
第一章 Linux 桌面虚拟化技术 KVM

本节所讲内容：

- 1.1 虚拟化产品对比介绍
- 1.2 安装 KVM 虚拟机
- 1.3 实战 1：配置 KVM 网络桥接功能
- 1.4 实战 2：使用 KVM 安装虚拟机
- 1.5 实战 3：解决 centos6 下 shutdown 关不了 KVM 虚拟机的问题

实验环境：

第四阶段以 centos7.4 为主

centos7.4 镜像下载：http://mirrors.163.com/centos/7.4.1708/isos/x86_64/

注：安装系统时，把图形界面和开发工具安装好

1.1 虚拟化产品对比介绍

1.1.1 虚拟化产品对比介绍

vmware

KVM rhel6_x64

xen [kernel-xen] rhel5

虚拟化技术：

仿真虚拟化 [对系统硬件没有要求,性能最低]

半虚拟化 [虚拟机可以使用真机物理硬件，性能高，需要改内核]

全虚拟化 直接使用物理硬件，性能高

vmware 支持仿真虚拟化

xen 半虚 REHL5 自带 xen, 安装时需要安装内核 rpm -ivh kernel-xen-xxx.rpm

kvm 全虚拟化 RHEL6 自带 kvm

KVM 概述：

KVM 即 Kernel-based Virtual Machine 基于内核的虚拟机。

KVM，是一个开源的系统虚拟化模块，自 Linux 2.6.20 之后集成在 Linux 的各个主要发行版本中。它使用 Linux 自身的调度器进行管理，所以相对于 Xen，其核心源码很少。KVM 目前已成为学术界的主流 VMM(虚拟机监控器)之一。**KVM 的虚拟化需要硬件支持 (如 Intel VT 技术或者 AMD V 技术)。是基于硬件的完全虚拟化。**而 Xen 早期则是基于软件模拟的 Para-Virtualization。

查看 CPU 是否支持虚拟化

Intel 酷睿 i5 4590 参数

在中关村在线：

<http://detail.zol.com.cn/382/381245/param.shtml> 查：Intel 酷睿 i5 4590 参数

技术参数	指令集 ⓘ	SSE 4.1/4.2 , AVX 2.0
	内存控制器	双通道 : DDR3 1333/1600
	支持最大内存	32GB
	超线程技术 ⓘ	不支持
	<u>虚拟化技术</u>	<u>Intel VT-x</u> 
	64位处理器	是
	Turbo Boost技术	支持

KVM : 是指基于 Linux 内核的虚拟机(Kernel-based Virtual Machine)。是第一个整合到 Linux 内核的虚拟化技术。在 KVM 模型中, 每一个虚拟机都是一个由 Linux 调度程序管理的标准进程, 你可以在**用户空间启动客户机操作系统**。

一个普通的 Linux 进程有两种运行模式: 内核和用户。KVM 增加了第三种模式: 客户模式(有自己的内核和用户模式)

XEN 需要升级内核, 只能支持和物理机系统一样的操作系统。 xen 虚拟上, 可以运行 windows ?

KVM : 支持 linux 以外的其它系统。比如: windows

QEMU : 是一套由 Fabrice Bellard 所编写的以 GPL 许可证分发源码的**模拟处理器**, 在 GNU/Linux 平台上使用广泛。QEMU 具有高速度和跨平台的特性, QEMU 能模拟至接近真实电脑的速度。

QEMU 能模拟整个电脑系统, 包括中央处理器及其他周边设备。

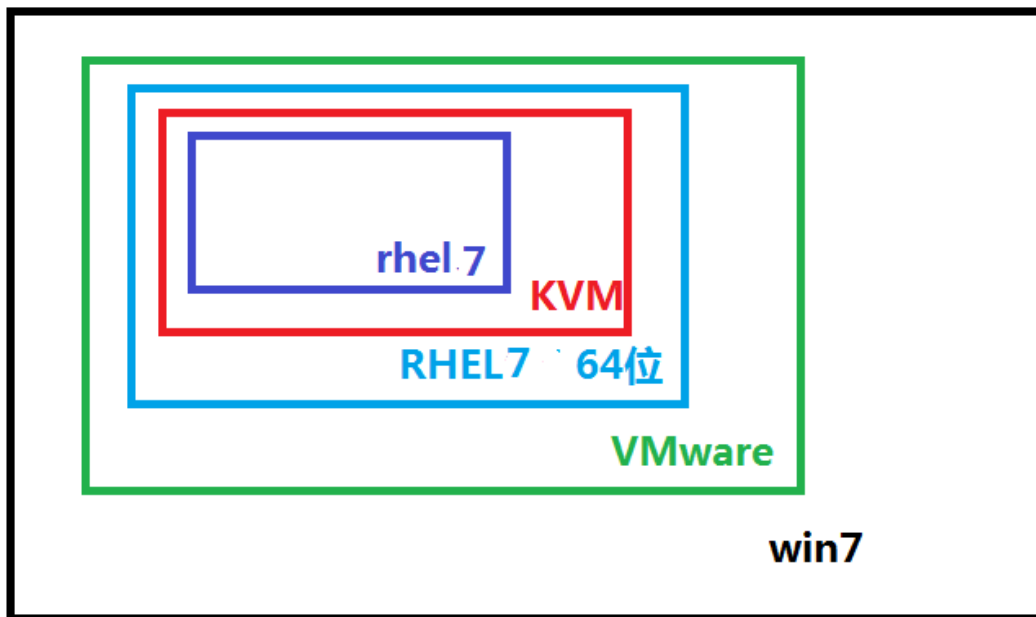
QEMU 和 vmware 一样都是支持仿真虚拟化, 效率比较低。

扩展 :

网络模拟器 : GNS3 , Packet tracer , IOU (考 CCIE)

1.1.2 配置 KVM 前期准备工作 :

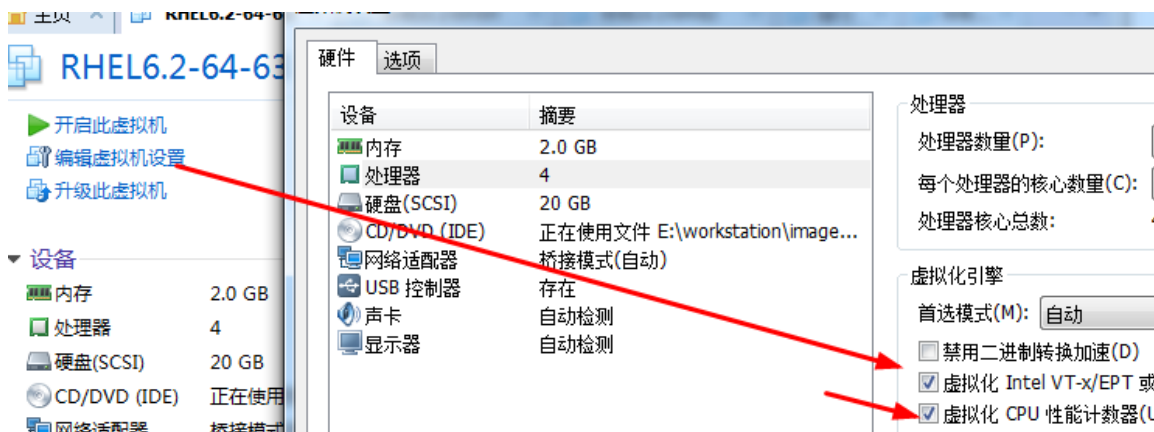
KVM 实验拓扑图 :



