

openstack应用

一、云平台类型

1、根据服务范围不同

公有云、私有云、混合云

2、根据提供服务内容不同

IaaS

基础设施即服务

Infrastructure As A Service

提供IT基础设备(计算、存储、网络)

PaaS

平台即服务

Platform As A Service

提供操作平台

SaaS

软件即服务

Software As a Service

二、openstack核心组件

1、keystone

提供identity认证服务

2、nova

提供compute计算服务

作用: 提供云服务器(实例 instance)的管理操作

调用Hypervisor实现云服务器管理

3、glance

提供image镜像服务, 用于为虚拟机提供磁盘文件模板

作用: 接收上传、删除、修改磁盘文件的请求, 处理请求

4、neutron

提供network网络服务, 提供网络连接功能
借助网络插件

5、Cinder

提供block块存储服务, 提供块存储能力
作用: 提供块存储设备管理

6、swift

提供磁盘文件、镜像存储功能

7、cellometer

提供计费功能

8、Dashboard

提供web管理界面

三、openstack-rocky版安装部署

1、基本环境的配置

1) 关闭防火墙、SELinux

2) 修改主机名、添加主机名解析

```
1 [root@controller ~]# cat /etc/hosts
2 127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
3 ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
4
5 192.168.183.10 controller
6 192.168.183.11 compute
```

3) 确保机器时间同步

```
1 [root@controller ~]# ntpdate 120.25.115.20
2 [root@controller ~]# crontab -l
3 */30 * * * * /usr/sbin/ntpdate 120.25.115.20 &> /dev/null
```

4) 配置openstack软件源

```
1 [root@controller ~]# cat /etc/yum.repos.d/openstack.repo
2 [openstack-r]
3 name=openstack-r
4 baseurl=ftp://172.31.2.252/upload/software/openstack-r
5 enabled=1
6 gpgcheck=0
```

5) 安装配置MySQL

```
1 [root@controller ~]# yum install -y mariadb-server python2-PyMySQL
2 [root@controller ~]# cat /etc/my.cnf.d/openstack.cnf
3 [mysqld]
4 bind-address = 192.168.183.10
5
6 default-storage-engine = innodb
7 innodb_file_per_table = on
8 max_connections = 4096
9 collation-server = utf8_general_ci
10 character-set-server = utf8
11
12 [root@controller ~]# systemctl start mariadb
13 [root@controller ~]# systemctl enable mariadb
14
15 [root@controller ~]# netstat -antp | grep mysql
16 tcp 0 0 192.168.183.10:3306 0.0.0.0:* LISTEN 1529/mysqld
17
18 [root@controller ~]# mysql_secure_installation
```

6) 安装配置rabbit消息队列

```
1 [root@controller ~]# yum install -y rabbitmq-server
2 [root@controller ~]# systemctl enable rabbitmq-server.service
3 [root@controller ~]# systemctl start rabbitmq-server.service
4
5 [root@controller ~]# netstat -antp | grep beam
6 tcp 0 0 0.0.0.0:25672 0.0.0.0:* LISTEN 1780/beam.smp
7 tcp 0 0 127.0.0.1:54614 127.0.0.1:4369 ESTABLISHED 1780/beam.smp
```

```

8 tcp6 0 0 :::5672 :::* LISTEN 1780/beam.smp
9
10 [root@controller ~]# rabbitmqctl add_user openstack redhat
11
12 [root@controller ~]# rabbitmqctl set_permissions openstack ".*" ".*" ".
*"

```

7) 安装配置memcached, 用于保存token

```

1 [root@controller ~]# yum install -y memcached python-memcached
2
3 [root@controller ~]# cat /etc/sysconfig/memcached
4 PORT="11211"
5 USER="memcached"
6 MAXCONN="1024"
7 CACHESIZE="64"
8 OPTIONS="-l 127.0.0.1,:::1,192.168.183.10"
9 [root@controller ~]#
10 [root@controller ~]# systemctl start memcached.service
11 [root@controller ~]# systemctl enable memcached.service
12 [root@controller ~]#
13 [root@controller ~]# netstat -antp | grep 11211
14 tcp 0 0 192.168.183.10:11211 0.0.0.0:* LISTEN 13125/memcached
15 tcp 0 0 127.0.0.1:11211 0.0.0.0:* LISTEN 13125/memcached
16 tcp6 0 0 :::11211 :::* LISTEN 13125/memcached

```

8) 安装etcd数据库,

```

1 [root@controller ~]# yum install -y etcd
2
3 [root@controller ~]# vim /etc/etcd/etcd.conf
4 #[Member]
5 ETCD_DATA_DIR="/var/lib/etcd/default.etcd"
6 ETCD_LISTEN_PEER_URLS="http://192.168.183.10:2380"
7 ETCD_LISTEN_CLIENT_URLS="http://192.168.183.10:2379"
8 ETCD_NAME="controller"
9
10 #[Clustering]
11 ETCD_INITIAL_ADVERTISE_PEER_URLS="http://192.168.183.10:2380"
12 ETCD_ADVERTISE_CLIENT_URLS="http://192.168.183.10:2379"

```

```
13 ETCD_INITIAL_CLUSTER="controller=http://192.168.183.10:2380"
14 ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster-01"
15 ETCD_INITIAL_CLUSTER_STATE="new"
16
17 [root@controller ~]# systemctl enable etcd
18 [root@controller ~]# netstat -antp | grep 2379
19 tcp 0 0 192.168.183.10:2379 0.0.0.0:* LISTEN 13687/etcd
20 tcp 0 0 192.168.183.10:2379 192.168.183.10:39224 ESTABLISHED 13687/etcd
21 tcp 0 0 192.168.183.10:39224 192.168.183.10:2379 ESTABLISHED 13687/etcd
```

