# 云计算部署与管理

什么是云计算

• 基于互联网的相关服务的增加、使用和交付模式

• 这种模式提供可用的、便捷的、按需的网络访问,

进入可配置的计算资源共享池(资源包括网络,服务器,存储,应用软件,服务)

• 这些资源能够被快速提供,只需投入很少的管理工作,或不服务供应商迚行很少的交互

• 通常涉及通过互联网来提供劢态易扩展且经常是虚拟化的资源

IaaS

• IaaS(Infrastructure as a Service),即基础设施即服务

• 提供给消费者的服务是对所有计算基础设施的利用,

包括处理CPU、内存、存储、网络和其它基本的计算

资源,用户能够部署和运行任意软件,包括操作系统

和应用程序

• 消费者不管理或控制任何云计算基础设施,但能控制

操作系统的选择、存储空间、部署的应用

• IaaS通常分为三两种用法:公有云共的和、私有云的和混合云

PaaS

• PaaS是Platform-as-a-Service的缩写,意思是平台即服务

• 云计算时代相应的服务器平台戒者开发环境作为服务

进行提供就成为了PaaS

• PaaS运营商所需提供的服务,丌仅仅是单纯的基础

平台,而且包括针对该平台的技术支持服务,甚至针

对该平台而迚行的应用系统开发、优化等服务

• 简单地说,PaaS平台就是指云环境中的应用基础设

施服务,也可以说是中间件即服务

SaaS

• SaaS是Software-as-a-Service(软件即服务)的简称

• 它是一种通过Internet提供软件的模式,厂商将应用软

件统一部署在自己的服务器上,客户可以根据自己实际

需求,通过互联网向厂商定购所需的应用软件服务

• 用户不用再购买软件,而改用向提供商租用基于Web的

软件,来管理企业经营活劢,且无需对软件进行维护,

服务提供商会全权管理和维护软件,软件厂商在向客户

提供互联网应用的同时,也提供软件的离线操作和本地

数据存储,让用户随时随地都可以使用其定购的软件和

服务

# Openstack简介

什么是Openstack

• OpenStack是一个由NASA(美国国家航空航天局)

和Rackspace合作研发并发起的项目

• OpenStack是一套IaaS解决方案

• OpenStack是一个开源的云计算管理平台

• 以Apache许可证为授权

Openstack主要组件

• Horizon

– 用于管理Openstack各种服务的、基于web的管理接口

– 通过图形界面实现创建用户、管理网络、启动实例等操作

• Keystone

– 为其他服务提供认证和授权的集中身份管理服务

– 也提供了集中的目录服务

– 支持多种身份认证模式,如果密码认证、令牌认证、以及AWS(亚马逊Web服务)登陆

– 为用户和其他服务提供了SSO认证服务

• Neutron

– 一种软件定义网络服务

– 用于创建网络、子网、路由器、管理浮动IP地址

– 可以实现虚拟交换机、虚拟路由器

– 可用于在项目中创建VPN

• Cinder

– 为虚拟机管理存储卷的服务

– 为运行在Nova中的实例提供永久的块存储

– 可以通过快照进行数据备份

– 经常应用在实例存储环境中,如果数据库文件

• Nova

– 在节点上用于管理虚拟机的服务

– Nova是一个分布式的服务,能够与Keystone交互实现认证,与Glance交互实现镜像管理

– Nova被设计成在标准硬件上能够进行水平扩展

– 启动实例时,如果有需要则下载镜像

• Glance

– 扮演虚拟机镜像注册的角色

– 允许用户为直接存储拷贝服务器镜像

– 这些镜像可以用于新建虚拟机的模板

配置 openstack 环境

物理机：

安装dns，能作为转发dns服务器即可

安装配置chronyd server

添加一个网桥 vbr1

配置2个虚拟机

第一个安装管理节点，最少8G内存

单独添加一块硬盘给openstack cinder使用 20G

单独添加一块网卡 使用vbr1

第二个安装nova节点，最少5G内存

单独添加一块网卡 使用vbr1

一、真机:

1、物理机安装配置dns

yum -y install bind-chroot

vim /etc/named.conf

options{

listen-on port 53 { 192.168.1.254; };

//listen-on-v6 port 53 { ::1; };

allow-query {any;};

forwarders { 176.19.0.26; };

dnssec-enable no;

dnssec-validation no;

};

systemctl restart named

2、安装配置时间服务器

vim /etc/chrony.conf

server ntp1.aliyun.com iburst

bindacqaddress 0.0.0.0

allow 0/0

systemctl restart chronyd

systemctl enable chronyd

验证查看

chronyc sources -v

^? 176.19.7.250 0 9 0 - +0ns[ +0ns] +/- 0ns

这里是问号就是不成功

这是\* 表示同步成功

这是+ 表示备用，有多台同步服务器

3、添加一块新网桥

vim /etc/libvirt/qemu/networks/vbr1.xml

和vbr.xml类似，改下名字和网段

virsh net-define vbr1.xml

virsh net-start vbr1

virsh net-autostart vbr1

4、修改两个虚拟机的配置文件

sed '/centos7.5/s/centos7.5/node1/' demo.xml > node1.xml

sed '/centos7.5/s/centos7.5/node2/' demo.xml > node2.xml

vim node1.xml

<memory unit='GiB'>8</memory>

<currentMemory unit='GiB'>8</currentMemory>

添加一部分

<disk type='file' device='disk'>

<driver name='qemu' type='qcow2'/>

<source file='/var/lib/libvirt/images/disk.img'/>

<target dev='vdb' bus='virtio'/>

</disk>

添加一部分

<interface type='network'>

<source network='vbr1'/>

<model type='virtio'/>

</interface>

vim node2.xml

<memory unit='GiB'>5</memory>

<currentMemory unit='GiB'>5</currentMemory>

添加一部分

<interface type='network'>

<source network='vbr1'/>

<model type='virtio'/>

</interface>

cd /var/lib/libvirt/images/

qemu-img create -b centos7\_module.qcow2 -f qcow2 node1.img 200G

qemu-img create -b centos7\_module.qcow2 -f qcow2 node2.img

qemu-img create -f qcow2 disk.img 20G

二、虚拟机修改

1、修改IP为静态

vim ifcfg-eth0 eth0是公共网络

TYPE=Ethernet

BOOTPROTO=none

IPV6INIT=no

DEVICE=eth0

ONBOOT=yes

IPADDR=192.168.1.10/11

PREFIX=24

GATEWAY=192.168.1.254

vim ifcfg-eth1 eth1是隧道接口

TYPE=Ethernet

BOOTPROTO=none

IPV6INIT=no

DEVICE=eth1

ONBOOT=yes

IPADDR=192.168.2.10/11

PREFIX=24

systemctl restart network

2、将node1的/分区空间扩容

node1上扩容/

df -h /分区只有16G

yum -y install cloud-utils-growpart

growpart /dev/vda 1 先将分区空间扩容 1 表示全部

xfs\_growfs / 扩容文件系统

df -h

######在使用growpart /dev/vda 1命令时，如果是ssh连接的虚拟机会报错，unexpected output in sfdisk。这是因为物理机的默认语言是中文，ssh连接使用的还是物理机的终端，所以修改真机的LANG=en\_US.utf-8 就可以了。使用virsh console 连接虚拟机不会出现这个问题，这样连接使用的是虚拟机的终端，虚拟机默认是英文，所以没有问题

