Openstack搭建简单应用

简介：

Openstack版本更替块每个版本的命令都有所不同（版本：3.2.1 系统centos7.4）

IaaS云：基础设施及服务，提供给用户计算资源CPU，内存，存储，网络等。

Openstack一套IaaS云的解决方案，开源云计算管理平台。

Openstack主要组建：

Horizon：基于web管理接口，管理openstack的各种服务

Keystone：为其他服务提供了集中管理认证授权的服务

Neutron：软件定义网络，用于创建虚拟交换虚拟路游等

Cinder：管理虚拟机的存储卷，提供永久的块存储

Nova：分布式服务管理虚拟机的服务，可水平扩展

Glance：用于存创建江管理虚拟机的镜像

Openstack实验基础环境搭建

物理机：ip 192.168.1.254 ntp服务器 ftp发布yum源 开启路由转发作网关

Dns服务器 后端盘模板：（1. 禁用 selinux2. 卸载 firewalld3. 卸载 NetworkManager4. 配置主机IP地址5. 配置主机yum客户端6. 创建cinder-volumes的卷组7. 系统yum导入公钥8. 安装依赖的软件包）

虚拟机1（计算节点）：1cpu 8G内存 50G+20G存储 eth0 192.168.1.1

Eth1 192.168.4.1 主机名：openstack

虚拟机2（存储节点）：1cpu 6G内存 50G存储 eth0 192.168.1.2

Eth1 192.168.4.2 主机名：nova

部署物理机：

一、安装ftp部署yum源(CentOS7-1708.iso RHEL7-extra.iso RHEL7OSP-10.iso)三个官网提供镜像，有12个yum源

Vim /etc/fstab/root/桌面/包/CentOS7-1708.iso /var/ftp/centos7/ iso9660 defaults 0 0

/root/桌面/包/RHEL7-extras.iso /var/ftp/RHEL7-ectras/ iso9660 defaults 0 0

/root/桌面/包/RHEL7OSP-10.iso /var/ftp/openstack/ iso9660 defaults 0 0

Mount -a

Yum -y install vsftpd

安装kvm虚拟化环境：

yum -y install qemu-kvm(仿真支持) libvirt-daemon（守护进程管理虚拟机） libvirt-client（用户端软件提供管理命令） libvirt-daemon-driver-qemu（提供连接qemu的驱动）

创建虚拟网络（类似交换机）

Vim vbr.xml 定义虚拟网络网段dhcp （uuid可删除自动生成）

<network>

<name>vbr</name>

<uuid>891d4bc4-9080-471e-aba6-c42212e182e6</uuid>

<forward mode='nat'/>

<bridge name='vbr' stp='on' delay='0'/>

<mac address='52:54:00:a3:04:d3'/>

<ip address='192.168.1.254' netmask='255.255.255.0'>

<dhcp>

<range start='192.168.1.100' end='192.168.1.200'/>

</dhcp>

</ip>

</network>

Vim private1.xml

<network>

<name>private1</name>

<uuid>559de4a7-724c-4eec-ba19-3c7c2e69b4ad</uuid>

<bridge name='private1' stp='on' delay='0'/>

<mac address='52:54:00:37:78:11'/>

<domain name='private1'/>

<ip address='192.168.4.254' netmask='255.255.255.0'>

</ip>

</network>

Virsh net-define vbr.xml

Virsh net-define private1.xml

Virsh net-autostart vbr

Virsh net-autostart private1

#定义网络设置开机启动，其中vbr支持nat。

三、制作后端盘模板

[root@nsd1803 images]# qemu-img create -f qcow2 num.qcow2 10G

安装时系统选择使用存储num.qcow2，软件选择 minimal ,关闭 KDUMP，英文模式，使用标准分区格式只分一个根分区。

1、禁用 selinux /etc/selinux/config

SELINUX=disabled

卸载防火墙NetworkManager

Yum -y remove NetworkManager-\* \*firewall\*

配置yum源

[root@openstack ~]# cat /etc/yum.repos.d/centos.repo

[centos]

name=centos7

baseurl=ftp://192.168.1.254/centos7/

enabled=1

gpgcheck=1

[extras]

name=extras

baseurl=ftp://192.168.1.254/RHEL7-ectras/

enabled=1

gpgcheck=0

[rhel-7-server-openstack-10-devtools-rpms]

name=rhel-7-server-openstack-10-devtools-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-openstack-10-devtools-rpms

