## 安装KVM

1、安装需要的包

yum install -y bridge-utils wget lrzsz gcc gcc-c++ net-tools telnet #系统基础包

yum install -y qemu-kvm libvirt virt-install bridge-utils docbook-utils.noarch virt-viewer

yum -y install kvm python-virtinst libvirt  bridge-utils virt-manager qemu-kvm-tools  virt-viewer  virt-v2v tunctl

####################桥接网卡需要tunctl ##########################

下载 tunctl-1.5.tar.gz

tar xzvf tunctl-1.5.tar.gz

cd tunctl-1.5

make

make install

2、 禁用网络管理器：

chkconfig NetworkManager off ##和桥接有冲突，要关闭

service NetworkManager stop

3、桥接网卡

在/etc/sysconfig/network-scripts/ 目录新建 ifcfg-br0

DEVICE=br0

TYPE=Bridge

BOOTPROTO=static

IPV6INIT=no

ONBOOT=yes

IPADDR=192.168.1.66

NETMASK=255.255.255.0

GATEWAY=192.168.1.1

DNS1=8.8.8.8

DELAY=0

[root@vfeelit network-scripts]# cat ifcfg-eth0

DEVICE=eth0

TYPE=Ethernet

BRIDGE=br0

BOOTPROTO=none

IPV6INIT=no

ONBOOT=yes

附加说明：这里配置成桥接方式，实际上本机的数据包发送是通过br0发送的，br0的TYPE是Bridge,其它的信息跟配置一个物理网卡一样，本机的eth0通过桥接到br0（通过BRIDGE指定）进行通信，它根本不需要配置IP信息，实际上数据发送还是通过本机的物理网卡。

[root@localhost centos7]# virt-install \

--name=VM-centos7 \

--ram 1024 \

--vcpus=1 \

--disk path=/KVM/VM-centos-7.img,size=50 \

--accelerate \

--cdrom /KVM/centos7/CentOS-7-x86\_64-DVD-1511.iso \

--graphics vnc,listen=0.0.0.0,port=5920, \

--network bridge=br0 \

--force \

--autostart

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

[root@localhost centos7]# virt-install \

> --name=VM-centos7 \ #虚拟机名

> --ram 1024 \ #分配内存大小，MB

> --vcpus=1 \ #配置虚拟机的vcpu 数目

> --disk path=/KVM/VM-centos-7.img,size=50 \ #虚拟机所用磁盘或镜像文件，size大小G

> --accelerate \

> --cdrom /KVM/centos7/CentOS-7-x86\_64-DVD-1511.iso \ #安装iso文件，如果使用网络安装则用： --pxe                    #网络安装

> --graphics vnc,listen=0.0.0.0,port=5920, \ #启用vnc ，监听IP和端口

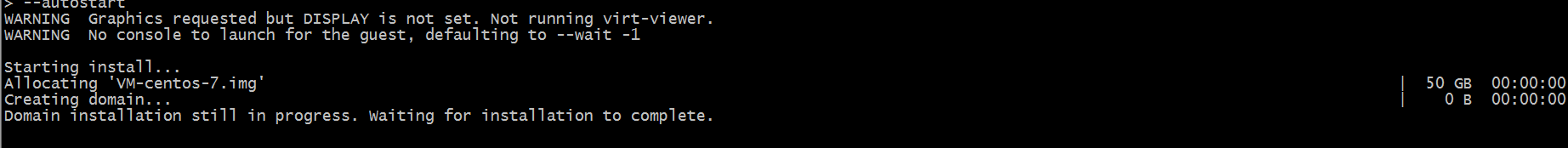
> --network bridge=br0 \ #指定网络，采用透明网桥

> --force \

> --autostart

提示则虚拟机正在创建，netstat 查看5920是否在监听。在监听则可以使用vnc连接IP:5920。

VNC连接出现闪退情况的解决：需修改vnc option里面Advanced-->expert-->ColourLevel的值为“rgb222” or “full”即可。



## 克隆虚拟机

### 1、使用命令克隆：

virsh shutdown VM-centos7 #先关源虚拟机

virsh destroy VM-centos7 #强行关闭虚拟机电源

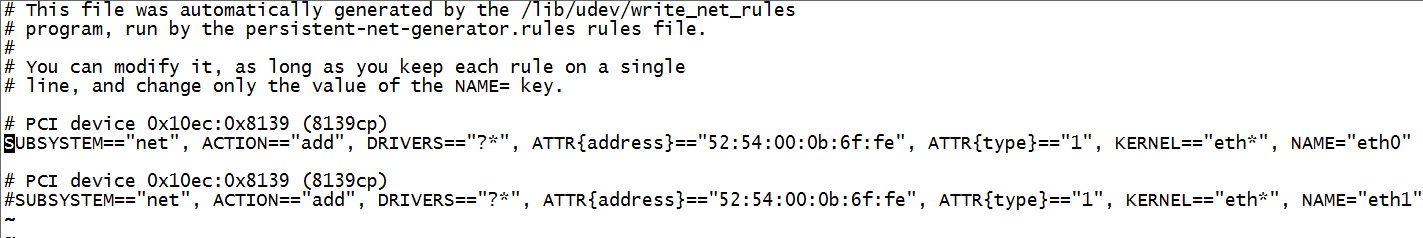
virt-clone -o VM-centos7 -n VM-centos7-1 -f /KVM/VM-centos7-1.img #执行克隆

virsh edit VM-centos7 #使用命令virsh edit 修改新虚拟机的配置文件（使用virsh-clone命令克隆的虚拟机会自动修改虚拟机名，mac，uuid。所以只需要修改vnc端口和socket文件，特别注意如果不修改socket文件路径名，则启动时只有当克隆源的机器启动时，克隆的机器才能正常启动。）

1. 修改vnc端口port='5921' autoport='no'，
2. Centos7需要修改socket文件路径：<source mode='bind' path='/var/lib/libvirt/qemu/channel/target/domain-VM-centos7-1/org.qemu.guest\_agent.0'/>
3. 修改虚拟机的名称<name>W2k3\_x64\_02</name>
4. 修改uuid编号 <uuid>de8d962a-6334-a652-f66b-4da87d852343</uuid>
5. 修改mac地址，如：<mac address='52:54:00:11:12:1f'/>

#### 克隆后网卡eth0找不到问题解决

1. 编辑/etc/udev/rules.d/70-persistent-net.rules文件
2. 将eth1行注释掉，将eth0的mac地址修改为eth1的mac地址



1. 编辑/etc/sysconfig/network-scripts/ifcfg-eth0 文件，将mac地址修改为与mac70-persistent-net.rules中的mac匹配的mac地址。

### 2、复制文件克隆

* 复制模板配置文件为w2k3\_x64\_02

1. 复制镜像文件

#cp VM-centos-7.img VM-centos7-Jira.img

1. cd /etc/libvirt/qemu #复制配置文件

#cp -av VM-centos-7.xml  VM-centos7-Jira.xml

1. 修改模板配置文件

virsh edit VM-centos7-Jira

1. 修改vnc端口port='5921' autoport='no'，
2. 修改socket文件路径：<source mode='bind' path='/var/lib/libvirt/qemu/channel/target/domain-VM-centos7-1/org.qemu.guest\_agent.0'/>
3. 修改虚拟机的名称<name>VM-centos7-Jira</name>
4. 修改uuid编号 <uuid>de8d962a-6334-a652-f66b-4da87d852343</uuid>
5. 修改mac地址，如：<mac address='52:54:00:11:12:1f'/>

