# 用kvm安装虚拟机

192.168.200.146(centos6.8):源+ks文件:

源: /var/www/html/os/centos6.8

ks文件: /var/www/html/ks/ centos6.8.cfg

宿主机(centos7):192.168.200.180

要安装的虚拟机(centos6.8):

eth0:192.168.200.227

# 安装准备

## 2.1源的配置

192.168.200.146上:

挂载光盘到/mnt下

mount /dev/sr0 /mnt

cp /mnt/\* /var/www/html/os/centos6.8

## 2.2ks的配置

192.168.200.146上:

vim /var/www/html/ks/ centos6.8.cfg

# ks文件

## 3.1centos6.8.cfg

|  |
| --- |
| #platform=x86, AMD64, or Intel EM64T  # System authorization information  auth --useshadow --enablemd5  # System bootloader configuration  bootloader --location=mbr  # Clear the Master Boot Record  zerombr  # Partition clearing information  clearpart --all --initlabel  # Use text mode install  text  # Firewall configuration  firewall --disabled  # Run the Setup Agent on first boot  firstboot --disable  # System keyboard  keyboard us  # System language  lang en\_US  # Installation logging level  logging --level=info  # Use network installation  #url --url=ftp://172.16.1.10  #指定源和virt-install中-l 指定源一样  url --url=<http://192.168.200.146/os/centos6.8>  #这个网卡是要能访问源的,如果不配置,则默认是自动获取ip地址  network --bootproto=static --device=eth0 --ip=192.168.200.227 --netmask=255.255.255.0 --onboot=on --hostname SN413  #network --bootproto=dhcp --device=eth0 --onboot=on  #Root password  rootpw 123456  reboot  # SELinux configuration  selinux --disabled  # Do not configure the X Window System  skipx  # System timezone  timezone Asia/Shanghai  # Install OS instead of upgrade  install  # Disk partitioning information  part /boot --fstype="ext4" --size=512  part / --fstype="ext4" --size=4000  part swap --fstype="swap" --size=512  %packages  @core  %end  %post  #!/bin/bash  mkdir /data  #关闭iptables  service iptables stop  chkconfig --level 2345 iptables off  %end |

## 3.2centos7.0.cfg

|  |
| --- |
| #version=DEVEL  # System authorization information  auth --enableshadow --passalgo=sha512  # Install OS instead of upgrade  install  # Do not configure the X Window System  skipx  # Use text install  text  # Keyboard layouts  keyboard --vckeymap=us --xlayouts='us'  # System language  lang en\_US.UTF-8  #指定源  url --url=http://192.168.200.146/os/centos7.0  # Network information  network --bootproto=static --device=eth0 --ip=192.168.200.226 --netmask=255.255.255.0 --gateway=192.168.200.96 --onboot=on  #network --bootproto=static --device=ens34 --ip=172.16.4.21 --netmask=255.255.0.0 --onboot=on  network --hostname=VM5  # Root password  rootpw 123456  reboot  # System services  services --disabled="chronyd"  # System timezone  timezone Asia/Shanghai --isUtc --nontp  # System bootloader configuration  bootloader --location=mbr  # Clear the Master Boot Record  zerombr  # Partition clearing information  clearpart --all --initlabel  # Firewall configuration  firewall --disabled  # SELinux configuration  selinux --disabled  # Disk partitioning information  part /boot --fstype="xfs" --size=512  part / --fstype="xfs" --size=8096  part swap --fstype="swap" --size=512  #part pv.01 --fstype='lvmpv' --grow --size=1 --ondisk=sdb  #volgroup vg01 --pesize=4096 pv.01  %packages  @core  #%addon com\_redhat\_kdump --disable --reserve-mb='auto'  %end |

