

第3章 分布式OpenStack Ocata VXLAN 模式云计算系统 运维与管理

Ocata 版是 OpenStack 社区于 2017 年 2 月 22 日正式发布的第 15 个版本。Ocata 版在单个 网络上集成裸机,在虚拟机和容器方面进行了实质性的创新;其侧重于稳定性,包括核心计 算和网络服务的可扩展性和性能;可以在网络层为基于容器的应用程序框架提供更大的支持,在 OpenStack 容器化部署方面也更加简便。

在这个发布周期中,OpenStack 社区见证了越来越多的多云趋势,企业在公有云和私有云 上采用了更加复杂的工作负载分配策略。特别是 OpenStack 用户看到了 OpenStack 私有云成本的 显著节省和合规性优势后,随着远程管理的私有云新模型的出现,用户也更容易在专用的环 境中体验公有云的好处。

本章主要介绍的是分布式 Ocata 版 OpenStack (Controller + 1 Compute + 1 Cinder)的搭建 过程。搭建的时候,请读者严格按照本章中的讲解内容进行配置,在不熟悉的情况下,严禁 自行添加额外的配置和设置。



为了方便学习,本例将 OpenStack 部署在三台虚拟机中,如果三台虚拟机被部署在一台 物理机中,物理机建议配置 16GB 的内存和 100GB 以上的固态硬盘,故在正式部署 OpenStack 之前,最需要厘清的是 OpenStack 分布式部署的环境。

3.1.1 物理网络拓扑规划

安装 VMware Workstation 12.5.0 虚拟机软件,在 VMware Workstation 中虚拟出三台虚拟 机,三台虚拟机的基本配置规划如表 3.1 所示。

			西舟	网络规划			
卫总有你	CFU rlly 心	1/317		名称	用途	IP 地址	
				eth0	external	1.1.1.128/24	
Controller	4 核心 4 核心	 核心 4GB 核心 4GB 核心 4GB 	200GB 200GB 200GB+100GB	eth1	admin mgt	10.1.1.128/24	
				eth2	tunnel	10.2.2.128/24	
Commute				eth1	admin mgt	10.1.1.129/24	
Compute				eth2	tunnel	10.2.2.129/24	
Cindar	4 核心、			eth1	admin mgt	10.1.1.130/24	
Cilder	4 核心			eth2	tunnel	10.2.2.130/24	

表 3.1 虚拟机配置规划表

三个网络的规划说明如下。

- external:这个网络是连接外网的,也就是说 OpenStack 环境里的虚拟机要让用户访问,则必须有个网段是连接外网的,用户通过这个网络能访问虚拟机。如果搭建的是公有云,那么这个 IP 段一般是公网的。
- admin mgt: 这个网段是用来管理网络的。OpenStack 环境里面各模块之间需要交互, 像连接数据库、连接 Message Queue 等都需要一个网络去支撑,该网段就是起这个作用的,简而言之,就是 OpenStack 自身用的 IP 段。
- tunnel: 隧道网络。OpenStack 里面使用 GRE 或 VXLAN 模式,需要有隧道网络; 隧道 网络采用点到点通信协议,从而代替了交换连接,在 OpenStack 里,这个 tunnel 就是 虚拟机走网络数据流量用的。

当然,这三个网络也可以放在一起,但是只能用于测试学习环境,在真正的生产环境中, 三者是要分开的。所以在创建完虚拟机后,请给虚拟机再添加两块网卡,根据生产环境的要 求搭建学习。

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,所以还要给存储 再添加一个网络——storage 段。网络分开的好处就是数据分流、安全、不相互干扰。

3.1.2 虚拟机网卡设置

在 VMware Workstation 中,单击"编辑"→"虚拟网络编辑器"命令。对网络的连接采 用如图 3.1 所示的虚拟网络编辑器设置。

呂称	类型	外部连接	主机连接	DHCP	子网地址	
Mnet1	NAT 模式	NAT 模式	已连接	已启用	1.1.1.0	
/Mnet2	仅主机	-	已连接	已启用	10.1.1.0	
Mnet3	仅主机	-	已连接	已启用	10.2.2.0	

图 3.1 虚拟网络编辑器设置

在 Controller 节点虚拟机的设置中删除默认的网卡, 重新添加三张网卡, 三张网卡分别自

定义连接到 VMnet1、VMnet2、VMnet3 上,虚拟机网卡设置如图 3.2 所示。

E	副初初设置	C-P Dwild
	硬件 选项	
	设备 ■ 处理器 ● 硬盘(SCSI) ● CD/DVD (IDE) ■ 一 一 一 一 一 使盘(SCSI) ● CD/DVD (IDE) ■ 一 一 一 一 一 一 使盘(SCSI) ● ○ CD/DVD (IDE) ■ 一 四 名 适面器 2 ■ 四 名 适面器 3 ■ ● USB 控制器 ● ● ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	摘要 4 GB 4 200 GB 正在使用文件 D:\BaiduYunDownloa 自定义(VMnet1) 自定义(VMnet2) 自定义(VMnet3) 存在 自动检测 存在 自动检测 存在
		□ =0 0£3(1

图 3.2 虚拟机网卡设置

在 Compute 节点虚拟机的设置中删除默认的网卡,重新添加两张网卡,两张网卡分别自 定义连接到 VMnet2 和 VMnet3 上。

在 Cinder 节点虚拟机的设置中删除默认的网卡,重新添加两张网卡,两张网卡分别自定 义连接到 VMnet2 和 VMnet3 上。

3.1.3 虚拟机系统与基本配置

第1步:所有节点安装 CentOS 7.2 系统(最小化安装,不要用【yum update】命令升级到 7.3 版本! Ocata 版在 7.3 版本下依然有虚拟机启动出现 iPXE 启动的问题)。

第2步:关闭防火墙和 SELinux。

【systemctl stop firewalld】停止防火墙服务。

【systemctl disable firewalld】禁用防火墙服务。

【vi /etc/sysconfig/selinux】设置 SELinux 的如下内容:

SELINUX=disable

第3步:安装相关工具。

因为安装系统时采用的是最小化安装,所以一些最基本的命令工具均未安装,如 ifconfig、 vim 等命令,所以需要运行下面的命令安装上述工具:

[yum install net-tools wget vim ntpdate bash-completion -y]

第4步:更改 hostname 主机名。

在 Controller 节点运行如下命令修改主机名:

[hostnamectl set-hostname controller]

在 Compute 节点运行如下命令修改 Compute 节点主机名:

[hostnamectl set-hostname compute]

在 Cinder 节点运行如下命令修改 Cinder 节点主机名:

[hostnamectl set-hostname cinder]

