5c6982-4910-471e-b864-1098015901b5/file |

| id | cc5c6982-4910-471e-b864-1098015901b5 |

| min\_disk | 0 |

| min\_ram | 0 |

| name | cirros |

| owner | ae7a98326b9c455588edd2656d723b9d |

| protected | False |

| schema | /v2/schemas/image |

| size | 13200896 |

| status | active |

| tags | |

| updated\_at | 2015-03-26T16:52:10Z |

| virtual\_size | None |

| visibility | public |

+------------------+------------------------------------------------------+

 确认镜像的上传并验证属性：

$ openstack image list

+--------------------------------------+--------+--------+

| ID | Name | Status |

+--------------------------------------+--------+--------+

| 38047887-61a7-41ea-9b49-27987d5e8bb9 | cirros | active |

+--------------------------------------+--------+--------+

(镜像的上传与此处相似)

# 计算服务（nova）

## 安装并配置控制节点

在安装和配置 Compute 服务前，你必须创建数据库服务的凭据以及 API endpoints。

1. 为了创建数据库，必须完成这些步骤：
   * 用数据库连接客户端以 root 用户连接到数据库服务器：

$ mysql -u root -p

* + 创建 nova\_api 和 nova 数据库：

CREATE DATABASE nova\_api;

CREATE DATABASE nova;

* + 对数据库进行正确的授权：

GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'localhost' \

IDENTIFIED BY 'NOVA\_DBPASS';

GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'%' \

IDENTIFIED BY 'NOVA\_DBPASS';

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' \

IDENTIFIED BY 'NOVA\_DBPASS';

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' \

IDENTIFIED BY 'NOVA\_DBPASS';

用合适的密码代替 NOVA\_DBPASS。

* + 退出数据库客户端。

 获得 admin 凭证来获取只有管理员能执行的命令的访问权限：

$ source admin-openrc

 要创建服务证书，完成这些步骤：

* 创建 nova 用户：

$ openstack user create --domain default \

--password-prompt nova

* User Password:
* Repeat User Password:
* +-----------+----------------------------------+
* | Field | Value |
* +-----------+----------------------------------+
* | domain\_id | e0353a670a9e496da891347c589539e9 |
* | enabled | True |
* | id | 8c46e4760902464b889293a74a0c90a8 |
* | name | nova |
* +-----------+----------------------------------+
* 给 nova 用户添加 admin 角色：

$ openstack role add --project service --user nova admin

注解

这个命令执行后没有输出。

* 创建 nova 服务实体：

$ openstack service create --name nova \

--description "OpenStack Compute" compute

* +-------------+----------------------------------+
* | Field | Value |
* +-------------+----------------------------------+
* | description | OpenStack Compute |
* | enabled | True |
* | id | 060d59eac51b4594815603d75a00aba2 |
* | name | nova |
* | type | compute |
* +-------------+----------------------------------+

 创建 Compute 服务 API 端点 ：

$ openstack endpoint create --region RegionOne \

compute public http://controller:8774/v2.1/%\(tenant\_id\)s

+--------------+-------------------------------------------+

| Field | Value |

+--------------+-------------------------------------------+

| enabled | True |

| id | 3c1caa473bfe4390a11e7177894bcc7b |

| interface | public |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | e702f6f497ed42e6a8ae3ba2e5871c78 |

| service\_name | nova |

| service\_type | compute |

| url | http://controller:8774/v2.1/%(tenant\_id)s |

+--------------+-------------------------------------------+

$ openstack endpoint create --region RegionOne \

compute internal http://controller:8774/v2.1/%\(tenant\_id\)s

+--------------+-------------------------------------------+

| Field | Value |

+--------------+-------------------------------------------+

| enabled | True |

| id | e3c918de680746a586eac1f2d9bc10ab |

| interface | internal |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | e702f6f497ed42e6a8ae3ba2e5871c78 |