2、在controller节点安装配置keystone

1、安装keystone相关软件

```
1 [root@controller ~]# yum install -y openstack-keystone.noarch httpd mod_wsgi python2-openstackclient
```

2、创建数据库

```
1 MariaDB [(none)]> create database keystone;
2 Query OK, 1 row affected (0.00 sec)
3
4 MariaDB [(none)]> grant all on keystone.* to 'keystone'@'localhost' identified by 'redhat';
5 Query OK, 0 rows affected (0.00 sec)
6
7 MariaDB [(none)]> grant all on keystone.* to 'keystone'@'%' identified by 'redhat';
8 Query OK, 0 rows affected (0.00 sec)
9
10 MariaDB [(none)]> flush privileges;
11 Query OK, 0 rows affected (0.00 sec)
```

3、编辑keystone配置文件

```
1 [root@controller ~]# vim /etc/keystone/keystone.conf
2
3 723 [database]
4 742 connection = mysql+pymysql://keystone:redhat@controller/keystone
5
6 //指定生成token的方法为fernet
7 2806 [token]
8 2829 provider = fernet
```

4、生成keystone组件需要的表

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

5、fernet初始化，生成fernet需要的key

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

6、定义keystone的访问地址、定义云平台区域名称、定义admin用户的密码

```
1 访问地址 endpoint
2 public 公共地址，用于面向最终用户、互联网用户
3 internal 内部地址，用于云平台内部组件相互通信的地址
4 admin 管理地址，类似于后台管理地址
5
6 内部自带名称为admin的管理账号
7
8 [root@controller ~]# keystone-manage bootstrap --bootstrap-password redhat \
9 > --bootstrap-admin-url http://controller:5000/v3/ \
10 > --bootstrap-internal-url http://controller:5000/v3/ \
11 > --bootstrap-public-url http://controller:5000/v3/ \
12 > --bootstrap-region-id RegionOne
```

7、配置httpd提供5000端口的虚拟主机

```
1 [root@controller ~]# grep -i "ServerName" /etc/httpd/conf/httpd.conf
2 ServerName controller
3
4 [root@controller ~]# ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/
5
6 [root@controller ~]# systemctl start httpd
7 [root@controller ~]# systemctl enable httpd
8
9 [root@controller ~]# netstat -antp | grep http
10 tcp6 0 0 :::80 :::* LISTEN 3494/httpd
11 tcp6 0 0 :::5000 :::* LISTEN 3494/httpd
```

8、定义admin用户相关的环境变量

```
1 [root@controller ~]# export OS_USERNAME=admin
2 [root@controller ~]# export OS_PASSWORD=redhat
3 [root@controller ~]# export OS_PROJECT_NAME=admin
4 [root@controller ~]# export OS_USER_DOMAIN_NAME=Default
5 [root@controller ~]# export OS_PROJECT_DOMAIN_NAME=Default
6 [root@controller ~]# export OS_AUTH_URL=http://controller:5000/v3
7 [root@controller ~]# export OS_IDENTITY_API_VERSION=3
8 [root@controller ~]#
```

9、创建域、项目、用户、角色

```
1 [root@controller ~]# openstack domain create --description "An Example Do
main" example
2 +-----+-----+
3 | Field | Value |
4 +-----+-----+
5 | description | An Example Domain |
6 | enabled | True |
7 | id | 3a3e8c4916de4705b380024f0c923285 |
8 | name | example |
9 | tags | [] |
10 +-----+-----+
11
12 [root@controller ~]# openstack project create --domain default \
13 > --description "Service project" service
14 +-----+-----+
15 | Field | Value |
16 +-----+-----+
17 | description | Service project |
18 | domain_id | default |
19 | enabled | True |
20 | id | 0a1c7b5c8d0c4a9e96508228368de00a |
21 | is_domain | False |
22 | name | service |
23 | parent_id | default |
24 | tags | [] |
25 +-----+-----+
26
27 [root@controller ~]# openstack project create --domain default --descrip
tion "Demo project" myproject
```

```

28 +-----+
29 | Field | Value |
30 +-----+
31 | description | Demo project |
32 | domain_id | default |
33 | enabled | True |
34 | id | 8441273ff7b14d6c92265abb04256f43 |
35 | is_domain | False |
36 | name | myproject |
37 | parent_id | default |
38 | tags | [] |
39 +-----+
40
41 [root@controller ~]# openstack user create --domain default \
42 > --password-prompt myuser
43 User Password:
44 Repeat User Password:
45 +-----+
46 | Field | Value |
47 +-----+
48 | domain_id | default |
49 | enabled | True |
50 | id | 8d5bc3f85b4445df887fb02cf55b1858 |
51 | name | myuser |
52 | options | {} |
53 | password_expires_at | None |
54 +-----+
55
56 [root@controller ~]# openstack role create myrole
57 +-----+
58 | Field | Value |
59 +-----+
60 | domain_id | None |
61 | id | 31307567cfea44daac08f795ecd76e5f |
62 | name | myrole |
63 +-----+
64 [root@controller ~]# openstack role add --project myproject --user myuse
r myrole