1、把虚拟机内存调成 2G 以上，因为我们要在 VMware 虚拟中安装 KVM,然后在 KVM 中再安装虚拟机



2、开启 vmware 中虚拟化支持



互动：虚拟化 cpu 性能计数器作用是什么？ 百度很难找到，怎么办？

你和百度上的大牛在多大差距？ 就差一个查看官方文档过程

虚拟化 cpu 性能计数器作用：

<http://pubs.vmware.com/workstation-12/index.jsp>

VMware 官网手册里有说明



虚拟化cpu性能计数器



目录

索引

搜索结果

显示搜索结果

在所有主题 中找到 1 [更改范围](#)

配置虚拟机处理器设置

您可以配置虚拟机的处理器设置，包括处理器数量、每个处理器的核心数，以及虚拟化引擎的首选执行模式。

禁用二进制转换加速

虚拟化 Intel VT-x/EPT 或 AMD-V/RVI

虚拟化 CPU 性能计数器

虚拟化 Intel VT-x/EPT 或 AMD-V/RVI

Workstation Pro 强制将虚拟机执行模式设为 VT-x/EPT 或 AMD-V/RVI。必须启用物理地址扩展 (PAE) 模式才能使用虚拟化 AMD-V/RVI。

如果执行模式不受主机系统支持，虚拟化 VT-x/EPT 或 AMD-V/RVI 将不可用。如果您将虚拟机迁移到其他 VMware 产品，虚拟化 VT-x/EPT 或 AMD-V/RVI 也可能不可用。

注

无法为共享或远程虚拟机配置此设置。

虚拟化 CPU 性能计数器

如果您打算使用性能监控应用程序（如 VTune 或 OProfile）优化或调试虚拟机中运行的软件，可以打开此功能。

仅在虚拟机兼与 Workstation 9 或更高版本兼容时，才可以使用该功能。

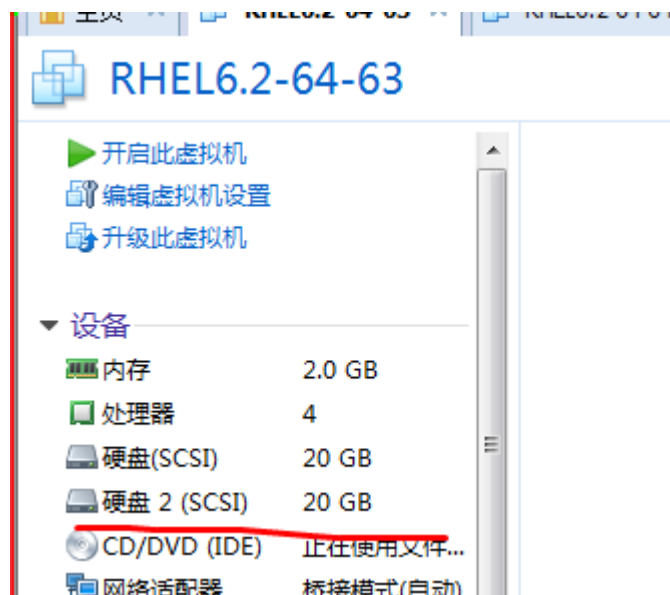
大家安装 KVM 时，不需要把这个打上对勾。

自选模式(M): 自动

☐ 禁用二进制转换加速(D)☐ 虚拟化 Intel VT-x/EPT 或 AMD-V/RVI(V)☐ 虚拟化 CPU 性能计数器(U)

扩展：后期大家遇到很多课堂中没有讲到的知识，你可以看官方文档。

3、添加一个 20G 的硬盘，用于存 KVM 虚拟机



注：只有 64 位 RHEL6 以上系统支持 KVM。32 位系统不支持。

4、查看 CPU 是否支持硬件虚拟化技术。 CPU 要支持

查看自己的 CPU 是否支持全虚拟化虚拟化技术且是 64 位的

Intel : `cat /proc/cpuinfo | grep --color vmx`

AMD : `cat /proc/cpuinfo | grep --color svm`

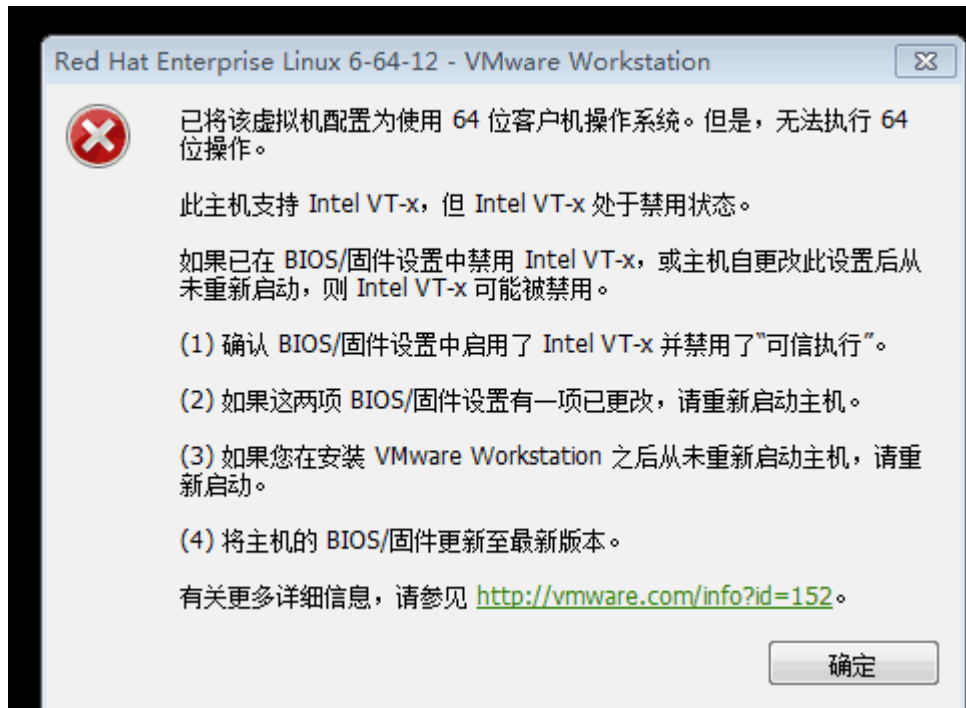
看看 flag 有没有上面的 vmx 或者是 svm，有的话就是支持全虚拟化技术