| service\_name | nova |

| service\_type | compute |

| url | http://controller:8774/v2.1/%(tenant\_id)s |

+--------------+-------------------------------------------+

$ openstack endpoint create --region RegionOne \

compute admin http://controller:8774/v2.1/%\(tenant\_id\)s

+--------------+-------------------------------------------+

| Field | Value |

+--------------+-------------------------------------------+

| enabled | True |

| id | 38f7af91666a47cfb97b4dc790b94424 |

| interface | admin |

| region | RegionOne |

| region\_id | RegionOne |

| service\_id | e702f6f497ed42e6a8ae3ba2e5871c78 |

| service\_name | nova |

| service\_type | compute |

| url | http://controller:8774/v2.1/%(tenant\_id)s |

+--------------+-------------------------------------------+

安全并配置组件

1. 安装软件包：
2. # yum install openstack-nova-api openstack-nova-conductor \

openstack-nova-console openstack-nova-novncproxy \

openstack-nova-scheduler

1. 编辑``/etc/nova/nova.conf``文件并完成下面的操作：
   * 在``[DEFAULT]``部分，只启用计算和元数据API：
   * [DEFAULT]
   * ...

enabled\_apis = osapi\_compute,metadata

* + 在``[api\_database]``和``[database]``部分，配置数据库的连接：
  + [api\_database]
  + ...

connection = mysql+pymysql://nova:NOVA\_DBPASS@controller/nova\_api

* + [database]
  + ...

connection = mysql+pymysql://nova:NOVA\_DBPASS@controller/nova

用你为 Compute 数据库选择的密码来代替 NOVA\_DBPASS。

* + 在 “[DEFAULT]” 和 “[oslo\_messaging\_rabbit]”部分，配置 “RabbitMQ” 消息队列访问：
  + [DEFAULT]
  + ...

rpc\_backend = rabbit

* + [oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = RABBIT\_PASS

用你在 “RabbitMQ” 中为 “openstack” 选择的密码替换 “RABBIT\_PASS”。

* + 在 “[DEFAULT]” 和 “[keystone\_authtoken]” 部分，配置认证服务访问：
  + [DEFAULT]
  + ...

auth\_strategy = 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 = NOVA\_PASS

使用你在身份认证服务中设置的``nova`` 用户的密码替换``NOVA\_PASS``。

注解

在 [keystone\_authtoken] 中注释或者删除其他选项。

* + 在 [DEFAULT 部分，配置``my\_ip`` 来使用控制节点的管理接口的IP 地址。
  + [DEFAULT]
  + ...

my\_ip = 10.0.0.11

* + 在 [DEFAULT] 部分，使能 Networking 服务：
  + [DEFAULT]
  + ...

use\_neutron = True

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

* + 在``[vnc]``部分，配置VNC代理使用控制节点的管理接口IP地址 ：
  + [vnc]
  + ...

vncserver\_listen = $my\_ip

vncserver\_proxyclient\_address = $my\_ip

* + 在 [glance] 区域，配置镜像服务 API 的位置：
  + [glance]
  + ...

api\_servers = http://controller:9292

* + 在 [oslo\_concurrency] 部分，配置锁路径：
  + [oslo\_concurrency]
  + ...

lock\_path = /var/lib/nova/tmp

1. 同步Compute 数据库：
2. # su -s /bin/sh -c "nova-manage api\_db sync" nova
3. # su -s /bin/sh -c "nova-manage db sync" nova

注解

忽略输出中任何不推荐使用的信息。

* 启动 Compute 服务并将其设置为随系统启动：

# systemctl enable openstack-nova-api.service \

openstack-nova-consoleauth.service openstack-nova-scheduler.service \

openstack-nova-conductor.service openstack-nova-novncproxy.service

# systemctl start openstack-nova-api.service \

openstack-nova-consoleauth.service openstack-nova-scheduler.service \

openstack-nova-conductor.service openstack-nova-novncproxy.service

验证操作

注解

在控制节点上执行这些命令。

 获得 admin 凭证来获取只有管理员能执行的命令的访问权限：

$ source admin-openrc

 列出服务组件，以验证是否成功启动并注册了每个进程：

$ openstack compute service list

+----+--------------------+------------+----------+---------+-------+----------------------------+