4、#virsh define VM-centos7-Jira.xml #用配置文件定义虚拟机

5、#virsh list --all    //可查看虚拟机以看到新添加的VM已经添加了

6、#virsh start VM-centos7-Jira   //启动VM

7、vnc连接服务器后修改IP

8、删除/etc/udev/rules.d/70-persistent-net.rules 文件

##################################################################

kvm虚拟机管理  
一、环境  
role         hostname    ip                  OS  
kvm\_server   target      192.168.32.40/24    rhel6.0\_x64  
vir\_guest1   node4       192.168.32.34/24    rhel5.5\_i386  
vir\_guest2   node5       192.168.32.35/24    rhel5.5\_i386  
manager      manager     192.168.32.33/24    rhel5.5\_i386  
ESXi         ESXi        192.168.2.251/24    ESXi 3.5  
二、安裝kvm  
[root@target ~]# yum install -y qemu-kvm.x86\_64 qemu-kvm-tools.x86\_64  #安装kvm内核  
[root@target ~]# yum install libvirt.x86\_64 libvirt-cim.x86\_64 libvirt-client.x86\_64 libvirt-java.noarch  libvirt-python.x86\_64 #安装virt管理工具  
[root@target ~]# modprobe kvm         #加载kvm内核  
[root@target ~]# modprobe kvm-intel   #intel cpu加载kvm-intel内核，要支持全虚拟化，必须开启；要求cpu支持，通过bios可设置。  
[root@target ~]# modprobe kvm-amd     #amd cpu加载kvm-intel

[root@target ~]# modprobe -ls | grep kvm   #查看内核是否开启  
kernel/arch/x86/kvm/kvm.ko  
kernel/arch/x86/kvm/kvm-intel.ko  
kernel/arch/x86/kvm/kvm-amd.ko  
[root@target ~]# modprobe -ls | grep kvm-intel  
kernel/arch/x86/kvm/kvm-intel.ko

三、安装guest虚拟机  
1、直接通过virt-manager安装、管理虚拟机（略）

2、通过命令行安装guest虚拟机  
[root@target ~]# yum install virt-viewer   #开启图形控制台安装虚拟客户机需要，  
[root@target ~]# virt-install \            #安装选项可用virt-install --help查看  
--name node4 \           #虚拟机名  
--ram=1024 \             #分配内存大小，MB  
--arch=x86\_64 \          #模拟的CPU 构架  
--vcpus=1 \              #配置虚拟机的vcpu 数目  
--check-cpu \            #检查确定vcpu是否超过物理 CPU数目，如果超过则发出警告。  
--os-type=linux \        #要安装的操作系统类型，例如：'linux'、'unix'、'windows'  
--os-variant=rhel5 \     #操作系统版本，如：'fedora6', 'rhel5', 'solaris10', 'win2k'  
--disk path=/virhost/node7.img,device=disk,bus=virtio,size=20,sparse=true \   #虚拟机所用磁盘或镜像文件，size大小G  
--bridge=br0 \           #指定网络，采用透明网桥  
--noautoconsole \        #不自动开启控制台  
--pxe                    #网络安装

四、利用virsh对虚拟机管理  
1、开关机  
[root@target ~]# virsh start node4      #开机  
[root@target ~]# virsh create /etc/libvirt/qemu/node4.xml #直接通过主机配置文档启动主机  
[root@target ~]# virsh shutdown node4   #关机  
[root@target ~]# virsh destroy node4    #强制关闭电源  
[root@target ~]# virsh list --all       #查看虚拟机状态           
 Id 名称               状态  
----------------------------------  
 18 node4                running  
  - node5                关闭  
  - win8                 关闭

2、添加删除虚拟机  
[root@target ~]# virsh define /etc/libvirt/qemu/node5.xml   #根据主机配置文档添加虚拟机  
[root@target ~]# virsh list --all        #node5已经添加  
 Id 名称               状态  
----------------------------------  
 18 node4                running  
  - node5                关闭  
  - win8                 关闭  
   
[root@target ~]# virsh undefine node5     #移除虚拟机  
[root@target ~]# ls /etc/libvirt/qemu  
networks  node4.xml  win8.xml  
[root@target ~]# virsh list --all         #node5已经被移除  
 Id 名称               状态  
----------------------------------  
 18 node4                running  
  - win8                 关闭

3、远程管理虚拟机（qemu+ssh连接）  
[root@target ~]# yum install virt-viewer  
[root@target ~]# export DISPLAY=192.168.40.18:0.0  
[root@target ~]# virt-viewer -c qemu///system node4                          #本地管理虚拟机，system：获取system权限，注意qemu后是三个/  
[root@manager ~]# virt-viewer -c qemu+ssh://root@192.168.32.40/system node4  #远程linux通过virt-viewer+ssh管理虚拟机  
Xlib:  extension "RANDR" missing on display "192.168.40.18:0.0".  
root@192.168.32.40's password:  
root@192.168.32.40's password:  
#会弹出virt-viwer的gtk管理界面

4、使用已存在的虚拟机配置文档安裝新的虚拟机  
[root@target ~]# qemu-img create -f qcow2 /virhost/kvm\_node/node6.img 20G  
#为新建虚拟机生产磁盘镜像文件

[root@target ~]# virsh list  
 Id 名称               状态  
----------------------------------  
 18 node4                running

[root@target ~]# virsh dumpxml node4 >/etc/libvirt/qemu/node6.xml  
#导出虚拟机node6的硬件配置信息为/etc/libvirt/qemu/node6.xml

[root@target ~]# vim /etc/libvirt/qemu/node6.xml   
<domain type='kvm' id='20'>    #修改node6的id号  
  <name>node6</name>           #虚拟机node6的name  
  <uuid>4b7e91eb-6521-c2c6-cc64-c1ba72707fc7</uuid>  #uuid必须修改，否则会和node4的冲突  
  <memory>524288</memory>  
  <currentMemory>524288</currentMemory>  
  <vcpu>1</vcpu>  
  <os>  
    <type arch='x86\_64' machine='rhel5.4.0'>hvm</type>  
    <boot dev='network'/>  
  </os>  
  <features>  
    <acpi/>  
    <apic/>  
    <pae/>  
  </features>  
  <clock offset='utc'/>  
  <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' cache='none'/>  
      <source file='/virhost/node4.img'/>         #指定新虚拟机的硬盘文件  
      <target dev='vda' bus='virtio'/>  
    </disk>  
    <interface type='bridge'>  
      <mac address='54:52:00:69:d5:c7'/>  
      <source bridge='br0'/>  
      <target dev='vnet0'/>  
      <model type='virtio'/>  
    </interface>  
    <interface type='bridge'>  
      <mac address='54:52:00:69:d5:d7'/>  
      <source bridge='br0'/>  
      <target dev='vnet1'/>  
      <model type='virtio'/>  
    </interface>  
    <serial type='pty'>  
      <source path='/dev/pts/4'/>  
      <target port='0'/>  
    </serial>  
    <console type='pty' tty='/dev/pts/4'>  
      <source path='/dev/pts/4'/>  
      <target port='0'/>  
    </console>  
    <input type='mouse' bus='ps2'/>  
    <graphics type='vnc' port='5900' autoport='yes' keymap='en-us'/>  
  </devices>  
</domain>

[root@target ~]# virsh define /etc/libvirt/qemu/node6.xml  
#使用虚拟描述文档建立虚拟机，可用virsh edit node6修改node6的配置文件

