手动搭建分布式OpenStack Ocata vxlan模式v1.0-发行版

前言 搭建前必须看我

本文档搭建的是分布式O版openstack（controller+ N compute + 1 cinder）的文档。
openstack版本为Ocata。

搭建的时候，请严格按照文档所描写的进行配置，在不熟悉的情况下，严禁自己添加额外的配置和设置！
学习这个文档能搭建基本的openstack环境，切记千万不能用于生产！要用于生产的环境，必须有严格的测试还有额外的高级配置！

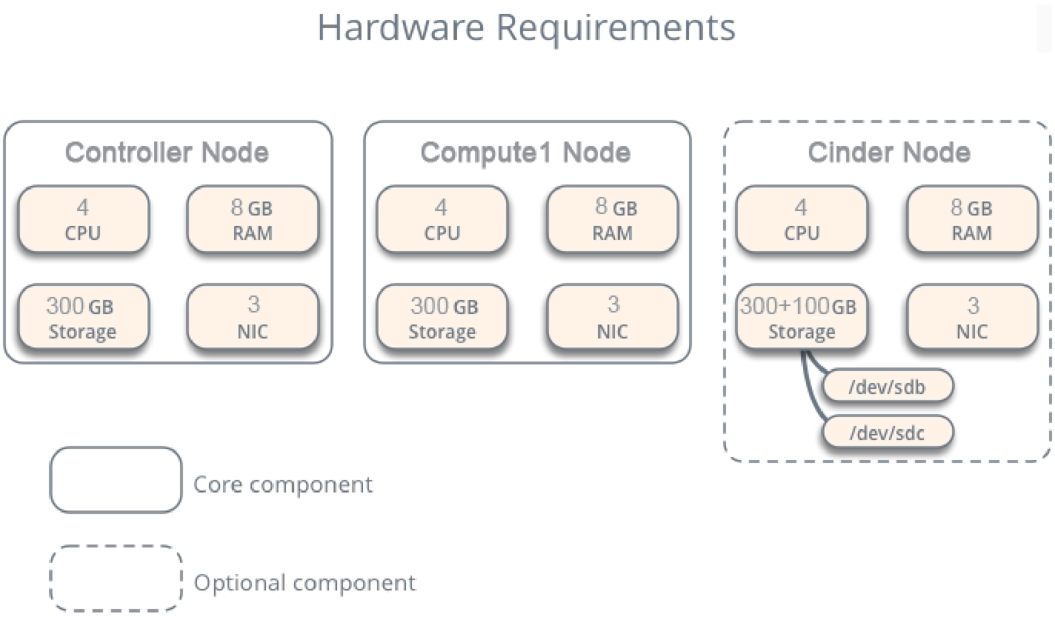
阅读文档注意，红色的部分是重要提示，另外其他加颜色的字体参数也要额外注意！
有些命令很长，注意有换行了，别只敲一半，每条命令前面都带有 #。

一、环境准备

1. 前提准备
安装vmware workstation12.5.0，虚拟出三台配置至少CPU 4c MEM 4G的虚拟机

Controller节点和Compute节点配置：
CPU:4c
MEM:4G
Disk:200G
Network: 3 (eth0 eth1 eth2， 第一块网卡就是extenel的网卡，第二块网卡是admin网卡，第三块是tunnel隧道)

Cinder节点配置：
CPU:4c
MEM:4G
Disk:200G+50G（这个50G可以根据自己需求调整大小）
Network: 2 (eth0 eth1， 第一块网卡就是extenel的网卡，第二块网卡是admin网卡，cinder节点不需要隧道)



注意：此架构设计只适合测试学习环境！不可用于生产！

安装CentOS7.2系统（**最小化安装**，不要yum update升级到7.3！Ocata版7.3下依然有虚拟机启动出现PXE启动问题依旧）+ 关闭防火墙 + 关闭selinux
# systemctl stop firewalld.service
# systemctl disable firewalld.service

安装好相关工具，因为系统是最小化安装的，所以一些ifconfig vim等命令没有，运行下面的命令把它们装上：
# yum install net-tools wget vim ntpdate bash-completion -y

2. 更改hostname
# hostnamectl set-hostname controller

如果是compute就运行：
# hostnamectl set-hostname compute1
cinder节点就运行：
# hostnamectl set-hostname cinder

然后每个节点配置/etc/hosts文件如下
10.1.1.150 controller
10.1.1.151 compute1
10.1.1.152 cinder

3. NTP同步系统时间
# ntpdate cn.pool.ntp.org

然后查看运行date命令查看时间是否同步成功
注意，这个操作很重要，openstack是分布式架构的，每个节点都不能有时间差！
很多同学刚装完centos系统，时间会跟当前北京的时间不一致，所以必须运行下这个命令！
另外，也把这个命令加到开机启动里面去

```
# echo "ntpdate cn.pool.ntp.org" >> /etc/rc.d/rc.local
# chmod +x /etc/rc.d/rc.local
```

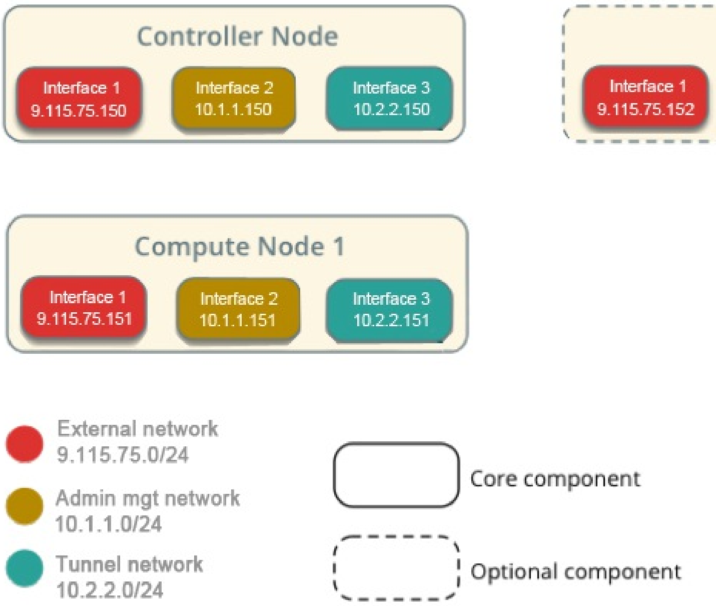
4. 配置IP 网络配置规划

网络配置：
external : 9.110.187.0/24
admin mgt : 10.1.1.0/24
tunnel：10.2.2.0/24
storage：10.3.3.0/24（我们环境没有，如果你集成了ceph就应该用到）

controller虚拟机第一块网卡external，请配置IP 9.110.187.150
第二块网卡admin，请配置IP 10.1.1.150
第三块网卡tunnel，请配置IP 10.2.2.150

compute1虚拟机第一块网卡external，请配置IP 9.110.187.151
第二块网卡admin，请配置IP 10.1.1.151
第三块网卡tunnel，请配置IP 10.2.2.151

cinder虚拟机第一块网卡external，请配置IP 9.110.187.152
第二块网卡admin，请配置IP 10.1.1.152
第三块网卡tunnel，请配置IP 10.2.2.152



三个网络解释：
1. external : 这个网络是链接外网的，也就是说openstack环境里的虚拟机要让用户访问，那必须有个网段是连外网的，用户通过这个网络能访问到虚拟机。如果是搭建的公有云，这个IP段一般是公网的（不是公网，你让用户怎么访问你的虚拟机？）
2. admin mgt：这个网段是用来做管理网络的。管理网络，顾名思义，你的openstack环境里面各个模块之间需要交互，连接数据库，连接Message Queue都是需要一个网络去支撑的，那么这个网段就是这个作用。最简单的理解，openstack自己本身用的IP段。
3. tunnel : 隧道网络，openstack里面使用gre或者vxlan模式，需要有隧道网络；隧道网络采用了点到点通信协议代替了交换连接，在openstack里，这个tunnel就是虚拟机走网络数据流量用的。

当然这3个网络你都放在一块也行，但是只能用于测试学习环境，真正的生产环境是得分开的。在自己学习搭建的时候，通常我们用的是vmware workstation虚拟机，有些同学创建虚拟机后，默认只有一块网卡，有些同学在只有一块网卡就不知道如何下手了，一看有三种网络就晕乎了... 所以，在创建完虚拟机后，请给虚拟机再添加2块网卡，根据生产环境的要求去搭建学习。

三种网络在生产环境里是必须分开的，有的生产环境还有分布式存储，所以还得额外给存储再添加一网络，storage段。网络分开的好处

就是数据分流、安全、不相互干扰。你想想，如果都整一块了，还怎么玩？用户访问虚拟机还使用你openstack的管理段，那太不安全了...

5. 搭建OpenStack内部使用源

关于内部源的搭建，请看视频。

二、 搭建Mariadb

1. 安装mariadb数据库

```
# yum install -y  MariaDB-server MariaDB-client
```

2. 配置mariadb

```
# vim /etc/my.cnf.d/mariadb-openstack.cnf
```

在mysqld区块添加如下内容：

```
[mysqld]
default-storage-engine = innodb
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8
bind-address = 10.1.1.150
```

3、启动数据库及设置mariadb开机启动

```
# systemctl enable mariadb.service
# systemctl restart mariadb.service
# systemctl status mariadb.service
# systemctl list-unit-files |grep mariadb.service
```

4. 配置mariadb，给mariadb设置密码

```
# mysql_secure_installation
```

先按回车，然后按Y，设置mysql密码，然后一直按y结束

这里我们设置的密码是devops

三、安装RabbitMQ

1. 每个节点都安装erlang

```
# yum install -y erlang
```

2. 每个节点都安装RabbitMQ

```
# yum install -y rabbitmq-server
```

3. 每个节点都启动rabbitmq及设置开机启动

```
# systemctl enable rabbitmq-server.service
# systemctl restart rabbitmq-server.service
# systemctl status rabbitmq-server.service
# systemctl list-unit-files |grep rabbitmq-server.service
```

4. 创建openstack，注意将PASSWOED替换为自己的合适密码

```
# rabbitmqctl add_user openstack devops
```

5. 将openstack用户赋予权限

```
# rabbitmqctl set_permissions openstack ".*" ".*" ".*"
# rabbitmqctl set_user_tags openstack administrator
# rabbitmqctl list_users
```

6. 看下监听端口 rabbitmq用的是5672端口

```
# netstat -ntlp |grep 5672
```

7. 查看RabbitMQ插件

```
# /usr/lib/rabbitmq/bin/rabbitmq-plugins list
```

8. 打开RabbitMQ相关插件

```
# /usr/lib/rabbitmq/bin/rabbitmq-plugins enable rabbitmq_management mochiweb webmachine rabbitmq_web_dispatch amqp_client rabbitmq_management_agent
```

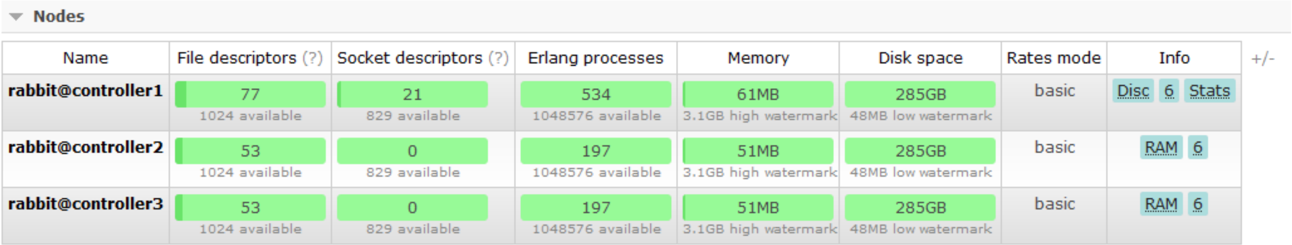
打开相关插件后，重启下rabbitmq服务

```
 systemctl restart rabbitmq-server
```