| Id | Binary | Host | Zone | Status | State | Updated At |

+----+--------------------+------------+----------+---------+-------+----------------------------+

| 1 | nova-consoleauth | controller | internal | enabled | up | 2016-02-09T23:11:15.000000 |

| 2 | nova-scheduler | controller | internal | enabled | up | 2016-02-09T23:11:15.000000 |

| 3 | nova-conductor | controller | internal | enabled | up | 2016-02-09T23:11:16.000000 |

+----+--------------------+------------+----------+---------+-------+-----------

## 安装和配置计算节点

1. 安装软件包：
2. # yum install openstack-nova-compute
3. 编辑``/etc/nova/nova.conf``文件并完成下面的操作：
   * 在``[DEFAULT]`` 和 [oslo\_messaging\_rabbit]部分，配置``RabbitMQ``消息队列的连接：
   * [DEFAULT]

rpc\_backend = rabbit

* + [oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = RABBIT\_PASS

用你在 “RabbitMQ” 中为 “openstack” 选择的密码替换 “RABBIT\_PASS”。

* + 在 “[DEFAULT]” 和 “[keystone\_authtoken]” 部分，配置认证服务访问：
  + [DEFAULT]

auth\_strategy = 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 = NOVA\_PASS

使用你在身份认证服务中设置的``nova`` 用户的密码替换``NOVA\_PASS``。

注解

在 [keystone\_authtoken] 中注释或者删除其他选项。

* 在 [DEFAULT] 部分，配置 my\_ip 选项：
* [DEFAULT]
* ...
* my\_ip = MANAGEMENT\_INTERFACE\_IP\_ADDRESS

将其中的 MANAGEMENT\_INTERFACE\_IP\_ADDRESS 替换为计算节点上的管理网络接口的IP 地址，例如 :ref:[`](https://docs.openstack.org/mitaka/zh_CN/install-guide-rdo/nova-compute-install.html#id1)example architecture <overview-example-architectures>`中所示的第一个节点 10.0.0.31 。

* 在 [DEFAULT] 部分，使能 Networking 服务：
* [DEFAULT]
* ...
* use\_neutron = True
* firewall\_driver = nova.virt.firewall.NoopFirewallDriver

在``[vnc]``部分，启用并配置远程控制台访问：

[vnc]

...

enabled = True

vncserver\_listen = 0.0.0.0

vncserver\_proxyclient\_address = $my\_ip

novncproxy\_base\_url = http://controller:6080/vnc\_auto.html

服务器组件监听所有的 IP 地址，而代理组件仅仅监听计算节点管理网络接口的 IP 地址。基本的 URL 指示您可以使用 web 浏览器访问位于该计算节点上实例的远程控制台的位置。

注解

如果你运行浏览器的主机无法解析``controller`` 主机名，你可以将 ``controller``替换为你控制节点管理网络的IP地址。

* 在 [glance] 区域，配置镜像服务 API 的位置：
* [glance]
* ...
* api\_servers = http://controller:9292
* 在 [oslo\_concurrency] 部分，配置锁路径：
* [oslo\_concurrency]
* ...
* lock\_path = /var/lib/nova/tmp

完成安装

1. 确定您的计算节点是否支持虚拟机的硬件加速。
2. $ egrep -c '(vmx|svm)' /proc/cpuinfo

如果这个命令返回了 one or greater 的值，那么你的计算节点支持硬件加速且不需要额外的配置。

如果这个命令返回了 zero 值，那么你的计算节点不支持硬件加速。你必须配置 libvirt 来使用 QEMU 去代替 KVM

* + 在 /etc/nova/nova.conf 文件的 [libvirt] 区域做出如下的编辑：
  + [libvirt]
  + ...
  + virt\_type = qemu

