## 1.win7系统qcow2格式镜像制作

### （1）概述

qcow2 格式的镜像获取可以直接网络下载相应系统镜像，也可以自己制作。

**使用qemu通过ISO镜像创建虚拟机，创建qcow2格式磁盘安装虚拟机系统，安装完成后该qcow2格式磁盘就是一个虚拟机的镜像文件。**

### （2）-失败-安装ISO镜像到qcow2磁盘。

安装virt-win驱动。无法直接使用OpenStack安装Windows的ISO镜像到卷的一大部分原因是windows无法直接使用virtio格式硬盘，需要驱动。

yum -y install wget

wget https://fedorapeople.org/groups/virt/virtio-win/virtio-win.repo -O /etc/yum.repos.d/virtio-win.repo

yum -y install virtio-win

安装完成后RPM包内容

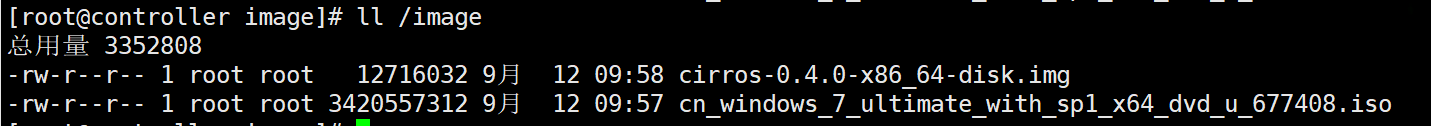
/usr/share/virtio-win/\*.iso: ISO镜像，包含所有驱动。

/usr/share/virtio-win/\*.vfd: 用于Windows XP下的VFD软驱镜像

/usr/share/virtio-win/drivers: 从VFD软驱镜像中提取的文件

/usr/share/guest-agent/\*.msi: QEMU Guest Agent 32bit 和 64bit 安装包

准备镜像到文件夹/image



创建qcow2格式磁盘，指定大小100G。磁盘文件放在/vhost 目录。

mkdir /vhost

qemu-img create -f qcow2 /vhost/win7\_xshell.qcow2 100G

查看磁盘信息：

qemu-img info /vhost/win7\_xshell.qcow2

修改两个目录的权限

chmod 777 /image/\*

chmod 777 /vhost/\*

开始将iso格式的镜像安装到刚刚创建的qcow2格式的文件中：

virt-install --name=win7\_xshell --ram=2048 --vcpus=2 --network network=default,model=virtio --cdrom=/image/cn\_windows\_7\_ultimate\_with\_sp1\_x64\_dvd\_u\_677408.iso --boot menu=on --disk=/vhost/win7\_xshell.qcow2,format=qcow2,bus=virtio --disk=/image/virtio-win-0.1.171.iso,device=cdrom --graphics vnc,listen=0.0.0.0,port=5901 --os-type=windows --os-variant=win7 --force

注意问题：

cdrom和disk都不要设置在/root中，防止没有权限访问；

有两个disk设置，一个是qcow2格式的虚拟机地址，一个是virtio-win驱动的地址；

os-variant需要设置为系统的版本win7；

通过vnc viewer连接虚拟机vnc 进行Win7操作系统的安装。

需一个可远程连接到CentOS7服务器主机的图形化系统，如windows mac安装vnc viewer软件，连接到服务器的相应端口对虚拟机系统进行安装设置。

sudo virt-install --connect qemu:///system -n win7 \

--vcpus=1 -r 2048 \

--disk path=/home/win7.img,format=qcow2,device=disk,size=64,bus=virtio,cache=none \

--disk path=/home/cn\_windows\_7\_ultimate\_with\_sp1\_x64\_dvd\_u\_677408.iso,device=cdrom,perms=rw \

--vnc --vnclisten=0.0.0.0 \

--os-type windows --os-variant=win7 \

--accelerate --network=default,model=virtio \

--disk path=/home/virtio-win-0.1.110.iso,device=cdrom,perms=rw \

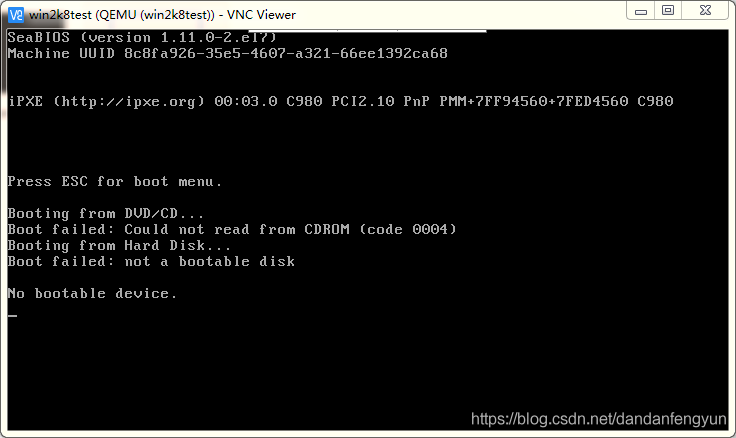
--disk=/home/virtio-win-0.1.110\_amd64.vfd,device=floppy

### （3）-失败-vnc viewer连接虚机

在PC机上安装了vnc viewer后用来连接vnc server，vnc server部署在Controller节点之上，我们在之前通过virt-install安装了qcow2虚机，端口设置为5901，因此我们直接通过vnc viewer连接vnc server即可。

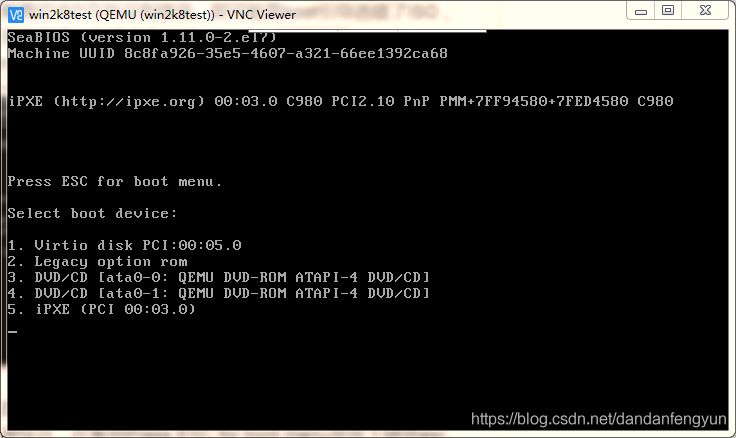


开启连接后可能会出现以下的情况：

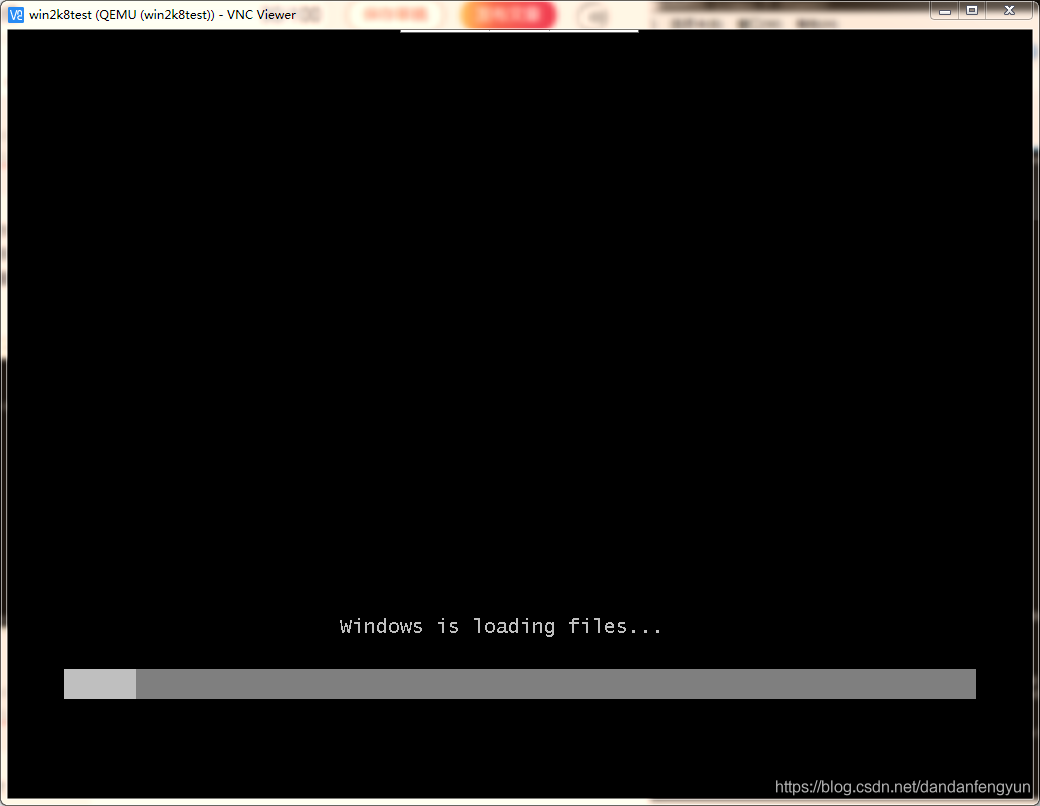


如果确定ISO文件没有错误。那可能是boot引导选错了ISO，需手动选择。右键点击VNC Viewer，然后选择send Ctrl+Alt+Del。然后重新连接。