5、开启 BIOS 虚拟化支持。 查找 VT 改为：enable 。BIOS 中也要打开，默认很多机器没有打开。

例：

排错：

安装虚拟机出现 intel vt -x 处于禁用状态打不开 处理方式



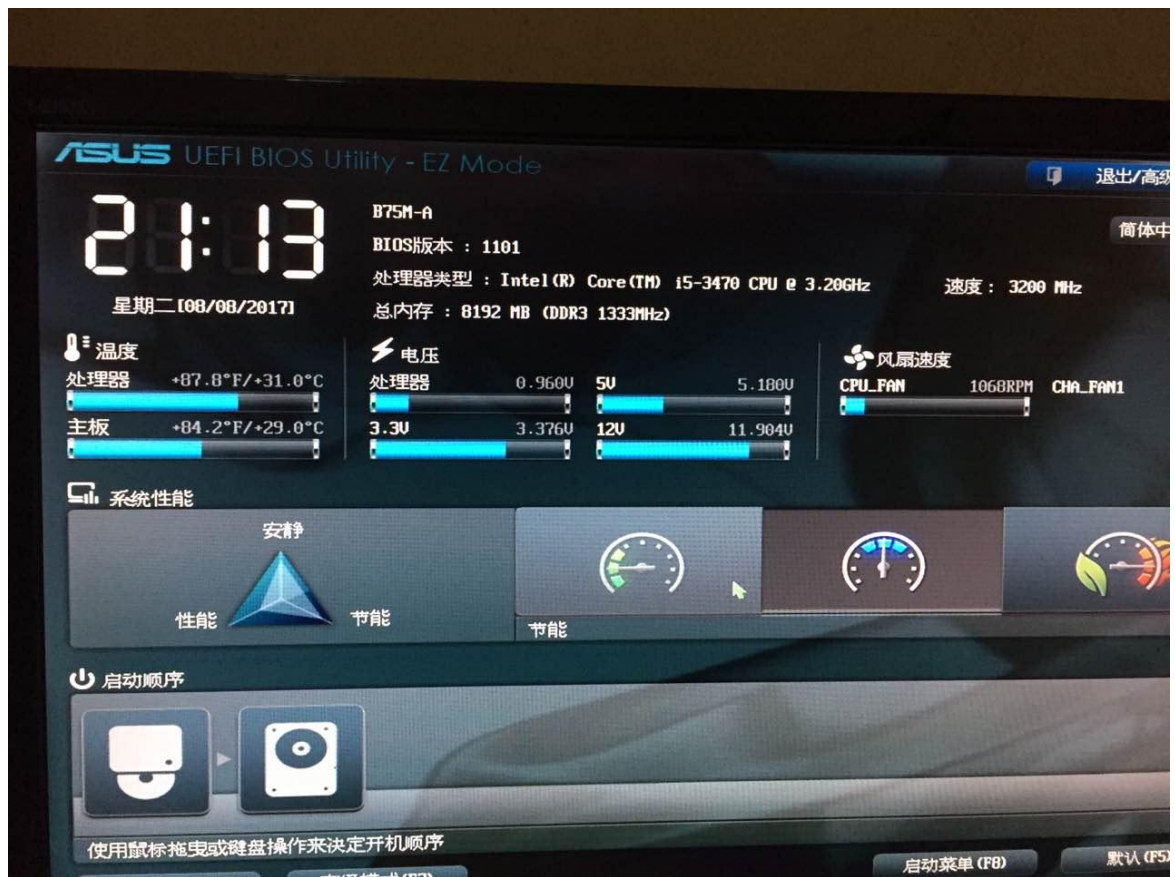
注：参加培训这 4 个月，除了掌握了知识？学习方法。报错后，先看懂错，然后再解决。差得是环境？

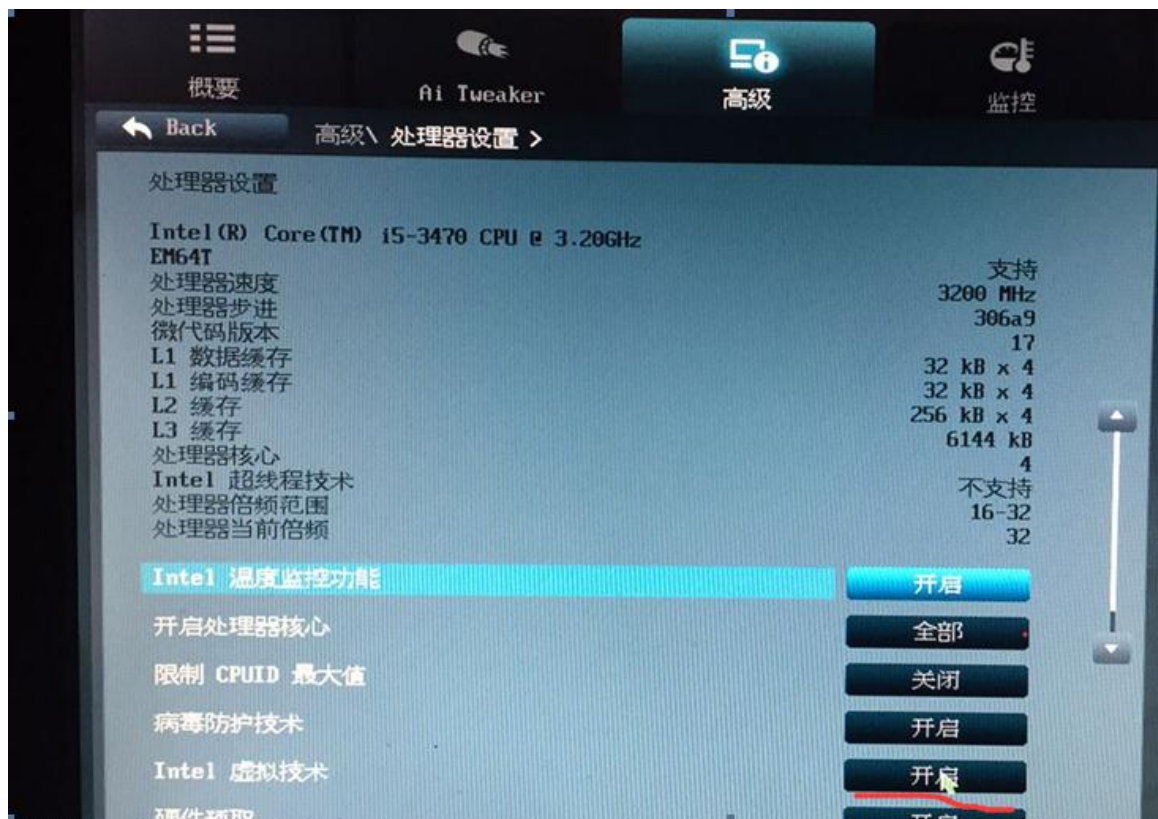
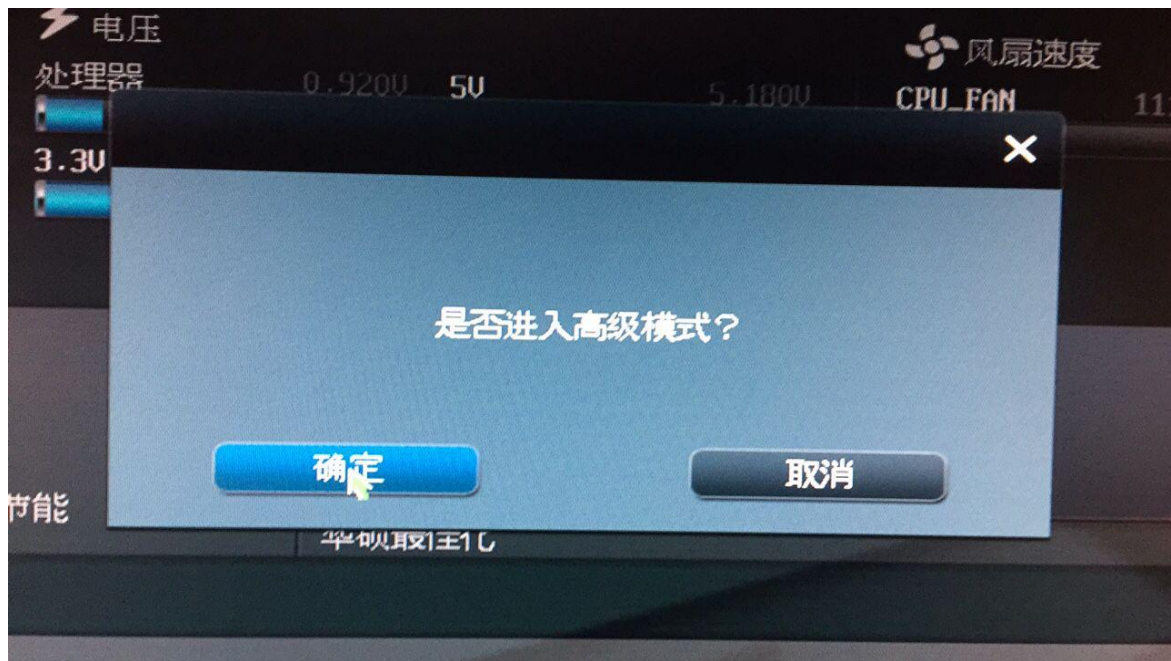
处理方式