第5步:修改 hosts 文件。

在每个节点运行【vim /etc/hosts】命令,为 hosts 配置文件增加如下代码,作为 IP 地址与 主机名的映射。

10.1.1.128	controller
10.1.1.129	compute1
10.1.1.130	cinder

hosts 文件配置效果如图 3.3 所示。



3.2.1 使用 NTP 服务同步系统时间

OpenStack 是分布式架构,每个节点都不能有时间差,但刚安装完 CentOS 系统,时间会 跟当前的北京时间不一致,所以必须使用 NTP 服务同步时间,命令如下:

【yum install ntp】安装时间服务。

【date】查询当前时间。

【ntpdate cn.pool.ntp.org】同步本机到当前北京时间。

在 3.1.1 节的网络拓扑规划中, Controller 节点是可以连接外网的,运行上述命令即可同步时间,但是,规划的 Compute 节点和 Cinder 节点是不可以连接外网的,因此,需要在 Controller 节点上配置和运行 NTP Server。

使用【vim /etc/ntp.conf】命令编辑 NTP Server 配置文件,修改文件中的第 21~24 行,即将 第 21 行修改为 "server ntpdate.pool.ntp.org iburst",注释掉第 22~24 行的内容,NTP 服务配置 文件效果如图 3.4 所示。



在 Controller 节点的 NTP Server 时间服务器搭建成功后, Compute 和 Cinder 节点即可直 接使用【ntpdate controller】命令同步时间。

另外,建议把如下命令添加到/etc/rc.d/rc.local中,使它们开机启动:

[echo "ntpdate cn.pool.ntp.org" >> /etc/rc.d/rc.local]

 $\left(\text{chmod} + \frac{x}{\text{chmod}} \right)$

搭建 OpenStack 内部使用 yum 源 3.2.2

yum 是"Yellow dog Updater, Modified"的缩写, 是一个软件包管理器, 它会从指定的 位置(相关网站的 RPM 包地址或本地的 RPM 路径)自动下载 RPM 包并且安装,能够很好 地解决依赖关系问题。

Linux 安装某个软件时往往需要安装很多其他特有的依赖软件, yum 就是为了解决依赖关 系而存在的。yum 源相当于是一个目录项,当我们使用 yum 机制安装软件时,若需要安装依 赖软件,系统就会根据在 yum 源中定义好的路径查找依赖软件,并将依赖软件安装好。

vum 的基本工作机制包括服务器端和客户机端,分别介绍如下。

- 服务器端: 在服务器中存放了所有的 RPM 软件包, 然后以相关的功能分析每个 RPM 文件的依赖性关系,将这些数据记录成文件并存放在服务器的某特定目录内。
- 客户机端: 如果需要安装某个软件时, 先下载服务器中记录的依赖性关系文件(可通 过 WWW 或 FTP 方式),通过对服务器端下载的记录数据进行分析,然后取得所有相 关的软件,一次性全部下载后进行安装。

自己搭建 yum 源相对来说比较安全,安装软件时从本地下载,速度快;另外,网络 yum 更新很快,但是生产中没有必要实时更新系统,这样搭建的 yum 还起到备份的作用,方便以 后重用。所以,搭建内部 yum 源非常重要。

OpenStack 搭建用到的源有 CentOS7 源、Epel7 源、MariaDB10.1 源、OpenStack Ocata 源。 搭建 yum 源的方式有很多种,可以通过 httpd、nginx、apache、Windows 本地 xampp、FTP 等搭建。

在本节中,我们通过 MyWebServer 在自己的物理机上搭建一个 Web 服务器,然后把相应 的 yum 源放到 Web 服务器的根目录下即可。

运行如下命令配置 yum 源的客户端文件。

【mkdir /etc/yum.repos.d/bak】建立备份文件夹。

【mv /etc/yum.repos.d/*.* /etc/yum.repos.d/bak】移动原配置文件到备份文件。

【vim /etc/yum.repos.d/centos epel_openstack_mariadb.repo】新建 yum 源客户端文件,具体 内容如下:

```
[centos]
name=centos7.2
baseurl= http://10.1.1.1/centos7.2/7.2/os/x86 64/
                                   版水子
enabled=1
gpgcheck=0
```

[epel]

```
name=epel
baseurl= http://10.1.1.1/epel/7/x86_64/
enabled=1
gpgcheck=0
[openstack]
name=ocata
baseurl=http://10.1.1.1/openstack-ocata/
enabled=1
gpgcheck=0
[mariadb]
name=mariadb10.1
baseurl=http://10.1.1.1/mariadb10.1/
enabled=1
gpgcheck=0
```

建立好 yum 源的配置文件后使用【yum clean all】命令清除原来的 yum 数据库,使用【yum makecache】命令重新查找 yum 源。如果没有报错, yum 源服务器和客户端便搭建成功。

3.2.3 搭建 MariaDB 数据库服务

MariaDB 数据库管理系统是 MySQL 的一个分支,主要由开源社区维护,采用 GPL 授权 许可。MariaDB 的目的是完全兼容 MySQL,包括 API 和命令行,使之能轻松成为 MySQL 的 替代品。在存储引擎方面,使用 XtraDB 代替 MySQL 的 InnoDB。MariaDB 由 MySQL 的创始 人米凯尔•维德纽斯(Michael Widenius)主导开发,早前,他曾以 10 亿美元的价格,将自 己创建的公司 MySQL AB 卖给了 SUN 公司,此后,随着 SUN 公司被 Oracle 公司收购,MySQL 的所有权也转入 Oracle 公司。

MariaDB 这个名称来自米凯尔·维德纽斯女儿的名字 Maria。MariaDB 基于事务的 Maria 存储引擎,替代了 MySQL 的 MyISAM 存储引擎,它使用了 Percona 的 XtraDB (InnoDB 的变体,分支的开发者希望提供访问 MySQL 5.4 InnoDB 的性能)。这个版本还包括了 PrimeBase XT (PBXT)和 FederatedX 存储引擎。

MariaDB 用于存储 OpenStack 中的所有信息,具体搭建方法如下。

(1) 使用【yum install -y MariaDB-server MariaDB-client】命令安装 MariaDB。

(2) 配置 MariaDB。使用【vim /etc/my.cnf.d/mariadb-openstack.cnf】命令创建配置文件, 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 = 1.1.1.128

(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。运行【mysql_secure_installation】命令设置 MariaDB 数据库的 root 密码。

先按 Enter 键, 然后按 Y 键, 设置 MySQL 密码, 然后一直按 Y 键结束, 这里我们设置 的密码是 yhy。需要注意, 输入密码时, 屏幕是没有任何显示的, 但主机系统已经接收到了 输入的密码, 如图 3.5 所示。



