需要编译安装的软件包要批量安装的时候，可以先编译安装一台，然后将源码打包成RPM包批量安装

vim /etc/vsftpd/vsftpd.conf

listen=yes

listen\_ipv=NO

cd ~ftp //ftp用户的家目录

验证ftp服务

yum -y install lftp

lftp 192.168.1.10 //默认连接ftp服务、

lftp 192.168.1.10 -u username //如果有用户名密码

lftp sftp://192.168.1.10 -u username //默认连接ssh服务

>put //上传，默认没有权限

>get //下载

创建自定义yum源

mkdir -p /var/ftp/public

cp xx.rpm /var/ftp/public

cd /var/ftp/public

createrepo . //创建

createrepo --update . //如果之前已经有了，更新

客户端配置

[local\_soft]

name=local soft

baseurl=ftp://192.168.1.10/public

enabled=1

gpgcheck=0

转发DNS安装配置

yum -y install bind bind-chroot

vim /etc/named.conf

listen-on port 53 { 192.168.1.10; };

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

allow-query { any; };

forwarders { 202.106.196.115; };

dnssec-enable no;

dnssec-validation no;

时间服务器，server、 client

yum install -y chrony

server:配置 /etc/chrony.conf

server ntp1.aliyun.com iburst

bindacqaddress 0.0.0.0

allow 0/0

检查状态

chronyc sources -v

\* 同步成功，+ 备胎 , ? 同步失败

KVM（内核模块）/QEMU（仿真工具）/LIBVIRTD（管理工具）

虚拟机组成

1、配置文件(虚拟机配置声明文件) /etc/libvirt/qemu/

2、镜像文件(虚拟机硬盘) /var/lib/libvirt/images

virsh nodeinfo //查看服务器信息

virsh list

virsh list --all //列出所有虚拟机

virsh net-list //列出虚拟网络

virsh net-list --all

virsh dominfo node1 //查看指定虚拟机的信息

virsh start node1

virsh start|reboot|shutdown node1

virsh destroy node1 //强制关机

virsh autostart|disable node1

echo $LANG

LANG=zh\_CN.UTF-8

创建新的镜像盘文件

qemu-img create -f 格式 磁盘路径 大小

qemu-img create -f qcow2 disk.img 50G

cd /var/lib/libvirt/images

qemu-img create -f qcow2 testnode.qcow2 8G

查询镜像盘文件的信息

qemu-img info 磁盘路径

qemu-img info disk.img

(backing file: disk.img，代表是前端盘，指向后端盘)

-b 使用后端模板文件

qemu-img create -b disk.img -f qcow2 disk1.img

后端盘 前端盘

不指定大小，创建的前端盘和后端盘一样大

COPY ON WRITE 写时复制

– 直接映射原始盘的数据内容

– 当原始盘的旧数据有修改时,在修改之前自动将旧数据存入前端盘

– 对前端盘的修改会回写到原始盘

本地连接

virsh

连接远程

virsh -c qemu+ssh://user@ip.xx.xx.xx:port/system

使用 virt-manager 客户端工具

需要注意的是 virt-manager 需要使用 ssh 免密码登录如果没有免密码登录需要单独安装 python 的 ssh 相关模块,这里我们可以使用部署 key 的方法解决

安装虚拟化服务器平台必备软件

1、qemu-kvm

为 kvm 提供底层仿真支持

2、libvirt-daemon

libvirtd 守护进程,管理虚拟机

3、libvirt-client

用户端软件,提供客户端管理命令

4、libvirt-daemon-driver-qemu

libvirtd 连接 qemu 的驱动

-------------------------------------------------------------------------------

真机创建虚拟交换机 （NAT上网）

1.vim /etc/libvirt/qemu/networks/vbr.xml

<network>

<name>vbr</name>

<bridge name="vbr"/>

<forward mode="nat"/> //代表可以上网

<ip address="192.168.1.254" netmask="255.255.255.0"> //设备IP、子网掩码

<dhcp>

<range start="192.168.1.100" end="192.168.1.200"/> //DHCP分配IP范围

</dhcp>

</ip>

</network>

virsh管理虚拟网络 （真机上操作）

cd /etc/libvirt/qemu/networks

net-list (--all)