1. 启动计算服务及其依赖，并将其配置为随系统自动启动：
2. # systemctl enable libvirtd.service openstack-nova-compute.service
3. # systemctl start libvirtd.service openstack-nova-compute.service

验证操作

注解

在控制节点上执行这些命令。

 获得 admin 凭证来获取只有管理员能执行的命令的访问权限：

$ source admin-openrc

 列出服务组件，以验证是否成功启动并注册了每个进程：

$ openstack compute service list

+----+--------------------+------------+----------+---------+-------+----------------------------+

| Id | Binary | Host | Zone | Status | State | Updated At |

+----+--------------------+------------+----------+---------+-------+----------------------------+

| 1 | nova-consoleauth | controller | internal | enabled | up | 2016-02-09T23:11:15.000000 |

| 2 | nova-scheduler | controller | internal | enabled | up | 2016-02-09T23:11:15.000000 |

| 3 | nova-conductor | controller | internal | enabled | up | 2016-02-09T23:11:16.000000 |

| 4 | nova-compute | compute1 | nova | enabled | up | 2016-02-09T23:11:20.000000 |

+----+--------------------+------------+----------+---------+-------+-----------

# networking服务（neutron）集成ovs

## 控制节点（以下安装的是控制节点的网络服务）

**数据库相关操作**：

$ mysql -u root -p

> CREATE DATABASE neutron;

> GRANT ALL PRIVILEGES **ON** neutron.\* **TO** 'neutron'@'localhost' IDENTIFIED **BY** 'NEUTRON\_DBPASS';

> GRANT ALL PRIVILEGES **ON** neutron.\* **TO** 'neutron'@'%' IDENTIFIED **BY** 'NEUTRON\_DBPASS';

替换**NEUTRON\_DBPASS**为需要创建的neutron数据库的密码

退出数据库

**创建neutron用户、分配admin权限、创建neutron服务实体**：

$ source admin-openrc

$ openstack user create --domain default --password-prompt neutron

User Password:

Repeat User Password:

+-----------+----------------------------------+

| Field | Value |

+-----------+----------------------------------+

| domain\_id | e0353a670a9e496da891347c589539e9 |

| enabled | True |

| id | b20a6692f77b4258926881bf831eb683 |

| name | neutron |

+-----------+----------------------------------+

$ openstack role add --project service --user neutron admin

$ openstack service create --name neutron --description "OpenStack Networking" network

+-------------+----------------------------------+

| Field | Value |

+-------------+----------------------------------+

| description | OpenStack Networking |

| enabled | True |

| id | f71529314dab4a4d8eca427e701d209e |

| name | neutron |

| type | network |

+-------------+----------------------------------+

**创建neutron服务的API endpoints**：

$ openstack endpoint create --region RegionOne network public http://controller:9696

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 85d80a6d02fc4b7683f611d7fc1493a3 |

| interface | public |

| region | RegionOne |

| region*\_id | RegionOne |*

*| service\_*id | f71529314dab4a4d8eca427e701d209e |

| service*\_name | neutron |*

*| service\_*type | network |

| url | http://controller:9696 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne network internal http://controller:9696

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 09753b537ac74422a68d2d791cf3714f |

| interface | internal |

| region | RegionOne |

| region*\_id | RegionOne |*

*| service\_*id | f71529314dab4a4d8eca427e701d209e |

| service*\_name | neutron |*

*| service\_*type | network |

| url | http://controller:9696 |

+--------------+----------------------------------+

$ openstack endpoint create --region RegionOne network admin http://controller:9696

+--------------+----------------------------------+

| Field | Value |

+--------------+----------------------------------+

| enabled | True |

| id | 1ee14289c9374dffb5db92a5c112fc4e |

| interface | admin |

| region | RegionOne |

| region*\_id | RegionOne |*

*| service\_*id | f71529314dab4a4d8eca427e701d209e |

| service*\_name | neutron |*

*| service\_*type | network |

| url | http://controller:9696 |

+--------------+----------------------------------+

**安装相关组件**：

**#** yum install openstack-neutron openstack-neutron-ml2 python-neutronclient which

**配置服务组件：**

编辑 vi/etc/neutron/neutron.conf文件：

[DEFAULT]

...