enabled=1

gpgcheck=0

[rhel-7-server-openstack-10-optools-rpms]

name=rhel-7-server-openstack-10-optools-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-openstack-10-optools-rpms

enabled=1

gpgcheck=0

[rhel-7-server-openstack-10-rpms]

name=rhel-7-server-openstack-10-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-openstack-10-rpms

enabled=1

gpgcheck=0

[rhel-7-server-openstack-10-tools-rpms]

name=rhel-7-server-openstack-10-tools-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-openstack-10-tools-rpms

enabled=1

gpgcheck=0

[rhel-7-server-rhceph-2-mon-rpms]

name=rhel-7-server-rhceph-2-mon-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-rhceph-2-mon-rpms

enabled=1

gpgcheck=0

[rhel-7-server-rhceph-2-osd-rpms]

name=rhel-7-server-rhceph-2-osd-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-rhceph-2-osd-rpms

enabled=1

gpgcheck=0

[rhel-7-server-rhceph-2-tools-rpms]

name=rhel-7-server-rhceph-2-tools-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-rhceph-2-tools-rpms

enabled=1

gpgcheck=0

[rhel-7-server-rhscon-2-agent-rpms]

name=rhel-7-server-rhscon-2-agent-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-rhscon-2-agent-rpms

enabled=1

gpgcheck=0

[rhel-7-server-rhscon-2-installer-rpms]

name=rhel-7-server-rhscon-2-installer-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-rhscon-2-installer-rpms

enabled=1

gpgcheck=0

[rhel-7-server-rhscon-2-main-rpms]

name=rhel-7-server-rhscon-2-main-rpms

baseurl=ftp://192.168.1.254/openstack/rhel-7-server-rhscon-2-main-rpms

enabled=1

gpgcheck=0

Centos系统光盘要求导入 gpg=1key验证

rpm - -import <ftp://192.168.1.254/centos7/RPM-GPG-KEY-CentOS-7>

4.安装常用命令

yum install -y net-tools vim-enhanced bridge-utils psmisc

5.配置网卡，禁用空路由

vim /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE="eth0"

ONBOOT="yes“

IPV6INIT="no"

TYPE="Ethernet"

BOOTPROTO=“dhcp“

禁用空路由

/etc/sysconfig/network

NOZEROCONF="yes

添加物理机通过console连接虚拟机

Vim /etc/default/grub

GRUB\_CMDLINE\_LINUX="biosdevname=0 net.ifnames=0

console=ttyS0,115200n8"

GRUB\_DISABLE\_LINUX\_UUID="true"

GRUB\_ENABLE\_LINUX\_LABEL="true“

grub2-mkconfig -o /boot/grub2/grub.cfg 生效去个性化vim /etc/fstab将uuid改为路径

安装分区扩展软件，因为后端盘大小设置10G，后面的前端盘分区大小会被限制安装软件后可扩展。

yum install -y cloud-utils-growpart

关闭虚拟机对后端盘的及其信息清理工作

Virt-sysprep -d 虚拟机名

保护后端盘

Virsh undefine 虚拟机名

四、创建虚拟机1和虚拟机2（根据虚拟机1制作）

创建虚拟机存储镜像文件，利用后端模板

Qemu-img create -b num.qcow2 -f qcow2 num1.img 50G

Qemu-img create -f qcow2 num1-1.img 50G

Qemu-img create -b num.qcow2 -f qcow2 num2.img 50G

准备xml文件直接定义创建虚拟机

Vim num1.xml 文件中定义了内存硬盘cpu网卡规格，uuid alias address等个性化标签可以删除

<!--

WARNING: THIS IS AN AUTO-GENERATED FILE. CHANGES TO IT ARE LIKELY TO BE

OVERWRITTEN AND LOST. Changes to this xml configuration should be made using:

virsh edit num1

or other application using the libvirt API.