浏览器输入：http://9.110.187.150:15672   默认用户名密码：guest/guest
通过这个界面，我们能很直观的看到rabbitmq的运行和负载情况

9. 查看rabbitmq状态

用浏览器登录http://9.110.187.150:15672 输入openstack/devops也可以查看状态信息：



Nodes

Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Rates mode	Info	+/-
rabbit@controller1	77 1024 available	21 829 available	534 1048576 available	61MB 3.1GB high watermark	285GB 48MB low watermark	basic	Disc 6 Stats	
rabbit@controller2	53 1024 available	0 829 available	197 1048576 available	51MB 3.1GB high watermark	285GB 48MB low watermark	basic	RAM 6	
rabbit@controller3	53 1024 available	0 829 available	197 1048576 available	51MB 3.1GB high watermark	285GB 48MB low watermark	basic	RAM 6	

四、安装配置Keystone

1、创建keystone数据库

```
CREATE DATABASE keystone;
```

2、创建数据库keystone用户&root用户及赋予权限

```
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY 'devops';
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'devops';
```

注意将devops替换为自己的数据库密码

3、安装keystone和memcached

```
# yum  -y install openstack-keystone httpd mod_wsgi python-openstackclient memcached python-memcached openstack-utils
```

4、启动memcache服务并设置开机自启动

```
# systemctl enable memcached.service
# systemctl restart memcached.service
# systemctl status memcached.service
```

5、配置/etc/keystone/keystone.conf文件

```
# cp /etc/keystone/keystone.conf /etc/keystone/keystone.conf.bak
# >/etc/keystone/keystone.conf
# openstack-config --set  /etc/keystone/keystone.conf DEFAULT transport_url rabbit://openstack:devops@controller
# openstack-config --set  /etc/keystone/keystone.conf database connection mysql://keystone:devops@controller/keystone
# openstack-config --set  /etc/keystone/keystone.conf cache backend oslo_cache.memcache_pool
# openstack-config --set  /etc/keystone/keystone.conf cache enabled true
# openstack-config --set  /etc/keystone/keystone.conf cache memcache_servers controller:11211
# openstack-config --set  /etc/keystone/keystone.conf memcache servers controller:11211
# openstack-config --set  /etc/keystone/keystone.conf token expiration 3600
# openstack-config --set  /etc/keystone/keystone.conf token provider fernet
```

6、配置httpd.conf文件&memcached文件

```
# sed  -i  "s/#ServerName www.example.com:80/ServerName controller/" /etc/httpd/conf/httpd.conf
# sed -i  's/OPTIONS*.*/OPTIONS="-l 127.0.0.1,::1,10.1.1.150"/' /etc/sysconfig/memcached
```

7、配置keystone与httpd结合

```
# ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/
```

8、数据库同步

```
# su -s /bin/sh -c "keystone-manage db_sync" keystone
```

9、初始化fernet

```
# keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
# keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

10、启动httpd，并设置httpd开机启动

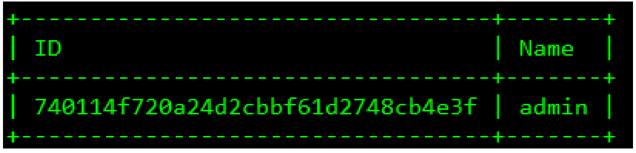
```
# systemctl enable httpd.service
# systemctl restart httpd.service
# systemctl status httpd.service
# systemctl list-unit-files |grep httpd.service
```

11、创建 admin 用户角色

```
# keystone-manage bootstrap \
   --bootstrap-password devops \
   --bootstrap-username admin \
   --bootstrap-project-name admin \
   --bootstrap-role-name admin \
   --bootstrap-service-name keystone \
   --bootstrap-region-id RegionOne \
   --bootstrap-admin-url http://controller:35357/v3 \
   --bootstrap-internal-url http://controller:35357/v3 \
   --bootstrap-public-url http://controller:5000/v3
```

验证：

```
# openstack project list --os-username admin --os-project-name admin --os-user-domain-id default --os-project-domain-id default --os-identity-api-version 3 --os-auth-url http://controller:5000 --os-password devops
```



```
+----------------------------------+-------+
| ID                               | Name  |
+----------------------------------+-------+
| 740114f720a24d2cbbf61d2748cb4e3f | admin |
+----------------------------------+-------+
```

12. 创建admin用户环境变量，创建/root/admin-openrc 文件并写入如下内容：

```
# vim /root/admin-openrc
```

添加以下内容：

```
export OS_USER_DOMAIN_ID=default
export OS_PROJECT_DOMAIN_ID=default
export OS_USERNAME=admin
export OS_PROJECT_NAME=admin
export OS_PASSWORD=devops
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
export OS_AUTH_URL=http://controller:35357/v3
```

13、创建service项目

```
# source /root/admin-openrc
# openstack project create --domain default   --description "Service Project" service
```

14、创建demo项目

```
# openstack project create --domain default   --description "Demo Project" demo
```

15、创建demo用户

```
# openstack user create --domain default  demo  --password devops
```

注意：devops为demo用户密码

16、创建user角色将demo用户赋予user角色

```
# openstack role create user
# openstack role add --project demo --user demo user
```

17、验证keystone

```
# unset OS_TOKEN OS_URL
# 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 --os-password devops
# openstack --os-auth-url http://controller:5000/v3   --os-project-domain-name default --os-user-domain-name default   --os-project-name demo --os-username demo token issue --os-password devops
```

五、安装配置glance

1、创建glance数据库

```
 CREATE DATABASE glance;
```

2、创建数据库用户并赋予权限

```
 GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' IDENTIFIED BY 'devops';
 GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' IDENTIFIED BY 'devops';
```

3、创建glance用户及赋予admin权限

```
# source /root/admin-openrc
# openstack user create  --domain default glance  --password devops
# openstack role add --project service --user glance admin
```

4、创建image服务

```
# openstack service create --name glance --description "OpenStack Image service" image
```

5、创建glance的endpoint

```
# openstack endpoint create --region RegionOne  image public http://controller:9292
# openstack endpoint create --region RegionOne  image internal http://controller:9292
# openstack endpoint create --region RegionOne  image admin http://controller:9292
```

6、安装glance相关rpm包

```
# yum install openstack-glance -y
```

7、修改glance配置文件/etc/glance/glance-api.conf

注意红色的密码设置成你自己的

```
# cp /etc/glance/glance-api.conf /etc/glance/glance-api.conf.bak
# >/etc/glance/glance-api.conf
# openstack-config --set  /etc/glance/glance-api.conf DEFAULT transport_url rabbit://openstack:devops@controller
# openstack-config --set  /etc/glance/glance-api.conf database connection mysql+pymysql://glance:devops@controller/glance
# openstack-config --set  /etc/glance/glance-api.conf keystone_authtoken auth_uri  http://controller:5000
# openstack-config --set  /etc/glance/glance-api.conf keystone_authtoken auth_url  http://controller:35357
# openstack-config --set  /etc/glance/glance-api.conf keystone_authtoken memcached_servers  controller:11211
# openstack-config --set  /etc/glance/glance-api.conf keystone_authtoken auth_type  password
# openstack-config --set  /etc/glance/glance-api.conf keystone_authtoken project_domain_name  default
# openstack-config --set  /etc/glance/glance-api.conf keystone_authtoken user_domain_name   default
# openstack-config --set  /etc/glance/glance-api.conf keystone_authtoken username  glance
# openstack-config --set  /etc/glance/glance-api.conf keystone_authtoken password  devops
# openstack-config --set  /etc/glance/glance-api.conf keystone_authtoken project_name  service
# openstack-config --set  /etc/glance/glance-api.conf paste_deploy flavor  keystone
# openstack-config --set  /etc/glance/glance-api.conf glance_store stores  file,http
# openstack-config --set  /etc/glance/glance-api.conf glance_store default_store  file
# openstack-config --set  /etc/glance/glance-api.conf glance_store filesystem_store_datadir  /var/lib/glance/images/
```

8、修改glance配置文件/etc/glance/glance-registry.conf：

```
# cp /etc/glance/glance-registry.conf /etc/glance/glance-registry.conf.bak
# >/etc/glance/glance-registry.conf
# openstack-config --set  /etc/glance/glance-registry.conf DEFAULT transport_url rabbit://openstack:devops@controller
# openstack-config --set  /etc/glance/glance-registry.conf database connection mysql+pymysql://glance:devops@controller/glance
# openstack-config --set  /etc/glance/glance-registry.conf keystone_authtoken auth_uri  http://controller:5000
# openstack-config --set  /etc/glance/glance-registry.conf keystone_authtoken auth_url  http://controller:35357
# openstack-config --set  /etc/glance/glance-registry.conf keystone_authtoken memcached_servers  controller:11211
# openstack-config --set  /etc/glance/glance-registry.conf keystone_authtoken auth_type  password
# openstack-config --set  /etc/glance/glance-registry.conf keystone_authtoken project_domain_name  default
# openstack-config --set  /etc/glance/glance-registry.conf keystone_authtoken user_domain_name  default
# openstack-config --set  /etc/glance/glance-registry.conf keystone_authtoken project_name  service
# openstack-config --set  /etc/glance/glance-registry.conf keystone_authtoken username  glance
# openstack-config --set  /etc/glance/glance-registry.conf keystone_authtoken password devops
# openstack-config --set  /etc/glance/glance-registry.conf paste_deploy flavor  keystone
```

9、同步glance数据库

```
# su -s /bin/sh -c "glance-manage db_sync" glance
```

10、启动glance及设置开机启动

```
# systemctl enable openstack-glance-api.service openstack-glance-registry.service
# systemctl restart openstack-glance-api.service openstack-glance-registry.service
# systemctl status openstack-glance-api.service openstack-glance-registry.service
```

12、下载测试镜像文件

```
# wget  http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img
```

13、上传镜像到glance

```
# source /root/admin-openrc

# glance image-create --name "cirros-0.3.4-x86_64" --file cirros-0.3.4-x86_64-disk.img  --disk-format qcow2 --container-format bare --visibility public --progress
```

如果你做好了一个CentOS6.7系统的镜像，也可以用这命令操作，例：

```
# glance image-create --name "CentOS7.1-x86_64" --file CentOS_7.1.qcow2  --disk-format qcow2 --container-format bare --visibility public --progress
```

查看镜像列表：

```
# glance image-list
```

六、安装配置nova

1、创建nova数据库

```
 CREATE DATABASE nova;
 CREATE DATABASE nova_api;
 CREATE DATABASE nova_cell0;
```

2、创建数据库用户并赋予权限

```
 GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' IDENTIFIED BY 'devops';
 GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY 'devops';
 GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'localhost'   IDENTIFIED BY 'devops';
 GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'%'   IDENTIFIED BY 'devops';
 GRANT ALL PRIVILEGES ON nova_cell0.* TO 'nova'@'localhost'   IDENTIFIED BY 'devops';
 GRANT ALL PRIVILEGES ON nova_cell0.* TO 'nova'@'%'   IDENTIFIED BY 'devops';
 GRANT ALL PRIVILEGES ON *.* TO 'root'@'controller' IDENTIFIED BY 'devops';
 FLUSH PRIVILEGES;
```

注：查看授权列表信息 SELECT DISTINCT CONCAT('User: ''',user,'''@''',host,''';') AS query FROM mysql.user;
　　取消之前某个授权 REVOKE ALTER ON *.* TO 'root'@'controller' IDENTIFIED BY 'devops';

3、创建nova用户及赋予admin权限