[root@target ~]# virsh start node6  
#启动虚拟机

5 为虚拟机开启vnc  
[root@target ~]# virsh edit node4     #编辑node4的配置文件；不建议直接通过vim node4.xml修改。  
<graphics type='vnc' port='-1' autoport='yes' listen='127.0.0.1' keymap='en-us'/>  
   #port='-1' autoport='yes'：port自动分配，监听回环网络（virt-manager管理需要listen='127.0.0.1'）,无密码  
改为  
<graphics type='vnc' port='5904' autoport='no' listen='0.0.0.0' keymap='en-us' passwd='xiaobai'/>  
   #固定vnc管理端口5904，不自动分配，vnc密码xiaobai，监听所有网络

2. 远程vnc访问地址：192.168.32.40:5904

五、存储池和存储卷的管理  
1.创建 KVM主机存储池  
1).创建基于文件夹（目录）的存储池  
[root@target virhost]# virsh pool-define-as vmware\_pool --type dir --target /virhost/vmware  
#定义存储池vmware\_pool  
或  
[root@target virhost]# virsh pool-create-as --name vmware\_pool --type dir --target /virhost/vmware      
#创建存储池vmware\_pool，类型为文件目录,/virhost/vmware，与pool-define-as结果一样

2).创建基于文件系统的存储池  
[root@target virhost]#virsh pool-define-as --name vmware\_pool --type fs --source-dev /dev/vg\_target/LogVol02 --source-format ext4 --target /virhost/vmware  
或  
[root@target virhost]#virsh pool-create-as --name vmware\_pool --type fs --source-dev /dev/vg\_target/LogVol02 --source-format ext4 --target /virhost/vmware

3).查看存储池信息  
[root@target virhost]# virsh pool-info vmware\_pool  #查看存储域（池）信息  
名称：       vmware\_pool  
UUID:           2e9ff708-241f-fd7b-3b57-25df273a55db  
状态：       running  
Persistent:     no  
自动启动： no  
容量：       98.40 GB  
分配：       18.39 GB  
可用：       80.01 GB  
4).启动存储池  
[root@target virhost]# virsh pool-start vmware\_pool   #启动存储池  
[root@target virhost]# virsh pool-list  
名称               状态     自动开始  
-----------------------------------------  
default              活动     yes        
virhost              活动     yes        
vmware\_pool          活动     no  
5)销毁存储域,取消存储池  
[root@target virhost]# virsh pool-destroy vmware\_pool  #销毁存储池  
[root@target virhost]# virsh pool-list --all  
名称               状态     自动开始  
-----------------------------------------  
default              活动     yes        
virhost              活动     yes        
vmware\_pool          不活跃  no           
[root@target virhost]# virsh pool-undefine vmware\_pool  #取消存储池的定义  
[root@target virhost]# virsh pool-list --all            
名称               状态     自动开始  
-----------------------------------------  
default              活动     yes        
virhost              活动     yes

2.创建了存储池后，就可以创建一个卷，这个卷是用来做虚拟机的硬盘  
[root@target virhost]# virsh vol-create-as --pool vmware\_pool --name node6.img --capacity 10G --allocation 1G --format qcow2  
#创建卷 node6.img,所在存储池为vmware\_pool，容量10G，初始分配1G，文件格式类型qcow2

[root@target virhost]# virsh vol-info /virhost/vmware/node6.img  #查看卷信息  
名称：       node6.img  
类型：       文件  
容量：       10.00 GB  
分配：       136.00 KB

3.在存储卷上安装虚拟主机  
[root@target virhost]#virt-install --connect qemu:///system \  
-n node7 \  
-r 512 \  
-f /virhost/vmware/node7.img \  
--vnc \  
--os-type=linux \  
--os-variant=rhel6  \  
--vcpus=1 \  
--network bridge=br0 \  
-c /mnt/rhel-server-6.0-x86\_64-dvd.iso

六、虚拟机的迁移（vmware 2 kvm）  
1.安装软件  
[root@target ~]# yum install -y virt-v2v.x86\_64  
[root@target ~]# rpm -ivh libguestfs-winsupport-1.0-7.el6.x86\_64.rpm  virtio-win-1.2.0-1.el6.noarch.rpm   
#虚拟机为windows时需要libguestfs-winsupport 包来支持 NTFS 文件系统和 virtio-win 包来支持 Windows para-virtualized 存储和网络设备驱动

2.创建 KVM 主机存储域（略）  
virt-v2v 在迁移虚拟机的过程中，需要拷贝被迁移虚拟机到 KVM 主机预先定义的存储池中。

3.创建 KVM 主机网络接口（略）  
虚拟机在迁移之后将连接 KVM 主机网络， 因此主机必须有一个与之相匹配的网络接口，比如说网桥。

4.在 KVM 主机上创建或修改 $HOME/.netrc 文件，添加 VMware ESXi 服务器的用户名和密码。  
[root@target ~]# cat ~/.netrc  
machine 192.168.2.251 login root password xxxxxx  
[root@target ~]# chmod 0600 ~/.netrc

5.Vmware ESXi到KVM的迁移  
[root@target ~]# virt-v2v -ic esx://192.168.2.251/?no\_verify=1 -op virhost -b br0 ipserver    
\*\* HEAD https://192.168.2.251/folder/tserver21/RHEL4.6-flat.vmdk?dcPath=ha-datacenter&dsName=ESX35-bak%3Astorage1 ==> 401 Unauthorized  
\*\* HEAD https://192.168.2.251/folder/tserver21/RHEL4.6-flat.vmdk?dcPath=ha-datacenter&dsName=ESX35-bak%3Astorage1 ==> 200 OK  
\*\* GET https://192.168.2.251/folder/tserver21/RHEL4.6-flat.vmdk?dcPath=ha-datacenter&dsName=ESX35-bak%3Astorage1 ==> 200 OK (2084s)  
unknown filesystem /dev/hda  
unknown filesystem /dev/fd0  
virt-v2v: Installation failed because the following files referenced in the configuration file are required, but missing: rhel/4/kernel-smp-2.6.9-89.EL.i686.rpm  
virt-v2v: 非 virtio 驱动程序配置的 tserver21

#所有选项可通过配置文件/etc/virt-v2v.conf来指定  
#-op：指定用于转换后的存储域（pool)，virhost  
#-b：指定网络为网桥br0  
#-ic：指定要转换的源地址

[root@target kvm\_node]# virsh list --all  
 Id 名称               状态  
----------------------------------  
  1 node4                running  
  - node5                关闭  
  - tserver21            关闭  
  - win8                 关闭  
[root@target kvm\_node]# virsh start tserver21

6.KVM到KVM迁移  
[root@target kvm\_node]# virt-v2v -ic qemu+ssh://192.168.32.179/system -op virhost -b br0 node6  
root@192.168.32.179's password:  
root@192.168.32.179's password:  
unknown filesystem label SWAP-vda3  
virt-v2v: The connected hypervisor does not support a machine type of rhel5.4.0. It will be set to the current default.  
virt-v2v: virtio 驱动程序配置的 node6

[root@target kvm\_node]# virsh list --all  
 Id 名称               状态  
----------------------------------  
  1 node4                running  
  - node5                关闭  
  - node6                关闭  
  - tserver21            关闭  
  - win8                 关闭

[root@target kvm\_node]# virsh start node6

六、管理kvm虚拟机  
virsh --connect qemu:///system  
a、如果你修改了一个客户机的xml文件（位于/etc/libvirt/qemu/ 目录），你必须重新定义客户机：  
define /etc/libvirt/qemu/vm10.xml  
b、启动和停止客户机，运行：  
start vm10  
c、停止一个客户机，运行  
shutdown vm10  
d、立即中断一个客户机（类似直接关电源），运行  
destroy vm10  
e、挂起一个客户机:  
suspend vm10  
f、恢复客户机:  
resume vm10  
  