-->

<domain type='kvm'>

<name>num1</name>

<uuid>04597c4f-8c7f-49c4-9bef-5b0f9e5cea07</uuid>

<memory unit='KiB'>8388608</memory>

<currentMemory unit='KiB'>8388608</currentMemory>

<vcpu placement='static'>2</vcpu>

<os>

<type arch='x86\_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>

<boot dev='hd'/>

<bootmenu enable='yes'/>

<bios useserial='yes'/>

</os>

<features>

<acpi/>

<apic/>

</features>

<cpu mode='host-passthrough' check='none'/>

<clock offset='localtime'/>

<on\_poweroff>destroy</on\_poweroff>

<on\_reboot>restart</on\_reboot>

<on\_crash>restart</on\_crash>

<devices>

<emulator>/usr/libexec/qemu-kvm</emulator>

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

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

<source file='/var/lib/libvirt/images/num1.img'/>

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

<address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>

</disk>

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

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

<source file='/var/lib/libvirt/images/num1-1.img'/>

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

<address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>

</disk>

<controller type='usb' index='0' model='piix3-uhci'>

<address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>

</controller>

<controller type='pci' index='0' model='pci-root'/>

<controller type='virtio-serial' index='0'>

<address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>

</controller>

<interface type='bridge'>

<mac address='52:54:00:d2:46:94'/>

<source bridge='vbr'/>

<model type='virtio'/>

<address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>

</interface>

<interface type='bridge'>

<mac address='52:54:00:f0:1c:ab'/>

<source bridge='private1'/>

<model type='virtio'/>

<address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>

</interface>

<serial type='pty'>

<target port='0'/>

</serial>

<console type='pty'>

<target type='serial' port='0'/>

</console>

<channel type='unix'>

<target type='virtio' name='org.qemu.guest\_agent.0'/>

<address type='virtio-serial' controller='0' bus='0' port='1'/>

</channel>

<input type='mouse' bus='ps2'/>

<input type='keyboard' bus='ps2'/>

<memballoon model='virtio'>

<address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>

</memballoon>

</devices>

</domain>

Virsh define num1.xml

Virsh start num1 #定义启动虚拟机

连接虚拟机扩展分区大小突破后端盘限制，配置网卡，修改主机名

Virsh console num1

growpart /dev/vda 1 扩展根分区空间大小

xfs\_growfs / 扩展根分区文件系统大小

DEVICE="eth0"

ONBOOT=yes

IPV6INIT=no

BOOTPROTO=static

TYPE=Ethernet

IPADDR=192.168.1.1

PREFIX=24

GATEWAY=192.168.1.254

DEVICE="eth1"

ONBOOT=yes

IPV6INIT=no

BOOTPROTO=static

TYPE=Ethernet

IPADDR=192.168.4.1

PREFIX=24

Systemctl restart network

Hostname set-hostname openstack

设置ntp服务器客户端，dns服务器客户端

物理机（作为转发dns服务器）：

Yum -y install bind-chroot chronyd

Vim /etc/named.conf

options {

listen-on port 53 { 192.168.1.254; };

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

directory "/var/named";

dump-file "/var/named/data/cache\_dump.db";

statistics-file "/var/named/data/named\_stats.txt";

memstatistics-file "/var/named/data/named\_mem\_stats.txt";

allow-query { any; };

recursion yes;

forwarders { 202.106.196.115; };

dnssec-enable no;

dnssec-validation no;