还是引导错误。再次Ctrl+Alt+Del，这一次不会退出该界面，手速快点，在看到Press ESC for boot menu是按下键盘esc。然后会让手动选择引导盘。（手速慢或者错过了还是会自动引导失败进入该界面，可以重复Ctrl+Alt+Del）

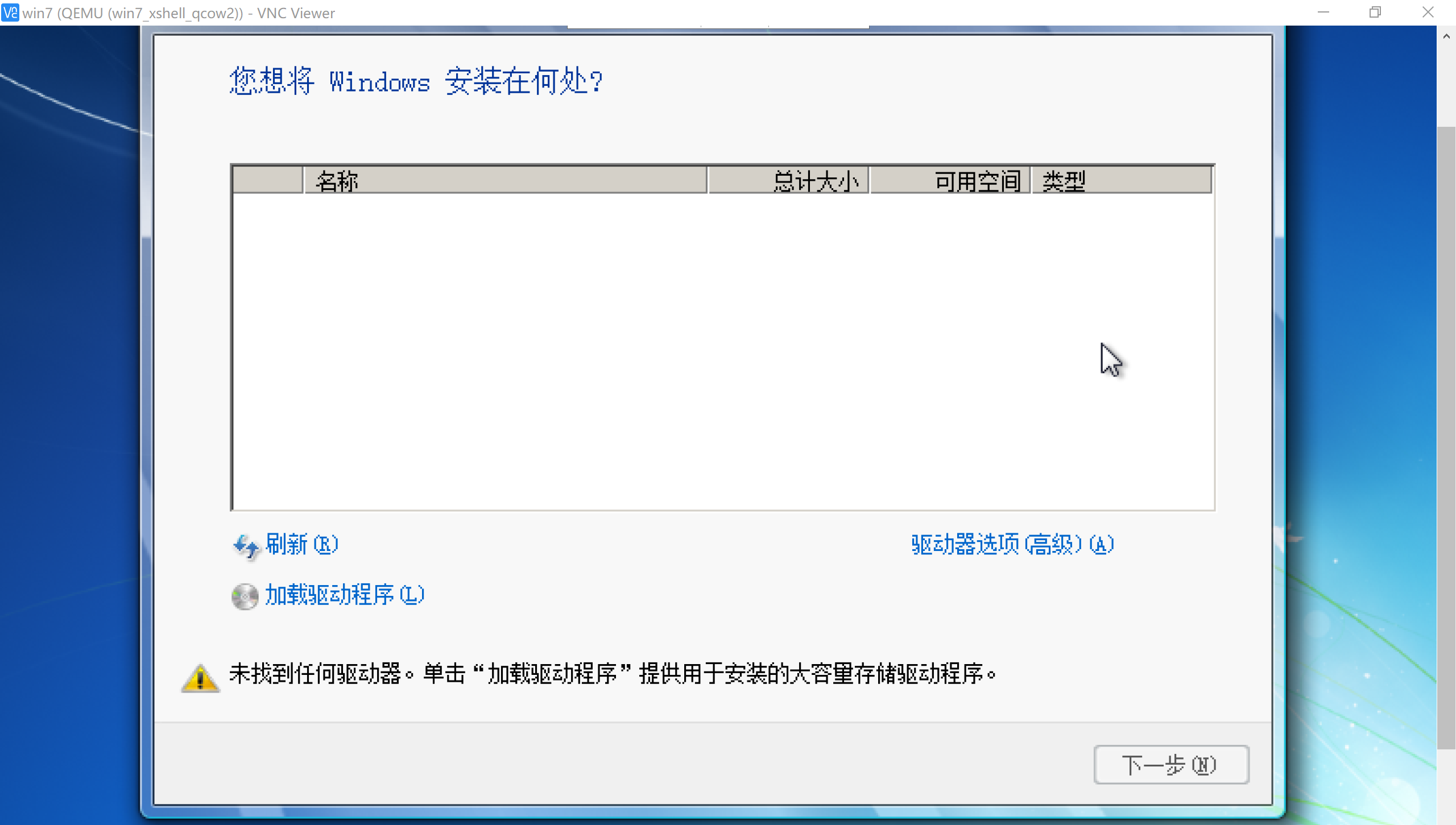


选择3或者4。进入引导安装界面（应该是选择4没错的，错了再次Ctrl+Alt+De）





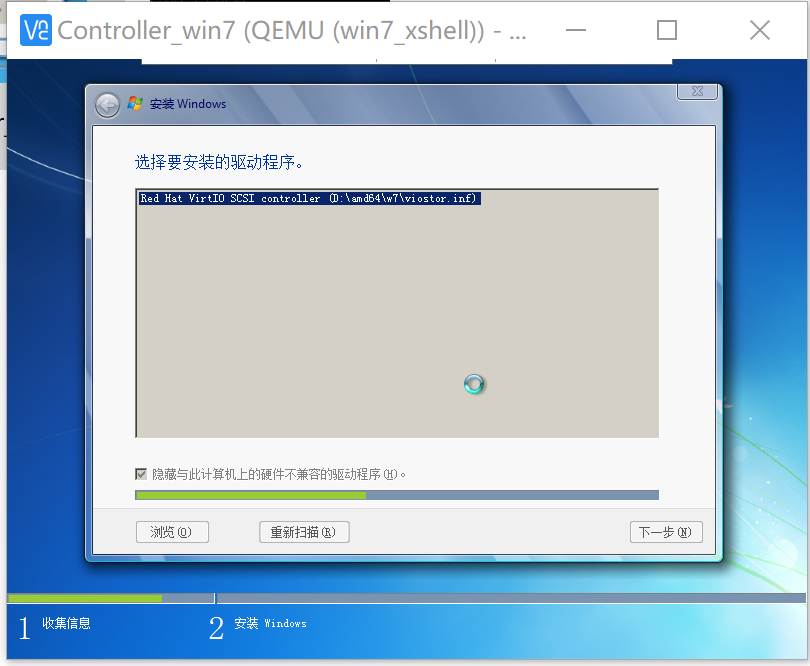
正常完成系统安装步骤直到出现硬盘驱动无法加载界面



选择加载驱动程序 自动扫描，然后选择win7的驱动。



等待驱动安装完成后即可选择磁盘，但是每一次都会卡在安装驱动上面，这是为什么呢？如下所示：



### （4）Centos安装图形化界面

感觉像是virt-install的参数设置问题导致的，因此尝试通过virt-manager图形化界面直接创建虚机；

因此尝试为centos安装图形化界面：

yum groupinstall -y "GNOME Desktop"

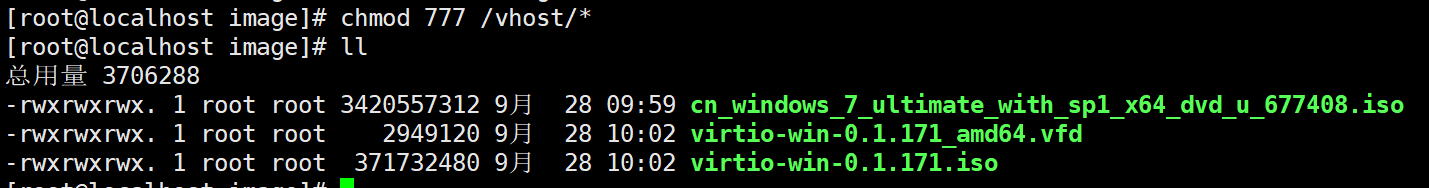
输入 init 5 进入图形化桌面

### （5）virt-manager创建虚拟机

查看是否支持了虚拟化：cat /proc/cpuinfo | grep -E 'vmx|svm'（可跳过），由于是新创建的虚拟机，因此需要在VMware中开启虚拟化；  
查看是否kvm的驱动已经加载：lsmod | grep kvm（可跳过）  
如果没有加载kvm驱动，利用命令加载驱动  
modprobe -a kvm  
modprobe -a kvm\_intel  
安装qemu-kvm: yum install -y qemu-kvm  
安装libvirt:  
yum install -y libvirt  
service libvirtd restart  
验证libvirt是否正常启动: virsh version  
安装virt-manager工具: yum install -y virt-manager  
关闭xshell，重新连接，输入命令：virt-manager，就可以自动弹出kvm管理软件