七、克隆  
virt-clone  --connect=qemu:///system \  
-o 原虚拟机 -n 新虚拟机 -f /data0/centos5.4/新虚拟机镜像  
新虚拟机配置文件需要注意：虚拟机克隆完毕后，需修改vnc端口；MAC地址也是需要注意的。  
  
八、加载磁盘空间  
#qemu-img create -f qcow2 disk1.img 50G  
#vi /etc/libvirt/qemu/centos1.xml  
增加的部分为：  
  
    <disk type='file' device='disk'>  
      <source file='/data0/disk1.img'/>  
      <target dev='hdb' bus='ide'/>  
    </disk>  
使配置文件生效  
virsh --connect qemu:///system  
virsh #define /etc/libvirt/qemu/centos1.xml  
重起虚拟机生效  
virsh # shutdown centos1  
virsh # start centos1  
进入虚拟机：  
mkfs.ext3 /dev/hdb  
mkdir /data  
mount /dev/hdb /data

### 3、KVM 链接克隆

KVM像其他的虚拟机产品有相同的克隆方式，一种是连接方式的克隆，另一种是完全克隆的方式。但是，一般情况下我们使用连接克隆，使用这种克隆方式的时候会节省硬盘的空间，而且还能够快速的进行虚拟机的克隆，而采用完全克隆的方式很明显会增加硬盘的使用空间。   
下面对虚拟机的连接克隆的方式进行总结，通过相同的模式，来进行虚拟机的克隆，然后通过脚本的方式来进行这一系列的操作的自动化进行。借此巩固知识，提升自己。   
虚拟机克隆方式为连接克隆时：

一、在模板虚拟机内部操作的步骤   
1)将模板虚拟机的MAC删掉

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

2)删除网卡的UDEV

#sed -i '1,$d' /etc/udev/rules.d/70-persistent-net.rules

二、在物理机上操作的步骤   
0)克隆前确定模板的磁盘文件是否为qcow2格式,如果不是需要转换

[root@kvm qemu]#qemu-img info /var/lib/libvirt/images/shiwei1.img

image: /var/lib/libvirt/images/shiwei1.img

file format: qcow2

virtual size: 8.0G (8589934592 bytes)

disk size: 1.3G

cluster\_size: 65536

如果显示上边的信息，表示磁盘格式为qcow2

[root@kvm qemu]#qemu-img convert -f raw -O qcow2 test.raw test.qcow2

[root@kvm qemu]#qemu-img info test.qcow2 #查看格式命令

1)克隆模板的磁盘文件

[root@kvm qemu]#qemu-img create -f qcow2 -b test.qcow2 test2.qcow2

2)克隆模板的配置文件

[root@kvm qemu]#virsh dumpxml test > /etc/libvirt/qemu/test2.xml

3)删掉网卡的MAC

[root@kvm qemu]#vim /etc/libvirt/qemu/test2.xml

<mac address='52:54:00:d2:55:41'/>

4)清除模板虚拟机的UUID

<uuid>9c60d018-5216-ccdd-1d87-0d2915b4a4da</uuid>

5)修改磁盘名称

<source file='/var/lib/libvirt/images/shiwei1.img'/>

6)虚拟机name

<name>shiwei</name>

7)重新定义一个虚拟机

[root@kvm qemu]#virsh define /etc/libvirt/qemu/test2.xml

下面介绍一下这个过程中的注意的问题，我们知道了如何进行操作，也要明白其中的道理，为什么要这样进行操作，以及这样操作我们可以实现什么样的效果。   
在虚拟机中的操作：将模板虚拟机内部的MAC地址删除，我们知道在同一个网络中我们必须保证不存在相同的两个MAC地址，如果存在相同的mac地址的话，我们就不能保证两个主机间的通信，而克隆，我们知道它的含义就是和原来的相同，不经过修改的话就是一样的，所以我们要删除MAC地址。   
删除网卡的UDEV：UDEV是什么东西呢？它是在我们系统中记录系统硬件变化的一个文件夹，我们知道当我们安装好一个有网卡的系统时，我们的网卡的信息就记录在相应的这个70-persistent-net.rules文件下，而此时的这个网卡的编号为eth0，但是如果我们的网卡坏掉了，我们将自己的网卡更换了一块，这时我们在进入系统时，看到的网卡就已经不再是eth0了，而是eth1，原因就是我们的网卡的mac地址不同，而这个文件是通过MAC地址来进行区别的。如果是磁盘，与这种情况类似，只不过编号不同，而是sda，sdb等，它是通过总线来进行区分的。（以上问题引申出来的一个东西，sed，强大工具的学习，积累和巩固）   
在实体机上的操作：我们建立的虚拟机其实主要有两个文件，一个是磁盘文件，另外一个是配置文件，其中磁盘文件一般默认位置在/var/lib/libvirt/images/文件夹下，我们可以复制这里面的磁盘文件，以此来克隆虚拟机。但是需要注意的是，我们如果不知道我们磁盘文件的格式的话，最好是查看一下磁盘文件的格式，然后在进行磁盘文件的复制。另外的配置文件，每一个虚拟机都有唯一个配置文件，这个配置文件定义了虚拟机的磁盘的位置，虚拟机中的网络配置，硬件配置等信息。我们对文件进行相应的修改作为虚拟机模板。   
通过对配置文件中四个地方的修改来完成克隆虚拟主机的配置文件生成，但是这里我们发现我们再次修改了mac地址，但是在上面的虚拟机中我们已经修改了相应的配置文件了，为什么这里我们还需要进行修改呢？原因是：我们都知道，在我们的硬件的中是存在着MAC地址的，而通过操作系统我们可以修改MAC地址，但是如果我们重新启动系统，我们的MAC地址有回到了原来，原因是，这个mac地址是写到了硬件中的，不能被修改的。这里其实是相同的道理，我们在内部修改mac，在实体机上也要进行修改。

创建的虚拟机的几种访问的方式：   
直接进行访问，使用图形化界面；   
使用xshell等远程连接的方式；   
通过console的方式，这种方式我们需要在相应的虚拟机中的/etc/grub.conf中进行配置这样的一行参数，console=ttyS0，注意是大写的S和数字0

