

首先Hyper-V 上安装Centos虚拟机

安装完成后查看是否支持虚拟化

```
egrep -o 'vmx|svm' /proc/cpuinfo
```

```
[root@localhost ~]# egrep -o 'vmx|svm' /proc/cpuinfo
[root@localhost ~]#
```

没有输出说明不支持

下面进行设置，参考[Hyper-v 开启嵌套虚拟化的方法](#)

hyber-V 机器开启嵌套虚拟化：

关闭虚拟机，管理员权限打开powershell

Get-VM ##列出虚拟机

Get-VMProcessor -VMName KVM主机 | fl ##查看虚拟化选项参数

```
PS C:\Users\Administrator> Get-VMProcessor -VMName 192.168.1.108 | fl

ResourcePoolName      : Primordial
Count                 : 1
CompatibilityForMigrationEnabled : False
CompatibilityForOlderOperatingSystemsEnabled : False
HvThreadCountPerCore  : 0
ExposeVirtualizationExtensions : False
EnablePerfmonPmu      : False
EnablePerfmonLbr       : False
EnablePerfmonPebs      : False
Maximum               : 100
Reserve               : 0
RelativeWeight         : 100
MaximumCountPerNumaNode : 12
MaximumCountPerNumaSocket : 1
EnableHostResourceProtection : False
OperationalStatus      : {OK, HostResourceProtectionDisabled}
StatusDescription      : [确定, 主机资源保护已禁用。]
Name                  : 处理器
Id                    : {Microsoft:1510FDE5-4DA7-4042-9653-5D561B59C416\b637f346-6a0e-4dec-af52-b470cb80a21d}0
VMId                  : 1510fde5-4da7-4042-9653-5d561b59c416
VMName                : 192.168.1.108
VMSnapshotId          : 00000000-0000-0000-0000-000000000000
VMSnapshotName        :
CimSession             : CimSession: .
ComputerName           : LG-H-1-06
IsDeleted              : False
VMCheckpointId         : 00000000-0000-0000-0000-000000000000
VMCheckpointName       :
https://blog.csdn.net/rockstics
```

关闭虚拟机进行设置

Set-VMProcessor -ExposeVirtualizationExtensions \$true -VMName KVM主机 ##将其设置为True

重启虚拟机，查看已支持虚拟化

```
PS C:\Users\Administrator> Set-VMProcessor -ExposeVirtualizationExtensions $true -VMName 192.168.1.112

PS C:\Users\Administrator> Get-VMProcessor -VMName 192.168.1.112 | fl

ResourcePoolName      : Primordial
Count                 : 4
CompatibilityForMigrationEnabled : False
CompatibilityForOlderOperatingSystemsEnabled : False
HvThreadCountPerCore  : 0
ExposeVirtualizationExtensions : True
EnablePerfmonPmu      : False
EnablePerfmonLbr       : False
EnablePerfmonPebs      : False
Maximum               : 100
Reserve               : 0
RelativeWeight         : 100
MaximumCountPerNumaNode : 12
MaximumCountPerNumaSocket : 1
EnableHostResourceProtection : False
OperationalStatus      : {}
StatusDescription      : {}
Name                  : 处理器
Id                    : {Microsoft:549D823E-60C5-4E86-B579-B7ADP284B44A\b637f346-6a0e-4dec-af52-b470cb80a21d}0
VMId                  : 1510fde5-4da7-4042-9653-5d561b59c416
VMName                : 192.168.1.112
VMSnapshotId          : 00000000-0000-0000-0000-000000000000
VMSnapshotName        :
CimSession             : CimSession: .
ComputerName           : LG-H-1-06
IsDeleted              : False
VMCheckpointId         : 00000000-0000-0000-0000-000000000000
VMCheckpointName       :
https://blog.csdn.net/rockstics
```

```
[root@localhost ~]#
[root@localhost ~]# egrep -o 'vmx|xvm' /proc/cpuinfo
VMX
[root@localhost ~]#
```