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

rpc\_backend = rabbit

auth\_strategy = keystone

notify\_nova\_on\_port\_status\_changes = True

notify\_nova\_on\_port\_data\_changes = True

[database]

...

connection = mysql+pymysql://neutron:NEUTRON\_DBPASS@controller/neutron

输入的密码即为**NEUTRON\_PASS**，即需要创建的neutron用户的密码

[oslo\_messaging\_rabbit]

...

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = RABBIT\_PASS

[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 = neutron

password = NEUTRON\_PASS

[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 = NOVA\_PASS

[oslo\_concurrency]

...

lock\_path = /var/lib/neutron/tmp

替换**NEUTRON\_DBPASS**为创建的neutron数据库的密码

替换**RABBIT\_PASS**RABBITMQ的密码

替换**NEUTRON\_PASS**为创建的用户neutron的密码

替换**NOVA\_PASS**为创建的用户nova的密码

**注意：将[keystone\_authtoken]中的其余选项都注释掉！！！**

**配置ML2组件：**

编辑 **/etc/neutron/plugins/ml2/ml2\_conf.ini**文件：

[ml2]

...

type\_drivers = flat,vlan,gre,vxlan

tenant\_network\_types = gre

mechanism\_drivers = openvswitch,l2population

extension\_drivers = port\_security

[ml2\_type\_flat]

...

flat\_networks = provider

[ml2\_type\_gre]

...

tunnel\_id\_ranges = 1:1000

[securitygroup]

...

enable\_ipset = True

**配置计算服务，使其可以使用网络服务：**

编辑 **/etc/nova/nova.conf**文件：

[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 = NEUTRON\_PASS

service\_metadata\_proxy = True

metadata\_proxy\_shared\_secret = METADATA\_SECRET

替换**NEUTRON\_PASS**为创建的用户neutron的密码

替换**METADATA\_SECRET**为设置的metadata代理的密码

**创建链接**：

# ln -s /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini

**同步数据库**：

**#** 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

**重启nova服务**：

**#** systemctl restart openstack-nova-api.service

**启动neutron-server服务**：

**#** systemctl enable neutron-server.service

**#** systemctl start neutron-server.service

## 网络节点

**安装相关组件**：

**#** yum install openstack-neutron openstack-neutron-ml2 openstack-neutron-openvswitch

**配置服务组件**：

编辑 **/etc/neutron/neutron.conf**文件：

删除 **[DEFAULT]** 中对数据库的访问！！

[DEFAULT]

...

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

rpc\_backend = rabbit

auth\_strategy = keystone

[oslo\_messaging\_rabbit]

...

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = RABBIT\_PASS

[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 = neutron

password = NEUTRON\_PASS

[oslo\_concurrency]

lock\_path = /var/lib/neutron/tmp

替换**RABBIT\_PASS**为RABBITMQ的密码

替换**NEUTRON\_PASS**为neutron用户的密码

**注意：将[keystone\_authtoken]中的其余选项都注释掉！！！**

**配置ML2组件：**

编辑 **/etc/neutron/plugins/ml2/ml2\_conf.ini**文件：

[ml2]

...

type\_drivers = flat,vlan,gre,vxlan

tenant\_network\_types = gre

mechanism\_drivers = openvswitch,l2population

extension\_drivers = port\_security

[ml2\_type\_flat]

...

flat\_networks = provider

[ml2\_type\_gre]

...

tunnel\_id\_ranges = 1:1000

[securitygroup]

...