1 进入 bios

以华硕主板为例：UEFI 没见过：1 见过：6

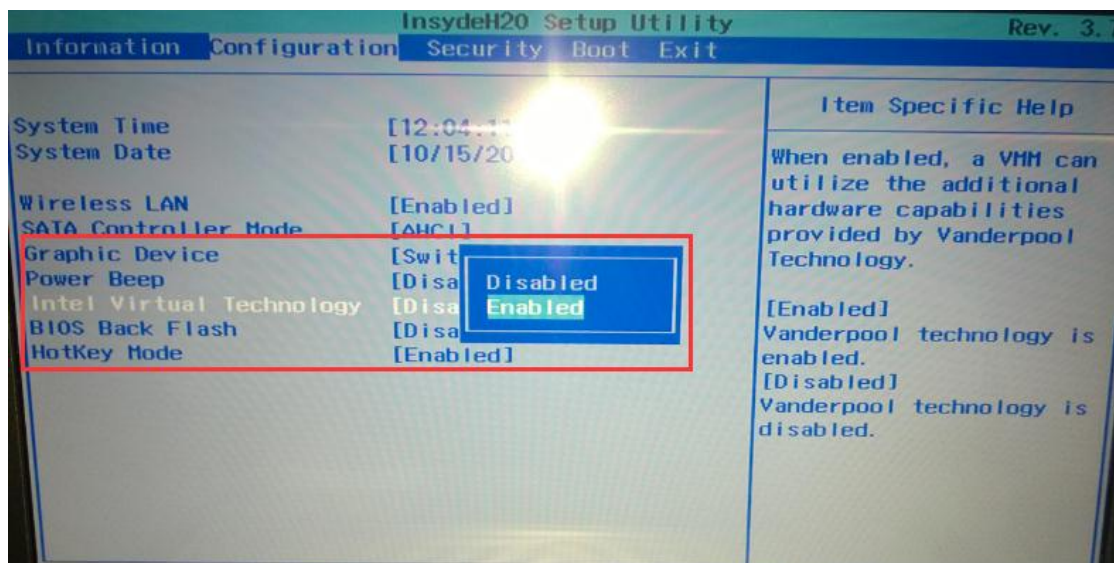
UEFI 支持 GPT 分区表，对 4T 以上硬盘支持比较好。intel 和微软 一起研发。





步骤：进入高级模式---进入高级菜单---开启 intel 虚拟技术

情况 2：传统的 BIOS 界面，找到这个



F10 保存退出。

2、操作系统必须是 RHEL6 64 位以上， RHEL6 32 不能安装 KVM

1.2 安装 KVM 虚拟机

1.2.1 开机后查看 CPU 是否支持全虚拟化

1、查看 CPU 是否支持硬件虚拟化技术。 CPU 要支持

例：

查看自己的 CPU 是否支持全虚拟化虚拟化技术且是 64 位的

Intel : `cat /proc/cpuinfo | grep --color vmx`

AMD : `cat /proc/cpuinfo | grep --color svm`

看看 flag 有没有上面的 vmx 或者是 svm，有的话就是支持全虚拟化技术

[root@xuegod63 Desktop]# `cat /proc/cpuinfo | grep --color vmx`

```
flags                : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov
pat pse36 clflush dts mmx fxsr sse sse2 ss ht syscall nx rdtscp lm constant_tsc arch_perfmon
pebs bts xtopology tsc_reliable nonstop_tsc aperfmperf unfair_spinlock pni pclmulqdq vmx
ssse3 cx16 sse4_1 sse4_2 x2apic popcnt aes xsave avx hypervisor lahf_lm ida arat epb
xsaveopt pln pts dts tpr_shadow vnmi ept vpid
```

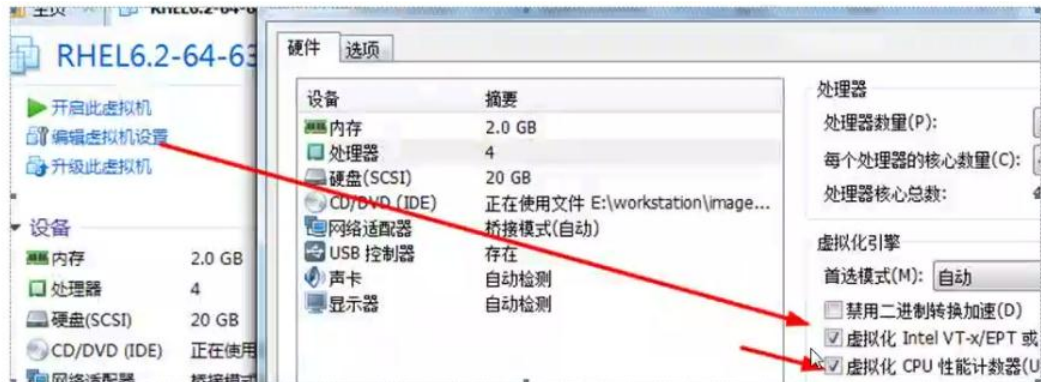
```
flags                : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov
pat pse36 clflush dts mmx fxsr sse sse2 ss ht syscall nx rdtscp lm constant_tsc arch_perfmon
pebs bts xtopology tsc_reliable nonstop_tsc aperfmperf unfair_spinlock pni pclmulqdq vmx
ssse3 cx16 sse4_1 sse4_2 x2apic popcnt aes xsave avx hypervisor lahf_lm ida arat epb
xsaveopt pln pts dts tpr_shadow vnmi ept vpid
```

```
flags                : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov
pat pse36 clflush dts mmx fxsr sse sse2 ss ht syscall nx rdtscp lm constant_tsc arch_perfmon
pebs bts xtopology tsc_reliable nonstop_tsc aperfmperf unfair_spinlock pni pclmulqdq vmx
ssse3 cx16 sse4_1 sse4_2 x2apic popcnt aes xsave avx hypervisor lahf_lm ida arat epb
xsaveopt pln pts dts tpr_shadow vnmi ept vpid
```

```
flags                : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov
pat pse36 clflush dts mmx fxsr sse sse2 ss ht syscall nx rdtscp lm constant_tsc arch_perfmon
```

pebs bts xtopology tsc_reliable nonstop_tsc aperfmperf unfair_spinlock pni pclmulqdq **vmx**
ssse3 cx16 sse4_1 sse4_2 x2apic popcnt aes xsave avx hypervisor lahf_lm ida arat epb
xsaveopt pln pts dts tpr_shadow vnmi ept vpid