##上图中可能会看到有多个VMX，有几个就意味着有几个CPU

至此，由Hyper-V创建的虚拟机支持KVM虚拟化的嵌套虚拟化设置完成

虚拟机安装KVM虚拟化工具

安装包	说明
-----	----

qemu-kvm	kvm提供的底层仿真软件
libvirt-daemon	虚拟机守护进程，管理虚拟机
libvirt-client	提供客户机管理命令
libvirt-deamon-driver-qemu	连接qemu的驱动
virt-manager	提供图形界面
qemu-img	使用qemu命令创建和管理磁盘，将作为libvirt 依赖安装
libvirt-python	可以使用Python来监控kvm虚拟机的资源使用情况

kvm的管理工具

- libvirt是一组提供了多种语言接口的API，为各种虚拟化技术提供一套方便、可靠的编程接口。它不仅支持KVM，也支持Xen、LXC、OpenVZ以及VirtualBox等其他虚拟化技术。利用libvirt API，用户可以创建、配置、监控、迁移或者关闭虚拟机。
- RHEL 7支持libvirt以及基于libvirt的各种管理工具，例如virsh和virt-manager等。
- virsh：是一个基于libvirt的命令行工具。利用virsh，用户可以完成所有的虚拟机管理任务，包括创建和管理虚拟机、查询虚拟机的配置和运行状态等。virsh工具包含在libvirt-client软件包中。
- Virt-Manager：是一套基于图形界面的虚拟化管理工具。同样，virt-manager也是基于libvirt API的，所以，用户可以使用virt-manager来完成虚拟机的创建、配置和迁移。此外，virt-manager还支持管理远程虚拟机。

QEMU、KVM、libvirt三者间的关系

虚拟化有关的软件包组：

软件包组	说明	必须软件包
Virtualization Client	安装和管理虚拟机的客户端工具	virt-install、virt-manager、virt-top、virt-viewer
Virtualization Platform	提供访问和控制虚拟机的接口	libvirt、virtwho
Virtualization Tools	提供离线管理虚拟机镜像的工具	libguestfs

软件安装方式：

至少需要安装：

- qemu-kvm 和 qemu-img 软件包。提供用户级 KVM 仿真器和磁盘映像管理器。
- libvirt 控制虚拟机的接口
- virt-manager 或者 virsh （安装包libvirt-client）管理工具： virt-manager 或者 virsh （安装包libvirt-client）

```
yum install qemu-kvm qemu-kvm libvirt virsh
```

完整安装：

方式一：

```
yum groupinstall "Virtualization Client" "Virtualization Tools" "Virtualization Platform"
```