## 给windows虚拟机新增磁盘

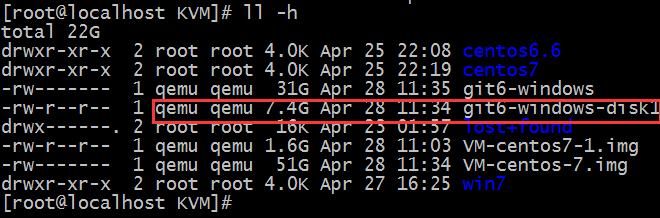
1. 关闭虚拟机

可以vnc登录到虚拟机关机或者

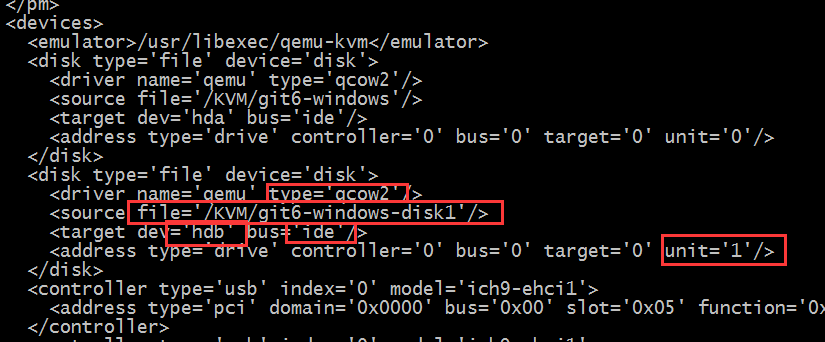
virsh destroy git6-windows #强制关闭电源

1. qemu-img create -f qcow2 git6-windows-disk1 50G #创建50G的新磁盘

刚创建时这个文件很小，只有100多k，随着往这个新建的磁盘中写入数据，硬盘也随之增大。



3、virsh edit git6-windows #使用virsh edit编辑配置文件。注意dev=’hdb’和unit不能跟原有设备重复。



4、virsh define git6-windows.xml #重新加载配置文件

5、virsh start git6-windows #启动虚拟机

6、在windows中添加磁盘即可

## 给linux虚拟机新增磁盘

1. 关闭虚拟机

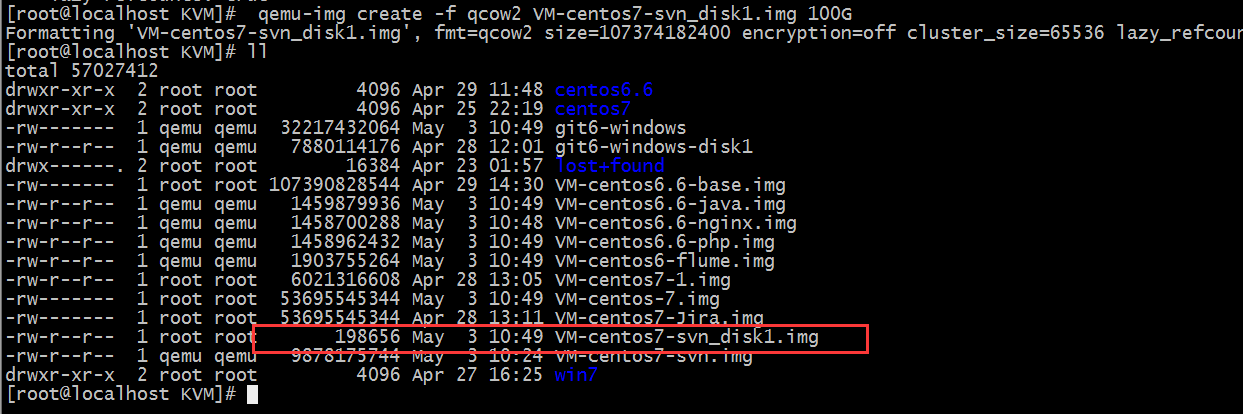
可以vnc登录到虚拟机关机或者

virsh destroy git6-windows #强制关闭电源

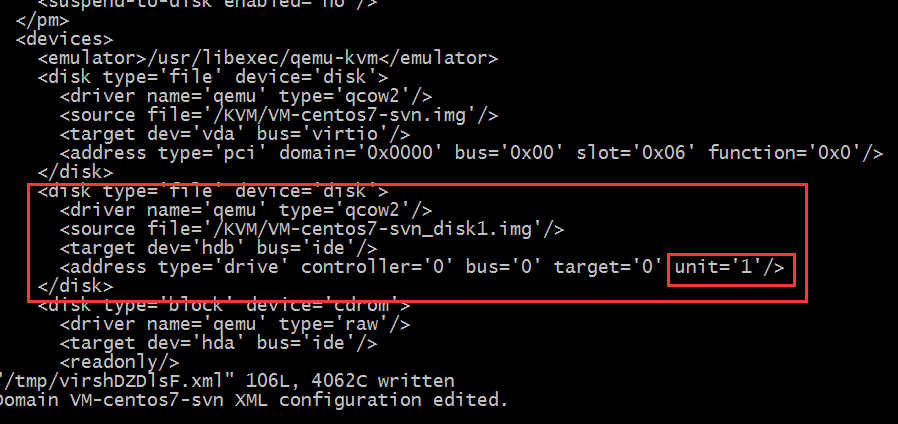
1. [root@localhost KVM]# qemu-img create -f qcow2 VM-centos7-svn\_disk1.img 100G

Formatting 'VM-centos7-svn\_disk1.img', fmt=qcow2 size=107374182400 encryption=off cluster\_size=65536 lazy\_refcounts=off #创建100G的新磁盘

刚创建时这个文件很小，只有100多k，随着往这个新建的磁盘中写入数据，硬盘也随之增大。

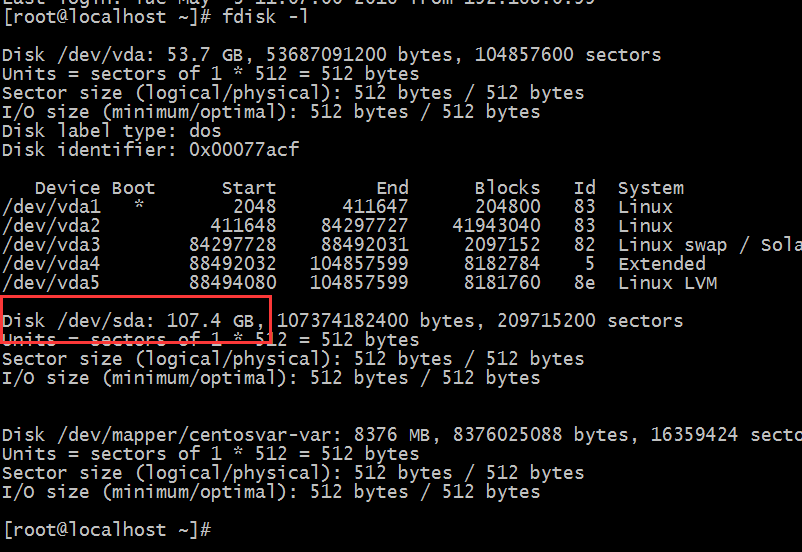


3、virsh edit VM-centos7-svn



4、virsh define VM-centos7-svn.xml #重新加载配置文件

5、[root@localhost KVM]# virsh start VM-centos7-svn #启动虚拟机



6、建lvm分区

[root@localhost ~]# fdisk /dev/sda

Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Device does not contain a recognized partition table

Building a new DOS disklabel with disk identifier 0x96647275.

Command (m for help): n

Partition type:

p primary (0 primary, 0 extended, 4 free)

e extended

Select (default p): p

Partition number (1-4, default 1):

First sector (2048-209715199, default 2048):

Using default value 2048

Last sector, +sectors or +size{K,M,G} (2048-209715199, default 209715199):