通过virt-manager图形化界面来安装一个Win7虚拟机

首先将image镜像转移到当前的虚拟机中，包括win7.iso镜像以及virtio镜像；



首先创建qcow2格式磁盘，指定虚拟磁盘大小100G。磁盘文件放在/vhost 目录。

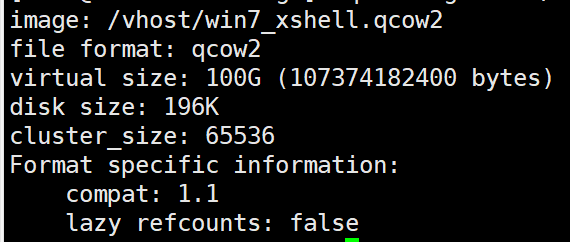
mkdir /vhost

qemu-img create -f qcow2 /vhost/win7\_xshell.qcow2 100G

查看磁盘信息：

qemu-img info /vhost/win7\_xshell.qcow2

可以看到，当前的实际磁盘大小是196k，虚拟磁盘大小是100G：



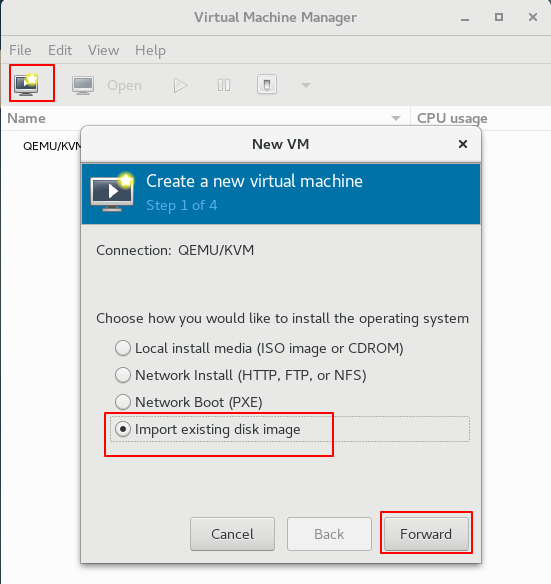
修改两个目录的权限

chmod 777 /image/\*

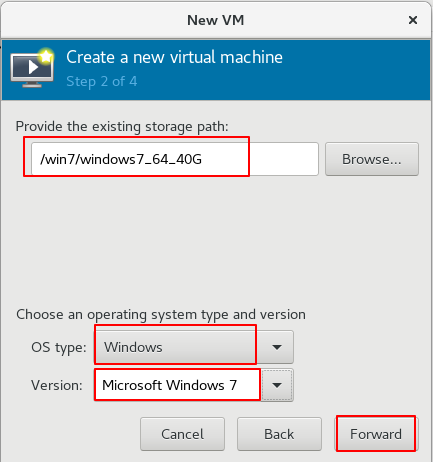
chmod 777 /vhost/\*

开始通过virt-manager图形化界面创建虚拟机：

从已存在的磁盘镜像中导入，这里对应的就是刚刚创建的 /vhost/win7\_xshell.qcow2

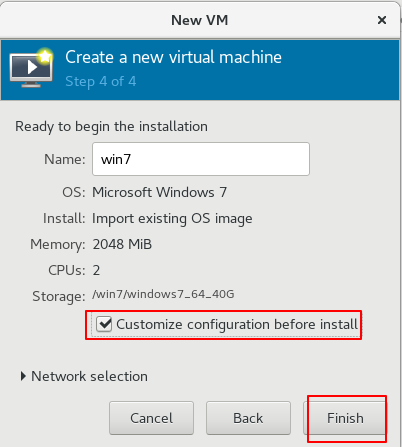


选择qcow2格式磁盘，以及设置Win7

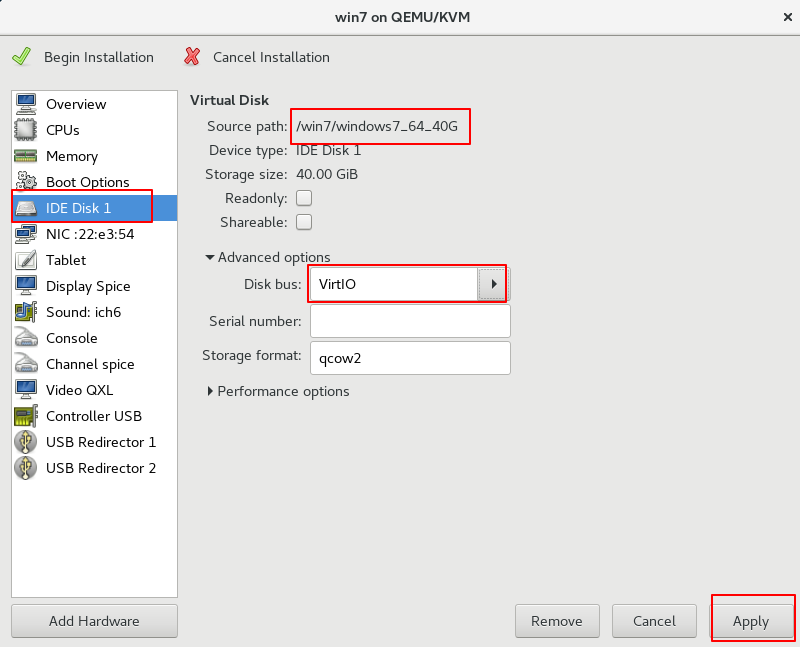


设置核心和内存大小，注意这里尽量设置的大一些，因为Win7的最低配置都是2GB内存，2核心，同时还要注意要在一个尽量大的宿主机中进行，之前的一直卡在安装virtio驱动这个过程中就是因为内存，核心设置不够导致的卡顿，因此新建了一个8G，6核心，100G的新的Centos虚拟机用来创建虚拟机。

