云操作系统应用

CONTENTS

目 录

第6章 计算服务Nova

6.1 Nova架构及原理

Nova 是 OpenStack 云中的计算组织控制器。 OpenStack 中云主机（ instances）生命周期的所有活动都由 Nova 处理。这样使得 Nova 成为一个负责管理计算资源、网络、认证、所需可扩展性的平台。但是， Nova 自身并没有提供任何虚拟化能力，而是使用 libvirt API 来与被支持的 Hypervisors 交互。 OpenStack 计算服务（ Nova）由下列组件构成。

（ 1） API Server

对外提供一个与云基础设施交互的接口，也是外部可用于管理基础设施的唯一组件。管理使用 EC2 API 通过 Web Services 调用实现。然后， API Server 通过消息队列（ Message Queue）轮流与云基础设施的相关组件通信。作为 EC2 API 的另外一种选择， OpenStack 也提供一个内部使用的“ OpenStack API”。

（ 2） Message Queue（ RabbitMQ Server）

OpenStack 节点之间通过消息队列使用 AMQP（ Advanced Message Queue Protocol，高级消息队列协议）完成通信。 Nova 通过异步调用请求响应，使用回调函数在收到响应时触发。因为使用了异步通信，所以不会有用户长时间卡在等待状态。这是有效的，因为许多 API 调用预期的行为都非常耗时，例如加载一个云主机，或者上传一个镜像。

（ 3） Compute Worker（ Nova-compute）

Compute Worker 处理管理云主机生命周期。它们通过消息服务接收云主机生命周期管理的请求，并承担操作工作。在一个典型生产环境的云部署中有一些 Compute Workers。云主机部署在哪个可用的 Compute Worker 上取决于调度算法。

（ 4） Network Controller（ Nova-network）

Network Controller 处理主机地网络配置。它包括 IP 地址分配、为项目配置 VLAN、实现安全组、配置计算节点网络。

6.2.1 数据库配置

1.登录 MySQL数据库：

```
# mysql -uroot -p000000
```

```
[root@controller ~]# mysql -uroot -p000000
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 15
Server version: 10.1.12-MariaDB MariaDB Server

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

2.创建 nova_api 和 nova 数据库：

```
# CREATE DATABASE nova;
# CREATE DATABASE nova_api;
```

```
MariaDB [(none)]> CREATE DATABASE nova;
Query OK, 1 row affected (0.06 sec)
MariaDB [(none)]> CREATE DATABASE nova_api;
Query OK, 1 row affected (0.02 sec)
```

6.2.1 数据库配置

3.设置授权用户和密码：

```
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY '000000';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY '00
0000';
Query OK, 0 rows affected (0.28 sec)
```

```
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' IDENTIFIED BY '000000';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' IDENTIFIE
D BY '000000';
Query OK, 0 rows affected (0.01 sec)
```

```
GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'%' IDENTIFIED BY '000000';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova_api'@'%' IDENTIFIED BY
 '000000';
Query OK, 0 rows affected (0.04 sec)
```

```
GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'localhost' IDENTIFIED BY '000000';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova_api'@'localhost' IDENT
IFIED BY '000000';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> exit
```

6.2.2 创建服务凭证和 API 端点

1．生效 admin 用户环境变量

```
# . admin-openrc
```

```
[root@controller ~]# . admin-openrc
```

2．创建服务凭证

创建名为 nova 的用户（user）：

```
# openstack user create --domain default --password-prompt nova
```

```
[root@controller ~]# openstack user create --domain default --password-prompt no
va
User Password:
Repeat User Password:
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | dafc8ccd6b0a469ea6faeb8a4c3620d3 |
| enabled   | True                             |
| id        | 8bbc6bad5f8b424da6a812b15b534152 |
| name      | nova                             |
+-----------+----------------------------------+
```

6.2.2 创建服务凭证和 API 端点

进行关联：给 nova 用户添加 admin 角色：

```
# openstack role add --project service --user nova admin
```

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

创建 nova服务实体认证：

```
# openstack service create --name nova --description "OpenStack Compute" compute
```

```
[root@controller ~]# openstack service create --name nova --description "OpenSta
ck Compute" compute
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OpenStack Compute                |
| enabled     | True                             |
| id          | 2b973331956b4e959e963ed467d23b83 |
| name        | nova                             |
| type        | compute                          |
+-------------+----------------------------------+
```

3.创建API 端点

创建公共端点：

```
# openstack endpoint create --region RegionOne compute public http://controller:8774/v2.1/%\(tenant_id\)s
```

6.2.2 创建服务凭证和 API 端点

```
[root@controller ~]# openstack endpoint create --region RegionOne compute public
 http://controller:8774/v2.1/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | ce49dafd23eb43a7a9e7f9860068153d        |
| interface    | public                                  |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 2b973331956b4e959e963ed467d23b83        |
| service_name | nova                                    |
| service_type | compute                                 |
| url          | http://controller:8774/v2.1/%(tenant_id)s |
+--------------+-----------------------------------------+
```

创建外部端点：