```
# source /root/admin-openrc
# openstack user create --domain default   nova  --password devops
# openstack role add --project service --user nova admin
```

4、创建computer服务

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

5、创建nova的endpoint

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

6、安装nova相关软件

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

7、配置nova的配置文件/etc/nova/nova.conf

```
# cp /etc/nova/nova.conf /etc/nova/nova.conf.bak
# >/etc/nova/nova.conf
# openstack-config --set  /etc/nova/nova.conf DEFAULT enabled_apis  osapi_compute,metadata
# openstack-config --set  /etc/nova/nova.conf DEFAULT auth_strategy  keystone
# openstack-config --set  /etc/nova/nova.conf DEFAULT my_ip 10.1.1.150
# openstack-config --set  /etc/nova/nova.conf DEFAULT use_neutron True
# openstack-config --set  /etc/nova/nova.conf DEFAULT firewall_driver  nova.virt.firewall.NoopFirewallDriver
# openstack-config --set  /etc/nova/nova.conf DEFAULT transport_url rabbit://openstack:devops@controller
# openstack-config --set  /etc/nova/nova.conf database connection mysql+pymysql://nova:devops@controller/nova
# openstack-config --set  /etc/nova/nova.conf api_database connection mysql+pymysql://nova:devops@controller/nova_api
# openstack-config --set  /etc/nova/nova.conf scheduler discover_hosts_in_cells_interval -1
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken auth_uri  http://controller:5000
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken auth_url  http://controller:35357
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken memcached_servers controller:11211
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken auth_type  password
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken project_domain_name   default
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken user_domain_name   default
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken project_name  service
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken username  nova
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken password devops
# openstack-config --set  /etc/nova/nova.conf keystone_authtoken service_token_roles_required True
# openstack-config --set  /etc/nova/nova.conf vnc vncserver_listen  10.1.1.150
# openstack-config --set  /etc/nova/nova.conf vnc vncserver_proxyclient_address  10.1.1.150
# openstack-config --set  /etc/nova/nova.conf glance api_servers  http://controller:9292
# openstack-config --set  /etc/nova/nova.conf oslo_concurrency lock_path  /var/lib/nova/tmp
```

注意：其他节点上记得替换IP，还有密码，文档红色以及绿色的地方。

8、设置cell（单元格）

关于cell（单元格）的介绍，引用出自于九州云分享的《Ocata组件Nova Cell V2 详解》& 有云的《引入Cells功能最核心要解决的问题就是OpenStack集群的扩展性》两篇文章的整合介绍：

OpenStack 在控制平面上的性能瓶颈主要在 Message Queue 和 Database 。尤其是 Message Queue，随着计算节点的增加，性能变的越来越差，因为openstack里每个资源和接口都是通过消息队列来通信的，有测试表明，当集群规模到了200，一个消息可能要在十几秒后才会响应；为了应对这种情况，引入Cells功能以解决OpenStack集群的扩展性。

同步下nova数据库

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

设置cell_v2关联上创建好的数据库nova_cell0

```
# nova-manage cell_v2 map_cell0 --database_connection mysql+pymysql://root:devops@controller/nova_cell0
```

创建一个常规cell，名字叫cell1，这个单元格里面将会包含计算节点

```
# nova-manage cell_v2 create_cell --verbose --name cell1 --database_connection mysql+pymysql://root:devops@controller/nova_cell0 --transport-url rabbit://openstack:devops@controller:5672/
```

检查部署是否正常

```
# nova-status upgrade check
```

创建和映射cell0，并将现有计算主机和实例映射到单元格中

```
# nova-manage cell_v2 simple_cell_setup
```

查看已经创建好的单元格列表

```
# nova-manage cell_v2 list_cells --verbose
```

注意，如果有新添加的计算节点，需要运行下面命令来发现，并且添加到单元格中

```
# nova-manage cell_v2 discover_hosts
```

当然，你可以在控制节点的nova.conf文件里[scheduler]模块下添加 discover_hosts_in_cells_interval=-1 这个设置来自动发现

9、安装placement

从Ocata开始，需要安装配置placement参与nova调度了，不然虚拟机将无法创建！

```
# yum install -y openstack-nova-placement-api
```

创建placement用户和placement 服务

```
# openstack user create --domain default  placement  --password devops
# openstack role add --project service --user placement admin
# openstack service create --name placement --description "OpenStack Placement" placement
```

创建placement endpoint

```
# openstack endpoint create --region RegionOne placement public http://controller:8778
# openstack endpoint create --region RegionOne placement admin http://controller:8778
# openstack endpoint create --region RegionOne placement internal http://controller:8778
```

把placement 整合到nova.conf里

```
# openstack-config --set  /etc/nova/nova.conf placement auth_url http://controller:35357
# openstack-config --set  /etc/nova/nova.conf placement memcached_servers controller:11211
# openstack-config --set  /etc/nova/nova.conf placement auth_type password
# openstack-config --set  /etc/nova/nova.conf placement project_domain_name default
# openstack-config --set  /etc/nova/nova.conf placement user_domain_name default
# openstack-config --set  /etc/nova/nova.conf placement project_name service
# openstack-config --set  /etc/nova/nova.conf placement username nova
# openstack-config --set  /etc/nova/nova.conf placement password devops
# openstack-config --set  /etc/nova/nova.conf placement os_region_name RegionOne
```

配置修改00-nova-placement-api.conf文件，这步没做创建虚拟机的时候会出现禁止访问资源的问题

```
# cd /etc/httpd/conf.d/
# cp 00-nova-placement-api.conf 00-nova-placement-api.conf.bak
# >00-nova-placement-api.conf
# vim 00-nova-placement-api.conf
```

添加以下内容：

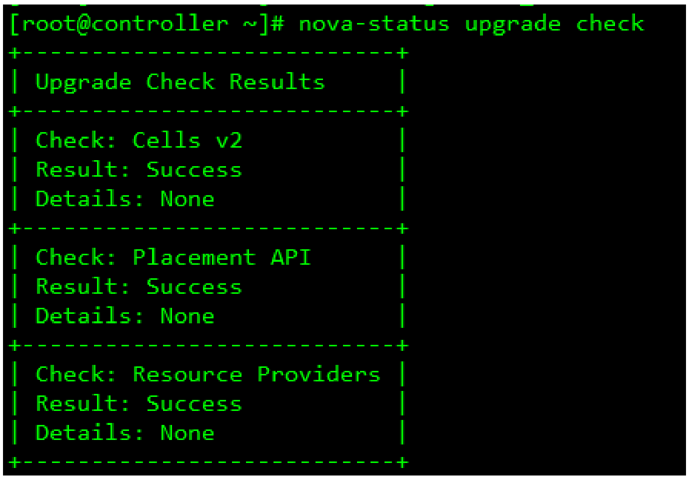
```
Listen 8778

<VirtualHost *:8778>
WSGIProcessGroup nova-placement-api
WSGIApplicationGroup %{GLOBAL}
WSGIPassAuthorization On
WSGIDaemonProcess nova-placement-api processes=3 threads=1 user=nova group=nova
WSGIScriptAlias / /usr/bin/nova-placement-api
<Directory "/">
   Order allow,deny
   Allow from all
   Require all granted
</Directory>
<IfVersion >= 2.4>
  ErrorLogFormat "%M"
</IfVersion>
  ErrorLog /var/log/nova/nova-placement-api.log
</VirtualHost>

Alias /nova-placement-api /usr/bin/nova-placement-api
<Location /nova-placement-api>
  SetHandler wsgi-script
  Options +ExecCGI
  WSGIProcessGroup nova-placement-api
  WSGIApplicationGroup %{GLOBAL}
  WSGIPassAuthorization On
</Location>
```

重启下httpd服务

```
# systemctl restart httpd
```

检查下是否配置成功

```
# nova-status upgrade check
```



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

10、设置nova相关服务开机启动

```
# systemctl enable openstack-nova-api.service openstack-nova-cert.service openstack-nova-consoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-novncproxy.service
```

启动nova服务：

```
# systemctl restart openstack-nova-api.service openstack-nova-cert.service openstack-nova-consoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-novncproxy.service
```

查看nova服务：

```
# systemctl status openstack-nova-api.service openstack-nova-cert.service openstack-nova-consoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-novncproxy.service
```

```
# systemctl list-unit-files |grep openstack-nova-*
```

11、验证nova服务

```
# unset OS_TOKEN OS_URL
# source /root/admin-openrc
# nova service-list
# openstack endpoint list  查看endpoint list
```

看是否有结果正确输出

七、安装配置neutron

1、创建neutron数据库

```
CREATE DATABASE neutron;
```

2、创建数据库用户并赋予权限

```
GRANT ALL PRIVILEGES ON neutron.* TO  'neutron'@'localhost' IDENTIFIED BY 'devops';
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' IDENTIFIED BY 'devops';
```

3、创建neutron用户及赋予admin权限

```
# source /root/admin-openrc
# openstack user create --domain default  neutron  --password devops
# openstack role add --project service --user neutron admin
```

4、创建network服务

```
# openstack service create --name neutron --description "OpenStack Networking" network
```

5、创建endpoint

```
# openstack endpoint create --region RegionOne  network public http://controller:9696
# openstack endpoint create --region RegionOne  network internal http://controller:9696
# openstack endpoint create --region RegionOne  network admin http://controller:9696
```

6、安装neutron相关软件

```
# yum install openstack-neutron openstack-neutron-ml2 openstack-neutron-linuxbridge ebtables -y
```

7、配置neutron配置文件/etc/neutron/neutron.conf

```
# cp /etc/neutron/neutron.conf /etc/neutron/neutron.conf.bak
# >/etc/neutron/neutron.conf
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT core_plugin  ml2
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT service_plugins  router
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT allow_overlapping_ips  True
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT auth_strategy  keystone
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT transport_url rabbit://openstack:devops@controller
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT notify_nova_on_port_status_changes True
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT notify_nova_on_port_data_changes True
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken auth_uri  http://controller:5000
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken auth_url  http://controller:35357
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken memcached_servers  controller:11211
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken auth_type  password
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken project_domain_name  default
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken user_domain_name  default
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken project_name  service
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken username  neutron
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken password  devops
# openstack-config --set  /etc/neutron/neutron.conf database connection
mysql+pymysql://neutron:devops@controller/neutron
# openstack-config --set  /etc/neutron/neutron.conf nova auth_url  http://controller:35357
# openstack-config --set  /etc/neutron/neutron.conf nova auth_type  password
# openstack-config --set  /etc/neutron/neutron.conf nova project_domain_name  default
# openstack-config --set  /etc/neutron/neutron.conf nova user_domain_name  default
# openstack-config --set  /etc/neutron/neutron.conf nova region_name  RegionOne
# openstack-config --set  /etc/neutron/neutron.conf nova project_name  service
# openstack-config --set  /etc/neutron/neutron.conf nova username  nova
# openstack-config --set  /etc/neutron/neutron.conf nova password devops
# openstack-config --set  /etc/neutron/neutron.conf oslo_concurrency lock_path  /var/lib/neutron/tmp
```

8、配置/etc/neutron/plugins/ml2/ml2_conf.ini

```
# openstack-config --set   /etc/neutron/plugins/ml2/ml2_conf.ini ml2 type_drivers flat,vlan,vxlan
# openstack-config --set   /etc/neutron/plugins/ml2/ml2_conf.ini ml2 mechanism_drivers linuxbridge,l2population
# openstack-config --set   /etc/neutron/plugins/ml2/ml2_conf.ini ml2 extension_drivers port_security
# openstack-config --set   /etc/neutron/plugins/ml2/ml2_conf.ini ml2 tenant_network_types vxlan
# openstack-config --set   /etc/neutron/plugins/ml2/ml2_conf.ini ml2 path_mtu 1500
# openstack-config --set   /etc/neutron/plugins/ml2/ml2_conf.ini ml2_type_flat flat_networks  provider
# openstack-config --set   /etc/neutron/plugins/ml2/ml2_conf.ini ml2_type_vxlan vni_ranges  1:1000
# openstack-config --set   /etc/neutron/plugins/ml2/ml2_conf.ini securitygroup enable_ipset  True
```

9、配置/etc/neutron/plugins/ml2/linuxbridge_agent.ini