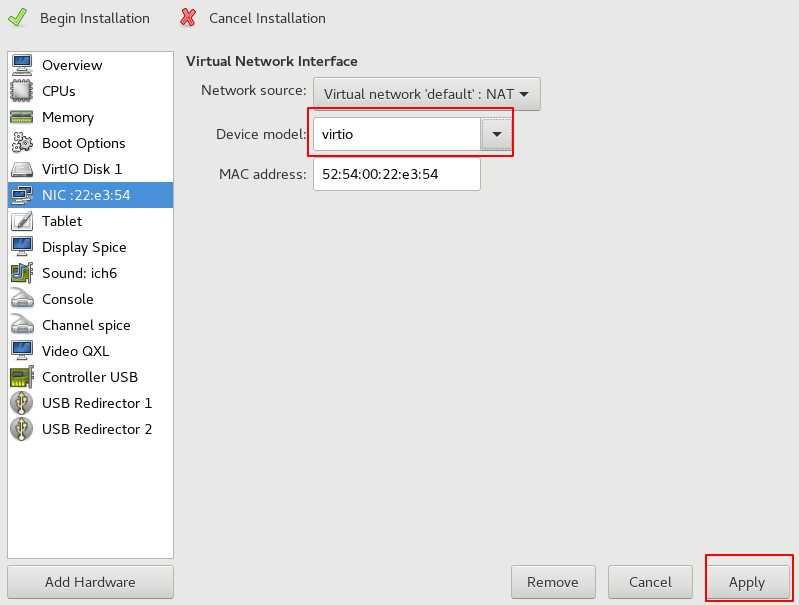
设置需要自定义配置



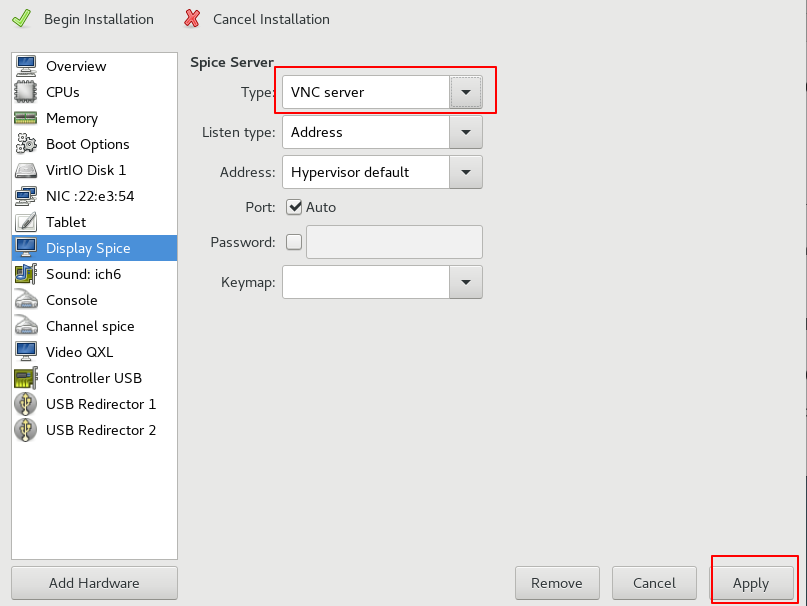
设置qcow2格式的磁盘的总线类型是virtio：



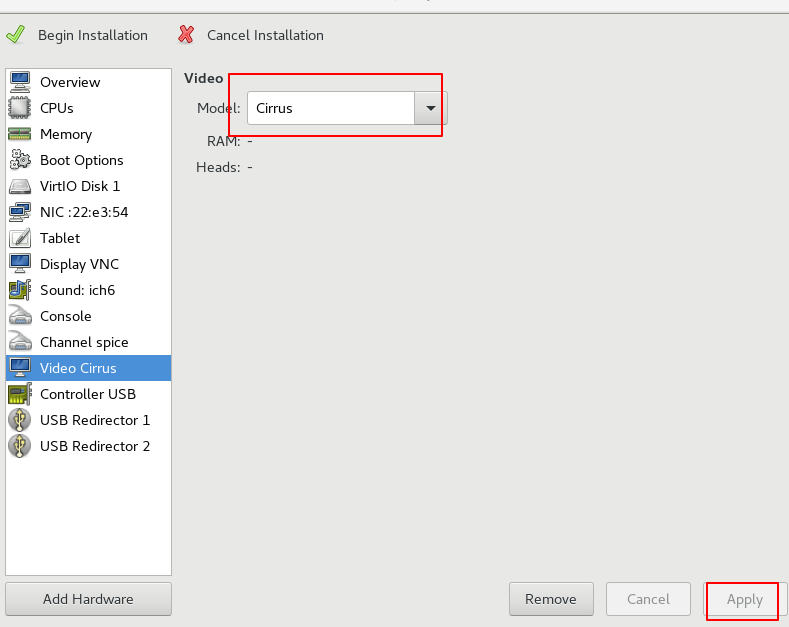
设置虚拟网络设备类型是virtio：



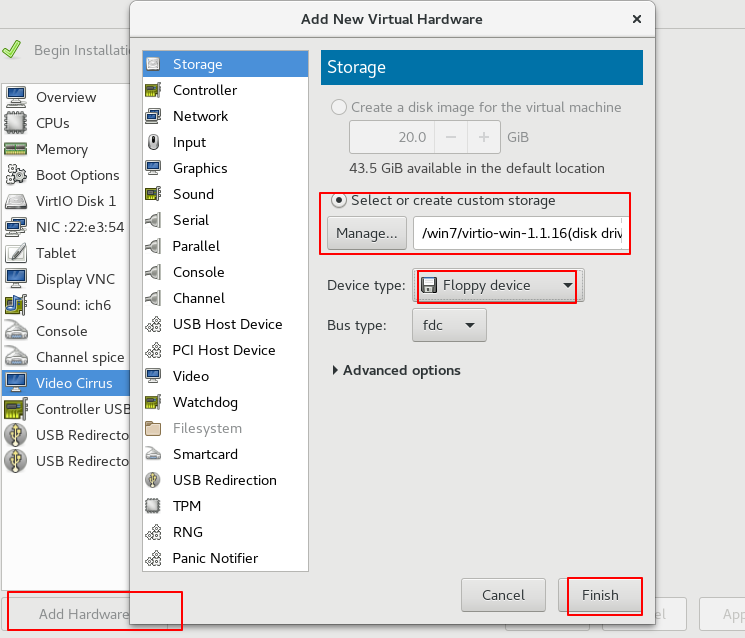
设置通过vnc连接虚拟机：



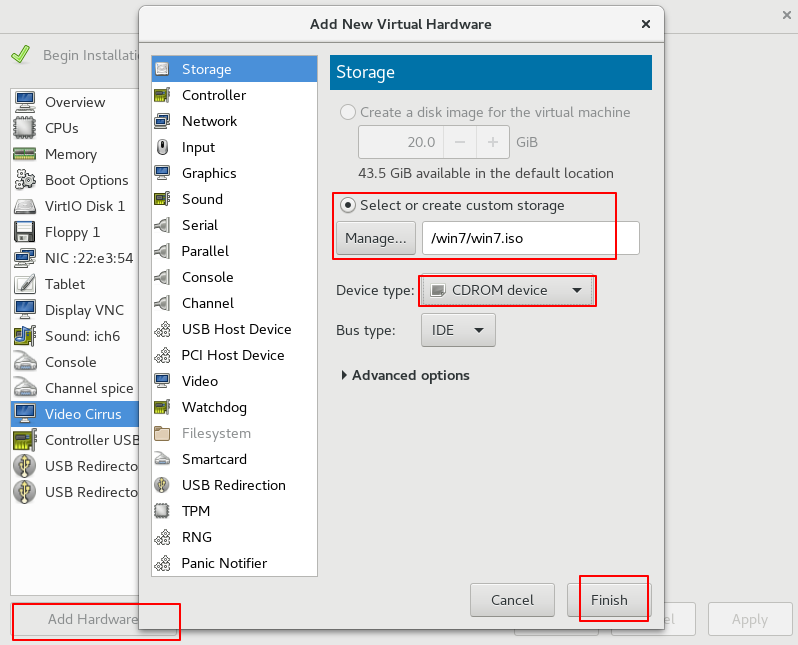
设置显卡类型是cirrus：

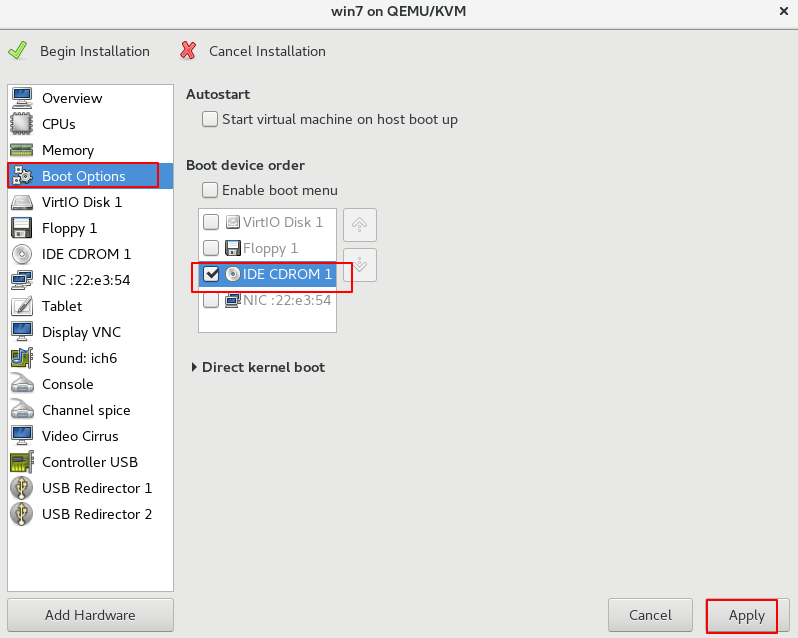


添加virtio\_amd64.vfd为软盘格式到虚拟机中：

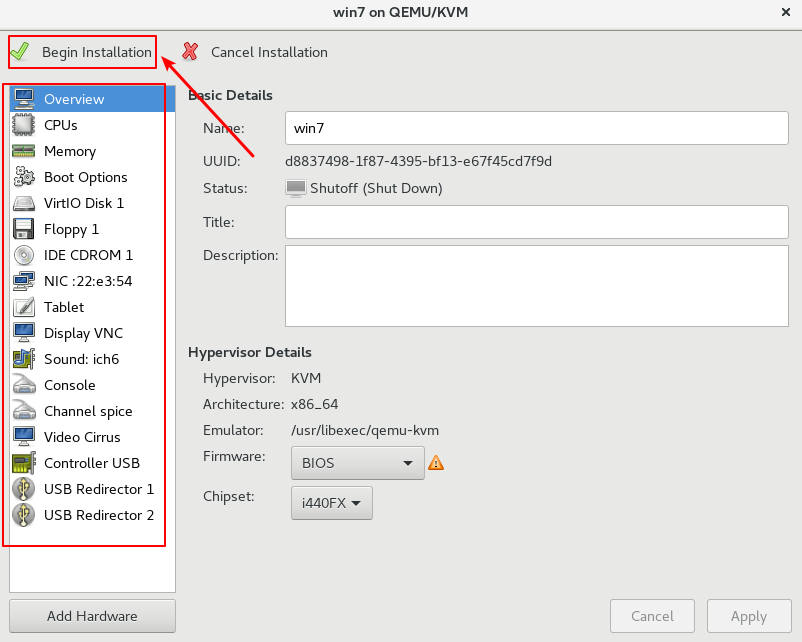