# virt-install安装过程

**yum install qemu-kvm qemu-img virt-install**

## 4.1virt-install

1. 在宿主机上建立网桥,目的是虚拟机将会桥接,通过此桥

宿主机(192.168.200.180):

enp1s0:网卡

brenp1s0:对应的网桥



|  |
| --- |
| virt-install -n lw\_2 -r 612 --vcpus=1 \  --os-variant=rhel6 \  -l http://192.168.200.146/os/centos6.8 \  --nographics --autostart \  -f /dev/vg01/lv2 \  -w bridge:brenp1s0 \  --extra-args='console=ttyS0 --device=eth0 ip=192.168.200.226 netmask=255.255.255.0 ks=http://192.168.200.146/ks/centos6.8.cfg ksdevice=eth0' --connect qemu:///system |

注意:

|  |
| --- |
| -l <http://192.168.200.146/os/centos6>.8 这个是指定源,其实这里主要目的在安装时提供一个最小的linux内核系统(initrd+vmlinuz+安装程序),当然如果在ks里不指定源,就以这个源为主来安装软件包,如果ks里指定了源则会在安装软件时的源会被ks里url –url=xxx 给覆盖掉  这一行参数有两个作用:  a.提供最小的linux内核系统(本人猜想)  b.作为软件安装的源(和ks里的url –url=xxx是一个作用),ks里指定了源,就以ks为主,没有则以-l 后面指定的源为主,ks的文件还可以作为pxe时使用,那时,没有virt-install来指定源,所以要在ks里指定源,所以综合来将,ks里最好写上源和virt-install 中的 -l 的地址一样.  c.既然提供了源,那么虚拟机肯定要有一个网卡来访问源,所以在ks里必定要设置一个网卡用来访问源,如果不设置网卡,则默认为dhcp来获取ip地址 |

--os-variant=rhel6,这里如果安装centos6.8可以不写这行,如果安装centos7.0必须写--os-variant=rhel7

ip=192.168.200.226 netmask=255.255.255.0 ks=http://192.168.200.146/ks/centos6.8.cfg ksdevice=eth0

这里这个ip是在安装系统后还没有读取ks文件前指定的ip地址,目的是用来访问ks文件,这个ip地址是临时配置的ip地址,最后就消失了,彻底安装后的ip地址是根据ks文件生成而来

的

console=ttyS0 有了这个配置就可以在virsh里面通过console id/虚拟机名称 能连入虚拟机了,如果没有这一行,在安装时,屏幕上不会出现任何信息,而且console连不进去

-n lw\_2 就是虚拟机的名称,不是主机名称啊,这个命令你在virsh里 list就能看到,和vmware里左侧显示的名称是一个类型的

-r 612指定内存为612M

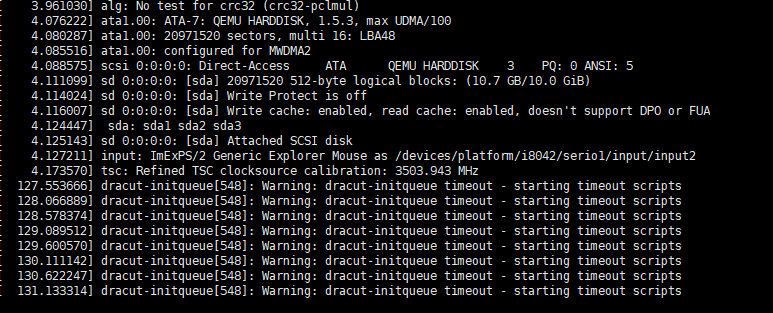
-l 指定源,这个-l必须要有,不然报错

-w bridge:brenp1s0 这里-w 表示第一块网卡并且桥接的是brenp1s0, 如果有多块网卡,则写 在一下行继续写 -w bridge:brenp2s0