图 3.5 设置 MariaDB 数据的 root 密码

3.2.4 安装 RabbitMQ

第1步:安装 erlang。 [yum install -y erlang] 第2步: 安装 RabbitMO。 vum install -v rabbitmg-server 第3步: 启动 RabbitMO 及设置开机启动。 [systemctl enable rabbitmg-server.service] [systemctl restart rabbitmq-server.service] [systemctl status rabbitmq-server.service] [systemctl list-unit-files |grep rabbitmq-server.service] 第4步: 创建用户,将用户名设置为 openstack,并将密码设置为 yhy。 [rabbitmqctl add user openstack yhy] 所有组件通过用户 openstack 与 RabbitMO 打交道。 第5步:为用户 openstack 赋予权限。 【rabbitmgctl set permissions openstack ".*" ".*" 】 赋予相应的权限。 【rabbitmqctl set user tags openstack administrator】定义成 administrator 角色。 【rabbitmqctl list users】查看添加的用户。 第6步:查看监听端口。 [netstat -ntlp |grep 5672] RabbitMQ 使用的是 5672 端口。 第7步:查看 RabbitMQ 插件。 [/usr/lib/rabbitmg/bin/rabbitmg-plugins list] 第8步:打开 RabbitMO 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,通过【systemctl restart rabbitmq-server】命令重启 RabbitMQ 服务。 在浏览器地址栏中输入 http://10.1.1.128:15672,以默认用户名 guest 和密码 guest 登录。 通过界面,我们能很直观地看到 RabbitMQ 的运行和负载情况。 第9步:查看 RabbitMQ 状态。

使用浏览器登录 http://10.1.1.128:15672, 然后以用户名 openstack 和密码 yhy 登录也可以 查看 RabbitMQ 的状态信息, 如图 3.6 所示。

山水东

王	
J	
C) D	
ñt.	
0	
ŝ	
旳	
Ā	
计	
算	
系	
统	
运	
従	
Ë	
う答	
^日	

▼ Node								
lode: rabbit@contro	ller (More about this n	ode)						
File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Rates mode	In	fo]-
58	0	231	53MB	194GB	basic	Disc 6	<u>Stats</u>	
1024 available	829 available	1048576 available	1.5GB high waterm	ar&8MB low watermark	¢			
Paths								
Config file	/etc/rabbitmg/rabbitm	q. config						
Database directory	/ /var/lib/rabbitmg/mnesia/rabbit@controller							
Log file	/var/log/rabbitmq/rab	bit@controller.log						
-								

图 3.6 查看 RabbitMQ 状态

3.2.5 安装配置 keystone

第1步: 创建 keystone 数据库。

【mysql-uroot-p】进入 MariaDB 数据库。

【CREATE DATABASE keystone;】创建数据库。

【show databases;】查看数据。

第2步: 创建数据库 keystone 用户及 root 用户并赋予权限。

【GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY 'yhy';】

GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'yhy';]

》注意.

yhy 为数据库密码。

第3步:安装 keystone 和 memcached。

[yum -y install openstack-keystone httpd mod_wsgi python-openstackclient memcached python-memcached openstack-utils]

第4步:启动 memcached 服务并设置开机自启动。

[systemctl enable memcached.service]

[systemctl restart memcached.service]

[systemctl status memcached.service]

第5步: 配置/etc/keystone/keystone.conf文件。

【cp /etc/keystone/keystone.conf /etc/keystone/keystone.conf.bak】备份原配置文件。

【>/etc/keystone/keystone.conf】清空配置文件。

使用如下命令配置 /etc/keystone/keystone.conf 文件:

[openstack-config --set /etc/keystone/keystone.conf DEFAULT transport_url rabbit://openstack: yhy@controller]

【openstack-config --set /etc/keystone/keystone.conf database connection mysql://keystone:yhy @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】 配置完的效果如图 3.7 所示。

```
[root@controller ~]# cat /etc/keystone/keystone.conf
[DEFAULT]
transport_url = rabbit://openstack:yhy@controller
[database]
connection = mysql://keystone:yhy@controller/keystone
[cache]
backend = oslo_cache.memcache_pool
enabled = true
memcache_servers = controller:11211
[memcache]
servers = controller:11211
[token]
expiration = 3600
provider = fernet
[root@controller ~]#
```

图 3.7 keystone 配置文件效果

第6步: 配置 httpd.conf 文件和 memcached 文件。

[sed -i "s/#ServerName www.example.com:80/ServerName controller/" /etc/httpd/conf/httpd. conf]

【sed -i 's/OPTIONS*.*/OPTIONS="-1127.0.0.1,::1,10.1.1.128"/' /etc/sysconfig/memcached】 第7步: 配置 keystone 与 httpd 结合。

[In -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/]

第8步:数据库同步。

[su -s /bin/sh -c "keystone-manage db_sync" keystone]

第9步:初始化 fernet。

[keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone]

[keystone-manage credential_setup --keystone-user keystone --keystone-group keystone]

初始化后会在/etc/keystone 下生成两个文件和一个文件夹,注意权限用户和用户组都是 keystone。

第10步: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】 第 11 步: 创建 admin 用户角色。 【keystone-manage bootstrap \ --bootstrap-password yhy \ --bootstrap-username admin \ --bootstrap-project-name admin \ --bootstrap-role-name admin \ --bootstrap-role-name keystone \ --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

验证命令:

【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 yhy】

返回结果如图 3.8 所示。

<pre>[root@controller keystone]# openstac</pre>	ck project listos-username adminos-project-name ad
minos-user-domain-id defaultos	s-project-domain-id defaultos-identity-api-version 3
os-auth-url http://controller:5000)os-password yhy
+	+
ID	Name
+ 625576540d8849089f9726d454c55d1a +	
[root@controller keystone]#	



第12步: 创建 admin 用户环境变量,创建/root/admin-openrc 文件并写入内容。 使用【vim /root/admin-openrc】命令创建 admin 用户环境变量,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=yhy export OS_IDENTITY_API_VERSION=3 export OS_IMAGE_API_VERSION=2 export OS_AUTH_URL=http://controller:35357/y3

第13步:创建 service 项目。

[source /root/admin-openrc]

【openstack project create --domain default --description "Service Project" service】 第 14 步: 创建 demo 项目。

Copenstack project create -- domain default -- description "Demo Project" demo

第 3 章 分布式 OpenStack Ocata VXLAN 模式云计算系统运维与管理

运行效果如图 3.9 所示。

<pre>++ Field Value ++ description Service Project domain_id default enabled True id ff04399fe86d456395b9b4dddf7f5cc5 is_domain False name service parent_id default </pre>	[root@controlle ice Project" se	er keystone]# openstack project ervice	create	domain	default	description	"Serv
description Service Project domain_id default enabled True id ff04399fe86d456395b9b4dddf7f5cc5 is_domain False name service parent_id default	+ Field +	Value	+ ++				
*	<pre> description domain_id enabled id is_domain name parent_id </pre>	Service Project default True ff04399fe86d456395b9b4dddf7f5c False service default	:c5 				