如果，看不到 **vmx**，是因为 VMware CPU 没有开启 VT 技术



说明：安装前：vmware 虚拟机内存要给的大一些。比如 2G。再添加一个 20G 硬盘，防止后期系统空间不够。安装虚拟机占用空间比较大。

1.2.2 安装 KVM：

(1).安装 KVM 模块、管理工具和 libvirt，命令行安装：

```
[root@xuegod63 ~]#
```

```
yum install qemu-kvm libvirt libguestfs-tools virt-install virt-manager libvirt-python -y
```

注：使用系统镜像，先配置好 yum 本地源

注：

qemu-kvm：kvm 主程序，KVM 虚拟化模块

virt-manager：KVM 图形化管理工具

libvirt：虚拟化服务

libguestfs-tools：虚拟机的系统管理工具

virt-install：安装虚拟机的实用工具。比如 virt-clone 克隆工具就是这个包安装的

libvirt-python：python 调用 libvirt 虚拟化服务的 api 接口库文件

1.2.3 查看安装完 KVM 后的服务：

```
[root@xuegod63 ~]# systemctl start libvirtd #开启虚拟化服务
```

```
[root@xuegod63 ~]# systemctl enable libvirtd #设置 libvirtd 服务开机启动
```

```
[root@xuegod63 ~]# systemctl is-enabled libvirtd #查看是不是开机启动 enabled
```

确定正确加载 kvm 模块

```
[root@xuegod63 ~]# lsmod | grep kvm
```

```
kvm_intel          170086  0
```

```
kvm                566340  1 kvm_intel
```

```
irqbypass         13503  1 kvm
```

检查 KVM 模块是否成功安装

1.2.4 使用命令：virt-manager 建立虚拟机，

将 kvm 管理工具从英文界面，切换成中文界面：

```
[root@xuegod63 ~]# echo $LANG
```

en_US.UTF-8

```
[root@xuegod63 ~]# LANG='zh_CN.UTF-8'
```

```
[root@xuegod63 ~]# virt-manager
```

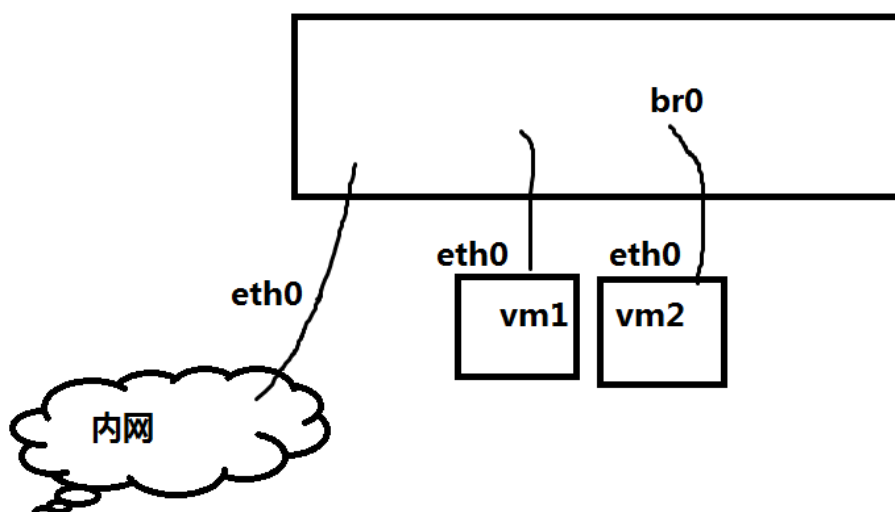
(1)、执行 virt-manager 后，弹出如下界面：



1.3 实战 1：配置 KVM 网络桥接功能

网桥介绍：我们经常所说的 Bridge 设备其实就是网桥设备，也就相当于现在的二层交换机，用于连接同一网段内的所有机器，所以我们的目的就是将网络设备 eth0 添加到 br0，此时 br0 就成为了所谓的交换机设备，我们物理机的 eth0 也是连接在上面的。

添加桥接设备 br0：相当于一个二层交换机



安装桥接设备工具：

```
[root@xuegod63 ~]# rpm -ivh /mnt/Packages/bridge-utils-1.2-9.el6.x86_64.rpm
```

把 eth0 绑定 br0 桥设备上：

```
[root@xuegod63 ~]# cd /etc/sysconfig/network-scripts/
[root@xuegod63 network-scripts]# cp ifcfg-ens33 /tmp/          #备份一下 eth0
[root@xuegod63 network-scripts]# vim ifcfg-ens33  #编辑配置文件为以下内容
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=c713acec-674b-411d-9e61-646482a292ca
DEVICE=ens33
IPADDR=192.168.1.63  #这些蓝色内容接下来，需要删除
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=202.106.46.151
ONBOOT=yes
IPV6_PRIVACY=no
PEERDNS=no
BRIDGE="br0"  #在文件最后插入这一行
```

生成桥设备的配置文件：

```
[root@xuegod63 network-scripts]# vim ifcfg-br0  #创建 ifcfg-br0 文件，并写入以下内容
DEVICE="br0"
NM_CONTROLLED="yes"
ONBOOT="yes"
TYPE="Bridge"
BOOTPROTO=none
IPADDR=192.168.1.63
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=192.168.1.1
注：TYPE="Bridge" ， B要大写
```

```
[root@xuegod63 network-scripts]# service network restart
```

测试 br0：

```
root@xuegod63 network-scripts]# ifconfig
```

```
[root@xuegod63 network-scripts]# ping g.cn
PING g.cn (203.208.37.20) 56(84) bytes of data.
64 bytes from 203.208.37.20: icmp_seq=1 ttl=57 time=12.3 ms
```

查看桥接的信息：

```
[root@xuegod63 network-scripts]# brctl show
bridge name    bridge id        STP enabled    interfaces
br0            8000.000c290283db  no            ens33
virbr0         8000.525400183a62  yes           virbr0-nic
```

1.4 实战 2：创建一台 KVM 虚拟机

1.4.1 创建一个分区，用于存放安装好的 Linux 操作系统：

```
[root@xuegod63 ~]# fdisk /dev/sdb  #创建 sdb1，大小 20G
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xe2316600.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').
```

Command (m for help): n

Command action

 e extended

 p primary partition (1-4)

p

Partition number (1-4): 1

First cylinder (1-2610, default 1):

Using default value 1

Last cylinder, +cylinders or +size{K,M,G} (1-2610, default 2610):