```


10、验证keystone是否工作正常

```
1 [root@controller ~]# openstack --os-auth-url http://controller:5000/v3 --  
os-project-domain-name Default --os-user-domain-name Default --os-project-n  
ame admin --os-usernsue  
2 Password:  
3  
4 [root@controller ~]# openstack --os-auth-url http://controller:5000/v3 --  
os-project-domain-name Default --os-user-domain-name Default --os-project-n  
ame myproject --os-username myuser token issue  
5 Password:
```

11、创建脚本保存分别保存admin, myuser用户信息

```
1 [root@controller ~]# cat admin  
2 export OS_PROJECT_DOMAIN_NAME=Default  
3 export OS_USER_DOMAIN_NAME=Default  
4 export OS_PROJECT_NAME=admin  
5 export OS_USERNAME=admin  
6 export OS_PASSWORD=redhat  
7 export OS_AUTH_URL=http://controller:5000/v3  
8 export OS_IDENTITY_API_VERSION=3  
9 export OS_IMAGE_API_VERSION=2  
10 [root@controller ~]#  
11 [root@controller ~]# cat myuser  
12 export OS_PROJECT_DOMAIN_NAME=Default  
13 export OS_USER_DOMAIN_NAME=Default  
14 export OS_PROJECT_NAME=myproject  
15 export OS_USERNAME=myuser  
16 export OS_PASSWORD=redhat  
17 export OS_AUTH_URL=http://controller:5000/v3  
18 export OS_IDENTITY_API_VERSION=3  
19 export OS_IMAGE_API_VERSION=2
```

3、在controller节点安装glance组件

glance组件

提供image镜像服务

glance-api服务

接收上传、删除、修改镜像的请求

glance-registry服务

保存处理镜像文件的元数据信息(名称、大小、类型)

1、创建glance数据库

```
1 MariaDB [(none)]> create database glance;
2 Query OK, 1 row affected (0.00 sec)
3
4 MariaDB [(none)]> grant all on glance.* to 'glance'@'localhost' identified by 'redhat';
5 Query OK, 0 rows affected (0.01 sec)
6
7 MariaDB [(none)]> grant all on glance.* to 'glance'@'%' identified by 'redhat';
8 Query OK, 0 rows affected (0.00 sec)
9
10 MariaDB [(none)]> flush privileges;
11 Query OK, 0 rows affected (0.00 sec)
```

2、创建glance用户，用于与其他组件通信进行认证

```
1 [root@controller ~]# openstack user create --domain default --password-prompt glance
2 User Password:
3 Repeat User Password:
4 +-----+-----+
5 | Field | Value |
6 +-----+-----+
7 | domain_id | default |
8 | enabled | True |
9 | id | 2619de68d5a4458ca721f3f038d8898f |
10 | name | glance |
11 | options | {} |
12 | password_expires_at | None |
13 +-----+-----+
14 [root@controller ~]#
15 [root@controller ~]# openstack role add --project service --user glance admin
```

3、创建image服务

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

```

2 +-----+-----+
3 | Field | Value |
4 +-----+-----+
5 | description | Openstack image |
6 | enabled | True |
7 | id | 80d8a474d51441f4837b6e23228bdd55 |
8 | name | glance |
9 | type | image |

```

4、创建image服务的访问地址 endpoint

```

1 [root@controller ~]# openstack endpoint create --region RegionOne image public http://controller:9292
2 +-----+-----+
3 | Field | Value |
4 +-----+-----+
5 | enabled | True |
6 | id | 6e5afaabde904103a3db7a73fcdea319 |
7 | interface | public |
8 | region | RegionOne |
9 | region_id | RegionOne |
10 | service_id | 80d8a474d51441f4837b6e23228bdd55 |
11 | service_name | glance |
12 | service_type | image |
13 | url | http://controller:9292 |
14 +-----+-----+
15 [root@controller ~]# openstack endpoint create --region RegionOne image internal http://controller:9292
16 +-----+-----+
17 | Field | Value |
18 +-----+-----+
19 | enabled | True |
20 | id | ad17669520a847b8b67634eeeb34c323 |
21 | interface | internal |
22 | region | RegionOne |
23 | region_id | RegionOne |
24 | service_id | 80d8a474d51441f4837b6e23228bdd55 |
25 | service_name | glance |
26 | service_type | image |
27 | url | http://controller:9292 |
28 +-----+-----+

```

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

5、安装openstack-glance软件

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

6、编辑glance-api配置文件

```
1 [root@controller ~]# vim /etc/glance/glance-api.conf
2 1882 [database]
3 1901 connection = mysql+pymysql://glance:redhat@controller/glance
4
5 //指定glance验证其他组件请求的方式
6 4389 [paste_deploy]
7 4414 flavor = keystone
8
9 3472 [keystone_auth_token]
10 3473 www_authenticate_uri = http://controller:5000
11 3474 auth_url = http://controller:5000
12 3475 memcached_servers = controller:11211
13 3476 auth_type = password
14 3477 username = glance
15 3478 password = redhat
16 3479 project_domain_name = Default
17 3480 user_domain_name = Default
18 3481 project_name = service
19
20 //指定glance所支持的存储磁盘镜像文件的方式
21 2007 [glance_store]
22 2043 stores = file,http
23 2096 default_store = file
24 2429 filesystem_store_datadir = /var/lib/glance/images
```

7、编辑glance-registry配置文件

```
1 [root@controller ~]# vim /etc/glance/glance-registry.conf
2 [database]
3 connection = mysql+pymysql://glance:redhat@controller/glance
```

```

4
5 [paste_deploy]
6 flavor = keystone
7
8 [keystone_authtoken]
9 www_authenticate_uri = http://controller:5000
10 auth_url = http://controller:5000
11 memcached_servers = controller:11211
12 auth_type = password
13 project_domain_name = Default
14 user_domain_name = Default
15 project_name = service
16 username = glance
17 password = redhat