```
# openstack-config --set  /etc/neutron/plugins/ml2/linuxbridge_agent.ini DEFAULT debug false
# openstack-config --set  /etc/neutron/plugins/ml2/linuxbridge_agent.ini  linux_bridge physical_interface_mappings
provider:eno16777736
# openstack-config --set  /etc/neutron/plugins/ml2/linuxbridge_agent.ini  vxlan  enable_vxlan  True
# openstack-config --set  /etc/neutron/plugins/ml2/linuxbridge_agent.ini  vxlan  local_ip  10.2.2.150
# openstack-config --set  /etc/neutron/plugins/ml2/linuxbridge_agent.ini  vxlan l2_population  True
# openstack-config --set  /etc/neutron/plugins/ml2/linuxbridge_agent.ini  agent  prevent_arp_spoofing  True
# openstack-config --set  /etc/neutron/plugins/ml2/linuxbridge_agent.ini  securitygroup  enable_security_group  True
# openstack-config --set   /etc/neutron/plugins/ml2/linuxbridge_agent.ini securitygroup  firewall_driver
neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
```

注意ens160是public网卡，一般这里写的网卡名都是能访问外网的，如果不是外网网卡，那么VM就会与外界网络隔离。
local_ip 定义的是隧道网络，vxLan下 vm-linuxbridge->vxlan ------tun-----vxlan->linuxbridge-vm

10、配置 /etc/neutron/l3_agent.ini

```
# openstack-config --set  /etc/neutron/l3_agent.ini  DEFAULT  interface_driver
neutron.agent.linux.interface.BridgeInterfaceDriver
# openstack-config --set  /etc/neutron/l3_agent.ini  DEFAULT  external_network_bridge
# openstack-config --set  /etc/neutron/l3_agent.ini  DEFAULT  debug false
```

11、配置/etc/neutron/dhcp_agent.ini

```
# openstack-config --set  /etc/neutron/dhcp_agent.ini  DEFAULT  interface_driver
```

```
neutron.agent.linux.interface.BridgeInterfaceDriver
# openstack-config --set  /etc/neutron/dhcp_agent.ini  DEFAULT dhcp_driver  neutron.agent.linux.dhcp.Dnsmasq
# openstack-config --set  /etc/neutron/dhcp_agent.ini  DEFAULT enable_isolated_metadata True
# openstack-config --set  /etc/neutron/dhcp_agent.ini  DEFAULT verbose True
# openstack-config --set  /etc/neutron/dhcp_agent.ini  DEFAULT debug false
```

12、重新配置/etc/nova/nova.conf，配置这步的目的是让compute节点能使用上neutron网络

```
# openstack-config --set  /etc/nova/nova.conf  neutron url  http://controller:9696
# openstack-config --set  /etc/nova/nova.conf  neutron auth_url  http://controller:35357
# openstack-config --set  /etc/nova/nova.conf  neutron auth_plugin  password
# openstack-config --set  /etc/nova/nova.conf  neutron project_domain_id  default
# openstack-config --set  /etc/nova/nova.conf  neutron user_domain_id  default
# openstack-config --set  /etc/nova/nova.conf  neutron region_name  RegionOne
# openstack-config --set  /etc/nova/nova.conf  neutron project_name service
# openstack-config --set  /etc/nova/nova.conf  neutron username  neutron
# openstack-config --set  /etc/nova/nova.conf  neutron password  devops
# openstack-config --set  /etc/nova/nova.conf  neutron service_metadata_proxy  True
# openstack-config --set  /etc/nova/nova.conf  neutron metadata_proxy_shared_secret  devops
```

13、将dhcp-option-force=26,1450写入/etc/neutron/dnsmasq-neutron.conf

```
# echo "dhcp-option-force=26,1450" >/etc/neutron/dnsmasq-neutron.conf
```

14、配置/etc/neutron/metadata_agent.ini

```
# openstack-config --set  /etc/neutron/metadata_agent.ini  DEFAULT nova_metadata_ip controller
# openstack-config --set  /etc/neutron/metadata_agent.ini  DEFAULT metadata_proxy_shared_secret devops
# openstack-config --set  /etc/neutron/metadata_agent.ini  DEFAULT metadata_workers 4
# openstack-config --set  /etc/neutron/metadata_agent.ini  DEFAULT verbose  True
# openstack-config --set  /etc/neutron/metadata_agent.ini  DEFAULT debug false
# openstack-config --set  /etc/neutron/metadata_agent.ini  DEFAULT nova_metadata_protocol http
```

15、创建硬链接

```
# ln -s /etc/neutron/plugins/ml2/ml2_conf.ini /etc/neutron/plugin.ini
```

16、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
```

17、重启nova服务，因为刚才改了nova.conf

```
# systemctl restart openstack-nova-api.service
# systemctl status openstack-nova-api.service
```

18、重启neutron服务并设置开机启动

```
# systemctl enable neutron-server.service  neutron-linuxbridge-agent.service neutron-dhcp-agent.service  neutron-metadata-agent.service
# systemctl restart neutron-server.service  neutron-linuxbridge-agent.service  neutron-dhcp-agent.service  neutron-metadata-agent.service
# systemctl status neutron-server.service  neutron-linuxbridge-agent.service  neutron-dhcp-agent.service  neutron-metadata-agent.service
```

19、启动neutron-l3-agent.service并设置开机启动

```
# systemctl enable neutron-l3-agent.service
# systemctl restart neutron-l3-agent.service
# systemctl status neutron-l3-agent.service
```

20、执行验证

```
# source /root/admin-openrc
# neutron ext-list
# neutron agent-list
```

21、创建vxLan模式网络，让虚拟机能外出

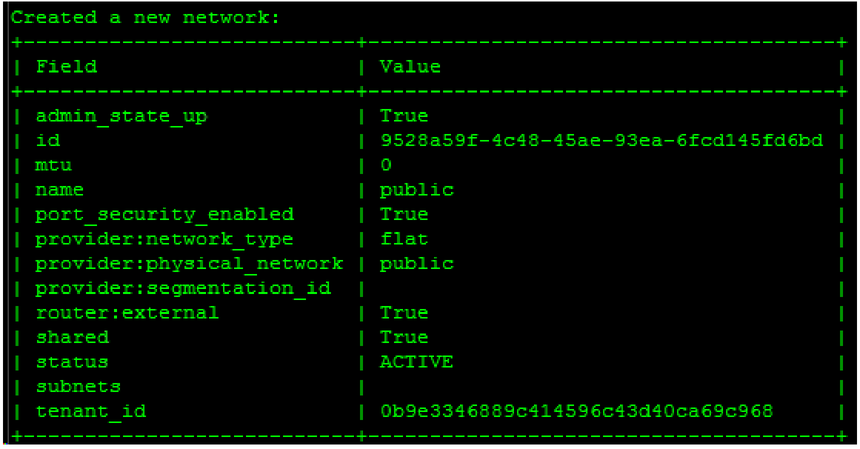
a. 首先先执行环境变量

```
# source /root/admin-openrc
```

b. 创建flat模式的public网络，注意这个public是外出网络，必须是flat模式的

# neutron --debug net-create --shared provider --router:external True --provider:network_type flat --provider:physical_network provider

执行完这步，在界面里进行操作，把public网络设置为共享和外部网络，创建后，结果为:



```
Created a new network:
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | True                                 |
| id                        | 9528a59f-4c48-45ae-93ea-6fcd145fd6bd |
| mtu                       | 0                                    |
| name                      | public                               |
| port_security_enabled     | True                                 |
| provider:network_type     | flat                                 |
| provider:physical_network | public                               |
| provider:segmentation_id  |                                      |
| router:external           | True                                 |
| shared                    | True                                 |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tenant_id                 | 0b9e3346889c414596c43d40ca69c968     |
+---------------------------+--------------------------------------+
```

c. 创建public网络子网，名为public-sub，网段就是9.110.187，并且IP范围是50-90（这个一般是给VM用的floating IP了），dns设置为8.8.8.8，网关为9.110.187.2
# neutron subnet-create provider 9.110.187.0/24 --name provider-sub --allocation-pool start=9.110.187.50,end=9.110.187.90 --dns-nameserver 8.8.8.8 --gateway 9.110.187.2

d. 创建名为private的私有网络, 网络模式为vxlan
# neutron net-create private --provider:network_type vxlan --router:external False --shared

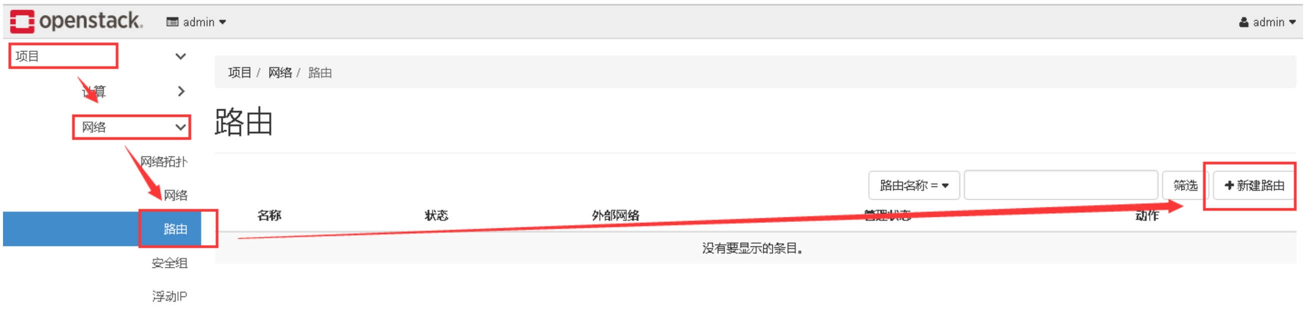
e. 创建名为private-subnet的私有网络子网，网段为192.168.1.0, 这个网段就是虚拟机获取的私有的IP地址
# neutron subnet-create private --name private-subnet --gateway 192.168.1.1 192.168.1.0/24

假如你们公司的私有云环境是用于不同的业务，比如行政、销售、技术等，那么你可以创建3个不同名称的私有网络
# neutron net-create private-office --provider:network_type vxlan --router:external False --shared
# neutron subnet-create private-office --name office-net --gateway 192.168.2.1 192.168.2.0/24