图 3.9 创建 demo 项目效果

第15步: 创建 demo 用户。

【openstack user create --domain default demo --password yhy】运行效果如图 3.10 所示。

root@controller keystone]# openstack user createdomain default demopassword yhy							
+ Field +	+	+ .+					
/ domain_id enabled id name options password_expires_at	, default True d5ae773cc44d4631b05a358db818d704 demo {} None						
[root@controller keyst	root@controller keystone]#						

图 3.10 创建 demo 用户效果

注意:

yhy为 demo 用户的密码。

第16步: 创建 user 角色并将 demo 用户赋予 user 角色。

[openstack role create user]

运行效果如图 3.11 所示。



图 3.11 创建 user 角色运行效果

[openstack project create --domain default --description "Demo Project" demo] [openstack role add --project demo --user demo user] [openstack project list]

运行效果如图 3.12 所示。

[root@controlle >descript:	er keystone]# openstad ion "Demo Project" dem	ck project no	createdo	omain defaul	t \	
 Field	 Value		+ +			
description domain_id enabled id is_domain name parent_id +	Demo Project default True 1dbd3fac81424f0e82bf False demo default	E3bbbd11cac	1 1 141 1 1 1			
[root@controlle [root@controlle	er keystone]# openstad er keystone]# openstad	ck role add ck project	1project list	demousei	: demo	user
+ ID +		 Name				
1dbd3fac81424 625576540d884 ff04399fe86d4	4f0e82bf3bbbd11cad41 49089f9726d454c55d1a 456395b9b4dddf7f5cc5	demo admin service				
[root@control]	er kevstonel#					

图 3.12 将 demo 用户赋予 user 角色效果

第17步:验证keystone。

[unset OS_TOKEN OS_URL]

通过 admin 用户验证命令如下:

【openstack --os-auth-url http://controller:35357/v3 --os-project-domain-name default --os-userdomain-name default --os-project-name admin --os-username admin token issue --ospassword yhy】 通过 demo 用户验证命令如下:

【openstack --os-auth-url http://controller:5000/v3 --os-project-domain-name default --os-userdomain-name default --os-project-name demo --os-username demo token issue --os-password yhy】 运行效果如图 3.13 所示。

[root@control]	ler keystone]# openstackos-auth-url http://controller:5000/v3os-projec
t-domain-name	defaultos-user-domain-name defaultos-project-name demoos-username
demo token iss	sueos-password yhy
Field	Value
expires	2018-01-30T23:11:39+0000
id	gAAAABacO2bhPwe2j0H_XuseIUxKWx_GNU3vrMDTSGNo2VIy549DzfHAxES1mM2JT6EV87U
	nP09Nbfva_01AUCCBOSX2o71rDYm2wZsX0E2CE9wwbssj2ILqVuZk8puf2RgxnDgAzzmuUUw
	KJxpQjljh2KtsMukXxkur2ltci4_rBCL4IE2ZQ4
project_id	1 dbd3fac81424f0e82bf3bbbd11cad41
user_id	d5ae773cc44d4631b05a358db818d704
froot@control	

图 3.13 keystone 验证效果

3.2.6 安装配置 glance

glance 在 OpenStack 里面负责镜像服务,镜像服务负责管理镜像模板。 第1步: 创建 glance 数据库。 首先,进入 MariaDB 数据库,命令如下:

[mysql -uroot -p]

然后, 创建 glance 数据库, 命令如下:

【CREATE DATABASE glance;】

第2步: 创建数据库用户、设置密码并赋予权限。

【GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' IDENTIFIED BY 'yhy';】 【GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' IDENTIFIED BY 'yhy';】 第 3 步: 创建 glance 用户并赋予其 admin 权限。

[source /root/admin-openre]

【openstack user create --domain default glance --password yhy】创建 glance 用户。

【openstack role add --project service --user glance admin】 给 glance 用户赋予 admin 权限。 第 4 步: 创建 image 服务。

【openstack service create --name glance --description "OpenStack Image service" image】运行命令后的效果如图 3.14 所示。

[root@controlle ck Image servio	er keystone]# openstack service cr ce" image	reatename	glance	description	"OpenSta
+ Field +		+ +			
description enabled id name type	OpenStack Image service True 9ccd6c85fa6043d381f96a360f59bf7e glance image				
[root@control]/	r koustonel#				

图 3.14 创建 image 服务效果

第5步: 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】 第 6 步:安装 glance 相关 RPM 包。

[yum install openstack-glance -y]

第7步:修改 glance 配置文件/etc/glance/glance-api.conf,注意密码的设置。

【cp /etc/glance/glance-api.conf /etc/glance/glance-api.conf.bak】备份原始配置文件。

【>/etc/glance/glance-api.conf】清空配置文件。

[openstack-config --set /etc/glance/glance-api.conf DEFAULT transport_url rabbit://openstack: yhy@controller]

[openstack-config --set /etc/glance/glance-api.conf database connection mysql+pymysql:// glance: yhy@controller/glance]

Copenstack-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】

Copenstack-config --set /etc/glance/glance-api.conf keystone_authtoken username glance

[openstack-config --set /etc/glance/glance-api.conf keystone_authtoken password yhy]

[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]

Copenstack-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/]

配置完成后的最终效果可以通过【cat】命令查看,如图 3.15 所示。

<pre>[root@controller ~]# cat /etc/glance/glance-api.conf [DEFAULT]</pre>
<pre>transport_url = rabbit://openstack:yhy@controller</pre>
[database]
connection = mysql+pymysql://glance:yhy@controller/glance
[keystone_authtoken]
auth_uri = http://controller:5000
autn_url = http://controller:3535/
auth type = password
project domain name = default
user domain name = default
username = glance
password = yhy
project_name = service
[paste deploy]
flavor = keystone
giance_store;
default store = file
filesystem store datadir = /var/lib/glance/images/
[root@controller~]#

图 3.15 glance 配置文件效果

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

【cp /etc/glance/glance-registry.conf /etc/glance/glance-registry.conf.bak】备份原始配置文件。

【>/etc/glance/glance-registry.conf】清空原始配置文件。

[openstack-config --set /etc/glance/glance-registry.conf DEFAULT transport_url rabbit://openstack: yhy@controller]