```

8、生成glance组件需要的表

```

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

```

9、启动glance

```

1 [root@controller ~]# systemctl start openstack-glance-api.service openstack-glance-registry.service
2 [root@controller ~]# systemctl enable openstack-glance-api.service openstack-glance-registry.service
3
4 [root@controller ~]# netstat -antp | grep 9292
5 tcp 0 0 0.0.0.0:9292 0.0.0.0:* LISTEN 14784/python2

```

10、测试上传镜像

```

1 [root@controller ~]# source admin
2
3 [root@controller ~]# openstack image create "cirrors-0.4.0-x64" \
4 > --file cirros-0.4.0-x86_64-disk.img \
5 > --disk-format qcow2 --container-format bare \
6 > --public
7
8 [root@controller ~]# openstack image list
9 +-----+-----+-----+
10 | ID | Name | Status |
11 +-----+-----+-----+
12 | a8d37904-1a20-4186-a018-079de1a73cef | cirrors-0.4.0-x64 | active |

```

4、安装部署nova组件

nova组件:

提供compute计算服务

nova-api: 接收、响应客户端创建云服务器的请求

nova-compute: 创建、销毁虚拟机; 调用物理机Hypervisor

nova-conductor: 实现在nova-compute服务与数据库之间的交互

nova-scheduler: 调用客户端的请求

nova-novncproxy: 通过vnc协议连接云服务器

nova-placement: 实现对物理资源的监控, 提高资源的利用率

nova-cell: 在不影响原有云环境下, 实现对云平台横向扩展; 方便后续扩展

1、在controller控制节点安装部署nova

1) 创建nova组件需要的数据库

```

1 MariaDB [(none)]> create database nova_api;
2 MariaDB [(none)]> create database nova;
3 MariaDB [(none)]> create database nova_cell0;
4 MariaDB [(none)]> create database placement;
5 Query OK, 1 row affected (0.00 sec)
6
7 MariaDB [(none)]> grant all on nova_api.* to 'nova'@'localhost' identified
  by 'redhat';
8 Query OK, 0 rows affected (0.00 sec)
9
10 MariaDB [(none)]> grant all on nova_api.* to 'nova'@'%' identified by 'r
  edhat';
11 Query OK, 0 rows affected (0.01 sec)
12
13 MariaDB [(none)]>
14 MariaDB [(none)]>
15 MariaDB [(none)]> grant all on nova.* to 'nova'@'localhost' identified b
  y 'redhat';
16 Query OK, 0 rows affected (0.00 sec)
17
18 MariaDB [(none)]> grant all on nova.* to 'nova'@'%' identified by 'redha
  t';

```

```

19 Query OK, 0 rows affected (0.00 sec)
20
21 MariaDB [(none)]>
22 MariaDB [(none)]> grant all on nova_cell0.* to 'nova'@'localhost' identified by 'redhat';
23 Query OK, 0 rows affected (0.00 sec)
24
25 MariaDB [(none)]> grant all on nova_cell0.* to 'nova'@'%' identified by 'redhat';
26 Query OK, 0 rows affected (0.00 sec)
27
28 MariaDB [(none)]> grant all on placement.* to 'nova'@'localhost' identified by 'redhat';
29 Query OK, 0 rows affected (0.00 sec)
30
31 MariaDB [(none)]> grant all on placement.* to 'nova'@'%' identified by 'redhat';

```

2) 创建nova用户

```

1 [root@controller ~]# source admin
2 [root@controller ~]# openstack user create --domain default --password-prompt nova
3 [root@controller ~]# openstack role add --project service --user nova admin

```

3) 创建compute服务

```

1 [root@controller ~]# openstack service create --name nova --description "Openstack compute" compute

```

4) 创建compute服务需要的访问地址

```

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

```

5) 创建placement用户, placement服务及访问地址; 用于实现对物理资源的监控

```

1 [root@controller ~]# openstack user create --domain default --password-prompt placement
2 [root@controller ~]# openstack role add --project service --user placement admin

```

```
3 [root@controller ~]# openstack service create --name placement --description "Placement API" placement
4 [root@controller ~]# openstack endpoint create --region RegionOne placement admin http://controller:8778
5 [root@controller ~]# openstack endpoint create --region RegionOne placement internal http://controller:8778
6 [root@controller ~]# openstack endpoint create --region RegionOne placement public http://controller:8778
```