2.net-define vbr.xml 创建虚拟网络

net-undefine vbr 删除虚拟网络

3.net-start vbr 启动虚拟网络

net-destroy vbr 停止虚拟网络

net-edit vbr 修改 vbr 网络的配置

4.net-autostart vbr 设置 vbr 虚拟网络开机自启动

-------------------------------------------------------------------------------

虚拟机管理

xml配置文件

定义了一个虚拟机的名称、CPU、内存、虚拟磁盘、

网卡等各种参数设置

默认位于 /etc/libvirt/qemu/虚拟机名.xml

导出xml配置文件 dumpxml

查看:virsh dumpxml 虚拟机名

备份:virsh dumpxml 虚拟机名 > 虚拟机名.xml

如：virsh dumpxml node1 > node1.xml

编辑虚拟机设置

对虚拟机的配置进行调整

编辑：virsh edit 虚拟机名

如：virsh edit node1

若修改 name、memory、disk、network,可自动保存为新虚拟机配置

定义虚拟机

根据修改后的独立xml文件定义新虚拟机

virsh define XML描述文件

删除虚拟机

virsh undefine 虚拟机名

自定义虚拟机

自定义虚拟机安装

-------------------------------------------------------------------------------

一、virt-manager安装虚拟机（后端盘）

• 启动 virt-manager 软件,选择新建虚拟机

• 选择网络安装源

– qemu-img create -f qcow2 testnode.qcow2 16G

2048M内存 2CPU /var/lib/libvirt/images/testnode.qcow2

• 网络选择自定义的 vbr

– 软件选择 minimal ,关闭 KDUMP

– 选择手工分区 //为以后扩展做准备

• 手动分区

• 使用标准分区格式,只分一个根分区

• 选取一个根分区,开启安装系统

二、虚拟机模板制作：（全在虚拟机操作，后端盘）

1、配置网络 yum 源，使用CentOS7-1708.iso

新装的虚拟机安装网络yum源ftp://192.168.1.254/CentOS7/

yum 源导入公钥验证配置：//和openstack冲突

gpgcheck=1

导入 gpg key

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

yum provides ifconfig //查询某个命令在哪个包里

安装常用工具包

2、禁用selinux

3、卸载防火墙、NetworkManager

1）先卸载防火墙yum remove -y firewalld-\* python-firewall

2）先修改配置文件，再卸载NetworkManager

删除网络配置里的个性化信息做到模板通用，如：MAC地址，IP，UUID

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

DEVICE=eth0

ONBOOT=yes

IPV6INIT=no

TYPE=Ethernet

BOOTPROTO=dhcp

systemctl restart network

yum remove -y NetworkManager-\* //和openstack冲突

4、禁用空路由

/etc/sysconfig/network

NOZEROCONF=yes

5、virsh console 连不上虚拟机：

1）vim /etc/sysconfig/grub

GRUB\_CMDLINE\_LINUX="biosdevname=0 net.ifnames=0 console=ttyS0,115200n8" //修改，打开串口终端GRUB\_DISABLE\_LINUX\_UUID="true" //添加

GRUB\_ENABLE\_LINUX\_LABEL="true" //添加

2）重新生成 grub.cfg

grub2-mkconfig -o /boot/grub2/grub.cfg

6、变成磁盘卷标挂载，而不是UUID

blkid

vim /etc/fstab

/dev/vda1 / xfs defaults 0 0

7、安装分区扩展软件

yum install -y cloud-utils-growpart

可以扩容根分区

设置第一次开机自动扩容根目录

chmod 755 /etc/rc.local

在 rc.local 里加入如下配置

###也可以手动执行以下两行 设置LANG＝en\_US.UTF-8

/usr/bin/growpart /dev/vda 1

/usr/sbin/xfs\_growfs /

/usr/bin/sed '/^###/,$d' -i /etc/rc.d/rc.local //只执行一次就删除该行

8、关闭虚拟机后在物理机上执行信息清理工作

virt-sysprep -d testcentos7（虚拟机名字）执行完这一步可以用console连了

三、快速创建虚拟机 （前端盘）

virsh undefine testcentos7 //取消定义虚拟机

cd /var/lib/libvirt/images/

创建前端文件

qemu-img create -f qcow2 -b testnode.qcow2 testnode1.img 30G

创建XML配置文件 根据模板XML文件demo.xml生成虚拟机XML文件

sed 's/demo/testnode1/' demo.xml > /etc/libvirt/qemu/testnode1.xml

虚拟机名 虚拟机名

将模板文件中demo替换成虚拟机名

virsh define /etc/libvirt/qemu/testnode1.xml

virsh start testnode1

virsh console testnode1

错误：操作失败: 这个域有活跃控制台会话

解决：

virsh

# shutdown testnode1

# list

# start testnode1

如果还不行就关闭virt-manager软件

-------------------------------------------------------------------------------

XML详解

virsh dumpxml testnode >aa.xml

<domain type='kvm' id='4'>

<name>centos7.0</name> //虚拟机名称

<uuid>8413f30a-c978-4796-a28f-8ba9fe74b759</uuid>

<memory unit='KiB'>2097152</memory> //内存

<currentMemory unit='KiB'>2097152</currentMemory>

– id=4 个性化设置 ,去掉

– uuid 去掉

– memory unit 虚拟机最大使用内存,可以手动调整 ？＊1024

– currentmemory 创建虚拟机使用内存，一般和上面值一样

<vcpu placement='static'>2</vcpu> //cpu数量,可以调整

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

– 重启,关机,强制关闭对应命令

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

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

<on\_crash>destroy</on\_crash>

– 仿真设备相关配置

<devices> ... ...

– 硬盘配置,需要修改

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

<disk ... ... </disk>

</devices>

– 其他配置

– 驱动安全标签,可删除

<seclabel> ... ... </seclabel>

– 可以添加硬盘，首先创建硬盘，复制<disk>

virsh edit 虚拟机

创建硬盘

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

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

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

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

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

</disk>

添加完重启 **poweroff**

– 网卡，可以添加网卡

virsh edit 虚拟机

ip link show 查看网卡

<interface type='bridge'>

<source bridge='private1'/> //从上到下，eth0、eth1，和哪个虚拟交换机相连

<model type='virtio'/>

</interface>

添加完重启 **poweroff**

ifconfig -a

cp ifcfg-eth0 ifcfg-eth1

vim ifcfg-eth1

DEVICE=eth1

ONBOOT=yes

IPV6INIT=no

TYPE=Ethernet

BOOTPROTO=dhcp

systemctl restart network

挂载虚拟盘

使用 guestmount 工具,离线修改前、后端盘

注意：如果修改了后端盘，所有引用后端盘的前端盘全部要删除重建

挂载虚拟盘:

guestmount -a /var/lib/libvirt/images/node1.img -i /mnt

-i (将系统挂载到一个空的文件夹)

ls /mnt

在/mnt修改完umount /mnt

chroot /mnt //一定要修改根目录，否则执行以下命令会在真机执行

ifconfig 报错就是虚拟机

grub2-mkconfig -o /boot/grub2/grub.cfg

设备 bus（总线）

hda pci 最差

sda scsi 一般用于服务器上

vda virtio 虚拟化专用