```
# openstack endpoint create --region RegionOne compute internal http://controller:8774/v2.1/%\(tenant_id\)s
```

```
[root@controller ~]# openstack endpoint create --region RegionOne compute intern
al http://controller:8774/v2.1/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | f108c0e008e84fbeac8f20c32e6bd186        |
| interface    | internal                                |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 2b973331956b4e959e963ed467d23b83        |
| service_name | nova                                    |
| service_type | compute                                 |
| url          | http://controller:8774/v2.1/%(tenant_id)s |
+--------------+-----------------------------------------+
```

6.2.2 创建服务凭证和 API 端点

创建管理端点：

```
# openstack endpoint create --region RegionOne compute admin http://controller:8774/v2.1/%\(tenant_id\)s
```

```
[root@controller ~]# openstack endpoint create --region RegionOne compute admin
http://controller:8774/v2.1/%\(tenant_id\)s
+--------------+-----------------------------------------+
| Field        | Value                                   |
+--------------+-----------------------------------------+
| enabled      | True                                    |
| id           | 0861d7ba8420412893ddfc7ad2f749c4        |
| interface    | admin                                   |
| region       | RegionOne                               |
| region_id    | RegionOne                               |
| service_id   | 2b973331956b4e959e963ed467d23b83        |
| service_name | nova                                    |
| service_type | compute                                 |
| url          | http://controller:8774/v2.1/%(tenant_id)s |
+--------------+-----------------------------------------+
```

6.2.3 安装并配置 Nova 组件

1．安装 Glance 所需软件包

```
# yum install openstack-nova-api openstack-nova-conductor openstack-nova-console openstack-nova-novncproxy openstack-nova-scheduler -y
```

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

2．配置 Nova 所需组件

使用vi命令编辑**/etc/nova/nova.conf**文件。

编辑**[DEFAULT]**部分，启用计算和元数据 API。

```
# vi /etc/nova/nova.conf
```

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

```
[DEFAULT]
enabled_apis = osapi_compute,metadata
```

```
[DEFAULT]
enabled_apis = osapi_compute,metadata
```

编辑**[api_database]**和**[database]**部分，配置数据库链接。

```
[api_database]
connection = mysql+pymysql://nova:000000@controller/nova_api
[database]
connection = mysql+pymysql://nova:000000@controller/nova
```

6.2.3 安装并配置 Nova 组件

```
[api_database]
connection = mysql+pymysql://nova:000000@controller/nova_api
[database]
connection = mysql+pymysql://nova:000000@controller/nova
```

编辑**[DEFAULT]**和**[oslo_messaging_rabbit]**部分，配置 RabbitMQ 消息服务器链接。

```
[DEFAULT]
rpc_backend = rabbit

[oslo_messaging_rabbit]
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = 000000
```

```
[DEFAULT]
rpc_backend = rabbit
[oslo_messaging_rabbit]
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = 000000
```

编辑**[DEFAULT]**和**[keystone_authtoken]**部分，配置 Keystone 身份认证。

```
[DEFAULT]
auth_strategy = keystone
```

6.2.3 安装并配置 Nova 组件

```
[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 = 000000
```

```
[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 = 000000
```

6.2.3 安装并配置 Nova 组件

编辑**[DEFAULT]**部分，配置管理 IP 地址和启用网络服务。

```
[DEFAULT]
my_ip = 192.168.100.10
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

```
[DEFAULT]
my_ip = 192.168.100.10
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

编辑**[vnc]**部分，配置 VNC 代理管理 IP 地址。

```
[vnc]
vncserver_listen = $my_ip
vncserver_proxyclient_address = $my_ip
```

```
[vnc]
vncserver_listen = $my_ip
vncserver_proxyclient_address = $my_ip
```

编辑[glance]部分，配置镜像服务 API 端点。

```
[glance]
api_servers = http://controller:9292
```

6.2.3 安装并配置 Nova 组件

```
[glance]
api_servers = http://controller:9292
```

编辑**[oslo_concurrency]**部分，配置 loca_path。

```
[oslo_concurrency]
lock_path = /var/lib/nova/tmp
```

3．同步数据库

```
# su -s /bin/sh -c "nova-manage api_db sync" nova
# su -s /bin/sh -c "nova-manage db sync" nova
```

```
[root@controller ~]# su -s /bin/sh -c "nova-manage api_db sync" nova
[root@controller ~]# su -s /bin/sh -c "nova-manage db sync" nova
/usr/lib/python2.7/site-packages/pymysql/cursors.py:146: Warning: Duplicate inde
x 'block_device_mapping_instance_uuid_virtual_name_device_name_idx' defined on t
he table 'nova.block_device_mapping'. This is deprecated and will be disallowed
in a future release.
  result = self._query(query)
```

注： 进入 nova 数据库查看是否有数据表，验证是否同步成功。

6.2.3 安装并配置 Nova 组件

4．启动并设置 Nova 服务开机自启