6) 安装nova相关软件

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

7) 编辑nova配置文件

```
1 [root@controller ~]# vim /etc/nova/nova.conf
2
3 1 [DEFAULT]
4 2738 enabled_apis=osapi_compute,metadata
5
6 3472 [api_database]
7 3487 connection=mysql+pymysql://nova:redhat@controller/nova_api
8
9 4558 [database]
10 4577 connection=mysql+pymysql://nova:redhat@controller/nova
11
12 [placement_database]
13 connection=mysql+pymysql://nova:redhat@controller/placement
14
15 //定义rabbit消息队列的访问地址
16 [default]
17 8297 transport_url=rabbit://openstack:redhat@controller
18
19 [api]
20 3196 auth_strategy=keystone
21
22 6060 [keystone_authtoken]
23 6061 auth_url = http://controller:5000/v3
24 6062 memcached_servers = controller:11211
25 6063 auth_type = password
26 6064 project_domain_name = default
```



```

27 6065 user_domain_name = default
28 6066 project_name = service
29 6067 username = nova
30 6068 password = redhat
31
32 // 在以下定义的IP上启动代表云服务器的vnc端口
33 [default]
34 1243 my_ip=192.168.183.10
35
36 10690 [vnc]
37 10707 enabled=true
38 10740 server_listen=$my_ip
39 10753 server_proxyclient_address=$my_ip
40
41 //由neutron组件提供网络连接，指定生成防火墙的驱动
42 1 [DEFAULT]
43 1707 use_neutron=true
44 2369 firewall_driver=nova.virt.firewall.NoopFirewallDriver
45
46 5239 [glance]
47 5262 api_servers=http://controller:9292
48
49 //指定nova锁文件的位置
50 7996 [oslo_concurrency]
51 8009 lock_path=/var/lib/nova/tmp
52
53 8815 [placement]
54 8816 os_region_name = RegionOne
55 8817 project_domain_name = Default
56 8818 project_name = service
57 8819 auth_type = password
58 8820 user_domain_name = Default
59 8821 auth_url = http://controller:5000/v3
60 8822 username = placement
61 8823 password = redhat

```

```

1 由于软件bug问题，避免由httpd软件版本引起bug
2 [root@controller ~]# vim /etc/httpd/conf.d/00-nova-placement-api.conf
3 <Directory /usr/bin>
4   <IfVersion >= 2.4>

```

```
5   Require all granted
6   </IfVersion>
7   <IfVersion < 2.4>
8   Order allow,deny
9   Allow from all
10  </IfVersion>
11  </Directory>
12  [root@controller ~]# systemctl restart httpd
```

8) 在数据库中生成nova需要的表

```
1  [root@controller ~]# su -s /bin/sh -c "nova-manage api_db sync" nova
2  [root@controller ~]# su -s /bin/sh -c "nova-manage cell_v2 map_cell0" nova
3  [root@controller ~]# su -s /bin/sh -c "nova-manage cell_v2 create_cell --
4  name=cell11 --verbose" nova
5  ab2a9fe5-c7c1-45e2-86df-122e996d91c2
6  [root@controller ~]# su -s /bin/sh -c "nova-manage db sync" nova
```

9) 启动nova服务

```
1  [root@controller ~]# systemctl start openstack-nova-api.service openstack-
2  nova-consoleauth.service openstack-nova-novncproxy.service openstack-nova-
3  scheduler.service openstack-nova-conductor.service
4  [root@controller ~]# systemctl enable openstack-nova-api.service openstac
5  k-nova-consoleauth.service openstack-nova-novncproxy.service openstack-nova-
6  scheduler.service openstack-nova-conductor.service
7  [root@controller ~]# systemctl is-active openstack-nova-api.service opens
8  tack-nova-consoleauth.service openstack-nova-novncproxy.service openstack-n
9  ova-scheduler.service openstack-nova-conductor.service
10 active
11 active
12 active
13 active
14 active
```

2、在compute节点安装nova-compute服务

1) 安装openstack-nova-compute软件

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

2) 编辑/etc/nova/nova.conf配置文件

```
1 [root@compute ~]# vim /etc/nova/nova.conf
2 1 [DEFAULT]
3 2738 enabled_apis=osapi_compute,metadata
4
5 3131 transport_url=rabbit://openstack:redhat@controller
6
7 3179 [api]
8 3196 auth_strategy=keystone
9
10 6058 [keystone_authtoken]
11 6059 auth_url = http://controller:5000/v3
12 6060 memcached_servers = controller:11211
13 6061 auth_type = password
14 6062 project_domain_name = Default
15 6063 user_domain_name = Default
16 6064 project_name = service
17 6065 username = nova
18 6066 password = redhat
19
20 1243 my_ip=192.168.183.11
21 10688 [vnc]
22 10704 enabled=true
23 10737 server_listen=0.0.0.0
24 10750 server_proxyclient_address=$my_ip
25 10771 novncproxy_base_url=http://controller:6080/vnc_auto.html
26
27 1 [DEFAULT]
28 1707 use_neutron=true
29 2369 firewall_driver=nova.virt.firewall.NoopFirewallDriver
30
31 5237 [glance]
32 5259 api_servers=http://controller:9292
33
34 8006 lock_path=/var/lib/nova/tmp
35
36 8812 [placement]
37 8813 region_name = RegionOne
38 8814 project_domain_name = Default
39 8815 project_name = service
```

```

40 8816 auth_type = password
41 8817 user_domain_name = Default
42 8818 auth_url = http://controller:5000/v3
43 8819 username = placement
44 8820 password = redhat

```

nova-compute默认调用kvm实现虚拟机的创建

nova.conf

virt_type=kvm

如果使用的机器CPU不支持虚拟化功能，将配置文件中的virt-type=qemu软件虚拟化

3) 启动libvirtd服务, openstack-nova-compute

```

1 [root@compute ~]# systemctl enable libvirtd openstack-nova-compute.service
2 [root@compute ~]# systemctl start libvirtd openstack-nova-compute.service
3 [root@compute ~]# systemctl is-active libvirtd openstack-nova-compute.service
4 active
5 active

```

4) 在controller节点验证是否可正常检测到compute节点存在

```

1 [root@controller ~]# source admin
2 [root@controller ~]# openstack compute service list --service nova-compute
3 +-----+-----+-----+-----+-----+-----+-----+
4 | ID | Binary | Host | Zone | Status | State | Updated At |
5 +-----+-----+-----+-----+-----+-----+-----+
6 | 6 | nova-compute | compute | nova | enabled | up | 2020-11-25T01:51:28.000000 |
7 +-----+-----+-----+-----+-----+-----+-----+