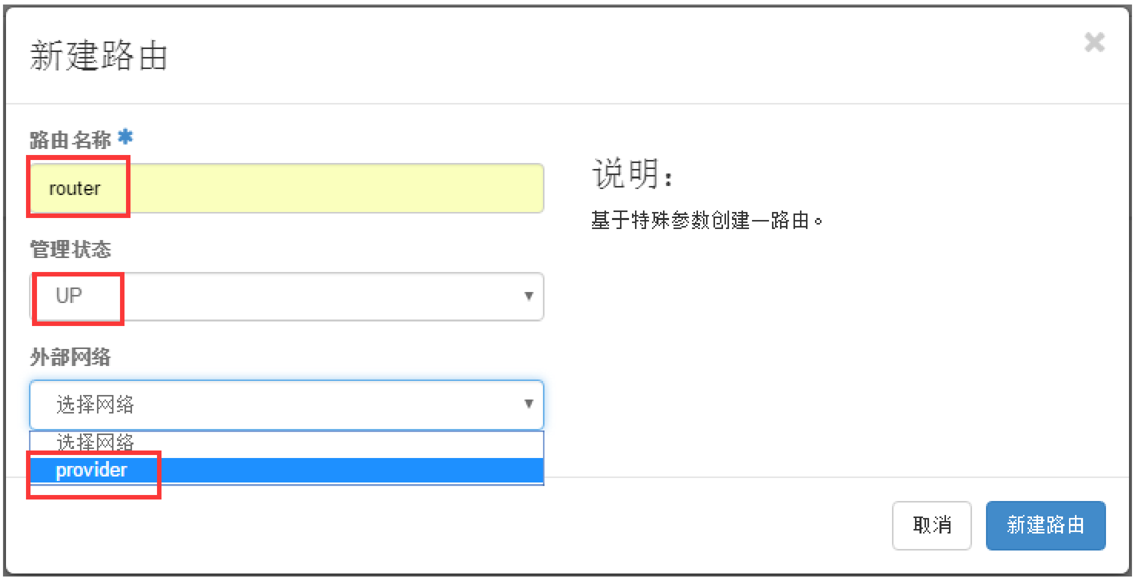
# neutron net-create private-sale --provider:network_type vxlan --router:external False --shared
# neutron subnet-create private-sale --name sale-net --gateway 192.168.3.1 192.168.3.0/24

# neutron net-create private-technology --provider:network_type vxlan --router:external False --shared
# neutron subnet-create private-technology --name technology-net --gateway 192.168.4.1 192.168.4.0/24

f. 创建路由，我们在界面上操作
点击项目-->网络-->路由-->新建路由



路由名称随便命名，我这里写"router", 管理员状态，选择"上"(up)，外部网络选择"provider"



新建路由

路由名称 *

router

管理状态

UP

外部网络

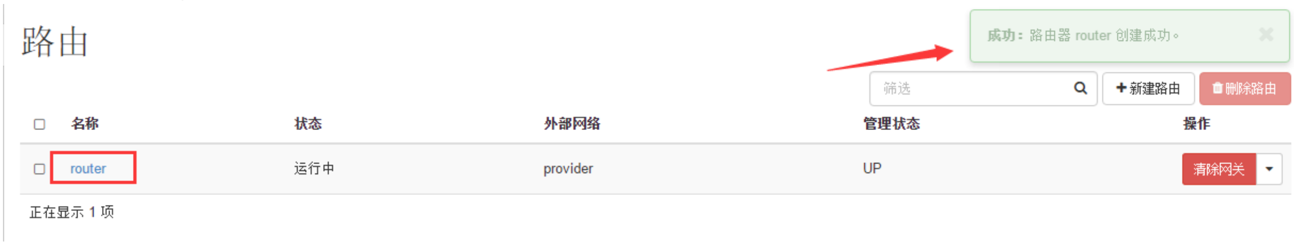
选择网络

provider

说明：

基于特殊参数创建一路由。

点击"新建路由"后，提示创建router创建成功



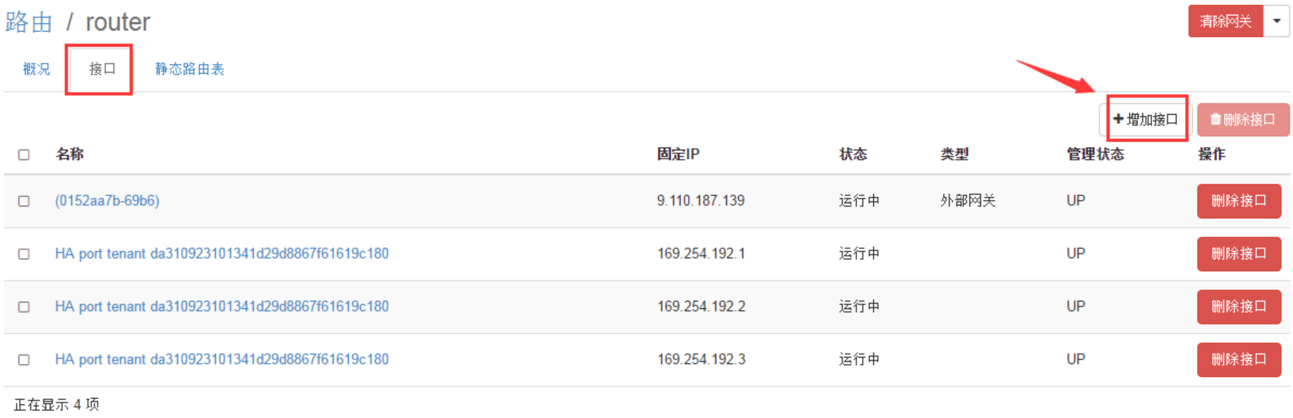
路由

成功：路由器 router 创建成功。

□	名称	状态	外部网络	管理状态	操作
□	router	运行中	provider	UP	清除网关

正在显示 1 项

接着点击"接口"-->"增加接口"

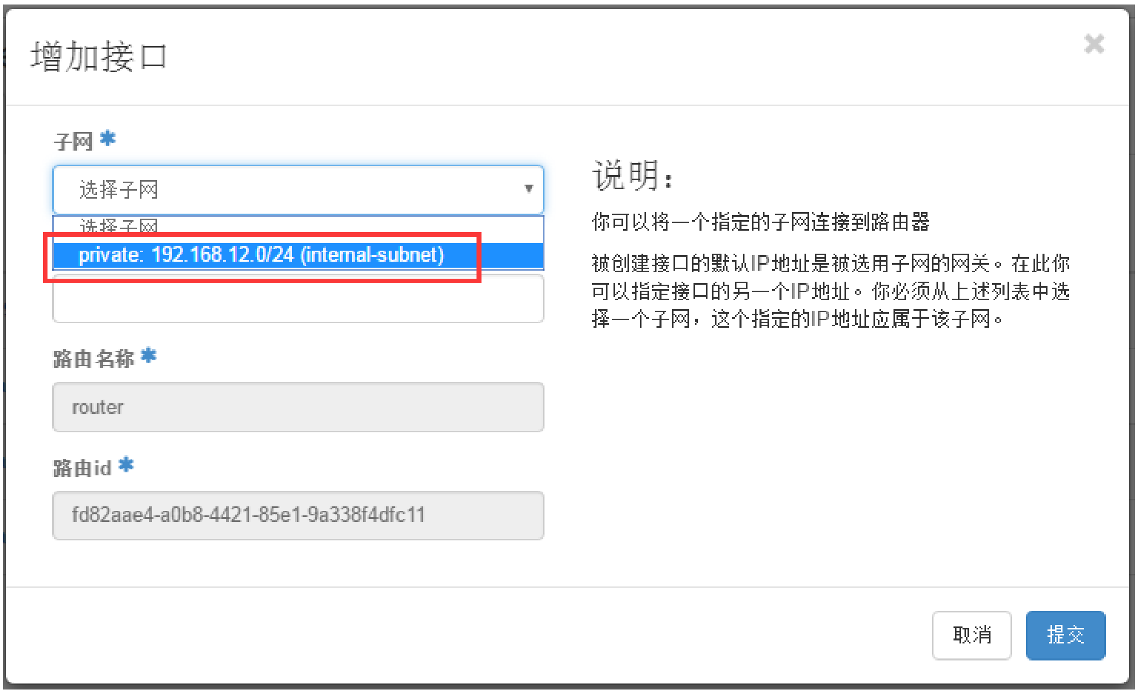


路由 / router

□	名称	固定IP	状态	类型	管理状态	操作
□	(0152aa7b-69b6)	9.110.187.139	运行中	外部网关	UP	删除接口
□	HA port tenant da310923101341d29d8867f61619c180	169.254.192.1	运行中		UP	删除接口
□	HA port tenant da310923101341d29d8867f61619c180	169.254.192.2	运行中		UP	删除接口
□	HA port tenant da310923101341d29d8867f61619c180	169.254.192.3	运行中		UP	删除接口

正在显示 4 项

添加一个连接私网的接口，选中"private: 192.168.12.0/24"



增加接口

×

子网 *

选择子网

选择子网

private: 192.168.12.0/24 (internal-subnet)

路由名称 *

router

路由id *

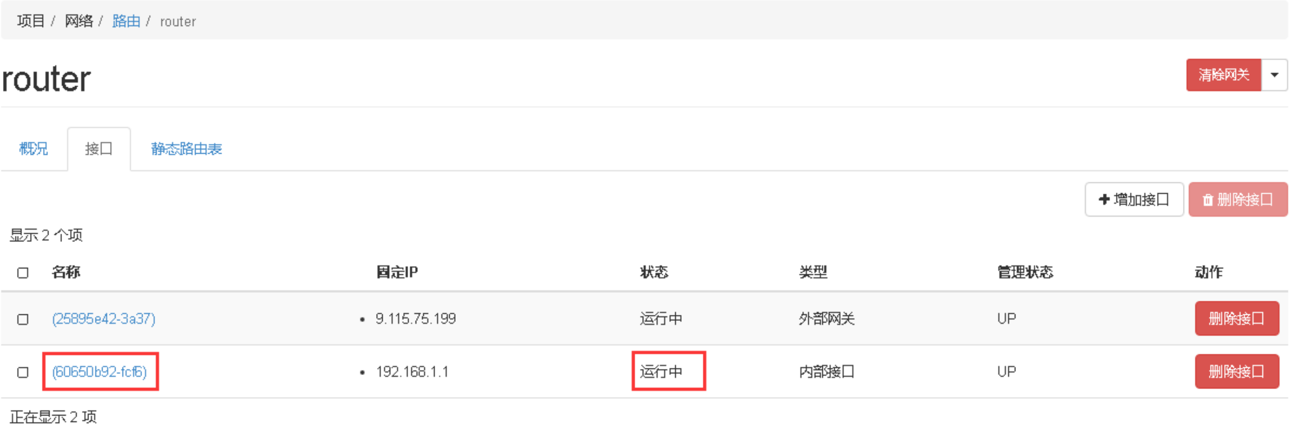
fd82aae4-a0b8-4421-85e1-9a338f4dfc11

说明：

你可以将一个指定的子网连接到路由器

被创建接口的默认IP地址是被选用子网的网关。在此你可以指定接口的另一个IP地址。你必须从上述列表中选择一个子网，这个指定的IP地址应属于该子网。

点击"增加接口"成功后，我们可以看到两个接口先是down的状态，过一会儿刷新下就是running状态（注意，一定得是运行running状态，不然到时候虚拟机网络会出不去）



router

显示 2 个项

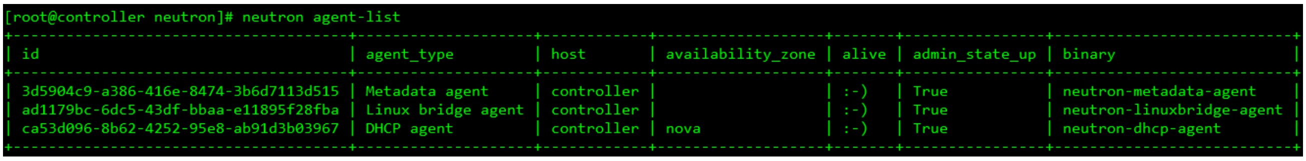
□	名称	固定IP	状态	类型	管理状态	动作
□	(25895e42-3a37)	• 9.115.75.199	运行中	外部网关	UP	删除接口
□	(60650b92-fcf6)	• 192.168.1.1	运行中	内部接口	UP	删除接口