3、配置openstack软件包的yum源（两台虚拟机都配置）

在虚拟机导入gpgcheck的公钥

yum -y install lftp lrzsz

lftp 192.168.1.254/centos7/

get RPM-GPG-KEY-CentOS-7

bye

rpm --import RPM-GPG-KEY-CentOS-7

将默认的源 gpgcheck改为=1

在真机上将两个镜像挂载

RHEL7OSP-10.iso /var/ftp/openstack

RHEL7-extras.iso /var/ftp/openstack-ext

配置镜像里面的yum源

使用脚本配置：

#!/bin/bash

cd /var/ftp/openstack/

for i in `ls -d rhel7\*`

do

cat >> /var/ftp/openstack.repo <<EOF

[$i]

name=$i

baseurl=ftp://192.168.1.254/openstack/$i

enabled=1

gpgcheck=0

EOF

done

从文件读取

cd /var/ftp/openstack/

while read l

do

cat >> /var/ftp/openstack.repo <<EOF

[$i]

name=$i

baseurl=ftp://192.168.1.254/openstack/$i

enabled=1

gpgcheck=0

EOF

done <./aa.txt

把openstack-ext的yum源 再追加进入 openstack.repo

将这个repo放入两个虚拟机中

4、修改两个虚拟机的主机名，让两个机器相互ping通

hostnamectl set-hostname openstack node1

hostnamectl set-hostname nova node2

5、初始化openstack的卷存储，卷组名称必须叫cinder-volumes，在openstack上操作

openstack 需要使用块设备，这里要给 openstack 添加一个块设备

安装逻辑卷分区软件 yum -y install lvm2

pvcreate /dev/vdb

vgcreate cinder-volumes /dev/vdb

vgs

6、安装openstack依赖的软件包（两个节点都安装）

qemu-kvm

libvirt-client

libvirt-daemon

libvirt-daemon-driver-qemu

python-setuptools

libvirt-daemon-driver-network

安装前再次检查基础环境

（1）禁用 selinux （物理机+2台虚拟机）

（2）卸载 firewalld （虚拟机）

（3）卸载 NetworkManager（虚拟机stop并卸载）

（4）配置主机IP地址 （虚拟机两个网段ip地址都互相ping通）

（5）配置主机yum客户端

（6）创建cinder-volumes的卷组

（7）导入公钥

（8）安装依赖的软件包

7、安装openstack 工具包 openstack-packstack（openstack节点上安装）

yum -y install openstack-packstack

8、创建应答文件及修改应答文件

packstack --gen-answer-file answer.ini

vim answer.ini

11 CONFIG\_DEFAULT\_PASSWORD=Taren1 //配置默认密码

42 CONFIG\_SWIFT\_INSTALL=n //不使用外界存储

75 CONFIG\_NTP\_SERVERS=192.168.1.254 //NTP服务器ip地址

98 CONFIG\_COMPUTE\_HOSTS=192.168.1.10,192.168.1.11 //计算节点的ip地址，同一网段

102 CONFIG\_NETWORK\_HOSTS=192.168.1.10,192.168.1.11 //网络配置节点的ip地址

554 CONFIG\_CINDER\_VOLUMES\_CREATE=n //cinder卷组，不自动创建，自己创

561 CONFIG\_CINDER\_VOLUMES\_SIZE=20G //cinder卷组大小，不能超过自己创建的

840 CONFIG\_NEUTRON\_ML2\_TYPE\_DRIVERS=flat,vxlan //两块网卡，配置两种类型

876 CONFIG\_NEUTRON\_ML2\_VXLAN\_GROUP=239.1.1.5 //用于通信的组播地址

910 CONFIG\_NEUTRON\_OVS\_BRIDGE\_MAPPINGS=physnet1:br-ex //网桥映射，把物理网络映射给一个

921 CONFIG\_NEUTRON\_OVS\_BRIDGE\_IFACES=br-ex:eth0 //绑定一个物理网络eth0：必须是1.10这ip

936 CONFIG\_NEUTRON\_OVS\_TUNNEL\_IF=eth1 //隧道通信的网卡

1179 CONFIG\_PROVISION\_DEMO=n //demo测试，改为不测试

9、安装openstack（packstack --help | grep answer）

在openstack节点上安装主程序

[root@openstack ~]# packstack --answer-file=answer.ini

出现 \*\*\*\* Installation completed successfully \*\*\*\*\*\*表示安装成功

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

在openstack节点上安装主程序的过程中，常见的报错汇总：

1、ERROR : Error appeared during Puppet run: 192.168.1.10\_controller.pp

Error: Cannot allocate memory - fork(2)

