1. 查看硬件是否支持虚拟化

#grep - E 'vmx|svm’ /proc/cpuinfo

2. VM 虚拟机的设置中打开了CPU 设置中的虚拟化引擎，选择 Intel VT-x/EPT 或 AMD-V/RVI(V) 这个选项。

3. 启用kvm

* 启用 KVM 模块

#modprobe kvm

* 功能区分 intel/amd 的启用：

#modprobe kvm-intel

# lsmod |grep kvm

kvm\_intel 55496 0

kvm 337772 1 kvm\_intel

4. 安装KVM所需组件

* yum 源安装qemu-kvm模拟器,libvert API，virt-manager图形管理:

#yum install -y qemu-kvm libvirt virt-manager

* 安装完成后启动 libvirtd 服务：

#service libvirtd start

Ifconfig能看到自动启动一个桥设备virbr0，这相当于 VMware Workstation 中的 host-only 仅主机的网络设备；

使用网桥管理命令查看：

# brctl show

bridge name bridge id STP enabled interfaces

virbr0 8000.525400305441 yes virbr0-nic

* 使用virsh创建物理桥接设备，关联网卡到桥接设备上

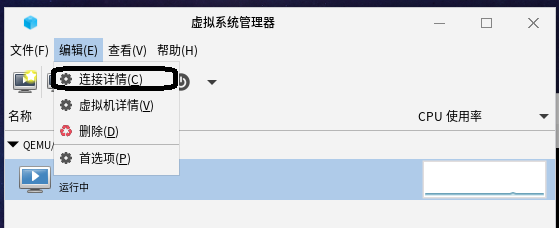
将 NetworkManager 服务关闭，开机启动也关闭：

# chkconfig NetworkManager off

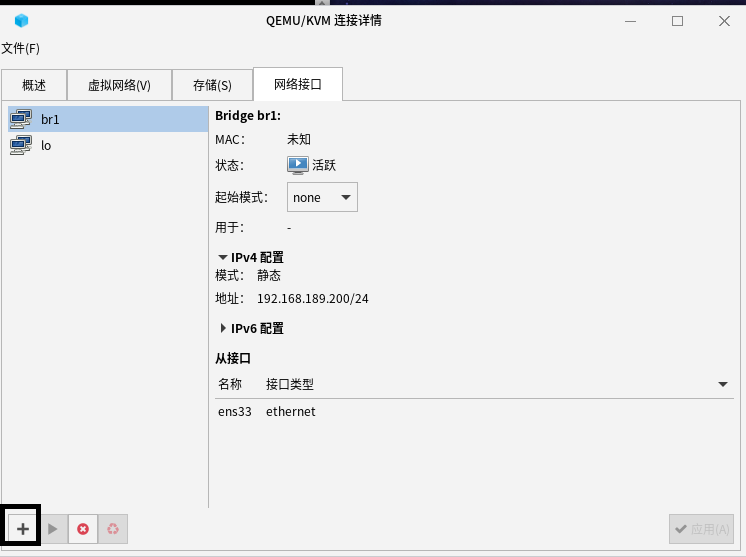
# service NetworkManager stop

创建桥接设备及关联网卡到桥接设备上：

1、# virsh iface-bridge eth0 br0

2、

选择连接详情





查看桥接设备：

# brctl show

bridge name bridge id STP enabled interfaces

br0 8000.000c293e6326 yes eth0

virbr0 8000.525400305441 yes virbr0-nic

至此，我们的虚拟化平台就构建完毕，下面就开始在 KVM 虚拟化平台上创建和管理虚拟机。

5. 使用 qemu-kvm 来创建和管理虚拟机

* 设置环境变量

qemu-kvm 位于 /usr/libexec 目录中。由于此目录不属于 PATH 环境变量，故无法直接使用，所以将 /usr/libexec/qemu-kvm 链接为 /usr/bin/qemu 。