Using default value 2610

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

格式化，挂载使用

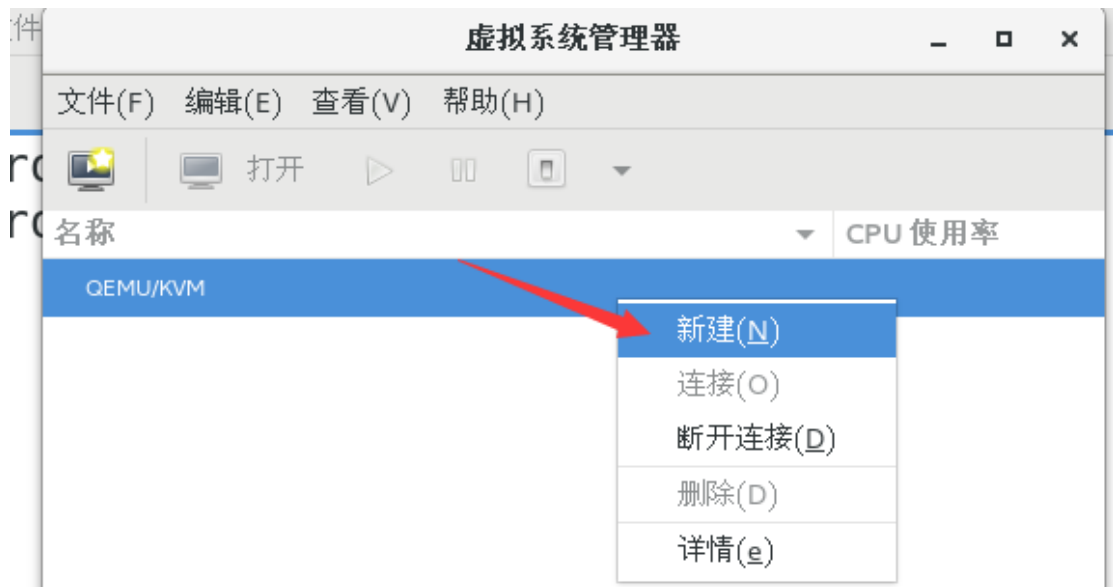

```
[root@xuegod63 network-scripts]# mkfs.xfs /dev/sdb1
```

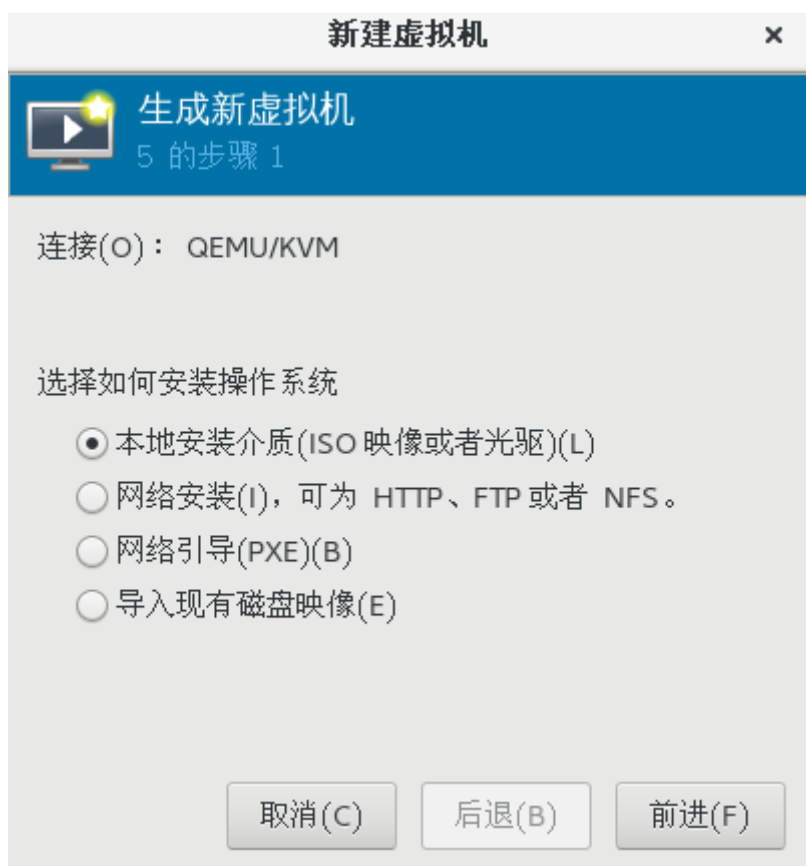
```
[root@xuegod63 network-scripts]# mount /dev/sdb1 /var/lib/libvirt/images/ #安装虚拟机，默认存放的路径：
```

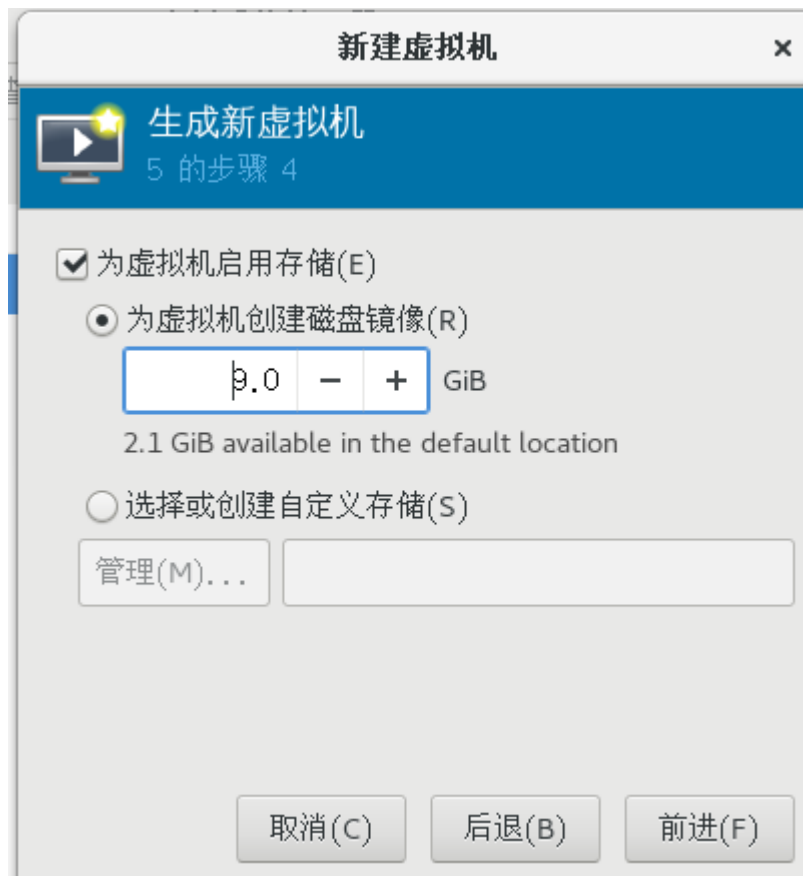
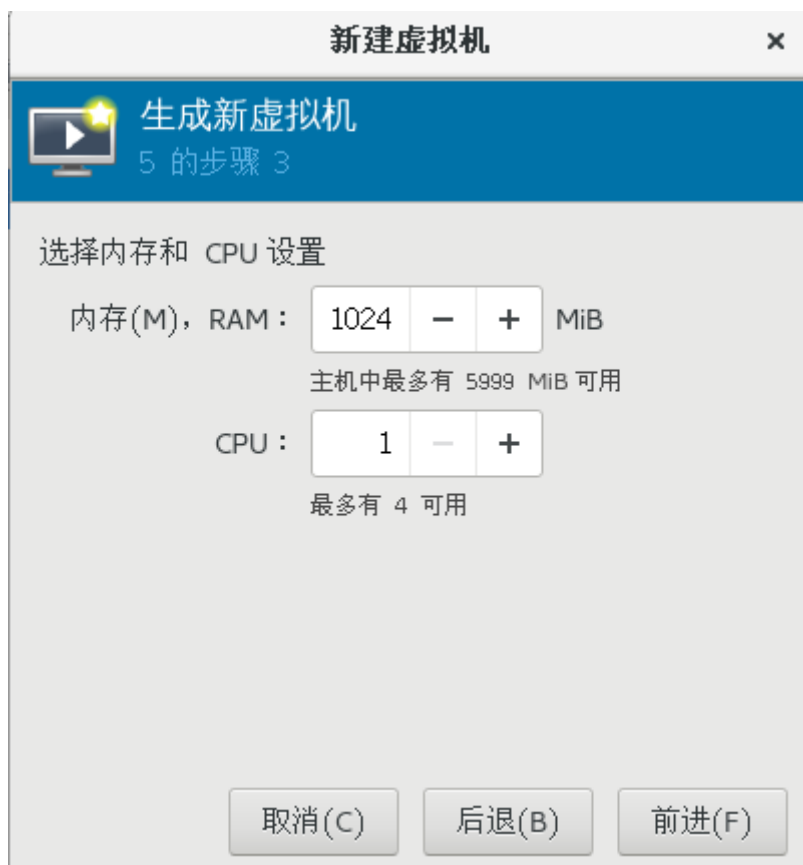
注：准备系统镜像：把 centos7.4 镜像上传到/var/lib/libvirt/images/目录下。