```

5) 将计算节点添加到nova_cell组件，方便后期扩展云平台架构

```

1 [root@controller ~]# su -s /bin/sh -c "nova-manage cell_v2 discover_hosts
--verbose" nova
2 Found 2 cell mappings.
3 Skipping cell0 since it does not contain hosts.
4 Getting computes from cell 'cell1': ab2a9fe5-c7c1-45e2-86df-122e996d91c2
5 Checking host mapping for compute host 'compute': 3fc84b1f-fa33-42db-b011
-41d877737d4c
6 Creating host mapping for compute host 'compute': 3fc84b1f-fa33-42db-b011
-41d877737d4c
7 Found 1 unmapped computes in cell: ab2a9fe5-c7c1-45e2-86df-122e996d91c2

```

6) 在controller节点验证nova相关的组件是否工作正常

a) 检测nova相关服务正常运行

```

1 [root@controller ~]# source admin
2 [root@controller ~]# openstack compute service list
3 +-----+-----+-----+-----+-----+-----+-----+
4 | ID | Binary | Host | Zone | Status | State | Updated At |
5 +-----+-----+-----+-----+-----+-----+-----+
6 | 1 | nova-consoleauth | controller | internal | enabled | up | 2020-11-2
5T01:58:48.000000 |
7 | 2 | nova-scheduler | controller | internal | enabled | up | 2020-11-25T
01:58:52.000000 |
8 | 3 | nova-conductor | controller | internal | enabled | up | 2020-11-25T
01:58:51.000000 |
9 | 6 | nova-compute | compute | nova | enabled | up | 2020-11-25T01:58:48.
000000 |
10 +-----+-----+-----+-----+-----+-----+-----+

```

b) 检测keystone, nova, glance, placement访问地址

```

1 [root@controller ~]# openstack catalog list
2 +-----+-----+-----+-----+-----+
3 | Name | Type | Endpoints |
4 +-----+-----+-----+-----+-----+
5 | keystone | identity | RegionOne |
6 | | public: http://controller:5000/v3/ |
7 | | RegionOne |
8 | | internal: http://controller:5000/v3/ |
9 | | RegionOne |

```

```

10 | | | admin: http://controller:5000/v3/ |
11 | | | |
12 | placement | placement | RegionOne |
13 | | | public: http://controller:8778 |
14 | | | RegionOne |
15 | | | internal: http://controller:8778 |
16 | | | RegionOne |
17 | | | admin: http://controller:8778 |
18 | | | |
19 | glance | image | RegionOne |
20 | | | admin: http://controller:9292 |
21 | | | RegionOne |
22 | | | public: http://controller:9292 |
23 | | | RegionOne |
24 | | | internal: http://controller:9292 |
25 | | | |
26 | nova | compute | RegionOne |
27 | | | admin: http://controller:8774/v2.1 |
28 | | | RegionOne |
29 | | | internal: http://controller:8774/v2.1 |
30 | | | RegionOne |
31 | | | public: http://controller:8774/v2.1 |
32 | | | |
33 +-----+-----+-----+

```

c) 检测glance上传的镜像是否正常

```

1 [root@controller ~]# openstack image list
2 +-----+-----+-----+
3 | ID | Name | Status |
4 +-----+-----+-----+
5 | a8d37904-1a20-4186-a018-079de1a73cef | cirrors-0.4.0-x64 | active |
6 +-----+-----+-----+

```

d) 检测cell, placement组件是否工作正常

```

1 [root@controller ~]# nova-status upgrade check
2 +-----+
3 | Upgrade Check Results |
4 +-----+
5 | Check: Cells v2 |

```

```

6 | Result: Success |
7 | Details: None |
8 +-----+
9 | Check: Placement API |
10 | Result: Success |
11 | Details: None |
12 +-----+
13 | Check: Resource Providers |
14 | Result: Success |
15 | Details: None |
16 +-----+
17 | Check: Ironi Flavor Migration |
18 | Result: Success |
19 | Details: None |
20 +-----+
21 | Check: API Service Version |
22 | Result: Success |
23 | Details: None |
24 +-----+
25 | Check: Request Spec Migration |
26 | Result: Success |
27 | Details: None |
28 +-----+
29 | Check: Console Auths |
30 | Result: Success |
31 | Details: None |
32 +-----+
33 [root@controller ~]#

```

5、安装neutron组件

neutron组件:

提供network网络服务

neutron-server

接收转发客户端创建网络、销毁网络的请求

借助网络插件

1、在controller节点安装部署neutron

1) 创建neutron数据库

```
1 MariaDB [(none)]> create database neutron;
2 Query OK, 1 row affected (0.00 sec)
3
4 MariaDB [(none)]> grant all on neutron.* to 'neutron'@'localhost' identified by 'redhat';
5 Query OK, 0 rows affected (0.01 sec)
6
7 MariaDB [(none)]> grant all on neutron.* to 'neutron'@'%' identified by 'redhat';
8 Query OK, 0 rows affected (0.00 sec)
9
10 MariaDB [(none)]> flush privileges;
11 Query OK, 0 rows affected (0.00 sec)
```

2) 创建neutron用户, 关联admin权限

```
1 [root@controller ~]# source admin
2 [root@controller ~]# openstack user create --domain default --password-prompt neutron
3 User Password:
4 Repeat User Password:
5 [root@controller ~]# openstack role add --project service --user neutron admin
```