正在显示 2 项

22、检查网络服务

neutron agent-list

看服务是否是笑脸



```
[root@controller neutron]# neutron agent-list
+--------------------------------------+--------------------+------------+-------------------+-------+----------------+---------------------------+
| id                                   | agent_type         | host       | availability_zone | alive | admin_state_up | binary                    |
+--------------------------------------+--------------------+------------+-------------------+-------+----------------+---------------------------+
| 3d5904c9-a386-416e-8474-3b6d7113d515 | Metadata agent     | controller |                   | :-)   | True           | neutron-metadata-agent    |
| ad1179bc-6dc5-43df-bbaa-e11895f28fba | Linux bridge agent | controller |                   | :-)   | True           | neutron-linuxbridge-agent |
| ca53d096-8b62-4252-95e8-ab91d3b03967 | DHCP agent         | controller | nova              | :-)   | True           | neutron-dhcp-agent        |
+--------------------------------------+--------------------+------------+-------------------+-------+----------------+---------------------------+
```

八、安装Dashboard

1、安装dashboard相关软件包

# yum install openstack-dashboard -y

2、修改配置文件/etc/openstack-dashboard/local_settings

# vim /etc/openstack-dashboard/local_settings

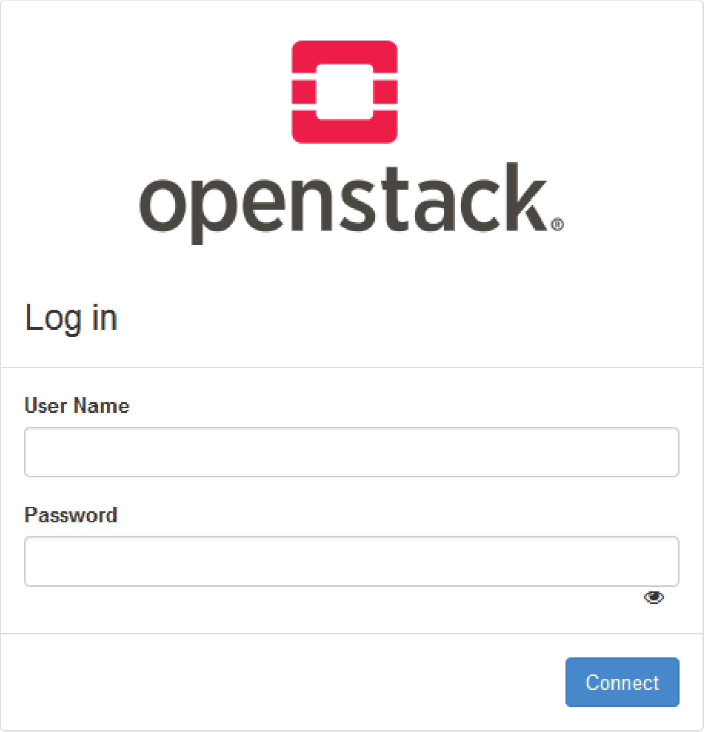
直接覆盖我给的local_settings文件也行（为了减少出错，大家还是用我提供的local_settings文件替换覆盖）

3、启动dashboard服务并设置开机启动

# systemctl restart httpd.service memcached.service

# systemctl status httpd.service memcached.service

到此，Controller节点搭建完毕，打开firefox浏览器即可访问**http://9.110.187.150/dashboard/** 可进入openstack界面！



九、安装配置cinder

1、创建数据库用户并赋予权限

```
CREATE DATABASE cinder;
GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'localhost' IDENTIFIED BY 'devops';
GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'%' IDENTIFIED BY 'devops';
```

2、创建cinder用户并赋予admin权限

```
# source /root/admin-openrc
# openstack user create  --domain default   cinder --password devops
# openstack role add --project service --user cinder admin
```

3、创建volume服务

```
# openstack service create --name cinder --description "OpenStack Block Storage" volume
# openstack service create --name cinderv2 --description "OpenStack Block Storage" volumev2
```

4、创建endpoint

```
# openstack endpoint create --region RegionOne  volume public http://controller:8776/v1/%\(tenant_id\)s
# openstack endpoint create --region RegionOne  volume internal http://controller:8776/v1/%\(tenant_id\)s
# openstack endpoint create --region RegionOne  volume admin http://controller:8776/v1/%\(tenant_id\)s
# openstack endpoint create --region RegionOne  volumev2 public http://controller:8776/v2/%\(tenant_id\)s
# openstack endpoint create --region RegionOne  volumev2 internal http://controller:8776/v2/%\(tenant_id\)s
# openstack endpoint create --region RegionOne  volumev2 admin http://controller:8776/v2/%\(tenant_id\)s
```

5、安装cinder相关服务

```
# yum install openstack-cinder -y
```

6、配置cinder配置文件

```
# cp /etc/cinder/cinder.conf /etc/cinder/cinder.conf.bak
# >/etc/cinder/cinder.conf
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT my_ip  10.1.1.150
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT auth_strategy  keystone
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT transport_url rabbit://openstack:devops@controller
# openstack-config --set /etc/cinder/cinder.conf  database connection  mysql+pymysql://cinder:devops@controller/cinder
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken auth_uri  http://controller:5000
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken auth_url  http://controller:35357
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken memcached_servers controller:11211
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken auth_type  password
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken project_domain_name   default
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken user_domain_name   default
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken project_name  service
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken username  cinder
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken password  devops
# openstack-config --set /etc/cinder/cinder.conf  oslo_concurrency lock_path  /var/lib/cinder/tmp
```

7、上同步数据库

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

8、在controller上启动cinder服务，并设置开机启动

```
# systemctl enable openstack-cinder-api.service openstack-cinder-scheduler.service
# systemctl restart openstack-cinder-api.service openstack-cinder-scheduler.service
# systemctl status openstack-cinder-api.service openstack-cinder-scheduler.service
```

9、安装Cinder节点，Cinder节点这里我们需要额外的添加一个硬盘（/dev/sdb)用作cinder的存储服务 (注意！这一步是在cinder节点操作的）

```
# yum install lvm2 -y
```

10、启动服务并设置为开机自启 (注意！这一步是在cinder节点操作的）

```
# systemctl enable lvm2-lvmetad.service
# systemctl start lvm2-lvmetad.service
# systemctl status lvm2-lvmetad.service
```

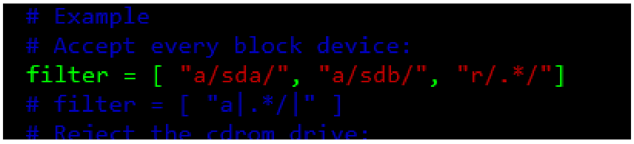
11、创建lvm, 这里的/dev/sdb就是额外添加的硬盘 (注意！这一步是在cinder节点操作的）

```
# fdisk -l
# pvcreate /dev/sdb
# vgcreate cinder-volumes /dev/sdb
```

12. 编辑存储节点lvm.conf文件 (注意！这一步是在cinder节点操作的）

```
# vim /etc/lvm/lvm.conf
```

在devices 下面添加 filter = ["a/sda/", "a/sdb/", "r/.*/"] ，130行 ，如图：



```
  # Example
  # Accept every block device:
  filter = [ "a/sda/", "a/sdb/", "r/.*/"]
  # filter = [ "a|.*/|" ]
  # Reject the cdrom drive:
```

然后重启下lvm2服务：

```
# systemctl restart lvm2-lvmetad.service
# systemctl status lvm2-lvmetad.service
```

13、安装openstack-cinder、targetcli (注意！这一步是在cinder节点操作的）

```
# yum install openstack-cinder openstack-utils targetcli python-keystone ntpdate -y
```

14、配置cinder配置文件 (注意！这一步是在cinder节点操作的）

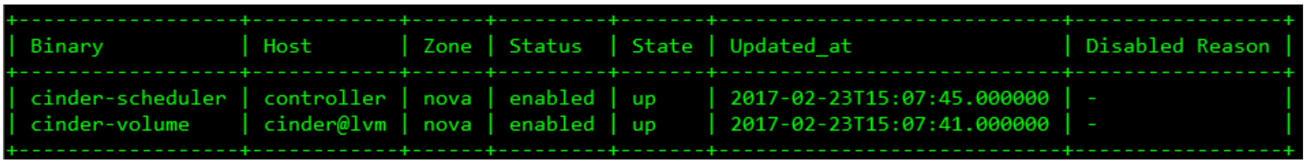
```
# cp /etc/cinder/cinder.conf /etc/cinder/cinder.conf.bak
# >/etc/cinder/cinder.conf
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT debug False
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT verbose True
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT auth_strategy keystone
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT my_ip 10.1.1.152
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT enabled_backends  lvm
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT glance_api_servers  http://controller:9292
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT glance_api_version 2
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT enable_v1_api True
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT enable_v2_api True
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT enable_v3_api True
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT storage_availability_zone nova
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT default_availability_zone nova
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT os_region_name RegionOne
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT api_paste_config /etc/cinder/api-paste.ini
# openstack-config --set /etc/cinder/cinder.conf  DEFAULT transport_url rabbit://openstack:devops@controller
# openstack-config --set /etc/cinder/cinder.conf  database connection  mysql+pymysql://cinder:devops@controller/cinder
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken auth_uri http://controller:5000
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken auth_url http://controller:35357
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken memcached_servers controller:11211
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken auth_type password
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken project_domain_name default
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken user_domain_name default
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken project_name service
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken username cinder
# openstack-config --set /etc/cinder/cinder.conf  keystone_authtoken password devops
# openstack-config --set /etc/cinder/cinder.conf  lvm volume_driver cinder.volume.drivers.lvm.LVMVolumeDriver
# openstack-config --set /etc/cinder/cinder.conf  lvm volume_group cinder-volumes
# openstack-config --set /etc/cinder/cinder.conf  lvm iscsi_protocol  iscsi
# openstack-config --set /etc/cinder/cinder.conf  lvm iscsi_helper lioadm
# openstack-config --set /etc/cinder/cinder.conf  oslo_concurrency lock_path /var/lib/cinder/tmp
```

15、启动openstack-cinder-volume和target并设置开机启动 （注意！这一步是在cinder节点操作的）

```
# systemctl enable openstack-cinder-volume.service target.service
# systemctl restart openstack-cinder-volume.service target.service
# systemctl status openstack-cinder-volume.service target.service
```

16、验证cinder服务是否正常

```
# source /root/admin-openrc
# cinder service-list
```



```
+------------------+-------------+------+---------+-------+----------------------------+-----------------+
| Binary           | Host        | Zone | Status  | State | Updated_at                 | Disabled Reason |
+------------------+-------------+------+---------+-------+----------------------------+-----------------+
| cinder-scheduler | controller  | nova | enabled | up    | 2017-02-23T15:07:45.000000 | -               |
| cinder-volume    | cinder@lvm  | nova | enabled | up    | 2017-02-23T15:07:41.000000 | -               |
+------------------+-------------+------+---------+-------+----------------------------+-----------------+
```

Compute节点部署

一、安装相关依赖包

```
# yum install openstack-selinux python-openstackclient yum-plugin-priorities openstack-nova-compute openstack-utils ntpdate -y
```