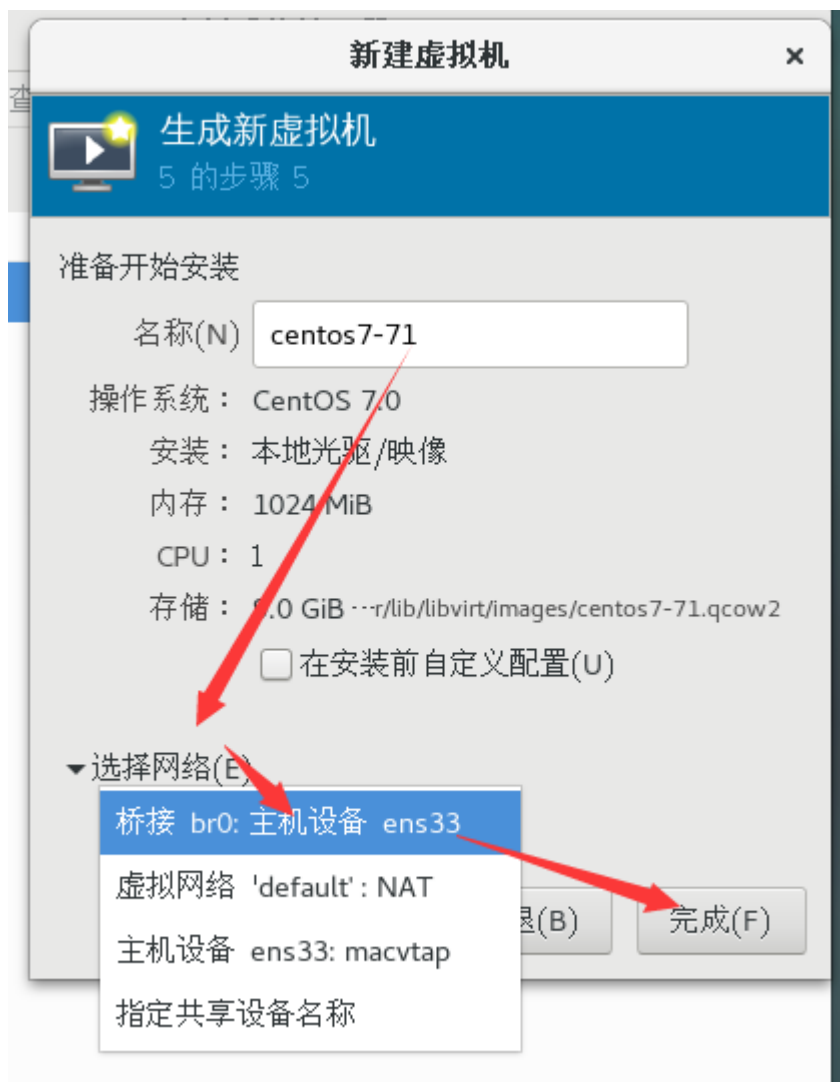
注：准备系统镜像：把 centos7.4 镜像上传到/var/lib/libvirt/images/目录下。

```
[root@xuegod63 ~]# virt-manager
```



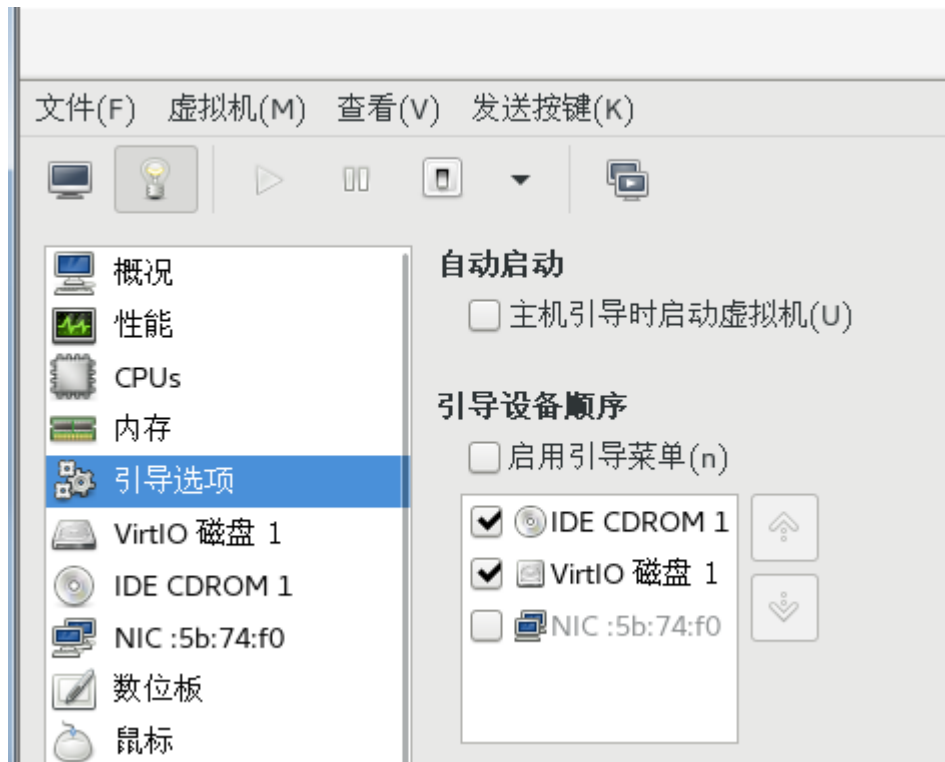




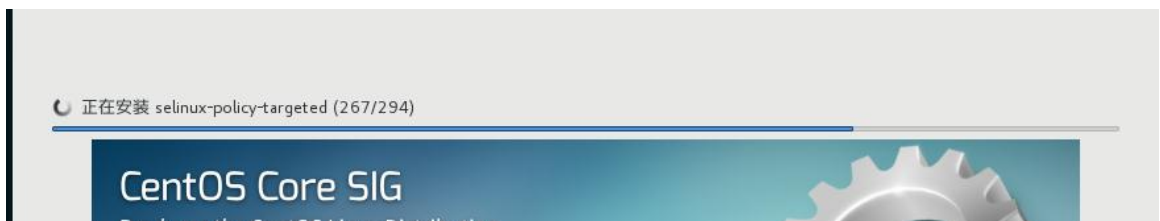


点“完成”到此创建好一个新的 KVM 虚拟机了。

1.4.2 修改 KVM 中 BIOS 引导顺序



这里为测试，选择最小化安装：



1.4.3 KVM 虚拟机常用命令

```
[root@xuegod63 ~]# virsh list           #列出在运行的虚拟机
[root@xuegod63 ~]# virsh start centos7-71 #启动 centos7-71 虚拟机
[root@xuegod63 ~]# virsh shutdown centos7-71 #关闭 centos7-71 虚拟机
[root@xuegod63 ~]# virsh autostart centos7-71 #设置 centos7-71 虚拟机为物理机开机
```