[openstack-config --set /etc/glance/glance-registry.conf database connection mysql+pymysql:// glance:yhy@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 yhy】 【openstack-config --set /etc/glance/glance-registry.conf paste_deploy flavor keystone】 第9步: 同步 glance 数据库。

【su -s /bin/sh -c "glance-manage db_sync" glance】 使用以下三条命令进行验证:

(mysql -uroot -p)

[use glance;]

(show tables;)

若出现如图 3.16 所示的表,则表示同步成功。

MariaDB [(none)]> use glance; Reading table information for completion of table and column names You can turn off this feature to get a quicker startup with -A
Database changed
MariaDB [glance]> show tables;
++
Tables_in_glance
++
alembic_version
artifact_blob_locations
artifact_blobs
artifact_dependencies
artifact_properties
artifact_tags
artifacts
image_locations
image_members
image_properties
image_tags
images
metadef_namespace_resource_types
metadef_namespaces
metadef_objects
metadef_properties
metadef_resource_types
metadef_tags
migrate_version
task_info
tasks
++
21 rows in set (0.01 sec)
MariaDB [glance]>
图 3.16 glance 数据库同步成功

基于 CentOS 的云计算系统运维与管理

第10步: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】 第 11 步: 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

第12步: 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】

如果制作好了一个 CentOS 7.5 系统的镜像,也可以通过如下命令进行操作:

【glance image-create --name "CentOS7.5-x86_64" --file CentOS_7.5.qcow2 --disk-format qcow2 --container-format bare --visibility public --progress】

使用【glance image-list】命令查看镜像列表,运行结果如图 3.17 所示。

[root@controller ~]# glance image-list	
ID	Name
+	cirros-0.3.4-x86_64
+ [root@controller ~]#	++

图 3.17 查看镜像列表

3.2.7 安装配置 nova

第1步: 创建 nova 数据库。
首先,进入 MariaDB 数据库:
【mysql-uroot-p】
然后,创建 nova 数据库:
【CREATE DATABASE nova;】
【CREATE DATABASE nova_api;】
【CREATE DATABASE nova_cell0;】
第2步: 创建数据库用户并赋予权限。
【GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' IDENTIFIED BY 'yhy';】
【GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY 'yhy';】
【GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'localhost' IDENTIFIED BY 'yhy';】
【GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'%' IDENTIFIED BY 'yhy';】
【GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'localhost' IDENTIFIED BY 'yhy';】
【GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'localhost' IDENTIFIED BY 'yhy';】

【GRANT ALL PRIVILEGES ON nova_cell0.* TO 'nova'@'%' IDENTIFIED BY 'yhy';】 【GRANT ALL PRIVILEGES ON *.* TO 'root'@'controller' IDENTIFIED BY 'yhy';】 【FLUSH PRIVILEGES;】 注意:

查看授权列表信息的命令为【SELECT DISTINCT CONCAT('User: "',user,""@"',host,"';') AS query FROM mysql.user; 】, 取消之前某个授权的命令为【REVOKE ALTER ON *.* TO 'root'@ 'controller' IDENTIFIED BY 'yhy'; 】。

第3步: 创建 nova 用户并赋予其 admin 权限。

[source /root/admin-openrc]

[openstack user create --domain default nova --password yhy]

(openstack role add --project service --user nova admin)

第4步: 创建 compute 服务。

【openstack service create --name nova --description "OpenStack Compute" compute】 第5步: 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]

第6步:安装 nova 的相关软件。

【yum install -y openstack-nova-api openstack-nova-conductor openstack-nova-cert openstack-nova-console openstack-nova-novncproxy openstack-nova-scheduler】

第7步: 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 10.1.1.128]

[openstack-config --set /etc/nova/nova.conf DEFAULT use_neutron True]

[openstack-config --set /etc/nova/nova.conf DEFAULT firewall_driver nova.virt.firewall.Noop FirewallDriver]

[openstack-config --set /etc/nova/nova.conf DEFAULT transport_url rabbit://openstack:yhy@ controller]

【openstack-config --set /etc/nova/nova.conf database connection mysql+pymysql://nova:yhy@ controller/nova】

[openstack-config --set /etc/nova/nova.conf api_database connection mysql+pymysql://nova: yhy@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 yhy]
[openstack-config --set /etc/nova/nova.conf keystone_authtoken password yhy]

True

[openstack-config --set /etc/nova/nova.conf vnc vncserver_listen 10.1.1.128]
[openstack-config --set /etc/nova/nova.conf vnc vncserver_proxyclient_address 10.1.1.128]
[openstack-config --set /etc/nova/nova.conf glance api_servers http://controller:9292]

 $\label{eq:constack-config} \ensuremath{\mathsf{-set}}\/etc/nova/nova.conf oslo_concurrency \ensuremath{\mathsf{lock}}\/var/lib/nova/tmp \ensuremath{\mathsflock}\/var/lib/nova/tmp \ensuremath{\mathsflock}\/var/lib/nova/tmp \ensuremath{\mathsflock}\/var/lib/nova/tmp \ensuremath{\mathsflock}\/var/lib/nova/tmp \ensuremath{\mathsflock}\/var/lib/nova/tmp \ensuremath{\mathsflock}\$

注意:

记得在其他节点上替换 IP,并注意密码、文档红色及绿色的地方。

第8步:设置 cell (单元格)。

OpenStack 在控制平面上的性能瓶颈主要集中于 Message Queue 和 Database 中。尤其是 Message Queue,随着计算节点的增加,其性能变得越来越差。因为 OpenStack 里每个资源和 接口都是通过消息队列通信的,有测试表明,当集群规模达到 200 时,一条消息可能要在十 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 关联创建好的数据库 nova_cell0:

[nova-manage cell_v2 map_cell0 --database_connection mysql+pymysql://root:yhy@controller/ nova_cell0]

创建一个名为 cell1 的常规 cell,这个 cell 里面将包含计算节点:

【nova-manage cell_v2 create_cell --verbose --name cell1 --database_connection mysql+pymysql:// root:yhy@controller/nova_cell0 --transport-url rabbit://openstack:yhy@controller:5672/】

检查部署是否正常:

[nova-status upgrade check]

创建和映射 cell0,并将现有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,需要运行下面的命令,并且添加到单元格中:

[nova-manage cell_v2 discover_hosts]

当然,读者可以通过在控制节点的 nova.conf 文件的[scheduler]模块下添加 discover_ hosts_in_cells_interval=-1 命令,使其自动发现。

第9步:安装 placement。

从 Ocata 版开始,需要安装配置 placement 参与 nova 调度,否则虚拟机将无法创建。 安装配置 placement 参与 nova 调度的命令如下:

[yum install -y openstack-nova-placement-api]

创建 placement 用户和 placement 服务,命令如下:

[openstack user create --domain default placement --password yhy]