添加win7.iso镜像为cdrom格式到虚拟机中：



设置系统启动顺序为cdrom磁盘：  


开始安装win7系统：

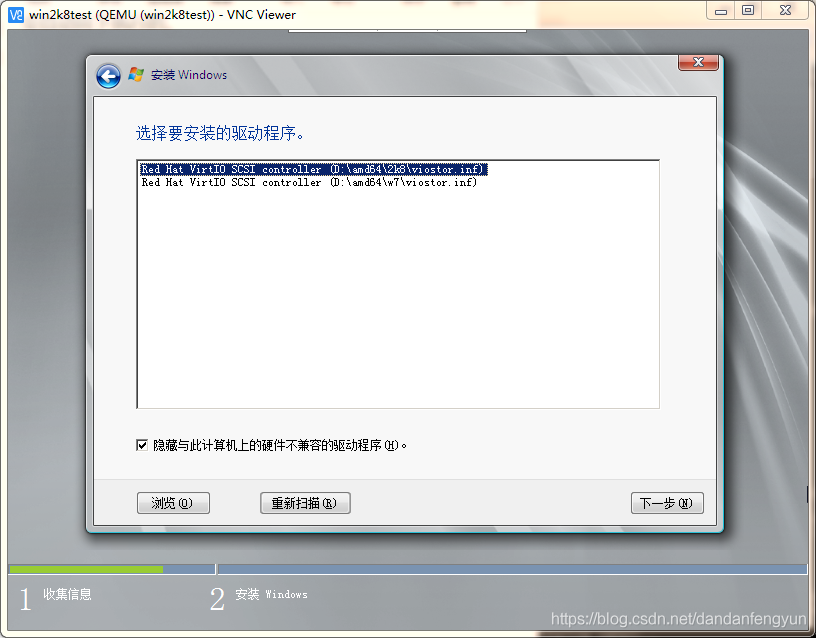


接下来就是非常快速地进入win7安装界面中，哭了，之前安装win7系统非常缓慢的原因就是宿主机的容量太小了，谨记啊，谨记啊！！！

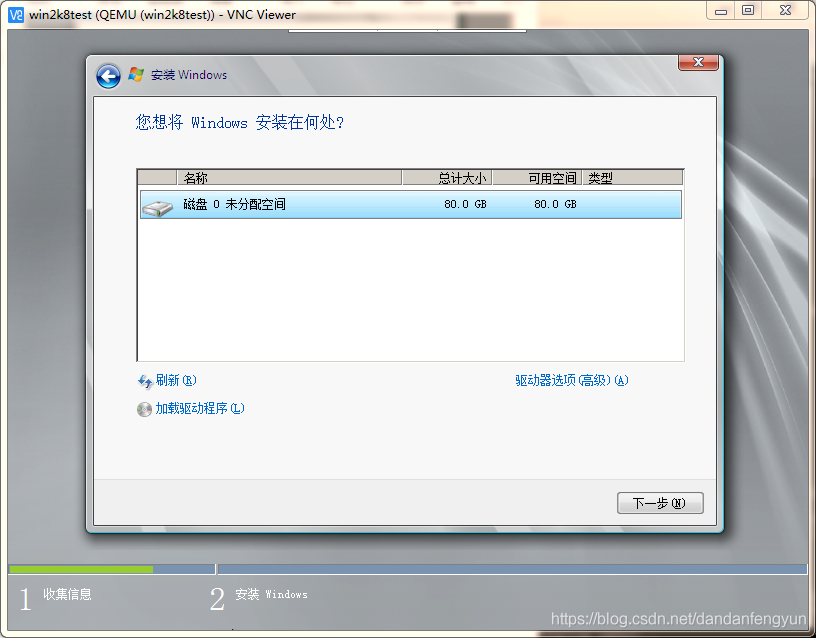
到了需要加载驱动的时候，需要选择之前挂载的virtio软盘：



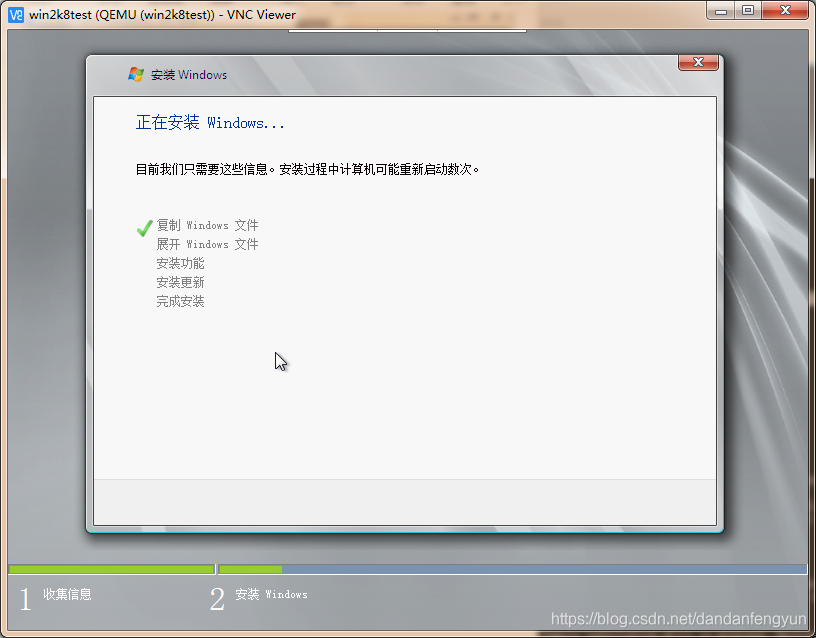
选择win7的amd64格式的驱动：



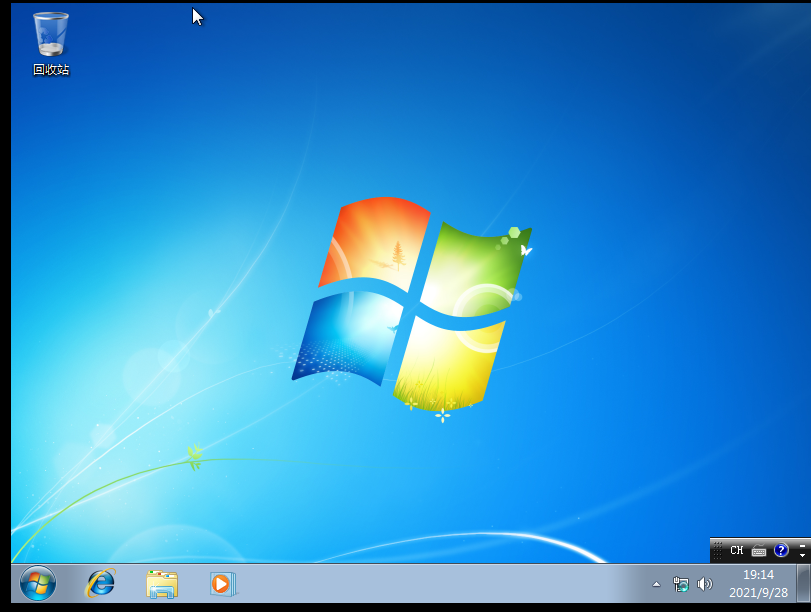
这次并没有卡顿，而是直接安装驱动完毕，然后显示了磁盘，该磁盘就是我们创建的qcow2格式磁盘