这是提示不能分配内存，产生的原因是192.168.1.10节点创建时内存分配不足，修改内存为8G以上重新安装

2、ERROR : Error appeared during Puppet run: 192.168.1.10\_controller.pp

Error: /usr/sbin/ntpdate 192.168.1.254 returned 1 instead of one of [0]

这里是因为物理主机上ntp服务配置不成功，重新查看物理主机chrony.conf的配置

需要注意的是，server的地址一定要正确，allow ip/子网掩码一定要开，因为物理主机要做服务端

3、ERROR : Error appeared during Puppet run: 192.168.1.10\_network.pp

Error: Could not start Service[openvswitch]: Execution of '/usr/bin/systemctl start openvswitch' returned 1: A dependency job for openvswitch.service failed. See 'journalctl -xe' for details.

在使用命令 journalctl -xe | grep -i error，输出的详细信息为

Jun 26 17:34:16 openstack ovs-ctl[18076]: ERROR: This system does not support "SSSE3".

这里的问题是系统的cpu模式不对，产生的原因是虚拟机配置文件xml中删除了cpu mode的部分，导致之后创建虚拟机自动生成的cpu mode与不删除时的模式不一致。

有两种解决方法

（1）修改openstack.xml和nova.xml的模板文件

将缺失的cpu mode部分，添加回去（可从其他人拷贝），然后重新创建虚拟机

（2）关闭openstack和nova节点虚拟机，使用virt-manager图形管理，点开2个虚拟机的小灯泡

修改cpus的配置，点选复制主机CPU配置，重新启动再次安装

4、ERROR : Failed to run remote script, stdout: Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

No package puppet available.

No package hiera available.

这里提示puppet hiera不可用，产生的原因是nova节点上未安装这两个包，在nova节点上安装后再次安装主程序

5、ERROR : Error appeared during Puppet run: 192.168.1.11\_compute.pp

Error: /Stage[main]/Packstack::Nova::Compute::Libvirt/Exec[virsh-net-destroy-

default]: Could not evaluate: Could not find command '/usr/bin/virsh'

这是在1.11节点上出现的问题，命令virsh未找到，在nova节点上安装libvirt-client软件包来提供此命令

之后openstack节点上再次安装主程序

6 当提示说某一项已经定义,不能重复定义.

原因是:第一次安装手动种植,这是第二次安装了,要恢复的话,得关闭虚拟机,重新生成img文件,重新安装依赖包,answer.ini可以用以前的

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

br

物理机A，物理网卡eth0，里面2台虚拟机A1和A2

两台虚拟机默认不能通讯，再虚拟一个交换机，2台虚拟机都连上交换机，就能通讯

物理机B，物理网卡eth0，里面2台虚拟机B1和B2

A和B的虚拟机之间不能通信，但是A和B主机可以通过物理交换机进行通信

openstack技术可以实现

vxlan技术 vxlan技术

A1-->虚拟交换机------------------->连接A物理网卡----->物理交换机------>连接B物理网卡------>虚拟交换机-->B1

br-int br-ex br-ex br-int

10、配置外部OVS网桥（自动生成的）

• 配置br-ex为外部OVS网桥 自动生成的

[root@openstack ~]# cat /etc/sysconfig/network-scripts/ifcfg-br-ex

ONBOOT="yes"

NM\_CONTROLLED="no"

IPADDR="192.168.1.66"

PREFIX="24"

GATEWAY="192.168.1.254"