[openstack role add --project service --user placement admin]

【openstack service create --name placement --description "OpenStack Placement" placement】 创建 placement 的 endpoint,命令如下:

Copenstack 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,命令如下:

[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 placement]

[openstack-config --set /etc/nova/nova.conf placement password yhy]

[openstack-config --set /etc/nova/nova.conf placement os_region_name RegionOne]

配置修改 00-nova-placement-api.conf 文件,这一步在没创建虚拟机的时候会出现禁止访问资源的问题。配置修改的具体命令如下。

【cd /etc/httpd/conf.d/】进入配置文件所在的目录。

【cp 00-nova-placement-api.conf 00-nova-placement-api.conf.bak】备份原配置文件。

【>00-nova-placement-api.conf】清空原配置文件。

【vim 00-nova-placement-api.conf】打开编辑配置文件。

在打开的配置文件中添加以下内容:

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>

【systemctl restart httpd】修改配置文件后, 需要重启 httpd 服务。

【nova-status upgrade check】检查是否配置成功,命令运行效果如图 3.18 所示。



图 3.18 nova 配置效果

第10步:设置 nova 相关服务开机启动。

[systemctl enable openstack-nova-api.service openstack-nova-cert.service openstack-novaconsoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstacknovanovncproxy.service] 启动 nova 服务:

[systemctl restart openstack-nova-api.service openstack-nova-cert.service openstack-novaconsoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstacknovanovncproxy.service]

查看 nova 服务:

[systemctl status openstack-nova-api.service openstack-nova-cert.service openstack-novaconsoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstacknovanovncproxy.service]

【systemctl list-unit-files |grep openstack-nova-*】 第 11 步:验证 nova 服务。 【unset OS_TOKEN OS_URL】 【source /root/admin-openrc】 【nova service-list】 【openstack endpoint list】查看 endpoint list 是否有结果正确输出。

3.2.8 安装配置 neutron

第1步: 创建 neutron 数据库。

【CREATE DATABASE neutron;】

第2步: 创建数据库用户并赋予权限。

【GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' IDENTIFIED BY 'yhy';】 【GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' IDENTIFIED BY 'yhy';】 第3步: 创建 neutron 用户并赋予其 admin 权限。

[source /root/admin-openrc]

[openstack user create --domain default neutron --password yhy]

[openstack role add --project service --user neutron admin]

第4步: 创建 network 服务。

【openstack service create --name neutron --description "OpenStack Networking" network】 第5步: 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】 第 6 步:安装 neutron 相关软件。

【 yum install -y openstack-neutron openstack-neutron-ml2 openstack-neutron-linuxbridge ebtables】

第7步: 修改 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]

基于 CentOS 的云计算系统运维与管理

[openstack-config --set /etc/neutron/neutron.conf DEFAULT allow overlapping ips True] [openstack-config --set /etc/neutron/neutron.conf DEFAULT auth strategy keystone]

Copenstack-config --set /etc/neutron/neutron.conf DEFAULT transport url rabbit://openstack: yhy@controller

Copenstack-config --set /etc/neutron/neutron.conf DEFAULT notify nova on port status changes True

Copenstack-config --set /etc/neutron/neutron.conf DEFAULT notify nova on port data changes True

Copenstack-config --set /etc/neutron/neutron.conf keystone authtoken auth uri http://controller: 5000 **J**

Copenstack-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]

Copenstack-config --set /etc/neutron/neutron.conf keystone authtoken project domain name default

Copenstack-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 yhy]

Copenstack-config --set /etc/neutron/neutron.conf database connection mysql+pymysql://neutron: yhy@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]

Copenstack-config --set /etc/neutron/neutron.conf nova password yhy

Copenstack-config --set /etc/neutron/neutron.conf oslo concurrency lock path /var/lib/neutron/ tmp

第8步: 配置文件/etc/neutron/plugins/ml2/ml2 conf.ini。

[openstack-config --set /etc/neutron/plugins/ml2/ml2 conf.ini ml2 type drivers flat,vlan,vxlan]

Copenstack-config --set /etc/neutron/plugins/ml2/ml2 conf.ini ml2 mechanism drivers linuxbridge,12population

Copenstack-config --set /etc/neutron/plugins/ml2/ml2 conf.ini ml2 extension drivers port security]

[penstack-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】 第9步: 配置文件/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:eno50332184]

[openstack-config --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini vxlan enable_vxlan True]

Copenstack-config --set /etc/neutron/plugins/ml2/linuxbridge_agent.ini vxlan local_ip 10.2.2.120

[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:

eno50332184 是外网网卡,通常情况下,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, 例如, VXLAN下 vm-linuxbridge->vxlan -----tun----vxlan-> linuxbridge-vm。

第10步: 配置文件/etc/neutron/l3_agent.ini。

【openstack-config --set /etc/neutron/l3_agent.ini DEFAULT interface_driver neutron.agent.linux. interface.BridgeInterfaceDriver】

[penstack-config --set /etc/neutron/l3_agent.ini DEFAULT external_network_bridge]

[openstack-config --set /etc/neutron/13_agent.ini DEFAULT debug false]

第11步: 配置文件/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】 第 12 步: 重新配置文件/etc/nova/nova.conf。 这一步配置的目的是让 Compute 节点能使用 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 project_name service】 【openstack-config --set /etc/nova/nova.conf neutron project_name service】

Copenstack-config --set /etc/nova/nova.conf neutron service_metadata_proxy True

[openstack-config --set /etc/nova/nova.conf neutron metadata_proxy_shared_secret yhy]

第13步:将 dhcp-option-force=26,1450 写入/etc/neutron/dnsmasq-neutron.conf。

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

第14步: 配置文件/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 yhy]

【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】

第15步:创建硬链接。

[ln -s /etc/neutron/plugins/ml2/ml2_conf.ini /etc/neutron/plugin.ini]

第16步: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]

第17步:重启 nova 服务。

因为前面修改了 nova.conf 文件,所以这里要重启 nova 服务。

[systemctl restart openstack-nova-api.service]

[systemctl status openstack-nova-api.service]

第18步: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]

[systemctl restart neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service]

[systemctl status neutron-server.service neutron-linuxbridge-agent.service neutron-dhcp-agent.service]

第19步:启动 neutron-13-agent.service 并设置开机启动。

[systemctl enable neutron-13-agent.service]

[systemctl restart neutron-13-agent.service]

[systemctl status neutron-13-agent.service]

第20步:执行验证。

[source /root/admin-openrc]

[neutron ext-list]

[neutron agent-list]