后面的就是安装Win7系统的过程了

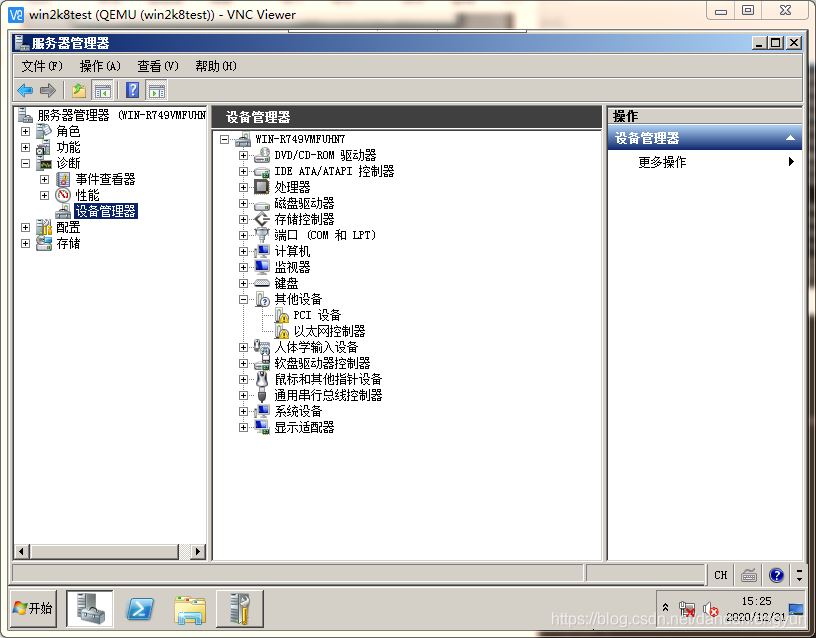


Win7系统成功地安装

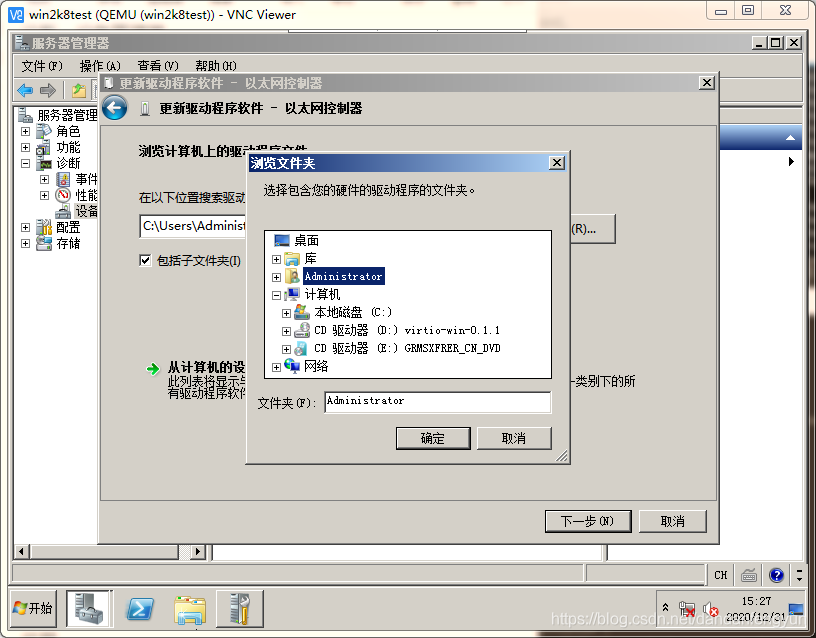


在该虚拟机中创建一个文档标明是图形化界面作为区分。

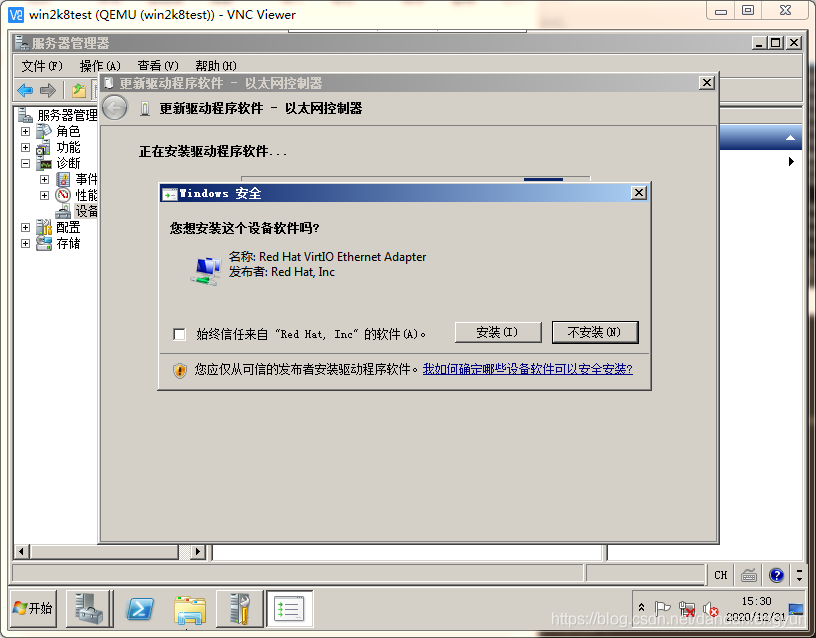
之后设置网卡驱动，此时没有网络驱动，需要手动安装。点击开始图标右边的服务管理器



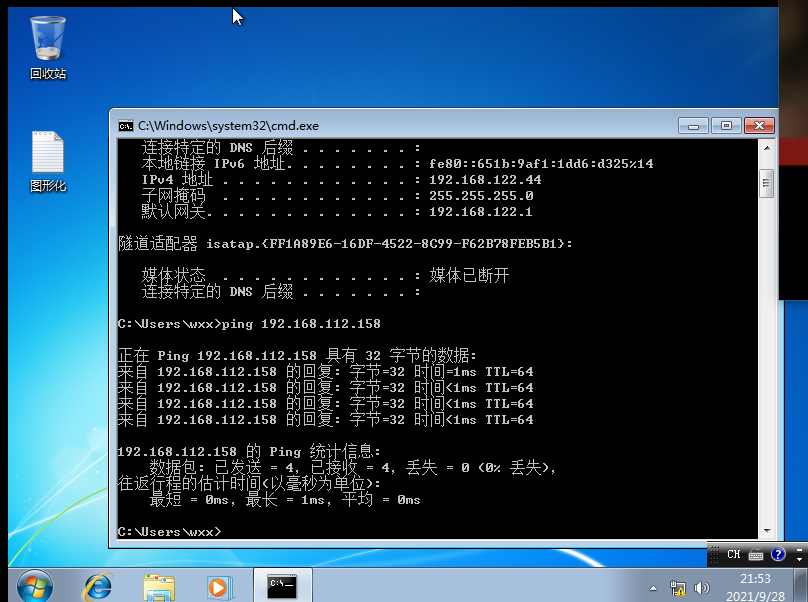
选择以太网控制器 更新驱动程序 手动查找驱动



选择win7下的驱动进行安装



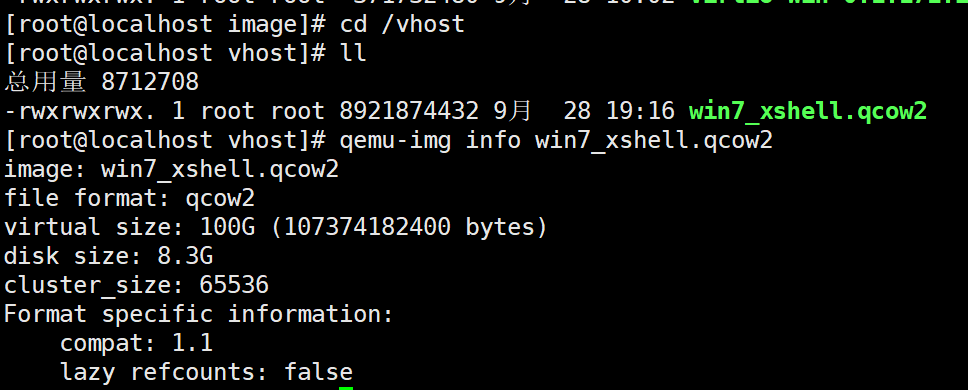
安装完成virtio-win的网卡驱动后，使用的virtio网络，查看当前虚拟机的网络情况：



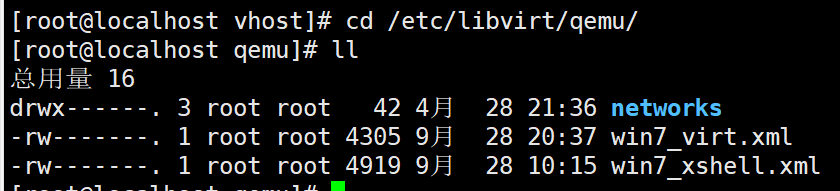
可以看到，IP地址设置为了192.168.122.142，同时尝试与宿主机ping，发现也可以ping通。

查看当前的qcow2格式Win7虚拟机镜像信息：

可以看到，当前的qcow2格式镜像大小为8.3G，原始的iso镜像大小为3.4G



注：只有配置了KVM虚拟机，libvirt就会生成一个与操作系统对应的xml文件，其记录了kvm虚拟机的状态。路径如下：/etc/libvirt/qemu



注：此文件只能通过“virsh edit”命令修改

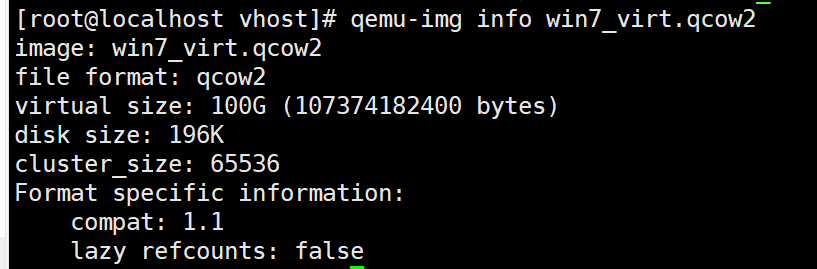
### （6）virt-install命令创建虚拟机

创建qcow2格式磁盘，指定大小100G。磁盘文件放在/vhost 目录。

mkdir /vhost

qemu-img create -f qcow2 /vhost/win7\_virt.qcow2 100G

查看该qcow2格式磁盘的信息：



开始将iso格式的镜像安装到刚刚创建的qcow2格式的文件中：

virt-install --name=win7\_virt --vcpus=2 --ram 2048 --disk path=/vhost/win7\_virt.qcow2,format=qcow2,device=disk,bus=virtio --disk path=/image/cn\_windows\_7\_ultimate\_with\_sp1\_x64\_dvd\_u\_677408.iso,device=cdrom,perms=rw --vnc --vnclisten=0.0.0.0 --os-type windows --os-variant=win7 --accelerate --network=default,model=virtio --disk=/image/virtio-win-0.1.171\_amd64.vfd,device=floppy --video cirrus --boot=cdrom --check path\_in\_use=off