Using default value 209715199

Partition 1 of type Linux and of size 100 GiB is set

Command (m for help): t

Selected partition 1

Hex code (type L to list all codes): L

0 Empty 24 NEC DOS 81 Minix / old Lin bf Solaris

1 FAT12 27 Hidden NTFS Win 82 Linux swap / So c1 DRDOS/sec (FAT-

2 XENIX root 39 Plan 9 83 Linux c4 DRDOS/sec (FAT-

3 XENIX usr 3c PartitionMagic 84 OS/2 hidden C: c6 DRDOS/sec (FAT-

4 FAT16 <32M 40 Venix 80286 85 Linux extended c7 Syrinx

5 Extended 41 PPC PReP Boot 86 NTFS volume set da Non-FS data

6 FAT16 42 SFS 87 NTFS volume set db CP/M / CTOS / .

7 HPFS/NTFS/exFAT 4d QNX4.x 88 Linux plaintext de Dell Utility

8 AIX 4e QNX4.x 2nd part 8e Linux LVM df BootIt

9 AIX bootable 4f QNX4.x 3rd part 93 Amoeba e1 DOS access

a OS/2 Boot Manag 50 OnTrack DM 94 Amoeba BBT e3 DOS R/O

b W95 FAT32 51 OnTrack DM6 Aux 9f BSD/OS e4 SpeedStor

c W95 FAT32 (LBA) 52 CP/M a0 IBM Thinkpad hi eb BeOS fs

e W95 FAT16 (LBA) 53 OnTrack DM6 Aux a5 FreeBSD ee GPT

f W95 Ext'd (LBA) 54 OnTrackDM6 a6 OpenBSD ef EFI (FAT-12/16/

10 OPUS 55 EZ-Drive a7 NeXTSTEP f0 Linux/PA-RISC b

11 Hidden FAT12 56 Golden Bow a8 Darwin UFS f1 SpeedStor

12 Compaq diagnost 5c Priam Edisk a9 NetBSD f4 SpeedStor

14 Hidden FAT16 <3 61 SpeedStor ab Darwin boot f2 DOS secondary

16 Hidden FAT16 63 GNU HURD or Sys af HFS / HFS+ fb VMware VMFS

17 Hidden HPFS/NTF 64 Novell Netware b7 BSDI fs fc VMware VMKCORE

18 AST SmartSleep 65 Novell Netware b8 BSDI swap fd Linux raid auto

1b Hidden W95 FAT3 70 DiskSecure Mult bb Boot Wizard hid fe LANstep

1c Hidden W95 FAT3 75 PC/IX be Solaris boot ff BBT

1e Hidden W95 FAT1 80 Old Minix

Hex code (type L to list all codes): 8e

Changed type of partition 'Linux' to 'Linux LVM'

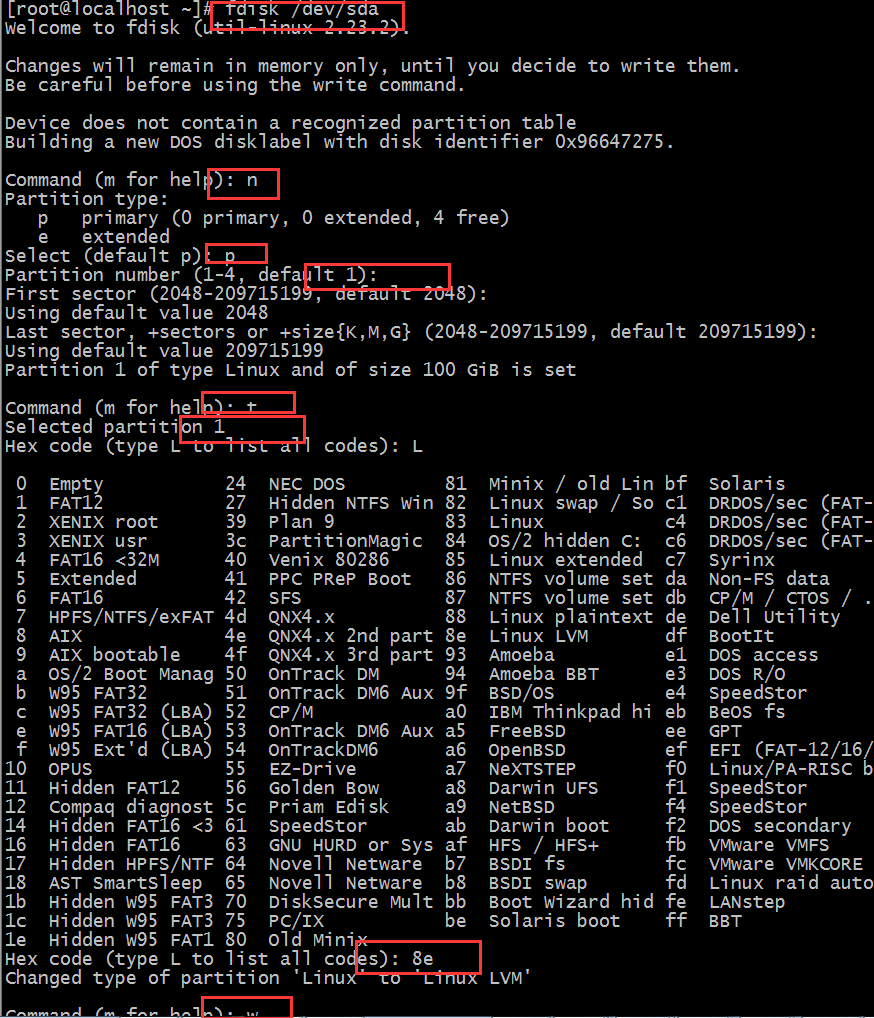
Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

[root@localhost ~]#



Disk /dev/vda: 53.7 GB, 53687091200 bytes, 104857600 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0x00077acf

Device Boot Start End Blocks Id System

/dev/vda1 \* 2048 411647 204800 83 Linux

/dev/vda2 411648 84297727 41943040 83 Linux

/dev/vda3 84297728 88492031 2097152 82 Linux swap / Solaris

/dev/vda4 88492032 104857599 8182784 5 Extended

/dev/vda5 88494080 104857599 8181760 8e Linux LVM

Disk /dev/sda: 107.4 GB, 107374182400 bytes, 209715200 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0x96647275

Device Boot Start End Blocks Id System

/dev/sda1 2048 209715199 104856576 8e Linux LVM

Disk /dev/mapper/centosvar-var: 8376 MB, 8376025088 bytes, 16359424 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

[root@localhost ~]# pvcreate /dev/sda1

Physical volume "/dev/sda1" successfully created

[root@localhost ~]# pvs

PV VG Fmt Attr PSize PFree

/dev/sda1 lvm2 --- 100.00g 100.00g

/dev/vda5 centosvar lvm2 a-- 7.80g 0

[root@localhost ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/vda2 40G 1.8G 36G 5% /

devtmpfs 488M 0 488M 0% /dev

tmpfs 497M 0 497M 0% /dev/shm

tmpfs 497M 6.6M 490M 2% /run

tmpfs 497M 0 497M 0% /sys/fs/cgroup

/dev/vda1 190M 99M 78M 56% /boot

/dev/mapper/centosvar-var 7.8G 813M 7.0G 11% /var

tmpfs 100M 0 100M 0% /run/user/0

[root@localhost ~]# vgextend centosvar /dev/sda1

Volume group "centosvar" successfully extended

[root@localhost ~]# pvs

PV VG Fmt Attr PSize PFree

/dev/sda1 centosvar lvm2 a-- 100.00g 100.00g

/dev/vda5 centosvar lvm2 a-- 7.80g 0

[root@localhost ~]# vgs

VG #PV #LV #SN Attr VSize VFree

centosvar 2 1 0 wz--n- 107.80g 100.00g

[root@localhost ~]# lvextend -l +100%FREE /dev/centosvar/var

Size of logical volume centosvar/var changed from 7.80 GiB (1997 extents) to 107.80 GiB (27596 extents).

Logical volume var successfully resized.