1. 配置nova.conf

```
# cp /etc/nova/nova.conf /etc/nova/nova.conf.bak
# >/etc/nova/nova.conf
# openstack-config --set /etc/nova/nova.conf DEFAULT auth_strategy  keystone
# openstack-config --set /etc/nova/nova.conf DEFAULT my_ip  10.1.1.151
# openstack-config --set /etc/nova/nova.conf DEFAULT use_neutron  True
# openstack-config --set /etc/nova/nova.conf DEFAULT firewall_driver  nova.virt.firewall.NoopFirewallDriver
# openstack-config --set /etc/nova/nova.conf DEFAULT transport_url rabbit://openstack:devops@controller
# openstack-config --set /etc/nova/nova.conf keystone_authtoken  auth_uri  http://controller:5000
# openstack-config --set /etc/nova/nova.conf keystone_authtoken  auth_url  http://controller:35357
# openstack-config --set /etc/nova/nova.conf keystone_authtoken  memcached_servers  controller:11211
# openstack-config --set /etc/nova/nova.conf keystone_authtoken  auth_type  password
# openstack-config --set /etc/nova/nova.conf keystone_authtoken  project_domain_name  default
# openstack-config --set /etc/nova/nova.conf keystone_authtoken  user_domain_name  default
# openstack-config --set /etc/nova/nova.conf keystone_authtoken  project_name  service
# openstack-config --set /etc/nova/nova.conf keystone_authtoken  username  nova
# openstack-config --set /etc/nova/nova.conf keystone_authtoken  password devops
# openstack-config --set /etc/nova/nova.conf placement auth_uri http://controller:5000
# openstack-config --set /etc/nova/nova.conf placement auth_url http://controller:35357
# openstack-config --set /etc/nova/nova.conf placement memcached_servers controller:11211
# openstack-config --set /etc/nova/nova.conf placement auth_type password
# openstack-config --set /etc/nova/nova.conf placement project_domain_name default
# openstack-config --set /etc/nova/nova.conf placement user_domain_name default
# openstack-config --set /etc/nova/nova.conf placement project_name service
# openstack-config --set /etc/nova/nova.conf placement username nova
# openstack-config --set /etc/nova/nova.conf placement password devops
# openstack-config --set /etc/nova/nova.conf placement os_region_name RegionOne
# openstack-config --set /etc/nova/nova.conf vnc enabled True
# openstack-config --set /etc/nova/nova.conf vnc keymap en-us
# openstack-config --set /etc/nova/nova.conf vnc vncserver_listen  0.0.0.0
# openstack-config --set /etc/nova/nova.conf vnc vncserver_proxyclient_address  10.1.1.151
# openstack-config --set /etc/nova/nova.conf vnc novncproxy_base_url  http://9.115.75.150:6080/vnc_auto.html
# openstack-config --set /etc/nova/nova.conf glance  api_servers  http://controller:9292
# openstack-config --set /etc/nova/nova.conf oslo_concurrency  lock_path  /var/lib/nova/tmp
# openstack-config --set /etc/nova/nova.conf libvirt virt_type  qemu
```

2. 设置libvirtd.service 和openstack-nova-compute.service开机启动

```
# systemctl enable libvirtd.service openstack-nova-compute.service
# systemctl restart libvirtd.service openstack-nova-compute.service
# systemctl status libvirtd.service openstack-nova-compute.service
```

3. 到controller上执行验证

```
# source /root/admin-openrc
# openstack compute service list
```

二、安装Neutron

1. 安装相关软件包

```
# yum install openstack-neutron-linuxbridge ebtables ipset -y
```

2. 配置neutron.conf

```
# cp /etc/neutron/neutron.conf /etc/neutron/neutron.conf.bak
# >/etc/neutron/neutron.conf
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT auth_strategy  keystone
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT advertise_mtu True
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT dhcp_agents_per_network 2
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT control_exchange neutron
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT nova_url http://controller:8774/v2
# openstack-config --set  /etc/neutron/neutron.conf DEFAULT transport_url rabbit://openstack:devops@controller
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken auth_uri  http://controller:5000
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken auth_url  http://controller:35357
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken memcached_servers  controller:11211
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken auth_type  password
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken project_domain_name  default
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken user_domain_name  default
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken project_name  service
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken username  neutron
# openstack-config --set  /etc/neutron/neutron.conf keystone_authtoken password  devops
# openstack-config --set  /etc/neutron/neutron.conf oslo_concurrency  lock_path  /var/lib/neutron/tmp
```

3. 配置/etc/neutron/plugins/ml2/linuxbridge_agent.ini

```
# openstack-config --set   /etc/neutron/plugins/ml2/linuxbridge_agent.ini  linux_bridge physical_interface_mappings provider:eno16777736
# openstack-config --set   /etc/neutron/plugins/ml2/linuxbridge_agent.ini  vxlan  enable_vxlan  True
# openstack-config --set   /etc/neutron/plugins/ml2/linuxbridge_agent.ini  vxlan  local_ip  10.2.2.151
# openstack-config --set   /etc/neutron/plugins/ml2/linuxbridge_agent.ini  vxlan l2_population  True
# openstack-config --set   /etc/neutron/plugins/ml2/linuxbridge_agent.ini securitygroup  enable_security_group  True
# openstack-config --set   /etc/neutron/plugins/ml2/linuxbridge_agent.ini securitygroup  firewall_driver neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
```

注意provider后面那个网卡名是第二块网卡的名称，我这里就是10.2.2.x网段网卡的名称

4. 配置nova.conf

```
# openstack-config --set  /etc/nova/nova.conf neutron url  http://controller:9696
# openstack-config --set  /etc/nova/nova.conf neutron auth_url  http://controller:35357
# openstack-config --set  /etc/nova/nova.conf neutron auth_type  password
# openstack-config --set  /etc/nova/nova.conf neutron project_domain_name  default
# openstack-config --set  /etc/nova/nova.conf neutron user_domain_name  default
# openstack-config --set  /etc/nova/nova.conf neutron region_name  RegionOne
# openstack-config --set  /etc/nova/nova.conf neutron project_name  service
# openstack-config --set  /etc/nova/nova.conf neutron username  neutron
# openstack-config --set  /etc/nova/nova.conf neutron password  devops
```

5. 重启和enable相关服务

```
# systemctl restart libvirtd.service openstack-nova-compute.service
# systemctl enable neutron-linuxbridge-agent.service
# systemctl restart neutron-linuxbridge-agent.service
# systemctl status libvirtd.service openstack-nova-compute.service neutron-linuxbridge-agent.service
```

三、计算节点结合Cinder

1.计算节点要是想用cinder,那么需要配置nova配置文件 (注意！这一步是在计算节点操作的)

```
# openstack-config --set /etc/nova/nova.conf  cinder os_region_name  RegionOne
# systemctl restart openstack-nova-compute.service
```

2.然后在controller上重启nova服务

```
# systemctl restart openstack-nova-api.service
# systemctl status openstack-nova-api.service
```

四. 在controler上执行验证

```
# source /root/admin-openrc
# neutron agent-list
# nova-manage cell_v2 discover_hosts
```

到此，Compute节点搭建完毕，运行nova host-list可以查看新加入的compute1节点
如果需要再添加另外一个compute节点，只要重复下第二大部即可，记得把计算机名和IP地址改下。

附-创建配额命令

```
# openstack flavor create m1.tiny --id 1 --ram 512 --disk 1 --vcpus 1
# openstack flavor create m1.small --id 2 --ram 2048 --disk 20 --vcpus 1
# openstack flavor create m1.medium --id 3 --ram 4096 --disk 40 --vcpus 2
# openstack flavor create m1.large --id 4 --ram 8192 --disk 80 --vcpus 4
# openstack flavor create m1.xlarge --id 5 --ram 16384 --disk 160 --vcpus 8
# openstack flavor list
```

另外，我们还有OpenStack高级培训和视频，欢迎咨询购买：

培训说明：本培训是全国首家Openstack高级运维实战培训，**大型云计算公司openstack专家亲自授课！**干货连连，价值连城！同学们看下面的培训内容就能发现，本培训的与众不同。 并不是教大家简简单单的openstack搭建！**而是深入浅出，彻彻底底的教会大家openstack在生产环境下的运维和实践！**培训时间为期**6**天（三个周六日）。

先基础，后高端！深入浅出，让你走在云计算时代技术的前列！

我们的课程全程干货，懂课程的人只要稍微看下我们的课程内容就会发现我们课程含金量有多高！

目前网上openstack资料和文章特别少，很多都是停留介绍基本的搭建；一般稍微遇到bug、问题，你很难从网上资料中得到解决...

终归原因是OpenStack涉及的知识面非常多，Linux、网络、python等等... 而且没有一定的基础绝对是玩不转的...

当然，openstack难，也是它的价值所在！

如果openstack很简单，就像vmware vsphere一样（封装太好），很快入手学会，那么，就没有值得研究和学习的价值了...

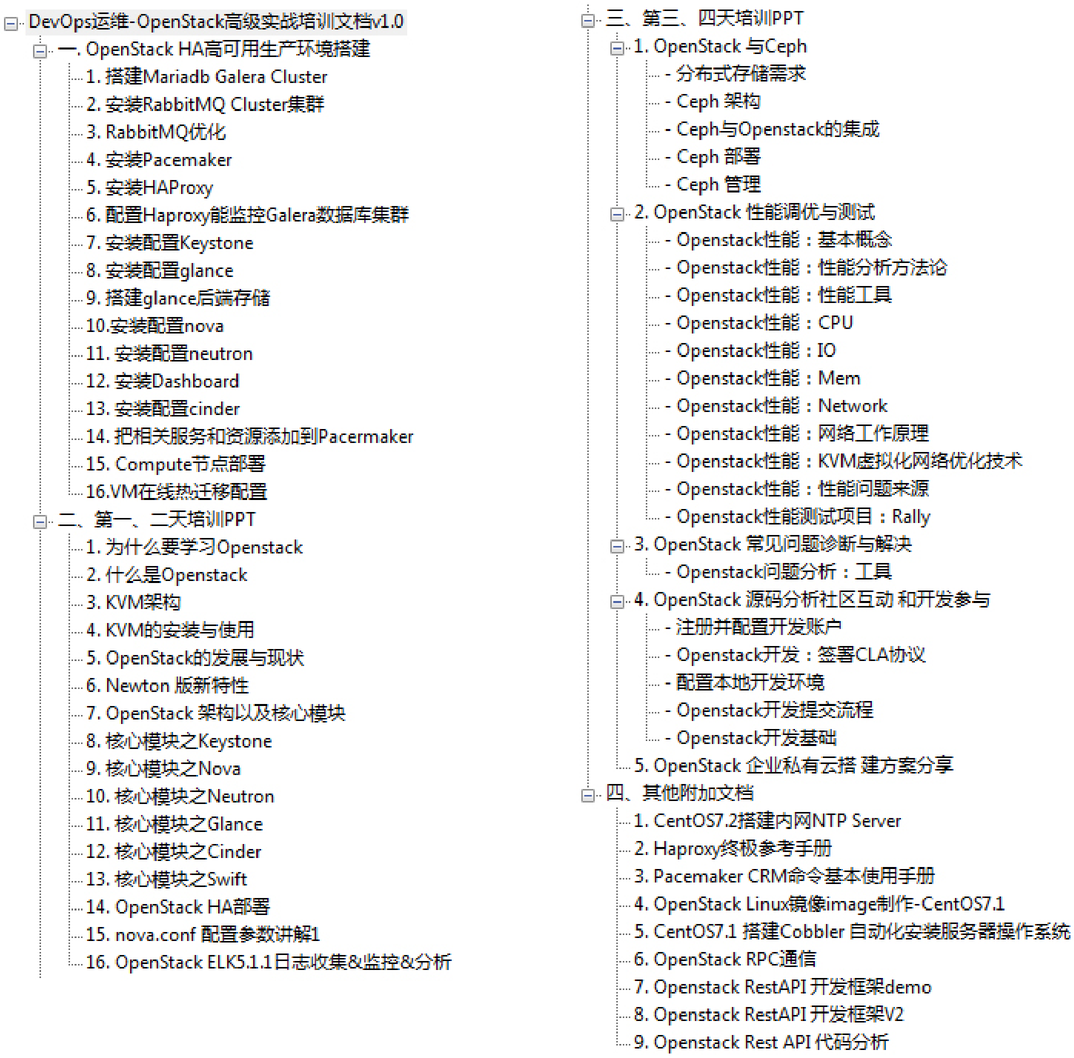
所以，趁现在真正了解openstack的人还不多，抓紧时间学习，赶在前面，尽快拉开距离，拿到高工资，这才是王道！