# ln -sv /usr/libexec/qemu-kvm /usr/bin/qemu-kvm

* 创建虚拟机的磁盘文件存放目录：

# mkdir /kvm/images -pv

* 创建虚拟磁盘文件：

# qemu-img create -o preallocation=metadata -f qcow2 /kvm/images/winxp.qcow2 60G

# ll -h /kvm/images/

total 9.6M

-rw-r--r-- 1 root root 61G Feb 7 23:50 winxp.qcow2

* 导入iso镜像

mkdir -p /kvm/tools

dd if=/dev/sr0 of=/kvm/tools/centos7.sio

* 安装 vnc-server 和 tigervnc 这个 vncviewer

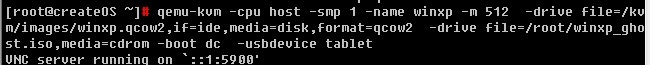
# yum install vnc-server tigervnc -y

* 安装winxp虚拟机

创建一个名为 winxp 的虚拟机，其 RAM 大小为 512MB ，有一颗 CPU 的 SMP 架构，默认引导设备是硬盘，有一个光驱设备和硬盘设备， usb 设备为 tablet ；

#qemu-kvm -cpu host -smp 1 -name winxp -m 512 -drive file=/kvm/images/winxp.qcow2,if=ide,meida=disk,format=qcow2 -drive file=/root/winxp\_ghost.iso,media=cdrom -boot dc -usbdevice tablet

启动后如图：



使用vncviewer进行连接：

#vncviewer :5900

启动后进入winxp安装界面，完成系统的安装。

* 虚拟机的网卡桥接

加入网卡设备，系统启动后桥接网络到 br0 上，我们通过脚本实现启动和停止系统时桥接网络和删除桥接的自动执行操作：

启动虚拟机将网卡桥接到桥接设备上：

# cat /etc/qemu-ifup

#!/bin/bash

#

bridge=br0

if [ -n "$1" ]; then

ip link set $1 up

sleep 1

brctl addif $bridge $1

[ $? -eq 0 ] && exit 0 || exit 1

else

echo "Error: no interfacespecified."

exit 1

fi

关闭虚拟机后自动将网卡从桥接设备上移除：

# cat/etc/qemu-ifdown

#!/bin/bash

#

bridge=br0

if [ -n "$1" ];then

brctl delif $bridge $1

ip link set $1 down

exit 0

else

echo "Error: no interfacespecified."

exit 1

fi

给予脚本执行权限；

#chmod +x /etc/qemu-ifup   
#chmod +x /etc/qemu-ifdown

启动虚拟机系统：

#qemu-kvm -cpu host -smp 1 -name winxp -m 512 -drive file=/kvm/images/winxp.qcow2,if=ide,media=disk,format=qcow2 -boot d -usbdevice tablet -net nic -net tap,ifname=vnet0,script=/etc/qemu-ifup,downscript=/etc/qemu-ifdown

启动完成后，通过 vncviewer 连接我们启动的虚拟机。

问题一：虚拟机不能ping通外网

宿主机编辑/etc/sysctl.conf 文件，增加 net.ipv4.ip\_forward = 1

sysctl -p

问题二：“unsupported configuration: CPU mode ‘custom’ for x86\_64 kvm domain on x86\_64 host is not supported”。

解决办法：重启主机。

sysctl -p

**问题二：**虚拟机如何打开串口。

**解决办法：**虚拟机的启动参数加入“console=tty0 console=ttyS0,115200”。可以通过virsh console [vm id] 命令，进入虚拟机串口操作。

**问题三：**非root用户ssh远程登录，执行命令：virt-manager 报如下错误：

在deepin操作系统上使用KVM虚拟机

**解决办法：**执行下面命令，把当前用户加入到组libvirt：

在deepin操作系统上使用KVM虚拟机

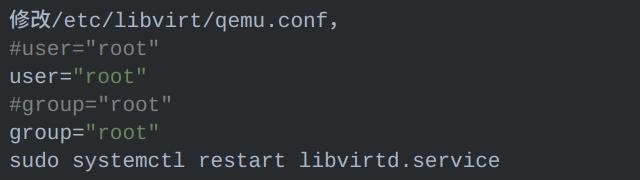
**问题四：**x11转发virt-manager图形界面报错误：(virt-manager:12205): Gtk-WARNING \*\*: cannot open display:

**解决办法：**使用如下方式连接 ：ssh -Y deepin@ServerIP virt-manager

**问题五：**

创建虚拟机时报权限错误：monitor: Could not access KVM kernel module: Permission denied failed to initialize KVM: Permission