###如果是xfs格式的磁盘需要使用xfs\_growfs命令扩展，否则使用resize2fs 命令扩展######

[root@localhost ~]# xfs\_info /dev/centosvar/var

meta-data=/dev/mapper/centosvar-var isize=256 agcount=4, agsize=511232 blks

= sectsz=512 attr=2, projid32bit=1

= crc=0 finobt=0

data = bsize=4096 blocks=2044928, imaxpct=25

= sunit=0 swidth=0 blks

naming =version 2 bsize=4096 ascii-ci=0 ftype=0

log =internal bsize=4096 blocks=2560, version=2

= sectsz=512 sunit=0 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

[root@localhost ~]# xfs\_growfs /dev/centosvar/var

meta-data=/dev/mapper/centosvar-var isize=256 agcount=4, agsize=511232 blks

= sectsz=512 attr=2, projid32bit=1

= crc=0 finobt=0

data = bsize=4096 blocks=2044928, imaxpct=25

= sunit=0 swidth=0 blks

naming =version 2 bsize=4096 ascii-ci=0 ftype=0

log =internal bsize=4096 blocks=2560, version=2

= sectsz=512 sunit=0 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

data blocks changed from 2044928 to 28258304

[root@localhost ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/vda2 40G 1.8G 36G 5% /

devtmpfs 488M 0 488M 0% /dev

tmpfs 497M 0 497M 0% /dev/shm

tmpfs 497M 6.6M 490M 2% /run

tmpfs 497M 0 497M 0% /sys/fs/cgroup

/dev/vda1 190M 99M 78M 56% /boot

/dev/mapper/centosvar-var 108G 815M 107G 1% /var

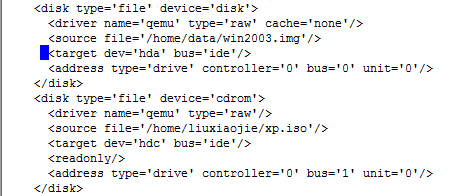
tmpfs 100M 0 100M 0% /run/user/0

[root@localhost ~]#

## KVM 中Windows虚拟机磁盘网卡优化virtio驱动

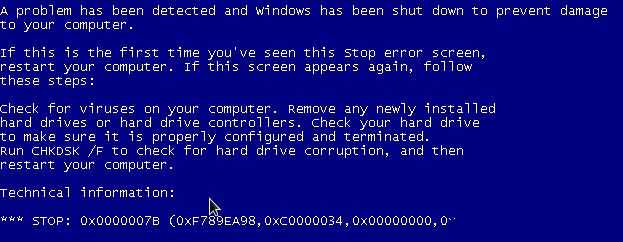
 1：默认情况下安装好[Windows](http://www.07net01.com/)[虚拟机](http://www.07net01.com/tags-%E8%99%9A%E6%8B%9F%E6%9C%BA-0.html)，采用普通的驱动，即硬盘和[网卡](http://www.07net01.com/r_s/)都采用默认[配置](http://www.07net01.com/r_s/)情况下，硬盘是 ide 模式，而[网卡](http://www.07net01.com/r_s/)工作在 模拟的rtl 8139 网卡下，速度为100M 全双工。采用 virtio 驱动后，网卡工作在 1000M 的模式下，硬盘工作是SCSI模式下。显然我想修改为virtio驱动！

2：开始我试着修改[配置文件](http://www.07net01.com/tags-%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6-0.html)（配置文件见附件）！然后重启虚拟机，结果蓝屏！



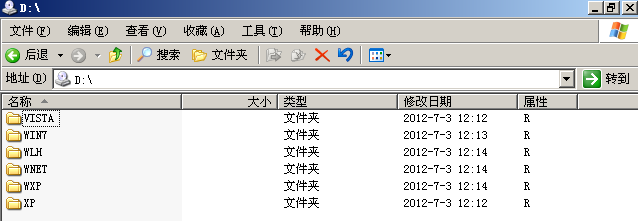
我试着device='disk' 部分的bus=‘ide’修改为virtio，并删除了<address type='drive' controller='0' bus='0' unit='0'/>，然后重启，结果蓝屏！

见附件



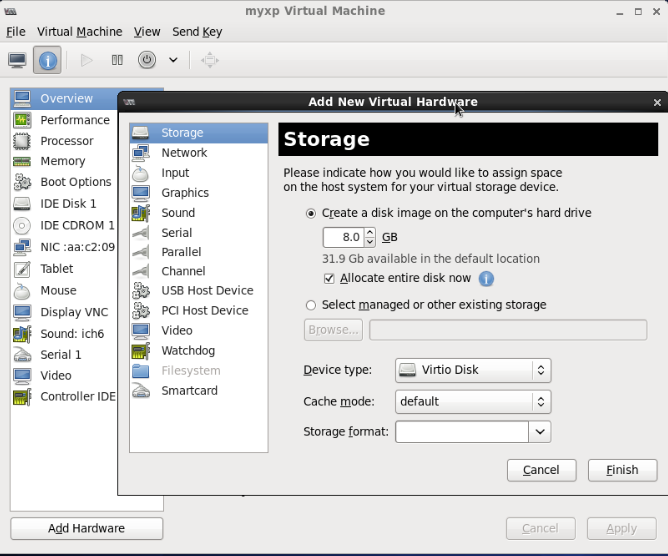
显然这样不行！

3：下载virtio.iso文件， 我使用的是 http://alt.fedoraproject.org/pub/alt/virtio-win/latest/images/[bin](http://www.07net01.com/linux/)/virtio-win-0.1-30.iso 这个iso文件支持几乎所有的windows系统。[挂载](http://www.07net01.com/linux/)后iso文件见附件！