后，自动启动

测试：

reboot 后，没有发现 kvm 虚拟机开机自动启动。原因是什么？

解决：

```
[root@xuegod63 ~]# chkconfig --list libvirtd #开机启动了
libvirtd      0:off    1:off    2:off    3:on 4:on 5:on 6:off
[root@xuegod63 ~]# vim /etc/fstab #记得设置开机自动挂载 sdb1，不然后开机启动不了虚
```

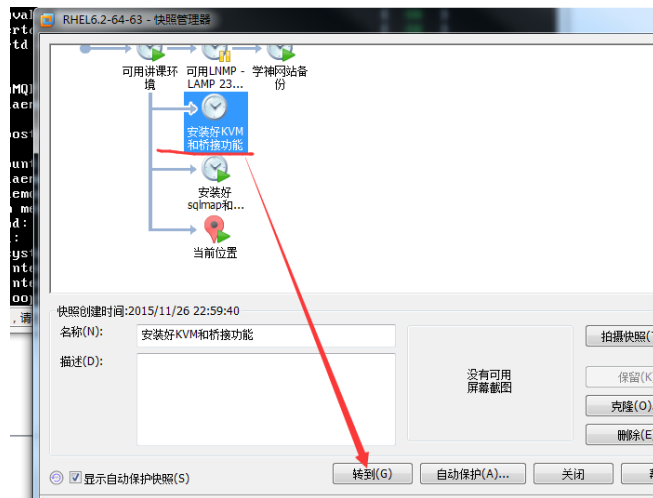
拟机

```
/dev/sdb1          /var/lib/libvirt/images xfs defaults 0 0
```

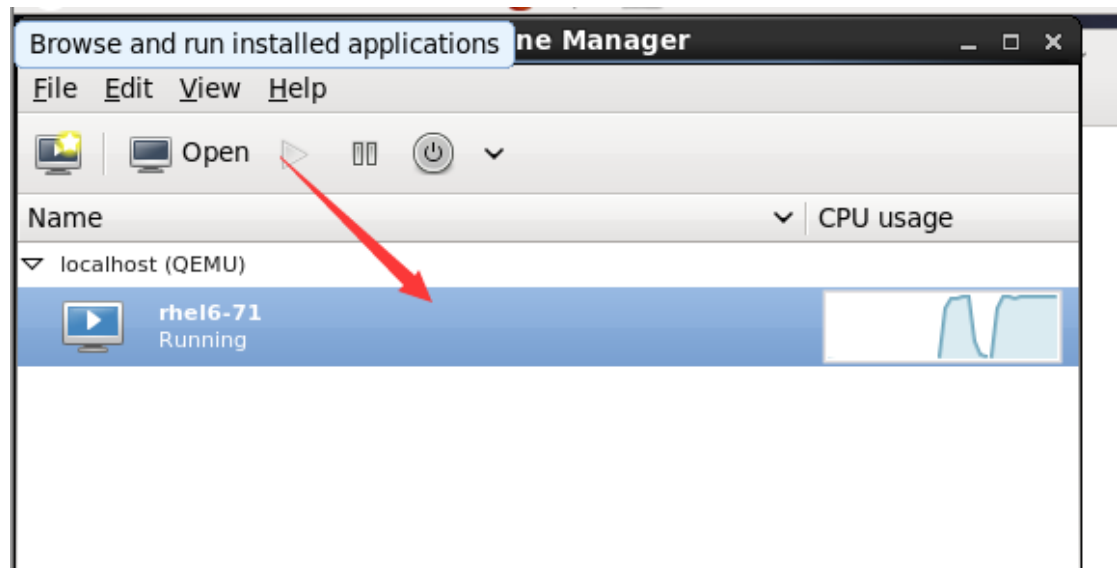
1.5 解决 centos6 系统上 shutdown 关不了虚拟机的问题

了解一下这个坑。

1.5.1 实验环境准备：



```
[root@xuegod63 ~]# mount /dev/sdb1 /var/lib/libvirt/images/
```



1.5.2. 解决方法：

在安装好的 KVM 的 Linux 虚拟机中**安装并启动 acpid 服务**

virsh shutdown 命令使用发送 acpi 指令来控制虚拟机的电源，

而 kvm 虚拟机**最小化安装的 centos6 系统**时默认是没有安装 acpi 服务的，所以并不会处理 shutdown 发来的关机信号。在 centos7 最小化安装的 linux 系统上有安装 acpid 服务。

只需要在虚拟机里安装和启动 acpid 服务即可。

做一件事：安装 acpid 服务 -> yum 源

1、给 KVM 虚拟机配置个 IP

安装

配置网卡 IP 地址：

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE="eth0"
```

```
HWADDR="52:54:00:95:89:DE"
```

```
IPADDR=192.168.1.71
```

NETMASK=255.255.255.0

GATEWAY=192.168.1.1

DNS1=192.168.1.1

NM_CONTROLLED="yes"

ONBOOT="yes"

[root@localhost ~]# service network restart #启动网卡

在 xuegod63 上搭建一个基于 http 协议 yum 源服务

[root@xuegod63 ~]# yum install httpd -y

[root@xuegod63 ~]# service httpd start

[root@xuegod63 ~]# mount /dev/cdrom /var/www/html/

配置 kvm 虚拟机 rhel6-71 的 yum 源

[root@xuegod63 ~]# ssh 192.168.1.71

[root@localhost ~]# cd /etc/yum.repos.d

[root@localhost ~]# vi rhel6.repo # 写入以下内容

[rhel6]

name= rhel6

baseurl=http://192.168.1.63/

#baseurl=file:///mnt/

enabled=1

gpgcheck=0

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[root@localhost ~]#rm -rf rhel-source.repo #删除多余的 yum 源配置文件

[root@localhost ~]# setenforce 0 #关闭 selinux

[root@localhost ~]#iptables -F #关闭防火墙

在 kvm 虚拟机 rhel6-71 安装 acpid 服务

[root@localhost yum.repos.d]# yum install acpid -y

[root@localhost yum.repos.d]# service acpid start //启动 acpic 服务 ,

[root@localhost yum.repos.d]# chkconfig --list acpid

acpid 0:off 1:off 2:on 3:on 4:on 5:on 6:off //安装后默认会加入到开机

启动的

在 xuegod63 KVM 真机上测试 :

[root@xuegod63 ~]# virsh shutdown rhel6-71 #关闭 rhel6-71 虚拟机

总结 :

1.1 虚拟化产品对比介绍

1.2 安装 KVM 虚拟机

1.3 实战 1 : 配置 KVM 网络桥接功能

1.4 实战 2 : 使用 KVM 安装虚拟机

1.5 实战 3：解决 centos6 下 shutdown 关不了 KVM 虚拟机的问题