目前还没有哪家培训机构的OpenStack课程像我们这样有深度！

认真学完后，绝对是中高级**OpenStack**云计算工程师！北上广月薪**20K**以上不是梦！

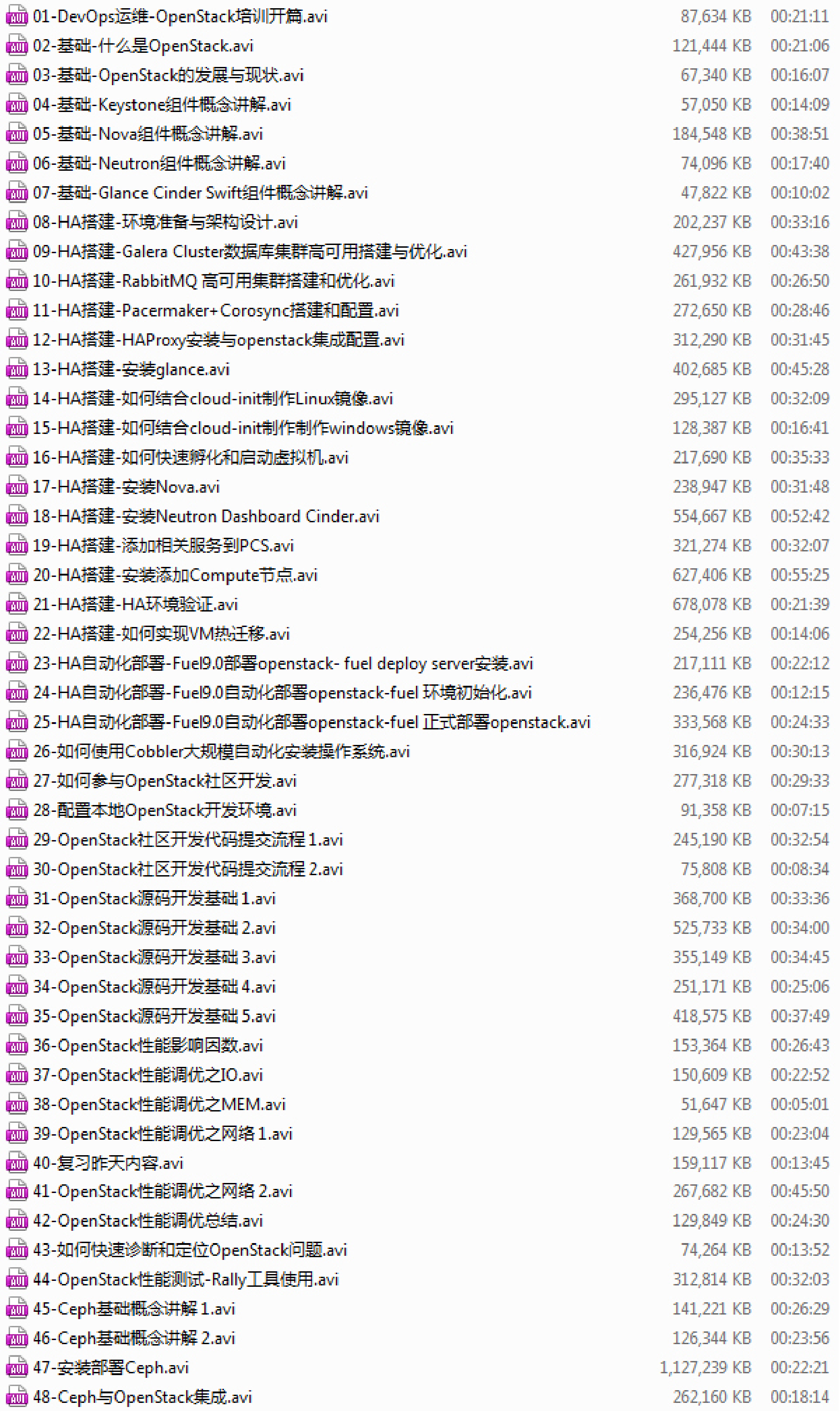
一次没听懂，可重听三次，在线VIP群答疑，让你学会为止！

OpenStack培训细节 业内最全



- DevOps运维-OpenStack高级实战培训文档v1.0
  - 一. OpenStack HA高可用生产环境搭建
    - 1. 搭建Mariadb Galera Cluster
    - 2. 安装RabbitMQ Cluster集群
    - 3. RabbitMQ优化
    - 4. 安装Pacemaker
    - 5. 安装HAProxy
    - 6. 配置Haproxy能监控Galera数据库集群
    - 7. 安装配置Keystone
    - 8. 安装配置glance
    - 9. 搭建glance后端存储
    - 10.安装配置nova
    - 11. 安装配置neutron
    - 12. 安装Dashboard
    - 13. 安装配置cinder
    - 14. 把相关服务和资源添加到Pacermaker
    - 15. Compute节点部署
    - 16.VM在线热迁移配置
  - 二、第一、二天培训PPT
    - 1. 为什么要学习Openstack
    - 2. 什么是Openstack
    - 3. KVM架构
    - 4. KVM的安装与使用
    - 5. OpenStack的发展与现状
    - 6. Newton 版新特性
    - 7. OpenStack 架构以及核心模块
    - 8. 核心模块之Keystone
    - 9. 核心模块之Nova
    - 10. 核心模块之Neutron
    - 11. 核心模块之Glance
    - 12. 核心模块之Cinder
    - 13. 核心模块之Swift
    - 14. OpenStack HA部署
    - 15. nova.conf 配置参数讲解1
    - 16. OpenStack ELK5.1.1日志收集&监控&分析
  - 三、第三、四天培训PPT
    - 1. OpenStack 与Ceph
      - - 分布式存储需求
      - - Ceph 架构
      - - Ceph与Openstack的集成
      - - Ceph 部署
      - - Ceph 管理
    - 2. OpenStack 性能调优与测试
      - - Openstack性能：基本概念
      - - Openstack性能：性能分析方法论
      - - Openstack性能：性能工具
      - - Openstack性能：CPU
      - - Openstack性能：IO
      - - Openstack性能：Mem
      - - Openstack性能：Network
      - - Openstack性能：网络工作原理
      - - Openstack性能：KVM虚拟化网络优化技术
      - - Openstack性能：性能问题来源
      - - Openstack性能测试项目：Rally
    - 3. OpenStack 常见问题诊断与解决
      - - Openstack问题分析：工具
    - 4. OpenStack 源码分析社区互动 和开发参与
      - - 注册并配置开发账户
      - - Openstack开发：签署CLA协议
      - - 配置本地开发环境
      - - Openstack开发提交流程
      - - Openstack开发基础
    - 5. OpenStack 企业私有云搭 建方案分享
  - 四、其他附加文档
    - 1. CentOS7.2搭建内网NTP Server
    - 2. Haproxy终极参考手册
    - 3. Pacemaker CRM命令基本使用手册
    - 4. OpenStack Linux镜像image制作-CentOS7.1
    - 5. CentOS7.1 搭建Cobbler 自动化安装服务器操作系统
    - 6. OpenStack RPC通信
    - 7. Openstack RestAPI 开发框架demo
    - 8. Openstack RestAPI 开发框架V2
    - 9. Openstack Rest API 代码分析

OpenStack高级学习路线图



01-DevOps运维-OpenStack培训开篇.avi	87,634 KB	00:21:11
02-基础-什么是OpenStack.avi	121,444 KB	00:21:06
03-基础-OpenStack的发展与现状.avi	67,340 KB	00:16:07
04-基础-Keystone组件概念讲解.avi	57,050 KB	00:14:09
05-基础-Nova组件概念讲解.avi	184,548 KB	00:38:51
06-基础-Neutron组件概念讲解.avi	74,096 KB	00:17:40
07-基础-Glance Cinder Swift组件概念讲解.avi	47,822 KB	00:10:02
08-HA搭建-环境准备与架构设计.avi	202,237 KB	00:33:16
09-HA搭建-Galera Cluster数据库集群高可用搭建与优化.avi	427,956 KB	00:43:38
10-HA搭建-RabbitMQ 高可用集群搭建和优化.avi	261,932 KB	00:26:50
11-HA搭建-Pacermaker+Corosync搭建和配置.avi	272,650 KB	00:28:46
12-HA搭建-HAProxy安装与openstack集成配置.avi	312,290 KB	00:31:45
13-HA搭建-安装glance.avi	402,685 KB	00:45:28
14-HA搭建-如何结合cloud-init制作Linux镜像.avi	295,127 KB	00:32:09
15-HA搭建-如何结合cloud-init制作制作windows镜像.avi	128,387 KB	00:16:41
16-HA搭建-如何快速孵化和启动虚拟机.avi	217,690 KB	00:35:33
17-HA搭建-安装Nova.avi	238,947 KB	00:31:48
18-HA搭建-安装Neutron Dashboard Cinder.avi	554,667 KB	00:52:42
19-HA搭建-添加相关服务到PCS.avi	321,274 KB	00:32:07
20-HA搭建-安装添加Compute节点.avi	627,406 KB	00:55:25
21-HA搭建-HA环境验证.avi	678,078 KB	00:21:39
22-HA搭建-如何实现VM热迁移.avi	254,256 KB	00:14:06
23-HA自动化部署-Fuel9.0部署openstack- fuel deploy server安装.avi	217,111 KB	00:22:12
24-HA自动化部署-Fuel9.0自动化部署openstack-fuel 环境初始化.avi	236,476 KB	00:12:15
25-HA自动化部署-Fuel9.0自动化部署openstack-fuel 正式部署openstack.avi	333,568 KB	00:24:33
26-如何使用Cobbler大规模自动化安装操作系统.avi	316,924 KB	00:30:13
27-如何参与OpenStack社区开发.avi	277,318 KB	00:29:33
28-配置本地OpenStack开发环境.avi	91,358 KB	00:07:15
29-OpenStack社区开发代码提交流程 1.avi	245,190 KB	00:32:54
30-OpenStack社区开发代码提交流程 2.avi	75,808 KB	00:08:34
31-OpenStack源码开发基础 1.avi	368,700 KB	00:33:36
32-OpenStack源码开发基础 2.avi	525,733 KB	00:34:00
33-OpenStack源码开发基础 3.avi	355,149 KB	00:34:45
34-OpenStack源码开发基础 4.avi	251,171 KB	00:25:06
35-OpenStack源码开发基础 5.avi	418,575 KB	00:37:49
36-OpenStack性能影响因数.avi	153,364 KB	00:26:43
37-OpenStack性能调优之IO.avi	150,609 KB	00:22:52
38-OpenStack性能调优之MEM.avi	51,647 KB	00:05:01
39-OpenStack性能调优之网络 1.avi	129,565 KB	00:23:04
40-复习昨天内容.avi	159,117 KB	00:13:45
41-OpenStack性能调优之网络 2.avi	267,682 KB	00:45:50
42-OpenStack性能调优总结.avi	129,849 KB	00:24:30
43-如何快速诊断和定位OpenStack问题.avi	74,264 KB	00:13:52
44-OpenStack性能测试-Rally工具使用.avi	312,814 KB	00:32:03
45-Ceph基础概念讲解 1.avi	141,221 KB	00:26:29
46-Ceph基础概念讲解 2.avi	126,344 KB	00:23:56
47-安装部署Ceph.avi	1,127,239 KB	00:22:21
48-Ceph与OpenStack集成.avi	262,160 KB	00:18:14