--connect qemu:///system这行参数加上和不加上好像并没有区别

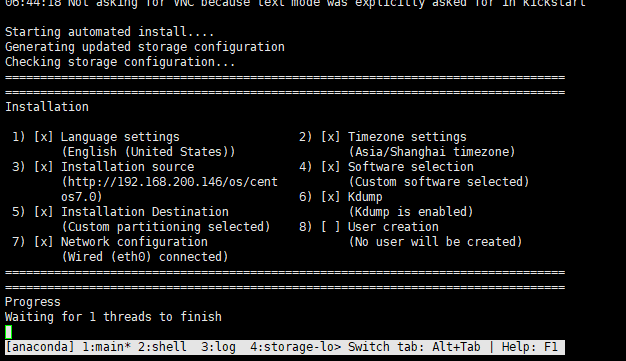
## 4.2安装遇到的问题

问题1.安装centos7.0



virt-install里需要加上参数: --os-variant=rhel7

问题2:安装centos7.0



解决方法:在含有源和ks的服务器上重启httpd 服务

屏幕剪辑

出现这个问题的原因是分给虚拟机的内存不够一般是centos7.3和centos7.5出现的问题,

centos7.3:安装此虚拟机内存必须大于1G

centos7.5: 安装此虚拟机内存必须大于1248M

# kvm和qemu的关系

KVM:运行在内核空间,提供cpu和内存的虚拟化,以及客户机的I/O拦截, Guest 的 I/O 被 KVM 拦截后，交给 QEMU 处理

qemu:修改过的为KVM虚拟机使用的QEMU代码,运行在用户空间,提供硬件I/O虚拟化,通过/dev/kvm和kvm进行交互

# 6.网卡的添加和删除

1. 查看网卡

domiflist lw\_2 #lw\_2为虚拟机名称

1. 动态(临时)添加网卡

attach-interface lw\_2 --type bridge --source brenp1s0 #brenp1s0为宿主机网桥

注意: 此时虚拟机lw\_2会立刻多了个网卡,动态的意思是这个网卡的配置是写在宿主机内存中,没有写入到配置文件中,即虚拟机如果shutdown了,则这个新建立的网卡就消失了

注意:

a.动态添加网卡的前提是虚拟机已经在运行了,不然添加失败

b.虚拟机reboot而不是shutdown,则虚拟机依然还有这个新建的网卡

c.虚拟机如果只有一个网卡,添加成功后不要想当然的认为第二块网卡的名称为eth1,

例子:

|  |
| --- |
| [root@SN413 ~]# dmesg | grep eth  eth0: RTL-8139C+ at 0xffffc9000067e000, 52:54:00:3a:e6:c1, IRQ 11  eth0: link up, 100Mbps, full-duplex, lpa 0x05E1  eth0: no IPv6 routers present  eth1: RTL-8139C+ at 0xffffc9000077a000, 52:54:00:bc:96:fc, IRQ 10  udev: renamed network interface eth1 to eth5 |

此时eth1被重命名为了eth5,所以要在/etc/sysconfig/netwrok-scripts/下新建ifcfg-eth5

1. 静态添加网卡

attach-interface lw\_2 --type bridge --source brenp1s0 –config

注意:

1. 这个操作会改变虚拟机的配置文件即/etc/libvirt/qemu/lw\_2.xml
2. 这个命名会在虚拟机关闭了的情况下生效(此时开机后就有了新建的网卡),即在运行中的虚拟机是没有作用的

3 所以对于运行中的虚拟机来说,要写两条命令来增加网卡(动态和静态),对于关机状态下的虚拟机来说只需要写一条静态增加网卡的命令

4 动态删除网卡(运行状态执行,不然报错)

首先查看

virsh # domiflist lw\_2

Interface Type Source Model MAC

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

vnet0 bridge brenp1s0 rtl8139 52:54:00:3a:e6:c1

vnet1 bridge brenp1s0 rtl8139 52:54:00:df:c9:23

vnet2 bridge brenp1s0 rtl8139 52:54:00:77:74:a1

detach-interface lw\_2 --type bridge --mac 52:54:00:e7:a7:05

5.静态删除网卡(关机下执行)

detach-interface lw\_2 --type bridge --mac 52:54:00:e7:a7:05 –config

注意:这个命令在虚拟机运行时也能执行,但是不会删除内存中网卡,会删除配置文件中的网卡,关机后会全部删除对应的网卡

# 7.硬盘的添加和删除

动态添加硬盘:

attach-disk lw\_2 /dev/vg01/lv1 sda

删除:detach-disk lw\_2 /dev/vg01/lv1

静态添加硬盘:

attach-disk lw\_2 /dev/vg01/lv1 sda --config

删除:detach-disk lw\_2 /dev/vg01/lv1 --config

# 8.系统的引导过程

mbr:分区信息保存到磁盘的第一个扇区(MBR扇区)中的64个字节中，每个分区占用16个字节，这16个字节中存有活动状态标志、文件系统标识、起止柱面号、磁头号、扇区号、隐含扇区数目(4个字节)、分区总扇区数目(4个字节)等内容。