第 21 步: 创建 VXLAN 模式网络,让虚拟机能访问外网。

【source /root/admin-openrc】先执行环境变量。

【neutron --debug net-create --shared provider --router:external True --provider:network_type flat --provider:physical_network provider】创建 flat 模式的 public 网络。注意,这个 public 是外 部网络,必须是 flat 模式的。

执行完这一步,在界面里进行操作,把 public 网络设置为共享和外部网络。

【neutron subnet-create provider 192.168.64.0/24 --name provider-sub --allocation-pool start=192.168.64.50,end=192.168.64.90 --dns-nameserver 8.8.8.8 --gateway 192.168.64.2】创建 public 网络子网,名为 public-sub,网段是 192.168.64.0/24,并且 IP 范围是 50~90 (一般是给 VM 用的 floating IP), DNS 设置为 8.8.8.8, 网关为 192.168.64.2。

【neutron net-create private --provider:network_type vxlan --router:external False --shared】创 建名为 private 的私有网络,网络模式为 VXLAN。

【neutron subnet-create private --name private-subnet --gateway 192.168.1.1 192.168.1.0/24】 创建名为 private-subnet 的私有网络子网,网段为 192.168.1.0,该网段就是虚拟机获取的私有 IP 地址。

假如客户公司的私有云环境用于不同的业务,比如行政、销售、技术等,那么就可以创 建三个不同名称的私有网络,命令如下:

[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]

创建路由。在界面上操作,单击"项目"下的"网络"→"路由"→"新建路由"命令,

路由名称可随便命名,这里命名为 router,管理员状态选择"上(up)",外部网络选择"provider", 单击"新建路由"命令后,提示 router 创建成功。 接着单击"接口"下的"增加接口"命令,添加一个连接私有网络的接口,选中"private: 192.168.12.0/24"。

单击"增加接口"命令后,可以看到两个接口先是 down 状态,过一会儿经过刷新就是 running 状态。注意,务必为 running (运行)状态,否则虚拟机网络无法访问外网。

第22步:检查网络服务。

执行【neutron agent-list】命令查看服务是否是"笑脸"状态。

3.2.9 安装 dashboard

【yum install -y openstack-dashboard】安装 dashboard 相关软件包。

【vim /etc/openstack-dashboard/local_settings】修改配置文件,内容如下:

```
SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
CACHES = {
    'default': {
        'BACKEND':'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION':'controller1:11211',
      }
}
```

启动 dashboard 服务并设置开机启动:

[systemctl restart httpd.service memcached.service]

[systemctl status httpd.service memcached.service]

至此, Controller 节点搭建完毕, 打开 Firefox 浏览器访问 http://1.1.1.128/dashboard/即可 进入 OpenStack 登录界面, 如图 3.19 所示。





3.3 Compute 节点部署

环境准备 3.3.1

第1步:新建虚拟机,CPU设为4核,内存设为4GB。 第2步:删除原来的网卡,新添加两张网卡。采用最小化安装系统。 第3步:设置 IP 地址,关闭防火墙、SELinux,设置主机名。 第4步:从Controller节点复制 yum 源。 登录 Controller 节点,执行如下命令: [cd /ete/yum.repo.d] 【rm-rf*】删除原来的 yum 源。 【scp -p yum7.repo 10.1.1.121:/etc/yum.repo.d/】从 Controller 节点复制 yum 源。 【scp -p ocata.repo 10.1.1.121:/etc/vum.repo.d/】从 Controller 节点复制 vum 源。 第5步:登录 Compute 节点,执行如下命令: 【yum clean all】清除原有 yum 源。 【yum makecache】把服务器的包信息下载到本地计算机缓存起来。 第6步:安装基础软件包。 [yum install -y net-tool wget vim ntpdate ntp base-completion] 第7步:同步时间。 【ntpdate 10.1.1.120】同步 Controller 节点的时间。 【vim /etc/ntp.conf】编辑时钟同步配置文件,添加如下代码: Server 10.1.1.120 iburst

【systemctl restart ntpd】重启时钟同步服务。

第8步:复制 Controller 节点的 hosts 文件到 Compute 节点。

登录 Controller 节点,运行如下命令将 Controller 节点的 hosts 文件复制到 Compute 节点。

[scp -p /etc/hosts 10.1.1.121:/etc/hosts]

第9步:做 SSH 互信。

【ssh-keygen -t rsa】运行此命令后会在/root/.ssh下生成一个 id rsa.pub 的公钥, 然后将其 复制到相互的机器中。

【ssh-copy-id -i /root/.ssh/id rsa.pub -p 22 root@10.1.1.121】在 Controller 节点中运行。

【ssh-copy-id -I /root/.ssh/id rsa.pub -p 22 root@10.1.1.120】在 Compute 节点中运行。

然后在 Compute 节点运行一次【ssh controller】命令,在 Controller 节点上运行一次【ssh compute】命令。今后,计算机之间 SSH 登录将不再需要密码,用户可以直接登录。 品版必多

第10步:建议给虚拟机做一个快照。

3.3.2 安装与配置相关依赖包

执行【yum install -y openstack-selinux python-openstackclient yum-plugin-priorities openstack-nova-compute openstack-utils ntpdate】命令安装软件。

第1步: 配置 nova.conf。

【cp /etc/nova/nova.conf /etc/nova/nova.conf.bak】备份原来的配置文件。

【>/etc/nova/nova.conf】清空原配置文件。

[openstack-config --set /etc/nova/nova.conf DEFAULT auth_strategy keystone]

[openstack-config --set /etc/nova/nova.conf DEFAULT my_ip 10.1.1.121]

[openstack-config --set /etc/nova/nova.conf DEFAULT use_neutron True]

【openstack-config --set /etc/nova/nova.conf DEFAULT firewall_driver nova.virt.firewall.Noop FirewallDriver】

[openstack-config --set /etc/nova/nova.conf DEFAULT transport_url rabbit://openstack:yhy@ 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]

Copenstack-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 yhy] [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 placement] [openstack-config --set /etc/nova/nova.conf placement password yhy] [openstack-config --set /etc/nova/nova.conf placement os region name RegionOne] Copenstack-config --set /etc/nova/nova.conf vnc enabled True [openstack-config --set /etc/nova/nova.conf vnc keymap en-us] 及版必

Copenstack-config --set /etc/nova/nova.conf vnc vncserver_listen 0.0.0.0

(openstack-config --set /etc/nova/nova.conf vnc vncserver_proxyclient_address 10.1.1.121)

[openstack-config --set /etc/nova/nova.conf vnc novncproxy_base_url http:// 10.1.1.121:6080/ 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】 第2步: 设置 libvirtd.service 和 openstack-nova-compute.service 开机启动。 【systemctl enable libvirtd.service openstack-nova-compute.service】设置开机启动。 【systemctl restart libvirtd.service openstack-nova-compute.service】重启相关服务。 【systemctl status libvirtd.service openstack-nova-compute.service】重启相关服务。 【systemctl status libvirtd.service openstack-nova-compute.service】 简易服务状态。