文件截取部分可根据真实环境配置。

Vim /etc/chrony.con

server ntp1.aliyun.com iburst

bindacqaddress 0.0.0.0

allow 0/0

local stratum 10

验证同步阿里的效果:chronyc sources -V \*号即正常

Systemctl enable named chronyd

Systemctl restart named chronyd

虚拟机：将dns的文件reslove和chrony.conf修改测试时间同步和dns解析

Openstack搭建前基本环境完成

一．Openstack虚拟机1环境部署

1.配置卷组为cinder组建提供该版本逻辑卷名必须为cinder-volumes

2.安装额外依赖包

yum install -y qemu-kvm libvirt-client libvirt-daemon

libvirt-daemon-driver-qemu python-setuptools

安装自动部署工具

Yum -y install openstack-packstack

packstack --gen-answer-file answer.ini

编辑 answer.ini

配置默认密码

11: CONFIG\_DEFAULT\_PASSWORD=Taren1

设置禁用 swift (对象存储) 模块

42: CONFIG\_SWIFT\_INSTALL=n

NTP 服务器地址

75: CONFIG\_NTP\_SERVERS=192.168.1.254

计算节点IP地址

98: CONFIG\_COMPUTE\_HOSTS=192.168.1.1，192.168.1.2

需要配置vxlan网络的 IP 地址

102: CONFIG\_NETWORK\_HOSTS=192.168.1.1，192.168.1.2

禁用自动创建 cinder-volumns 卷组

554: CONFIG\_CINDER\_VOLUMES\_CREATE=n

设置网络支持协议

840: CONFIG\_NEUTRON\_ML2\_TYPE\_DRIVERS=flat,vxlan

设置组播地址

876: CONFIG\_NEUTRON\_ML2\_VXLAN\_GROUP=239.1.1.5

设置虚拟交换机

910: CONFIG\_NEUTRON\_OVS\_BRIDGE\_MAPPINGS=physnet1:br-ex

设置虚拟交换机所连接的物理网卡

921: CONFIG\_NEUTRON\_OVS\_BRIDGE\_IFACES=br-ex:eth0

设置隧道网络使用的网卡

936: CONFIG\_NEUTRON\_OVS\_TUNNEL\_IF=eth1

禁用测试的DEMO

1179:CONFIG\_PROVISION\_DEMO=n

packstack --answer-file=answer.in 安装openstack

horizon 软件配置 bug

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

添加 WSGIApplicationGroup %{GLOBAL}

重新载入配置 apachectl graceful

使用浏览器连接初始帐号密码在/root/keystonerc\_admin

1. 命令行操控较为重要的（每个版本都会变化）

初始化环境变量

source ~/keystonerc\_admin

Openstack help # 查看命令帮助

Openstack user set --paswword 密码 帐户名 #修改用户密码

Openstack image save --file=/名 #保存镜像到本地

1. openstack简单使用web

openstack用户：创建用户加入到不同项目中

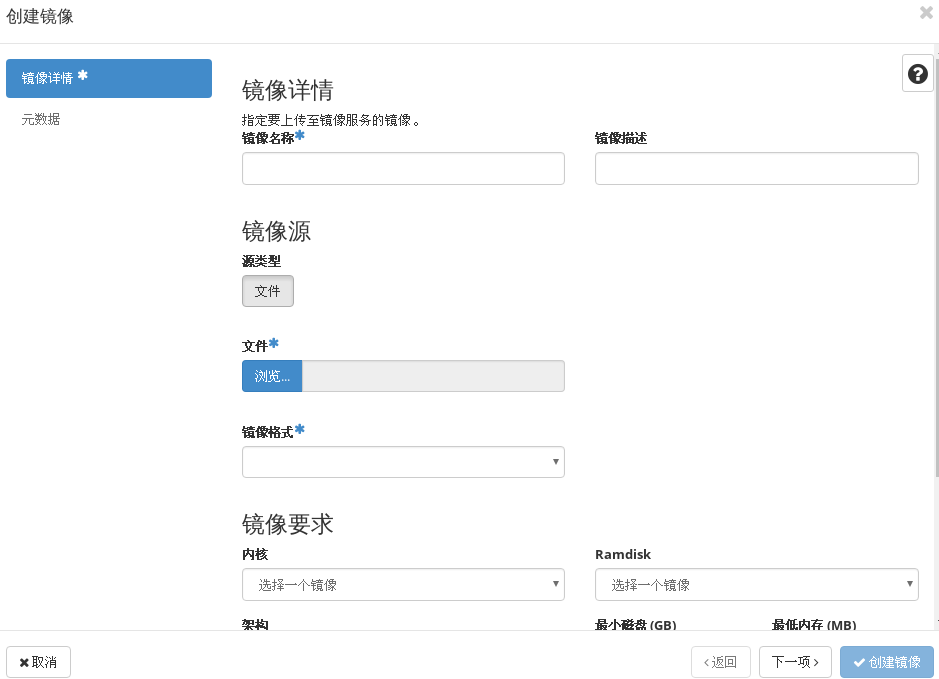
Openstack项目：一个项目对应一个公司，可分配计算资源配额





创建云主机的xml文件

创建镜像模板



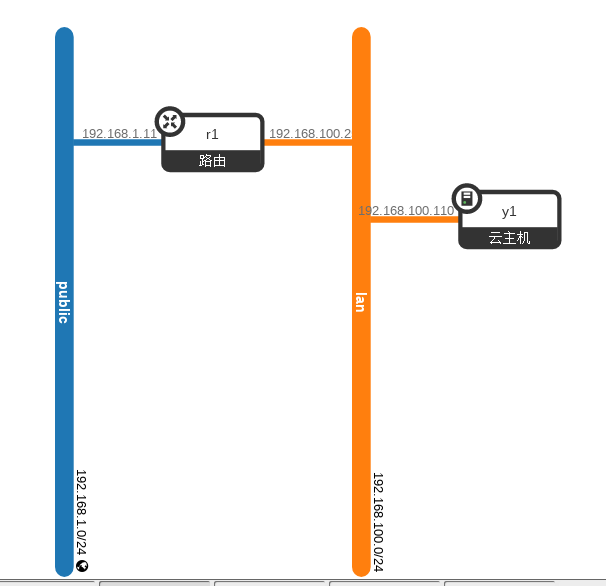
创建网络即公网的虚拟交换机：其中设备写phynet1为先前在自动应答文件中指定网段为主机ip192.168.1.0/24