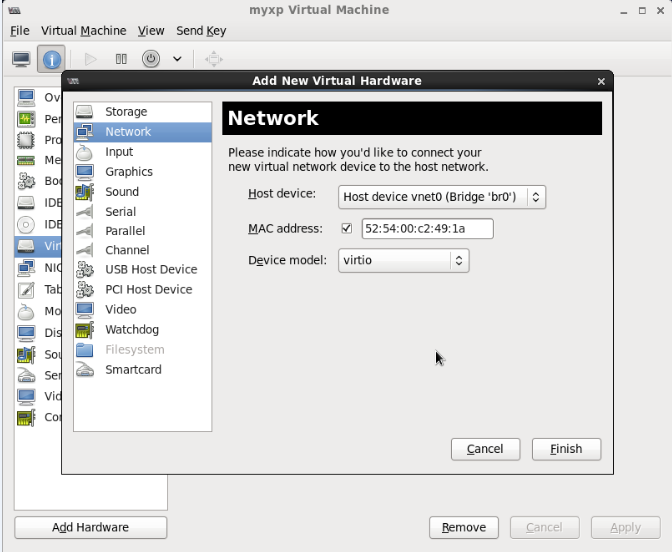


4：在配置中，添加virtio硬盘、virtio网卡，加载virtio-win.iso（cdrom）。启动vm

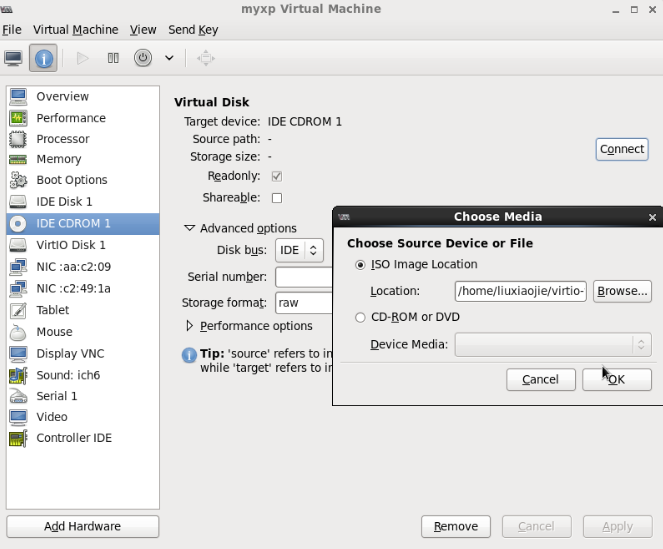
见附件：



尤其要注意Device type要选中Virtio Disk

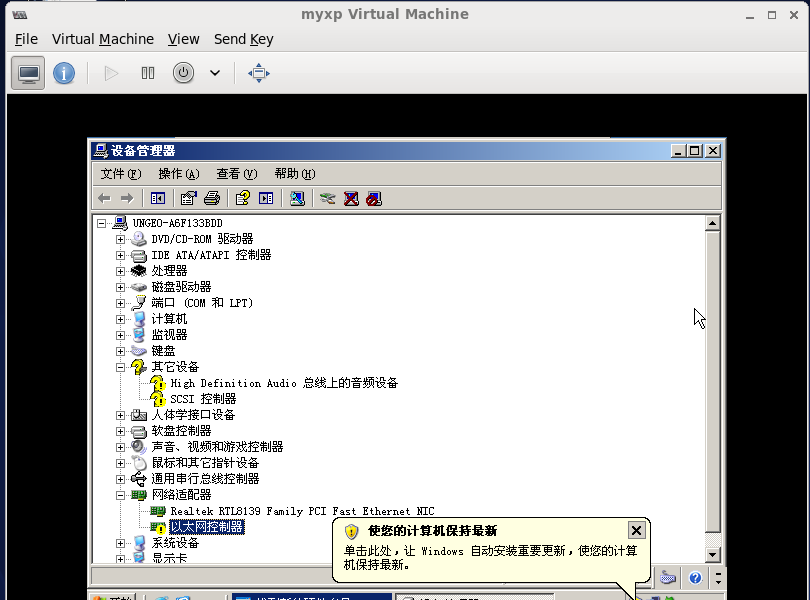


添加网卡的时候，尤其要注意Device model选中为virtio



选中/home/liuxiaojie/virtio-win-0.1-30.iso  待会重启vm后，需要通过光盘安装virtio的驱动！

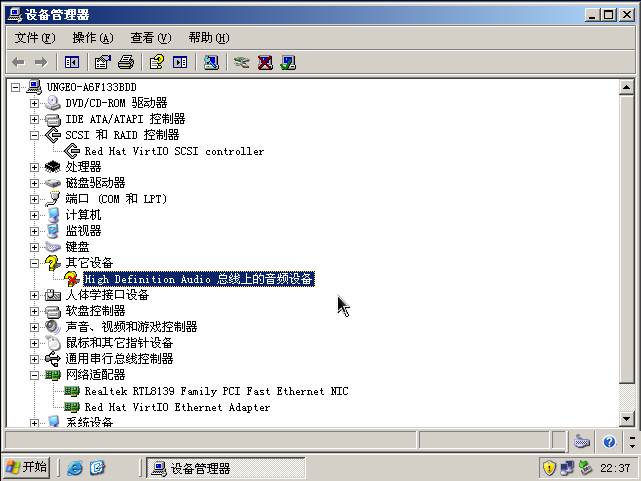
5：启动vm，通过设备管理器查看硬件信息，见附件，显然有两个硬件没有被驱动，我们需要光盘来安装驱动！系统会自动搜索 SCSI的驱动，安装即可，过程见附件



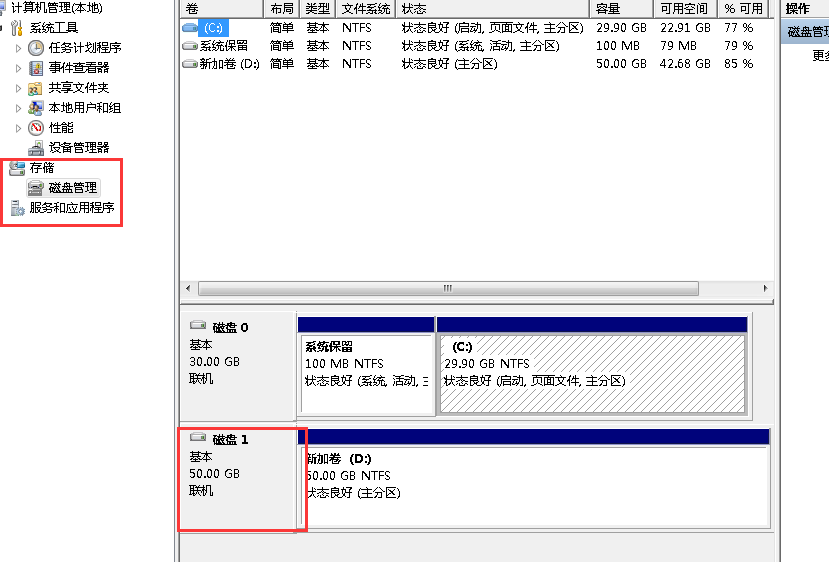
像安装普通的windows驱动一样，见附件



安装完成后，通过设备管理器查看如下：



6：删除掉新添加到硬盘和原来的网卡，修改ide硬盘为 virtio模式即可



## 快照

virsh snapshot-create VM-centos7-jira #创建

virsh snapshot-list VM-centos7-jira #查看

virsh snapshot-revert VM-centos7-jira 1465886025 #恢复

virsh snapshot-delete VM-centos7-jira 1465884367 #删除

## guestfish镜像工具套件

在宿主机管理虚拟机内部文件

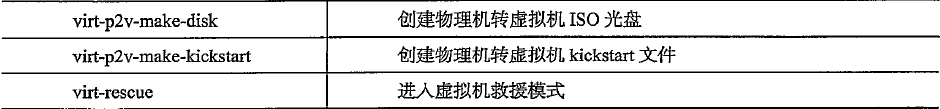
#查看虚拟机中的文件

virt-cat -d H231-centos6-Mysql-226 /etc/sysconfig/network-scripts/ifcfg-eth1

#编辑虚拟机中的文件（虚拟机必须处于关机状态）

virt-edit -d H231-centos6-Mysql-226 /etc/sysconfig/network-scripts/ifcfg-eth1







## 基本命令

常用

virsh vncdisplay VM-centos6.6-php #显示vcn端口

virsh autostart   VM-centos6.6-php     #设置开机自动启动虚拟机

virsh autostart --disable VM-centos7-jira #解除开机自动启动

virsh list --all #查看所有虚拟机状态

virsh destroy VM-centos6.6-php   #强行关机

virsh reboot VM-centos6.6-php          #重新启动一个虚拟机 

virsh参数如下：  
autostart         #开机自动启动虚拟机 virsh autostart Winxp  
connect           #重新连接到hypervisor    
console           #连接到客户会话    
create            #从一个SML文件创建一个虚拟机    
start             #开始一个非活跃的虚拟机    
destroy           #强行关闭虚拟机    
define            #从一个XML文件定义一个虚拟机    
domid             #把一个虚拟机名或UUID转换为ID    
domuuid           #把一个域名或ID转换为UUID    
dominfo           #查看虚拟机信息    
domstate          #查看虚拟机状态    
domblkstat        #获取虚拟机设备快状态    
domifstat         #获取虚拟机网络接口状态    
dumpxml           #XML中的虚拟机信息    
edit              #编辑某个虚拟机的XML文件    
list              #列出虚拟机    
migrate           #将虚拟机迁移到另一台主机    
quit              #退出非交互式终端    
reboot            #重新启动一个虚拟机    
resume            #重新恢复一个虚拟机    
save              #把一个虚拟机的状态保存到一个文件    
dump              #把一个虚拟机的内核dump到一个文件中以方便分析    
shutdown          #关闭一个虚拟机    
setmem            #改变内存的分配    
setmaxmem         #改变最大内存限制值    
suspend           #挂起一个虚拟机    
vcpuinfo          #虚拟机的cpu信息    
version           #显示virsh版本