DEVICE=br-ex

DEVICETYPE=ovs

OVSBOOTPROTO="static"

TYPE=OVSBridge

• 配置eth0为外部OVS网桥的端口(将虚拟br-ex绑定到eth0物理网卡上) 自动生成的

[root@openstack ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

NAME=eth0

DEVICETYPE=ovs

TYPE=OVSPort

OVS\_BRIDGE=br-ex

ONBOOT=yes

BOOTPROTO=none

Openstack操作基础

Horizon配置

概述

• Horizon是一个用以管理、控制OpenStack服务的

Web控制面板,也称之为Dashboard仪表盘

• 可以管理实例、镜像、创建密匙对,对实例添加卷、

操作Swift容器等。除此之外,用户还可以在控制面

板中使用终端(console)或VNC直接访问实例

• 基于python的django web框架迚行开发

功能与特点

• 实例管理:创建、终止实例,查看终端日志,VNC连接,添加卷等

• 访问不安全管理:创建安全群组,管理密匙对,设置浮动IP等

• 偏好设定:对虚拟硬件模板可以迚行丌同偏好设定

• 镜像管理:编辑或删除镜像

• 用户管理:创建用户等

• 卷管理:创建卷和快照

• 对象存储处理:创建、删除容器和对象

• Horizon BUG 处理

• 安装虽然没有报错,但默认无法打开 Horizon,这是

一个软件的配置 BUG

– /etc/httpd/conf.d/15-horizon\_vhost.conf

在 WSGIProcessGroup apache这一行下面加入

ServerAlias localhost

WSGIDaemonProcess apache group=apache processes=3

threads=10 user=apache

WSGIProcessGroup apache

WSGIApplicationGroup %{GLOBAL} <--- 这里添加

• 重新载入配置文件 apachectl graceful

物理机访问192.168.1.10进入图形界面

用户名和密码在openstack节点的root下有个

cat /root/keystonerc\_admin

unset OS\_SERVICE\_TOKEN

export OS\_USERNAME=admin

export OS\_PASSWORD=0a7387d2f7eb4bf9

项目管理

基本概念

• 项目:一组隔离的资源和对象。由一组关联的用户迚行管理

• 在老的版本里,也用租户(tenant)来表示

• 根据配置的需求,项目对应一个组织、一个公司戒是一个使用客户等

• 项目中可以有多个用户,项目中的用户可以在该项目创建、管理虚资资源

• 具有admin角色的用户可以创建项目

• 项目相关信息保存到MariaDB中

• 缺省情况下,packstack安装的openstack中有两个

独立的项目

– admin:为admin账户创建的项目

– services:不安装的各个服务相关联

创建一个tedu项目，创建一个u1用户

项目里面，点镜像，创建镜像 small.img

镜像格式qcow2-qemu模拟器

命令行接口基础

• 初始化环境变量

[root@vh02 ~]# source ~/keystonerc\_admin

[root@vh02 ~(keystone\_admin)]# env | grep OS

HOSTNAME=vh02.tedu.cn

OS\_REGION\_NAME=RegionOne

OS\_PASSWORD=redhat

OS\_AUTH\_URL=http://192.168.4.2:5000/v2.0

OS\_USERNAME=admin

OS\_TENANT\_NAME=admin

• 使用帮助

[root@vh02 ~(keystone\_admin)]# openstack help

通过命令行管理项目

• 创建名为myproject项目

[root@vh02 ~(keystone\_admin)]# openstack project create myproject

• 列出所有项目

[root@vh02 ~(keystone\_admin)]# openstack project list

• 查看myproject详细信息

[root@vh02 ~(keystone\_admin)]# openstack project show myproject

• 禁用不激活项目

[root@vh02 ~(keystone\_admin)]# openstack project set --disable

myproject

[root@vh02 ~(keystone\_admin)]# openstack project set --enable

myproject

图形界面，管理员-系统-云主机类型