特点:

1.MBR分区主分区数目不能超过4个，很多时候，4个主分区并不能满足需要

2.MBR分区方案无法支持超过2TB容量的磁盘。因为这一方案用4个字节存储分区的总扇区数，最大能表示2的32次方的扇区个数，按每扇区512字节计算，每个分区最大不能超过2TB。磁盘容量超过2TB以后，分区的起止位置也就无法表示了，BIOS将无法识别分区

GPT

：GUID磁碟分割表（GUID Partition Table）的缩写，含义“全局唯一标识磁盘分区表”，是一个实体硬盘的分区表的结构布局的标准。在GTP磁盘的第一个数据块中同样有一个与MBR（主引导记录）类似的标记，叫做PMBR。PMBR的作用是，当使用不支持GPT的分区工具时，整个硬盘将显示为一个受保护的分区，以防止分区表及硬盘数据遭到破坏。UEFI并不从PMBR中获取GPT磁盘的分区信息，它有自己的分区表，即GPT分区表。GPT的分区方案之所以比MBR更先进，是因为在GPT分区表头中可自定义分区数量的最大值，也就是说GPT分区表的大小不是固定的。在Windows中，微软设定GPT磁盘最大分区数量为128个。另外，GPT分区方案中逻辑块地址（LBA）采用64位二进制数表示，可以表示2^64个逻辑块地址。除此之外，GPT分区方案在硬盘的末端还有一个备份分区表，保证了分区信息不容易丢失。

特点:

1.支持2TB以上的大硬盘。

2.每个磁盘的分区个数几乎没有限制。为什么说“几乎”呢？是因为Windows系统最多只允许划分128个分区。

3.分区大小几乎没有限制。因为它用64位的整数表示扇区号。夸张一点说，一个64位整数能代表的分区大小已经是个“天文数字”了，若干年内你都无法见到这样大小的硬盘，更不用说分区了。

4.分区表自带备份。在磁盘的首尾部分分别保存了一份相同的分区表，其中一份被破坏后，可以通过另一份恢复。

5.支持唯一的磁盘和分区 ID，每个分区可以有一个名称(不同于卷标)。



磁盘的第一个扇区(512大小)包含:

1. 主引导分区(mbr),用来安装引导加载程序的地方,446bytes
2. 分区表:记录整个硬盘分区的状态,64bytes

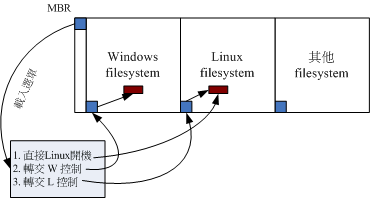
网卡包含ROM(只读存储器):这块存储空间包含PXE Client和tftp client

**操作系统的启动过程:**

1. **bios进行硬件的检测,并依据设置获取第一个可启动的设备(例如从磁盘1启动)**
2. **bios读取一个启动设备的mbr的引导程序boot loader(即是grub等程序):注意bios是借助硬件INT 13来读取mbr的引导程序到到内存中,然后bios执行引导程序**
3. **根据boot loader(grub)的设置加载内核,即/boot/vmlinuz,此时内核不认识磁盘,需要加载磁盘驱动才认识磁盘,但是磁盘驱动需要挂载后才能得到,此时是没有文件系统的,没地方挂载,所以Boot loader会加载/boot/initrd来模拟一个虚拟文件系统并且形成一个根目录，initrd来会在内存中的虚拟文件系统中提供一个程序用来加载和检测内核模块,即硬件驱动,加载硬件驱动后,就又开始进行硬件的检测,最后这个虚拟文件系统会被释放,并挂载实际的根目录文件系统,总结:boot loader加载vmlinuz和initrd**
4. **内核调用init 进程，init执行/etc/rc.d/rc.sysinit这个玩意是不管哪个run-level都要执行的程序**
5. **init 执行对应run-level上的service服务**
6. **init执行/etc/rc.d/rc.local文件**
7. **init执行终端模拟程序mingetty来启动Login进程等待用户的登陆**