3) 创建network服务, 访问地址

```
1 [root@controller ~]# openstack service create --name neutron --description "Openstack network" network
2 [root@controller ~]# openstack endpoint create --region RegionOne network admin http://controller:9696
3 [root@controller ~]# openstack endpoint create --region RegionOne network public http://controller:9696
4 [root@controller ~]# openstack endpoint create --region RegionOne network internal http://controller:9696
```

4) 安装neutron相关软件

```
1 [root@controller ~]# yum install -y openstack-neutron openstack-neutron-m12.noarch openstack-neutron-linuxbridge.noarch ebtables.x86_64
```

5) 编辑neutron配置文件


```
1 [root@controller ~]# vim /etc/neutron/neutron.conf
2 722 [database]
3 749 connection = mysql+pymysql://neutron:redhat@controller/neutron
4
5 1 [DEFAULT]
6 580 transport_url = rabbit://openstack:redhat@controller
7 27 auth_strategy = keystone
8
9 846 [keystone_authtoken]
10 847 www_authenticate_uri = http://controller:5000
11 848 auth_url = http://controller:5000
12 849 memcached_servers = controller:11211
13 850 auth_type = password
14 851 project_domain_name = default
15 852 user_domain_name = default
16 853 project_name = service
17 854 username = neutron
18 855 password = redhat
19
20 //启用网络插件，ml2二层网络插件，router三层网络插件
21 [default]
22 30 core_plugin = ml2
23 33 service_plugins = router
24 //允许不同的用户使用相同的网段构建自己的网络
25 85 allow_overlapping_ips = True
26
27 //启动通知nova端口状态、数据的变化；目的是让nova组件可及时了解虚拟机网络连接状态的变化
28 [default]
29 98 notify_nova_on_port_status_changes = true
30 102 notify_nova_on_port_data_changes = true
31
32 1094 [nova]
33 1095
34 1096 auth_url = http://controller:5000
35 1097 auth_type = password
36 1098 project_domain_name = default
37 1099 user_domain_name = default
38 1100 region_name = RegionOne
39 1101 project_name = service
40 1102 username = nova
```

```
41 1103 password = redhat
42
43 1214 [oslo_concurrency]
44 1227 lock_path = $state_path/lock
```

6) 编辑二层网络插件，实现二层网络

```
1 [root@controller ~]# vim /etc/neutron/plugins/ml2/ml2_conf.ini
2
3 128 [ml2]
4 // flat: 扁平网络，交换机上所有主机配置同网段地址通信
5 // vlan: 虚拟局域网，一个交换机创建多个vlan，隔离广播数据，安全；方便网络扩展
6 // vxlan: 高级vlan
7 136 type_drivers = flat,vlan,vxlan
8 141 tenant_network_types = vxlan
9 145 mechanism_drivers = linuxbridge,l2population
10 //启用端口安全机制
11 150 extension_drivers = port_security
12
13 177 [ml2_type_flat]
14 186 flat_networks = provider
15
16 231 [ml2_type_vxlan]
17 239 vni_ranges = 1:1000
18
19 //启用安全组，防火墙
20 247 [securitygroup]
21 263 enable_ipset = true
```

```
1 [root@controller ~]# vim /etc/neutron/plugins/ml2/linuxbridge_agent.ini
2 [linux_bridge]
3 //指定linuxbridge进行桥接时使用的网卡
4 physical_interface_mappings = provider:ens33
5
6 [vxlan]
7 enable_vxlan = true
8 local_ip = 192.168.183.10
9 l2_population = true
10
11 [securitygroup]
```

```
12 firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewall
Driver
13 enable_security_group = true
```

7) 配置l3三层插件

```
1 [root@controller ~]# vim /etc/neutron/l3_agent.ini
2 [DEFAULT]
3 interface_driver = linuxbridge
```

8) 配置dhcp插件

```
1 [root@controller ~]# vim /etc/neutron/dhcp_agent.ini
2 [DEFAULT]
3 [root@controller ~]# vim /etc/neutron/dhcp_agent.ini
4 interface_driver = linuxbridge
5 dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
6 enable_isolated_metadata = true
```

9) 配置网络元数据插件

```
1 [root@controller ~]# vim /etc/neutron/metadata_agent.ini
2 [default]
3 nova_metadata_host = controller
4 metadata_proxy_shared_secret = redhat
```

10) 修改nova的配置文件，指定neutron组件的连接信息

```
1 [root@controller ~]# vim /etc/nova/nova.conf
2 [neutron]
3 url = http://controller:9696
4 auth_url = http://controller:5000
5 auth_type = password
6 project_domain_name = default
7 user_domain_name = default
8 region_name = RegionOne
9 project_name = service
10 username = neutron
11 password = redhat
12 service_metadata_proxy = true
13 metadata_proxy_shared_secret = redhat
```

11) 创建neutron组件需要的软链接文件

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

12) 在数据库生成neutron需要的表

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

13) 重启nova-api服务

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

14) 启动neutron相关服务

```
1 [root@controller ~]# systemctl enable neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service neutron-l3-agent.service  
2 [root@controller ~]# systemctl start neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service neutron-l3-agent.service  
3 [root@controller ~]# systemctl is-active neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service neutron-l3-agent.service  
4 active  
5 active  
6 active  
7 active  
8 active
```