enable\_ipset = true

**配置openvswitch agent：**

编辑 **/etc/neutron/plugins/ml2/openvswitch\_agent.ini**文件：

**[ovs]**

local\_ip = TUNNEL\_INTERFACE\_IP\_ADDRESS

bridge\_mappings = provider:br-ex

**[agent]**

tunnel\_types = gre

l2\_population = true

prevent\_arp\_spoofing = true

**[securitygroup]**

enable\_security\_group = true

firewall\_driver=neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

替换**TUNNEL\_INTERFACE\_IP\_ADDRESS**为用于隧道网的网卡地址（租户网络）

**配置layer-3 agent：**

编辑 **/etc/neutron/l3\_agent.ini**文件：

[DEFAULT]

...

interface\_driver = neutron.agent.linux.interface.OVSInterfaceDriver

external\_network\_bridge = br-ex

官网上external\_network\_bridge为空???

**配置DHCP agent：**

编辑 **/etc/neutron/dhcp\_agent.ini**文件：

[DEFAULT]

...

interface\_driver = neutron.agent.linux.interface.OVSInterfaceDriver

dhcp\_driver = neutron.agent.linux.dhcp.Dnsmasq

enable\_isolated\_metadata = true

**配置metadata agent：**

编辑 **/etc/neutron/metadata\_agent.ini**文件：

[DEFAULT]

...

nova\_metadata\_ip = controller

metadata\_proxy\_shared\_secret = METADATA\_SECRET

替换**METADATA\_SECRET**为metada的密码

**创建链接：**

# ln -s /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini

**创建网桥br-ex，并将网卡加入其中：**

**#** systemctl enable openvswitch.service

**#** systemctl start openvswitch.service

**#** ovs-vsctl add-br br-ex

**#** ovs-vsctl add-port br-ex INTERFACE\_NAME(enp4s0f1)(此步骤暂时不加，是基于双网卡配置)加了这一步以后，ping不通百度，也ping不通192.168.200.2了；别加

替换**INTERFACE\_NAME**为用于外部网络的网卡名

**启用各项服务：**

# **systemctl** **start** **neutron-openvswitch-agent**.service **neutron-l3-agent**.service **neutron-dhcp-agent**.service **neutron-metadata-agent**.service

# **systemctl** **enable** **neutron-openvswitch-agent**.service **neutron-l3-agent**.service **neutron-dhcp-agent**.service **neutron-metadata-agent**.service

## 计算节点

**安装相关组件：**

**#** yum install openstack-neutron openstack-neutron-ml2 openstack-neutron-openvswitch

**配置各个组件：**

编辑 **/etc/neutron/neutron.conf**文件：

将[DEFAULT]中的关于连接数据库部分都注释掉！！！

[DEFAULT]

...

rpc\_backend = rabbit

auth\_strategy = keystone

[oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = RABBIT\_PASS

[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 = neutron

password = NEUTRON\_PASS

[oslo\_concurrency]

...

lock\_path = /var/lib/neutron/tmp

替换**RABBIT\_PASS**为RABBITMQ的密码

替换**NEUTRON\_PASS**为用户neutron的密码

将[keystone\_authtoken]中的其他选项全部注释掉！！！

**配置其ovs agent：**

编辑 **/etc/neutron/plugins/ml2/openvswitch\_agent.ini**文件：

[ovs]

...

local\_ip = TUNNEL\_INTERFACE\_IP\_ADDRESS

[agent]

...

tunnel\_types = gre

l2\_population = true

[securitygroup]

...