```
# 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
```

```
[root@controller ~]# systemctl enable openstack-nova-api.service openstack-nova-
consoleauth.service openstack-novascheduler.service openstack-nova-conductor.ser
vice openstack-nova-novncproxy.service
[root@controller ~]# systemctl start openstack-nova-api.service openstack-nova-c
onsoleauth.service openstack-novascheduler.service openstack-nova-conductor.serv
ice openstack-nova-novncproxy.service
```

6.3 安装并配置计算节点

6.3.1 安装并配置 Nova 组件

1．安装 Nova 组件所需软件包

```
# yum install openstack-nova-compute -y
```

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

2．配置 Nova 所需组件

使用vi命令编辑**/etc/nova/nova.conf** 文件。

编辑**[DEFAULT]**和**[oslo_messaging_rabbit]**部分，配置 RabbitMQ 消息服务器链接。

```
# vi /etc/nova/nova.conf
```

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

```
[DEFAULT]
rpc_backend = rabbit

[oslo_messaging_rabbit]
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = 000000
```

```
[DEFAULT]
rpc_backend = rabbit
```

6.3.1 安装并配置 Nova 组件

```
[oslo_messaging_rabbit]
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = 000000
```

编辑**[DEFAULT]**和**[keystone_authtoken]**部分，配置 Keystone 身份认证。

```
[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 = 000000
```

```
[DEFAULT]
auth_strategy = keystone
```

6.3.1 安装并配置 Nova 组件

```
[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 = 000000
```

编辑**[DEFAULT]**部分，配置管理 IP 地址和启用网络服务支持。

```
[DEFAULT]
my_ip = 192.168.100.20
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

```
[DEFAULT]
my_ip = 192.168.100.20
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

编辑**[vnc]**部分，启用并配置远程控制台的访问。

6.3.1 安装并配置 Nova 组件

```
[vnc]
enabled = True
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $my_ip
novncproxy_base_url = http://192.168.200.10:6080/vnc_auto.html
```

```
[vnc]
enabled = True
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $my_ip
novncproxy_base_url = http://controller:6080/vnc_auto.html
```

编辑**[glance]**部分，配置镜像服务 API 的位置。

```
[glance]
api_servers = http://controller:9292
```

```
[glance]
api_servers = http://controller:9292
```

编辑**[oslo_concurrency]**部分，配置 loca_path。

```
[oslo_concurrency]
lock_path = /var/lib/nova/tmp
```

```
[oslo_concurrency]
lock_path = /var/lib/nova/tmp
```

6.3.2 检查主机是否支持虚拟机硬件加速

1．执行命令

```
# egrep -c '(vmx|svm)' /proc/cpuinfo
```

编辑**/etc/nova/nova.conf** 文件。
编辑**[libvirt]**部分：

```
[libvirt]
virt_type = qemu
```

2．启动并设置 Nova 服务开机自启

```
# systemctl enable libvirtd.service openstack-nova-compute.service
# systemctl start libvirtd.service openstack-nova-compute.service
```

6.4 验证 Nova 服务

在控制节点执行

. admin-openrc

openstack compute service list

```
[root@controller ~]# openstack compute service list
+----+----------------+------------+----------+---------+-------+----------------+
| Id | Binary         | Host       | Zone     | Status  | State | Updated At     |
+----+----------------+------------+----------+---------+-------+----------------+
|  1 | nova-          | controller | internal | enabled | up    | 2021-11-08T13: |
|    | scheduler      |            |          |         |       | 20:35.000000   |
|  2 | nova-          | controller | internal | enabled | up    | 2021-11-08T13: |
|    | conductor      |            |          |         |       | 20:34.000000   |
|  3 | nova-          | controller | internal | enabled | up    | 2021-11-08T13: |
|    | consoleauth    |            |          |         |       | 20:35.000000   |
|  6 | nova-compute   | compute    | nova     | enabled | up    | 2021-11-08T13: |
|    |                |            |          |         |       | 20:33.000000   |
+----+----------------+------------+----------+---------+-------+----------------+
[root@controller ~]#
```

6.4 验证 Nova 服务

在控制节点执行

. admin-openrc

openstack compute service list

```
[root@controller ~]# openstack compute service list
+----+----------------+------------+----------+---------+-------+----------------+
| Id | Binary         | Host       | Zone     | Status  | State | Updated At     |
+----+----------------+------------+----------+---------+-------+----------------+
|  1 | nova-          | controller | internal | enabled | up    | 2021-11-08T13: |
|    | scheduler      |            |          |         |       | 20:35.000000   |
|  2 | nova-          | controller | internal | enabled | up    | 2021-11-08T13: |
|    | conductor      |            |          |         |       | 20:34.000000   |
|  3 | nova-          | controller | internal | enabled | up    | 2021-11-08T13: |
|    | consoleauth    |            |          |         |       | 20:35.000000   |
|  6 | nova-compute   | compute    | nova     | enabled | up    | 2021-11-08T13: |
|    |                |            |          |         |       | 20:33.000000   |
+----+----------------+------------+----------+---------+-------+----------------+
[root@controller ~]#
```

谢谢观看