登陆一个项目

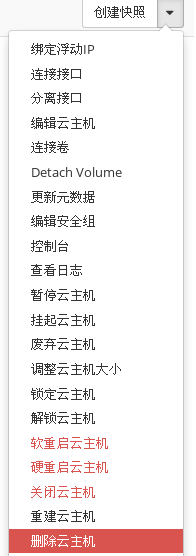
使用普通用户创建内部虚拟交换机:



创建路由和子接口：连通二者

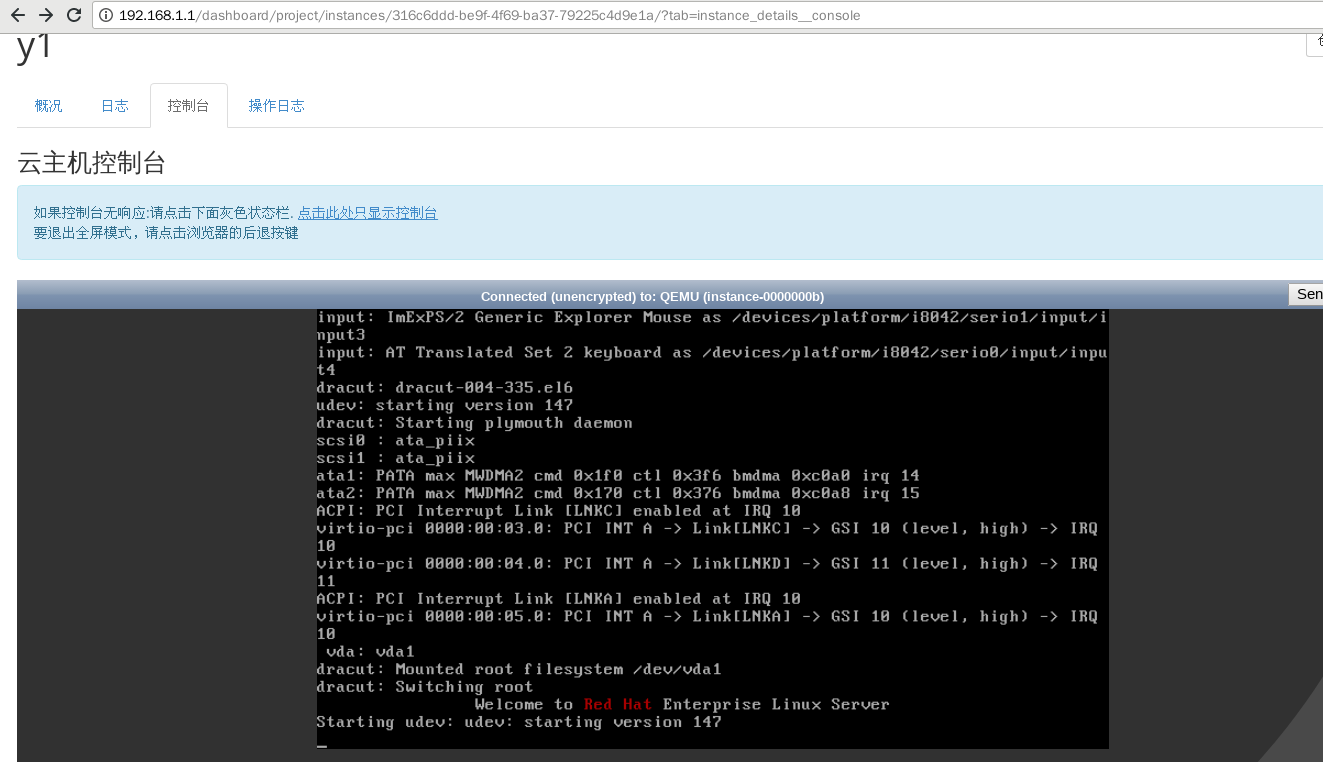


创建云主机：基于之前上传的镜像

云主机绑定浮动ip就是公网ip

云主机配置安全组允许入站策略

控制台连接云主机



虚拟机可以连通外网