# 9多重引导

每个磁盘的第一个扇区的前64字节的地方称为主引导分区mbr(含有引导加载程序boot loader),每个分区的前面有引导扇区(boot sector)(含有含有引导加载程序boot loader)



# 10.PEX网络安装系统

## 10.1流程

1.bios加载网卡的pxe client到内存中并执行pxe client,pxe client先开始自动获取ip地址,此时服务器会收到客户端的回应(即我已经得到ip地址了),服务器会将引导文件的名称(pxelinux.0)和tftp-server的ip地址发送给pxe client

2.pxe client根据tftp-server的ip地址下载引导文件pxelinux.0,然后在执行pxelinux.0进行引导,引导过程首先是下载tftp-server: /var/lib/tftp/下的除了pxelinux.0的其他文件

3.pxelinux.0会读取isolinux.cfg/default文件,根据里面的选项(lable标签) **加载vmlinuz和initrd和额外的参数ks文件路径,然后启动linux(此时网卡是自动获取ip的要和ks文件所在地处于同一个网段),然后下载ks文件,并启动安装程序开始安装,ks文件里指定了源的地址**

## 10.2具体操作

1.客户端先选择从网卡启动(当然这只是测试),安装完了后在改为从硬盘启动

注意:

如果是没有安装系统的电脑,是不用这么设置的,默认会从硬盘启动,发现没有系统,在从网络启动,安装玩操作系统后,因为默认是硬盘启动所以没问题了

2.服务器192.168.1.2身兼三个服务(dhcp + tftp(pxe-server,tftp-server由xinetd控制) + httpd(os+ks))

3.dhcp配置

|  |
| --- |
| allow booting;  allow bootp;  ddns-update-style interim;  ignore client-updates;  subnet 192.168.1.0 netmask 255.255.255.0{    option subnet-mask 255.255.255.0;  #dns服务器  option domain-name-servers 202.101.172.35;  default-lease-time 21600;  max-lease-time 43200;  option routers 192.168.1.2;  range dynamic-bootp 192.168.1.10 192.168.1.20;  next-server 192.168.1.2;  filename "pxelinux.0";  } |

4.httpd的配置请参考 标题1 的2

5.tftp-server的配置

yum install tftp-server syslinux

1. 将/var/www/html/os/centos7.0/\*拷贝到/var/lib/tftpboot下面
2. 到/var/lib/tftpboot下面建立pxelinux.cfg目录,将/var/lib/tftpboot下的isolinux.cfg拷贝到pxelinux.cfg目录下改名为default

default内容:

找到label linux这里:

label linux

menu label ^Install CentOS 7

#这里写default的作用是,安装时会进入到启动菜单,几个label就对应几个菜单项,default是默认选择这一项的意思

menu default

kernel vmlinuz

#注意这里写inst.ks=xxx 和ks=xxx都可以

append initrd=initrd.img inst.ks=http://192.168.1.2/ks/centos7.0.cfg

# append initrd=initrd.img inst.stage2=hd:LABEL=CentOS\x207\x20x86\_64 quiet

从这个菜单选项可以看出来,是先加载vmlinuz 再加载initrd.img 最后加载ks文件的

ks文件的话是先设置语言,键盘,时区等等,在分区,在通过源安装软件,之后安装grub,最后重启

从/var/lib/tftpboot/下有grub.conf.可以判断出pxelinux.0其实就是包装了grub的引导程序