**解决办法：**



**问题六：**.aarch64服务器创建支持aarch64架构的虚拟机时，报libvirt版本不支持UEFI。

**解决办法：**重启系统，再创建一次。

**附件：qemu-kvm介绍**

**1.qemu介绍**

Qemu 是一个广泛使用的开源计算机仿真器和虚拟机。当作为仿真器时，可以在一种架构 ( 如 PC 机 ) 下运行另一种架构 ( 如 ARM) 下的操作系统和程序。而通过动态转化，其可以获得很高的运行效率。当作为一个虚拟机时， qemu 可以通过直接使用真机的系统资源，让虚拟系统能够获得接近于物理机的性能表现。 qemu 支持 xen 或者 kvm 模式下的虚拟化。当用 kvm 时， qemu 可以虚拟 x86 、服务器和嵌入式 powerpc ，以及 s390 的系统。

QEMU 当运行与主机架构相同的目标架构时可以使用 KVM 。例如，当在一个 x86 兼容处理器上运行 qemu-system-x86时，可以利用 KVM加速，为宿主机和客户机提供更好的性能。

Qemu 有如下几个部分组成：

处理器模拟器(x86、PowerPC和Sparc)；

仿真设备(显卡、网卡、硬盘、鼠标等)；

用于将仿真设备连接至主机设备(真实设备)的通用设备；

模拟机的描述信息；

调试器；

与模拟器交互的用户接口；

基于 libvirt 的工具如 virt-manager 和 virt-install 提供了非常便捷的虚拟机管理接口，但它们事实上上经二次开发后又封装了 qemu-kvm 的工具。因此，直接使用 qemu-kvm 命令也能够完成此前的任务。

**2.Qemu-kvm 的使用**

qemu-kvm 命令使用格式为“ qemu-kvm [options] [disk\_image] ”，其选项非常多，不过，大致可分为如下几类。

标准选项；

USB选项；

显示选项；

i386平台专用选项；

网络选项；

字符设备选项；

蓝牙相关选项；

Linux系统引导专用选项；

调试/专家模式选项；

PowerPC专用选项；

Sparc32专用选项；

* **qemu-kvm的标准选项**

qemu-kvm的标准选项主要涉及指定主机类型、CPU模式、NUMA、软驱设备、光驱设备及硬件设备等。

-name name：设定虚拟机名称；

-M machine：指定要模拟的主机类型，如Standard PC、ISA-only PC或Intel-Mac等，可以使用“qemu-kvm -M ?”获取所支持的所有类型；

-m megs：设定虚拟机的RAM大小；

-cpu model：设定CPU模型，如coreduo、qemu64等，可以使用“qemu-kvm -cpu ?”获取所支持的所有模型；

-smp n[,cores=cores][,threads=threads][,sockets=sockets][,maxcpus=maxcpus]：设定模拟的SMP架构中CPU的个数等、每个CPU的核心数及CPU的socket数目等；PC机上最多可以模拟255颗CPU；maxcpus用于指定热插入的CPU个数上限；

-numa opts：指定模拟多节点的numa设备；

-fda file

-fdb file：使用指定文件(file)作为软盘镜像，file为/dev/fd0表示使用物理软驱；

-hda file

-hdb file

-hdc file

-hdd file：使用指定file作为硬盘镜像；

-cdrom file：使用指定file作为CD-ROM镜像，需要注意的是-cdrom和-hdc不能同时使用；将file指定为/dev/cdrom可以直接使用物理光驱；

-drive option[,option[,option[,...]]]：定义一个硬盘设备；可用子选项有很多。

file=/path/to/somefile：硬件映像文件路径；

if=interface：指定硬盘设备所连接的接口类型，即控制器类型，如ide、scsi、sd、mtd、floppy、pflash及virtio等；

index=index：设定同一种控制器类型中不同设备的索引号，即标识号；

media=media：定义介质类型为硬盘(disk)还是光盘(cdrom)；

snapshot=snapshot：指定当前硬盘设备是否支持快照功能：on或off；

cache=cache：定义如何使用物理机缓存来访问块数据，其可用值有none、writeback、unsafe和writethrough四个；

format=format：指定映像文件的格式，具体格式可参见qemu-img命令；

-boot [order=drives][,once=drives][,menu=on|off]：定义启动设备的引导次序，每种设`a、b表示软驱、c表示第一块硬盘，d表示第一个光驱设备，n-p表示网络适配器；默认为硬盘设备；