重新登录 Dashboard, 在"管理员"→"虚拟机管理器"下可以看到 Compute 节点。

3.3.3 安装 Neutron

第1步:安装相关软件包。

[yum install -y openstack-neutron-linuxbridge ebtables ipset]

第2步: 配置 neutron.conf。

【cp /etc/neutron/neutron.conf /etc/neutron/neutron.conf.bak】备份原配置文件。

【>/etc/neutron/neutron.conf】清空原配置文件。

[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: yhy@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]

Copenstack-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 yhy]

[openstack-config --set /etc/neutron/neutron.conf oslo_concurrency lock_path /var/lib/neutron/

tmp]

第3步: 配置/etc/neutron/plugins/ml2/linuxbridge_agent.ini 二层交换。

【 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 10.2.2.121]

【 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】

第4步: 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 yhy]

第5步:重启和开机自启动相关服务。

[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]

添加 Cinder 节点 34

将 Cinder 作为计算节点 3.4.1

第1步: 配置 nova。

要是想用 Cinder 作为计算节点,则需要修改 nova 配置文件(注意,这一步是在计算节点 Compute 上操作的)。

[openstack-config --set /etc/nova/nova.conf cinder os region name RegionOne]

[systemctl restart openstack-nova-compute.service]

第2步:在Controller上重启 nova 服务。

[systemctl restart openstack-nova-api.service]

systemctl status openstack-nova-api.service

在 Controller 上执行验证 3.4.2

[source /root/admin-openre]

[neutron agent-list]

[nova-manage cell v2 discover hosts]

至此, Compute 节点搭建完毕, 运行 nova host-list 可以查看新加入的 computel 节点。

如果需要再添加另外一个 Compute 节点,只要重复上述步骤即可,注意修改计算机名和 **IP**地址。

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)

安装配置 Cinder 3.4.3

反权所 第1步: 创建数据库用户并赋予权限。 登录数据库,执行如下命令: **CREATE DATABASE cinder;** GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'localhost' IDENTIFIED BY 'yhy'; GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'%' IDENTIFIED BY 'yhy'; 第2步: 创建 Cinder 用户并赋予 admin 权限。 [source /root/admin-openrc]

[openstack user create --domain default cinder --password yhy]

[openstack role add --project service --user cinder admin]

第3步: 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】 第4步: 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]

第5步:安装 Cinder 相关服务。

[yum install openstack-cinder -y]

第6步:修改 Cinder 配置文件。

【cp /etc/cinder.conf /etc/cinder.conf.bak】备份原配置文件。

【>/etc/cinder/cinder.conf】清空原配置文件。

[openstack-config --set /etc/cinder/cinder.conf DEFAULT transport_url rabbit://openstack:yhy@ controller]

[openstack-config --set /etc/cinder/cinder.conf DEFAULT my_ip 10.1.1.120]

[openstack-config --set /etc/cinder/cinder.conf DEFAULT auth_strategy keystone]

【openstack-config --set /etc/cinder/cinder.conf database connection mysql+pymysql://cinder: yhy@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.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 yhy]

【openstack-config --set /etc/cinder/cinder.conf oslo_concurrency lock_path /var/lib/cinder/tmp】 第7步: 同步数据库。

[su -s /bin/sh -c "cinder-manage db sync" cinder]

第8步: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]

第9步:安装 Cinder 节点。在这里,我们需要额外添加一个硬盘(/dev/sdb)用作 Cinder 的存储服务(注意,这一步是在 Cinder 节点上操作的)。

[yum install lvm2 -y]

第10步:启动服务并设置为开机自启(注意,这一步是在 Cinder 节点上操作的)。

[systemctl enable lvm2-lvmetad.service]

[systemctl start lvm2-lvmetad.service]

[systemctl status lvm2-lvmetad.service]

第11步: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 点上操作的)。

fdisk -1

[pvcreate /dev/sdb]

[vgcreate cinder-volumes /dev/sdb]

第12步:编辑存储节点 lvm.conf 文件(注意,这一步是在 Cinder 节点上操作的)。

[vim /etc/lvm/lvm.conf]

在 devices 下面添加 filter = ["a/sda/", "a/sdb/", "r/.*/"]。

然后重启下 lvm2 服务:

[systemctl restart lvm2-lvmetad.service]

[systemctl status lvm2-lvmetad.service]

第13步:安装 openstack-cinder、targetcli(注意,这一步是在 Cinder 节点上操作的)。

[yum install openstack-cinder openstack-utils targetcli python-keystone ntpdate -y]

第14步:修改 Cinder 配置文件(注意,这一步是在 Cinder 节点上操作的)。

[cp/etc/cinder/cinder.conf/etc/cinder/cinder.conf.bak]

[>/etc/cinder.conf]

[openstack-config --set /etc/cinder/cinder.conf DEFAULT auth_strategy keystone]

[openstack-config --set /etc/cinder/cinder.conf DEFAULT my_ip 10.1.1.122]

[openstack-config --set /etc/cinder/cinder.conf DEFAULT enabled_backends lvm]

Copenstack-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:yhy @controller]

【openstack-config --set /etc/cinder/cinder.conf database connection mysql+pymysql://cinder: yhy@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]

Copenstack-config --set /etc/cinder.conf keystone_authtoken memcached_servers controller:

[openstack-config --set /etc/cinder/cinder.conf keystone_authtoken auth_type password]

 $\$ openstack-config --set /etc/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 yhy]

[openstack-config --set /etc/cinder/cinder.conf lvm volume_driver cinder.volume.driver.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]

Copenstack-config --set /etc/cinder/cinder.conf oslo_concurrency lock_path /var/lib/cinder/tmp

第15步:启动 openstack-cinder-volume 和 target 并设置开机启动(注意,这一步是在 Cinder 节点上操作的)。

【systemctl enable openstack-cinder-volume.service target.service】 【systemctl restart openstack-cinder-volume.service target.service】 【systemctl status openstack-cinder-volume.service target.service】 第 16 步:验证 Cinder 服务是否正常。 [source /root/admin-openrc]
[cinder service-list]



本章详细介绍了在 CentOS 7 中安装部署 OpenStack 的方法,主要内容包括 OpenStack 的 基础知识, OpenStack 的体系架构, OpenStack 的部署工具,使用 RDO 部署 OpenStack 以及 管理 OpenStack 等。本章的重点是掌握 OpenStack 的体系架构,使用 RDO 部署 OpenStack 的 方法以及镜像、虚拟网络和实例的管理。