注意问题：

cdrom和disk都不要设置在/root中，防止没有权限访问；

有两个disk设置，一个是qcow2格式的虚拟机地址，一个是virtio-win驱动的地址；

os-variant需要设置为系统的版本win7；

完全按照图形化界面安装win7系统的方式添加参数；

使用同一个iso镜像，设置不检查iso镜像路径；

通过vnc viewer连接虚拟机vnc 进行Win7操作系统的安装。

需一个可远程连接到CentOS7服务器主机的图形化系统，如windows mac安装vnc viewer软件，连接到服务器的相应端口对虚拟机系统进行安装设置。

要想连接虚拟机就需要执行一条命令来查看刚才新建虚拟机的端口信息  
[root@compute ~]# netstat -lntup |grep kvm



但是即使设置了正确的IP地址，也无法连接上，考虑可能是防火墙的问题，因此将防火墙的开机自启动关闭了：

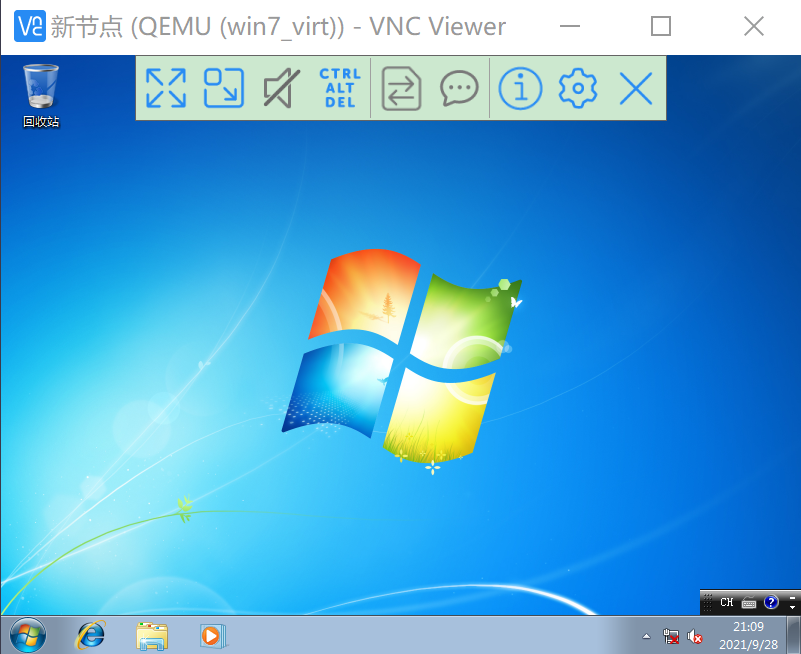
systemctl stop firewalld

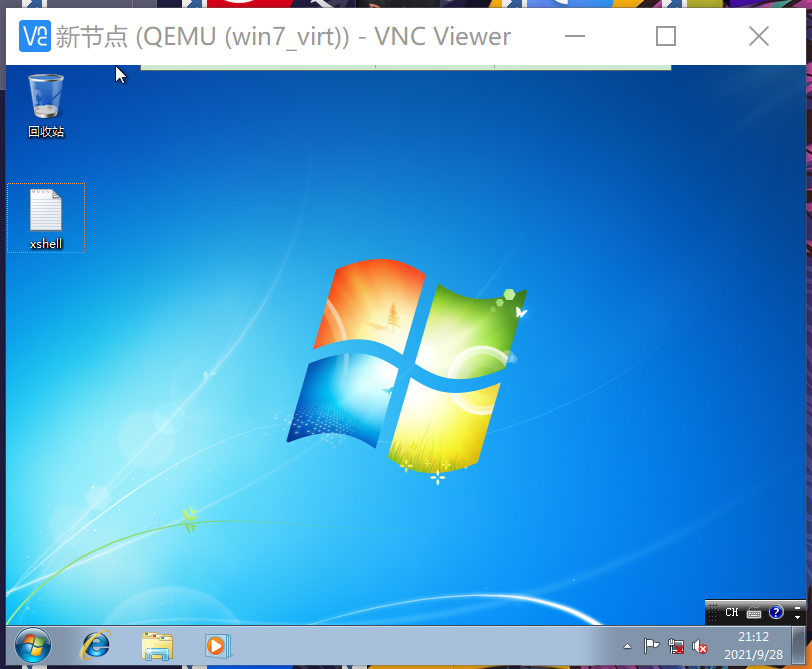
systemctl disable firewalld

再次通过vnc server连接就可以连接到虚拟机上了：

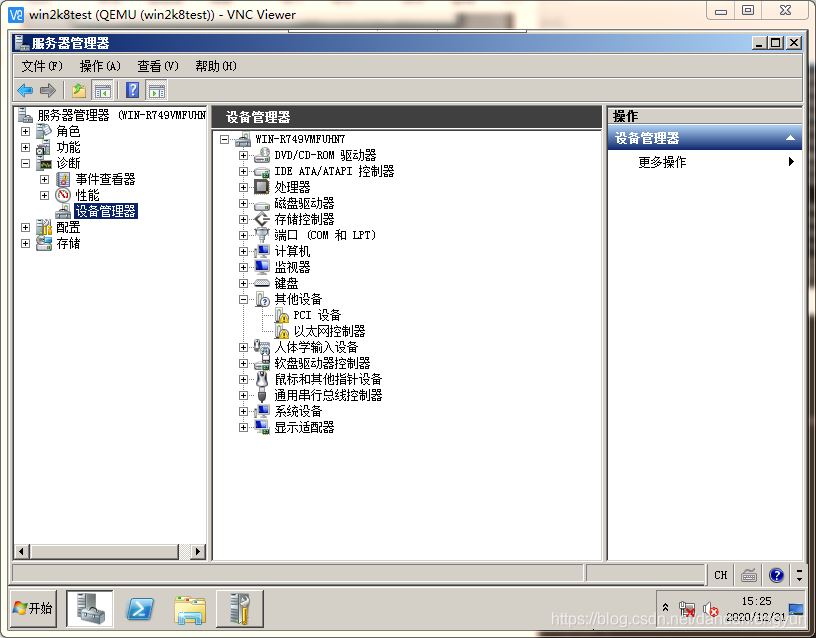


接下来就是正常的win7系统的安装过程；

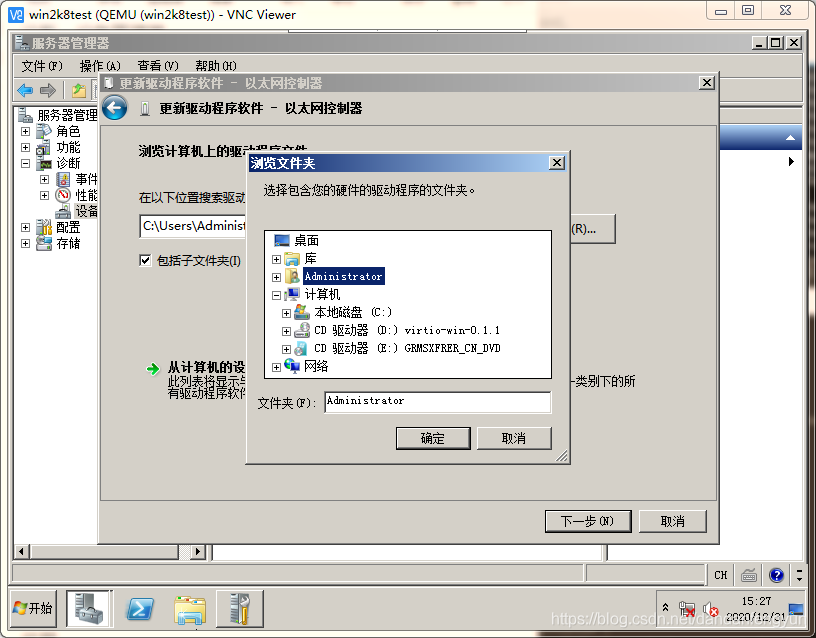


可以看到这是一个全新的win7虚拟机，并没有因为是通过同一个win7的iso镜像创建的就有所相同，说明iso镜像是公用的，而通过该iso镜像创建的qcow2格式镜像是互不相同的。在该虚拟机中再次创建创建一个下文档标识该虚拟机是xshell命令创建的：  


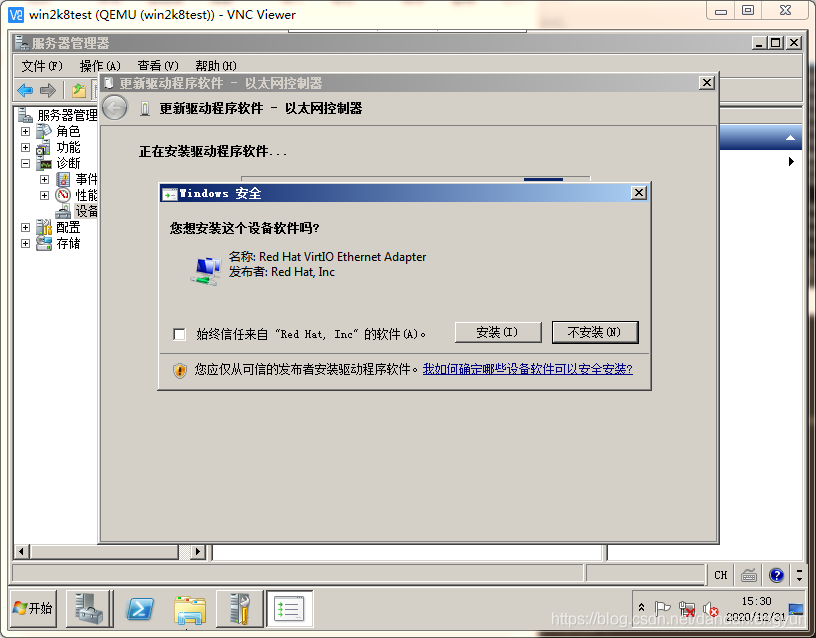
之后设置网卡驱动，此时没有网络驱动，需要手动安装。点击开始图标右边的服务管理器



选择以太网控制器 更新驱动程序 手动查找驱动



选择win7下的驱动进行安装



安装完成virtio-win的网卡驱动后，使用的virtio网络，查看当前虚拟机的网络情况：



可以看到，IP地址设置为了192.168.122.142，同时尝试与宿主机ping，发现也可以ping通。

### （7）qcow2格式镜像空间压缩

直接通过下面的方式即可压缩镜像空间：

qemu-img convert -O qcow2 win7.qcow2 win7\_new.qcow2

### （8）virt-install命令参数说明

virt-install 常用参数说明

# 新版本的写法

--name 指定虚拟机名称

--ram 虚拟机内存大小，以 MB 为单位

--vcpus 分配CPU核心数，最大与实体机CPU核心数相同

--network 虚拟机网络配置

--os-type 系统的类型如 linux windows

--os-variant 操作系统类型，更详细的表述。可以用osinfo-query os命令列出可选系统

--disk 指定虚拟机的磁盘存储位置

# size，初始磁盘大小，以 GB 为单位。

--location 指定安装介质路径，如光盘镜像的文件路径。

--graphics 图形化显示配置

# 全新安装虚拟机过程中可能会有很多交互操作，比如设置语言，初始化 root 密码等等。

# graphics 选项的作用就是配置图形化的交互方式，可以使用 vnc（一种远程桌面软件）进行链接。

# 如果使用命令行的方式安装，这里要设置为 none，但要通过 --extra-args 选项指定终端信息，

# 这样才能将安装过程中的交互信息输出到当前控制台。

--extra-args 根据不同的安装方式设置不同的额外选项

--cdrom 待安装的ISO镜像 也可使用--location

--boot 引导方式。

设置menu=on 表示可手动选择引导盘。

设置--boot=cdrom，引导直接通过cdrom启动。

--graphics 设置vnc的端口。linsten=0.0.0.0 允许除本机外主机连接。

从指定安装源创建新虚拟机。

optional arguments:

-h, --help show this help message and exit

--version show program's version number and exit

--connect URI 通过 libvirt URI 连接到虚拟机管理程序

通用选项:

-n NAME, --name NAME 客户机实例名称

--memory MEMORY Configure guest memory allocation. Ex:

--memory 1024 (in MiB)

--memory 512,maxmemory=1024

--memory 512,maxmemory=1024,hotplugmemorymax=2048,hotplugmemoryslots=2

--vcpus VCPUS Number of vcpus to configure for your guest. Ex:

--vcpus 5

--vcpus 5,maxvcpus=10,cpuset=1-4,6,8

--vcpus sockets=2,cores=4,threads=2

--cpu CPU CPU model and features. Ex:

--cpu coreduo,+x2apic

--cpu host-passthrough

--cpu host

--metadata METADATA 配置客户机元数据。例如：

--metadata name=foo,title="My pretty title",uuid=...

--metadata description="My nice long description"

安装方法选项:

--cdrom CDROM 光驱安装介质

-l LOCATION, --location LOCATION

安装源 (例如：nfs:host:/path, http://host/path,

ftp://host/path)

--pxe 使用 PXE 协议从网络引导

--import 在已有的磁盘镜像中构建客户机

--livecd 将光驱介质视为 Live CD

-x EXTRA\_ARGS, --extra-args EXTRA\_ARGS

将附加参数添加到由 --location

引导的内核中

--initrd-inject INITRD\_INJECT

添加指定文件到由 --location 指定的 initrd

根中

--os-variant DISTRO\_VARIANT

在客户机上安装的操作系统，例如：'fedor

a18'、'rhel6'、'winxp' 等。

--boot BOOT 配置客户机引导设置。例如：

--boot hd,cdrom,menu=on

--boot init=/sbin/init (针对容器)

--idmap IDMAP 为 LXC 容器启用用户名称空间。例如：

--idmap uid\_start=0,uid\_target=1000,uid\_count=10

设备选项:

--disk DISK 指定存储的各种选项。例如：

--disk size=10 (在默认位置创建 10GiB 镜像)

--disk /my/existing/disk,cache=none

--disk device=cdrom,bus=scsi

--disk=?

-w NETWORK, --network NETWORK

配置客户机网络接口。例如：

--network bridge=mybr0

--network network=my\_libvirt\_virtual\_net

--network network=mynet,model=virtio,mac=00:11...

--network none

--network help

--graphics GRAPHICS 配置客户机显示设置。例如：

--graphics vnc

--graphics spice,port=5901,tlsport=5902

--graphics none

--graphics vnc,password=foobar,port=5910,keymap=ja

--controller CONTROLLER

配置客户机控制器设备。例如：

--controller type=usb,model=ich9-ehci1

--input INPUT 配置客户机输入设备。例如：

--input tablet

--input keyboard,bus=usb

--serial SERIAL 配置客户机串口设备

--parallel PARALLEL 配置客户机并口设备

--channel CHANNEL 配置客户机通信通道

--console CONSOLE 配置文本控制台连接主机与客户机

--hostdev HOSTDEV 配置物理 USB/PCI 等主机设备与客户机共享

--filesystem FILESYSTEM

传递主机目录到客户机。例如：

--filesystem /my/source/dir,/dir/in/guest

--filesystem template\_name,/,type=template

--sound [SOUND] 配置客户机声音设备仿真

--watchdog WATCHDOG 配置客户机 watchdog 设备

--video VIDEO 配置客户机视频硬件。

--smartcard SMARTCARD

配置客户机智能卡设备。例如：

--smartcard mode=passthrough

--redirdev REDIRDEV 配置客户机重定向设备。例如：

--redirdev usb,type=tcp,server=192.168.1.1:4000

--memballoon MEMBALLOON

配置客户机 memballoon 设备。例如：

--memballoon model=virtio

--tpm TPM 配置客户机 TPM 设备。例如：

--tpm /dev/tpm

--rng RNG Configure a guest RNG device. Ex:

--rng /dev/urandom

--panic PANIC 配置客户机 panic 设备。例如：

--panic default

--memdev MEMDEV Configure a guest memory device. Ex:

--memdev dimm,target\_size=1024

客户机配置选项:

--security SECURITY 设置域安全驱动配置。

--cputune CPUTUNE Tune CPU parameters for the domain process.

--numatune NUMATUNE 为域进程调整 NUMA 策略。

--memtune MEMTUNE 为域进程调整内存策略。

--blkiotune BLKIOTUNE

为域进程调整 blkio 策略。

--memorybacking MEMORYBACKING

为域进程设置内存后备策略。例如：

--memorybacking hugepages=on

--features FEATURES 设置域 <features> XML。例如：

--features acpi=off

--features apic=on,eoi=on

--clock CLOCK 设置域 <clock> XML。例如：

--clock offset=localtime,rtc\_tickpolicy=catchup

--pm PM 配置 VM 电源管理功能

--events EVENTS 配置 VM 生命周期管理策略

--resource RESOURCE 配置 VM 资源分区(cgroups)

--sysinfo SYSINFO Configure SMBIOS System Information. Ex:

--sysinfo emulate

--sysinfo host

--sysinfo bios\_vendor=Vendor\_Inc.,bios\_version=1.2.3-abc,...

--sysinfo system\_manufacturer=System\_Corp.,system\_product=Computer,...

--sysinfo baseBoard\_manufacturer=Baseboard\_Corp.,baseBoard\_product=Motherboard,...

--qemu-commandline QEMU\_COMMANDLINE

Pass arguments directly to the qemu emulator. Ex:

--qemu-commandline='-display gtk,gl=on'

--qemu-commandline env=DISPLAY=:0.1

虚拟化平台选项:

-v, --hvm 这个客户机应该是一个全虚拟化客户机

-p, --paravirt 这个客户机应该是一个半虚拟化客户机

--container 这个客户机应该是一个容器客户机

--virt-type HV\_TYPE 要使用的管理程序名称 (kvm, qemu, xen, ...)

--arch ARCH 模拟 CPU 架构

--machine MACHINE 机器类型为仿真类型

其它选项:

--autostart 主机启动时自动启动域。

--transient Create a transient domain.

--wait WAIT 请等待数分钟以便完成安装。

--noautoconsole 不要自动尝试连接到客户端控制台

--noreboot 安装完成后不启动客户机。

--print-xml [XMLONLY]

打印生成的 XML 域，而不是创建客户机。

--dry-run 运行安装程序，但不创建设备或定义客户

机。

--check CHECK 启用或禁用验证检查。例如：

--check path\_in\_use=off

--check all=off

-q, --quiet 抑制非错误输出

-d, --debug 输入故障排除信息

### （9）virsh命令

上面通过virt-install来创建了虚机，可以通过virsh命令来查看：

virsh常用命令：

virsh start x                   启动名字为x的非活动虚拟机    
virsh list   --all                 列出虚拟机    
virsh create x.xml           创建虚拟机（创建后，虚拟机立即执行，成为活动主机）    
virsh suspend x              暂停虚拟机    
virsh resume x               启动暂停的虚拟机    
virsh shutdown x           正常关闭虚拟机    
virsh destroy x               强制关闭虚拟机    
# virsh undefine x          删除虚拟机    
virsh dominfo x                 显示虚拟机的基本信息    
virsh domname 2               显示id号为2的虚拟机名    
virsh domid x                    显示虚拟机id号    
virsh domuuid x                 显示虚拟机的uuid    
virsh domstate x                显示虚拟机的当前状态    
virsh dumpxml x                 显示虚拟机的当前配置文件（可能和定义虚拟机时的配置不同，因为当虚拟机启动时，需要给虚拟机 分配id号、uuid、vnc端口号等等）    
virsh setmem x 512000       给不活动虚拟机设置内存大小    
virsh edit x                        编辑配置文件（一般是在刚定义完虚拟机之后）