2、安装compute计算节点安装部署neutron

1) 安装neutron软件

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

2) 编辑neutron配置文件

```
1 # vim /etc/neutron/neutron.conf  
2 1 [DEFAULT]  
3 580 transport_url = rabbit://openstack:redhat@controller  
4 27 auth_strategy = keystone  
5  
6 845 [keystone_authtoken]
```

```
7 846 www_authenticate_uri = http://controller:5000
8 847 auth_url = http://controller:5000
9 848 memcached_servers = controller:11211
10 849 auth_type = password
11 850 project_domain_name = default
12 851 user_domain_name = default
13 852 project_name = service
14 853 username = neutron
15 854 password = redhat
16
17 1204 [oslo_concurrency]
18 1217 lock_path = $state_path/lock
```

3) 配置网络插件

```
1 [root@compute ~]# vim /etc/neutron/plugins/ml2/linuxbridge_agent.ini
2 [linux_bridge]
3 physical_interface_mappings = provider:ens33
4
5 [vxlan]
6 enable_vxlan = true
7 local_ip = 192.168.183.11
8 l2_population = true
9
10 [securitygroup]
11 firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewall
Driver
12 enable_security_group = true
```

4) 修改nova配置文件，指定neutron的连接信息

```
1 [root@compute ~]# vim /etc/nova/nova.conf
2 [neutron]
3 url = http://controller:9696
4 auth_url = http://controller:5000
5 auth_type = password
6 project_domain_name = default
7 user_domain_name = default
8 region_name = RegionOne
9 project_name = service
10 username = neutron
11 password = redhat
```

5) 重启nova-compute服务

```
1 [root@compute ~]# systemctl restart openstack-nova-compute.service
2 [root@compute ~]# systemctl is-active openstack-nova-compute.service
3 active
```

6) 启动linuxbridge插件

```
1 [root@compute ~]# systemctl start neutron-linuxbridge-agent.service
2 [root@compute ~]# systemctl enable neutron-linuxbridge-agent.service
3 Created symlink from /etc/systemd/system/multi-user.target.wants/neutron-
linuxbridge-agent.service to /usr/lib/systemd/system/neutron-linuxbridge-ag
ent.service.
4 [root@compute ~]# systemctl is-active neutron-linuxbridge-agent.service
5 active
```

7) 在controller节点验证neutron工作正常

```
1 [root@controller ~]# source admin
2 [root@controller ~]# openstack extension list --network
3
4 [root@controller ~]# openstack network agent list
5 +-----+-----+-----+-----+-----+-----+-----+
6 | ID | Agent Type | Host | Availability Zone | Alive | State | Binary |
7 +-----+-----+-----+-----+-----+-----+-----+
8 | 2d15d897-c43c-4a68-bd8a-4b5aa268a496 | Metadata agent | controller | No
ne | :- ) | UP | neutron-metadata-agent |
9 | 38a3eafb-8f9e-473c-8a61-f909fa8fa2b9 | L3 agent | controller | nova | :
-) | UP | neutron-l3-agent |
10 | 762aec55-9b04-41db-85c7-ff717e952dee | Linux bridge agent | compute |
None | :- ) | UP | neutron-linuxbridge-agent |
11 | 7c11322f-de66-40f7-8fe2-00665c8542d7 | Linux bridge agent | controller
| None | :- ) | UP | neutron-linuxbridge-agent |
12 | eed15344-e8eb-4533-9049-a95b9c90a6ba | DHCP agent | controller | nova
| :- ) | UP | neutron-dhcp-agent |
13 +-----+-----+-----+-----+-----+-----+-----+
  +-----+-----+-----+-----+-----+-----+-----+
```

安装dashboard提供web管理界面

1) 安装dashboard软件

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

2) 编辑dashboard配置

```
1 [root@controller ~]# vim /etc/openstack-dashboard/local_settings
2 OPENSTACK_HOST = "controller"
3 OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST
4
5 ALLOWED_HOSTS = ['*', ]
6
7 SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
8
9 CACHES = {
10     'default': {
11         'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
12         'LOCATION': 'controller:11211',
13     }
14 }
15
16 OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
17
18 OPENSTACK_API_VERSIONS = {
19     "identity": 3,
20     "image": 2,
21     "volume": 2,
22 }
23
24 OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = 'Default'
25
26 OPENSTACK_KEYSTONE_DEFAULT_ROLE
```

3) 修改httpd关于dashboard虚拟主机配置文件，添加wsgi机制

```
1 [root@controller ~]# vim /etc/httpd/conf.d/openstack-dashboard.conf
2 WSGIApplicationGroup %{GLOBAL}
```

4) 启动相关服务

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

访问web管理界面

http://controller/dashboard

1、创建实例类型

让云平台都支持创建哪些配置的机器

2、创建网络

3、创建实例

错误现象:

实例可正常创建，启动实例卡在GRUB界面

原因:

使用cirrors测试镜像，测试镜像中硬盘、网卡使用的驱动virtio半驱动化；vmware虚拟化不支持virtio驱动

解决办法:

将cirror镜像中的磁盘驱动修改为ide, 网卡驱动修改为e1000，再创建实例可正常启动

```
1 [root@controller ~]# openstack image list
2 +-----+-----+-----+
3 | ID | Name | Status |
4 +-----+-----+-----+
5 | cc71c2f9-fb93-4253-b7a9-565bdf87a24 | cirrors-0.3.0 | active |
6 | a8d37904-1a20-4186-a018-079de1a73cef | cirrors-0.4.0-x64 | active |
7 +-----+-----+-----+
8
9 [root@controller images]# openstack image set --property hw_disk_bus=ide
--property hw_vif_model=e1000 a8d37904-1a20-4186-a018-079de1a73cef
```