-boot order=dc,once=d

* **qemu-kvm的显示选项**

显示选项用于定义虚拟机启动后的显示接口相关类型及属性等。

-nographic：默认情况下，qemu使用SDL来显示VGA输出；而此选项用于禁止图形接口，此时,qemu类似一个简单的命令行程序，其仿真串口设备将被重定向到控制台；

-curses：禁止图形接口，并使用curses/ncurses作为交互接口；

-alt-grab：使用Ctrl+Alt+Shift组合键释放鼠标；

-ctrl-grab：使用右Ctrl键释放鼠标；

-sdl：启用SDL；

-spice option[,option[,...]]：启用spice远程桌面协议；其有许多子选项，具体请参照qemu-kvm的手册；

-vga type：指定要仿真的VGA接口类型，常见类型有：

cirrus：Cirrus Logic GD5446显示卡；

std：带有Bochs VBI扩展的标准VGA显示卡；

vmware：VMWare SVGA-II兼容的显示适配器；

qxl：QXL半虚拟化显示卡；与VGA兼容；在Guest中安装qxl驱动后能以很好的方式工作，在使用spice协议时推荐使用此类型；

none：禁用VGA卡；

-vnc display[,option[,option[,...]]]：默认情况下，qemu使用SDL显示VGA输出；使用-vnc选项，可以让qemu监听在VNC上，并将VGA输出重定向至VNC会话；使用此选项时，必须使用-k选项指定键盘布局类型；其有许多子选项，具体请参照qemu-kvm的手册；

**i386平台专用选项**

-no-acpi：禁用ACPI功能，GuestOS与ACPI出现兼容问题时使用此选项；

-balloon none：禁用balloon设备；

-balloon virtio[,addr=addr]：启用virtio balloon设备；

* **网络属性相关选项**

网络属性相关选项用于定义网络设备接口类型及其相关的各属性等信息。这里只介绍nic、tap和user三种类型网络接口的属性，其它类型请参照qemu-kvm手册。

-net nic[,vlan=n][,macaddr=mac][,model=type][,name=name][,addr=addr][,vectors=v]：创建一个新的网卡设备并连接至vlan n中；PC架构上默认的NIC为e1000，macaddr用于为其指定MAC地址，name用于指定一个在监控时显示的网上设备名称；emu可以模拟多个类型的网卡设备，如virtio、i82551、i82557b、i82559er、ne2k\_isa、pcnet、rtl8139、e1000、smc91c111、lance及mcf\_fec等；不过，不同平台架构上，其支持的类型可能只包含前述列表的一部分，可以使用“qemu-kvm -net nic,model=?”来获取当前平台支持的类型；

-net tap[,vlan=n][,name=name][,fd=h][,ifname=name][,script=file][,downscript=dfile]：通过物理机的TAP网络接口连接至vlan n中，使用script=file指定的脚本(默认为/etc/qemu-ifup)来配置当前网络接口，并使用downscript=file指定的脚本(默认为/etc/qemu-ifdown)来撤消接口配置；使用script=no和downscript=no可分别用来禁止执行脚本；

-net

user[,option][,option][,...]：在用户模式配置网络栈，其不依赖于管理权限；有效选项有：

vlan=n：连接至vlan n，默认n=0；

name=name：指定接口的显示名称，常用于监控模式中；

net=addr[/mask]：设定GuestOS可见的IP网络，掩码可选，默认为10.0.2.0/8；

host=addr：指定GuestOS中看到的物理机的IP地址，默认为指定网络中的第二个，即x.x.x.2；

dhcpstart=addr：指定DHCP服务地址池中16个地址的起始IP，默认为第16个至第31个，即x.x.x.16-x.x.x.31；

dns=addr：指定GuestOS可见的dns服务器地址；默认为GuestOS网络中的第三个地址，即x.x.x.3；

tftp=dir：激活内置的tftp服务器，并使用指定的dir作为tftp服务器的默认根目录；

bootfile=file：BOOTP文件名称，用于实现网络引导GuestOS；如：qemu -hda linux.img -boot n -net user,tftp=/tftpserver/pub,bootfile=/pxelinux.0