firewall\_driver

=neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

enable\_security\_group = true

替换**TUNNEL\_INTERFACE\_IP\_ADDRESS**为用于隧道网的网卡地址

**配置nova,使其能使用网络服务：**

编辑 **/etc/nova/nova.conf**文件：

[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 = NEUTRON\_PASS

替换**NEUTRON\_PASS**为用户neutron的密码

**启用各项服务：**

**#** systemctl restart openstack-nova-compute.service

**#** systemctl enable neutron-openvswitch-agent.service

**#** systemctl start neutron-openvswitch-agent.service

**验证是否安装成功**

source admin-openrc

$ neutron ext-list

+---------------------------+-----------------------------------------------+

| alias | name |

+---------------------------+-----------------------------------------------+

| default-subnetpools | Default Subnetpools |

| network-ip-availability | Network IP Availability |

| network\_availability\_zone | Network Availability Zone |

| auto-allocated-topology | Auto Allocated Topology Services |

| ext-gw-mode | Neutron L3 Configurable external gateway mode |

| binding | Port Binding |

| agent | agent |

| subnet\_allocation | Subnet Allocation |

| l3\_agent\_scheduler | L3 Agent Scheduler |

| tag | Tag support |

| external-net | Neutron external network |

| net-mtu | Network MTU |

| availability\_zone | Availability Zone |

| quotas | Quota management support |

| l3-ha | HA Router extension |

| provider | Provider Network |

| multi-provider | Multi Provider Network |

| address-scope | Address scope |

| extraroute | Neutron Extra Route |

| timestamp\_core | Time Stamp Fields addition **for** core resources |

| router | Neutron L3 Router |

| extra\_dhcp\_opt | Neutron Extra DHCP opts |

| dns-integration | DNS Integration |

| security-group | security-group |

| dhcp\_agent\_scheduler | DHCP Agent Scheduler |

| router\_availability\_zone | Router Availability Zone |

| rbac-policies | RBAC Policies |

| standard-attr-description | standard-attr-description |

| port-security | Port Security |

| allowed-address-pairs | Allowed Address Pairs |

| dvr | Distributed Virtual Router |

+---------------------------+-----------------------------------------------+

$ neutron agent-list

+--------------------------------------+--------------------+----------+-------------------+-------+----------------+---------------------------+

| id | agent\_type | host | availability\_zone | alive | admin\_state\_up | binary |

+--------------------------------------+--------------------+----------+-------------------+-------+----------------+---------------------------+

| 250ffcfd-afb1-43ed-b23b-77297cdf842b | L3 agent | network | nova | :-) | True | neutron-l3-agent |

| 2cc4f859-cf2f-4238-9053-210583ed96d5 | DHCP agent | network | nova | :-) | True | neutron-dhcp-agent |

| 7382d15a-8a75-405b-b829-748d5a93dd94 | Metadata agent | network | | :-) | True | neutron-metadata-agent |

| 8d504da9-5d70-4fd9-b8f6-5520fa7c7a5f | Open vSwitch agent | network | | :-) | True | neutron-openvswitch-agent |

| a09e5522-a4fc-4d21-be9a-968826386f3c | Open vSwitch agent | compute1 | | :-) | True | neutron-openvswitch-agent |

+--------------------------------------+--------------------+----------+----

# Dashboard（控制节点）

1. 安装软件包：
2. # yum install openstack-dashboard
3. 编辑文件 /etc/openstack-dashboard/local\_settings 并完成如下动作：
   * 在 controller 节点上配置仪表盘以使用 OpenStack 服务：

OPENSTACK\_HOST = "controller"

* + 允许所有主机访问仪表板：

ALLOWED\_HOSTS = ['\*', ]

* + 配置 memcached 会话存储服务：

SESSION\_ENGINE = 'django.contrib.sessions.backends.cache'

* + CACHES = {

'default': {

'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',

* + 'LOCATION': 'controller:11211',
  + }
  + }

注解

将其他的会话存储服务配置注释。

 启用第3版认证API:

OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v3" % OPENSTACK\_HOST

 启用对域的支持

OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True

 配置API版本:

OPENSTACK\_API\_VERSIONS = {

"identity": 3,

"image": 2,

"volume": 2,

}

 通过仪表盘创建用户时的默认域配置为 default :

OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = "default"

 通过仪表盘创建的用户默认角色配置为 user ：

OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "user"

可以选择性地配置时区：

TIME\_ZONE = " Asia/Shanghai"

完成安装

重启web服务器以及会话存储服务：

# systemctl restart httpd.service memcached.service

验证操作

在谷歌或者火狐浏览器访问<http://controller/dashboard>

把controller替换为ip地址