方式二：安装桌面版 （安装桌面版本centos自带KVM）

```
yum -y groupinstall GNOME Desktop
```

查看KVM 是否安装成功：

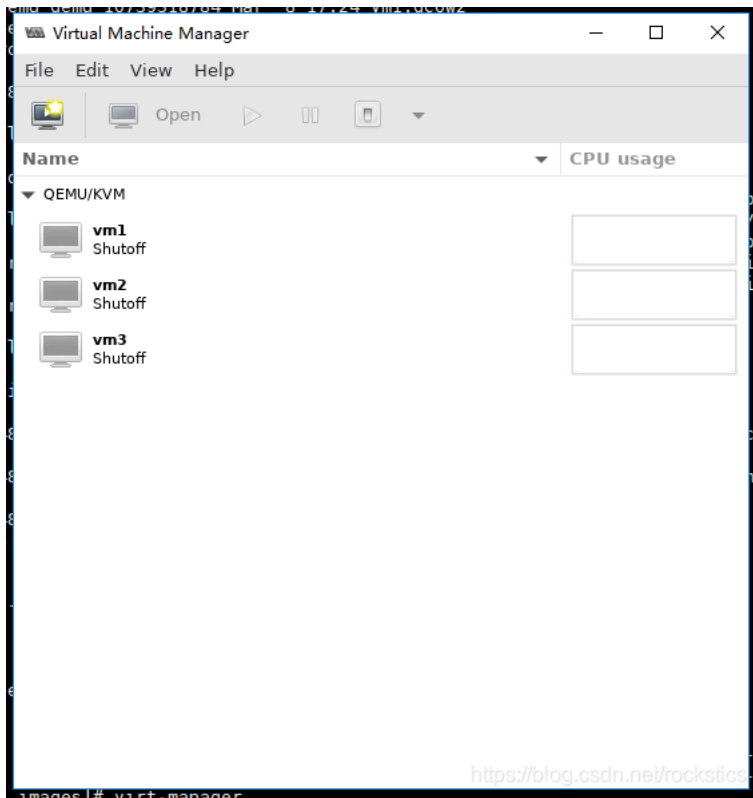
```
lsmod | grep kvm
```

也可以使用virsh命令来验证libvirtd服务是否正常启动：

```
virsh -c qemu:///system list
```

虚拟机创建：

virt-manager 启动图形管理界面



virt-manager启动报错解决

报错：

```
[root@localhost ~]# virt-manager
[root@localhost ~]# libGL error: unable to load driver: swrast_dri.so
libGL error: failed to load driver: swrast
```

解决

```
yum -y install mesa-libGLES-devel.x86_64 mesa-dri-drivers
```

图形界面创建KVM虚拟机见 [CSDN](https://blog.csdn.net/rocksitics)

命令模式创建KVM 虚拟机：

```
virt-install --name test --vcpus 1 --memory 1024 --disk /var/lib/libvirt/images/test.qcow2,bus=virtio,size=5 --network bridge=virbr0,model=virtio -
```

创建虚拟机镜像文件

qemu-img create -f 类型 -b 母盘路径名称 快照路径名称

```
qemu-img create -f qcow2 -b /var/lib/libvirt/images/westos.qcow2 /var/lib/libvirt/images/nodel.qcow2
```

利用虚拟机镜像文件qcow2安装虚拟机

```
virt-install --name vm3 --ram 1024 --disk /var/lib/libvirt/images/vm3.qcow2 --import
```

##--ram 为必要参数，指定虚拟机运行内存,其余参数如果不指定会根据系统默认值进行设置（不是母镜像文件的值），可通过 `virsh edit [vmname]` 方式去修改,

删除虚拟机

virsh shutdown vm1

virsh destroy vm1

删除虚拟机的硬件信息

命令： **virsh undefine**

/etc/libvirt/qemu

一次性打开虚拟机

命令： **virsh create vmname.xml**

永久性恢复虚拟机

命令： **virsh define vmname.xml**

virt-install 命令参数：

- -n --name= 客户端虚拟机名称
- -r --ram= 客户端虚拟机分配的内存
- -u --uuid= 客户端UUID 默认不写时，系统会自动生成

- --vcpus= 客户端的vcpu个数
- -v --hvm 全虚拟化
- -p --paravirt 半虚拟化
- -l --location=localdir 安装源, 有本地、nfs、http、ftp几种, 多用于ks网络安装
- --vnc 使用vnc, 另有--vnclient = 监听的IP --vncport = VNC监听的端口
- -c --cdrom= 光驱 安装途径
- --disk= 使用不同选项作为磁盘使用安装介质
- -w NETWORK, --network=NETWORK 连接客户机到主机网络
- -s --file-size= 使用磁盘映像的大小 单位为GB
- -f --file= 作为磁盘映像使用的文件
- --cpuset=设置哪个物理CPU能够被虚拟机使用
- --os-type=OS_TYPE 针对一类操作系统优化虚拟机配置 (例如: 'linux', 'windows')
- --os-variant=OS_VARIANT 针对特定操作系统变体 (例如 'rhel6', 'winxp', 'win2k3') 进一步优化虚拟机配置
- --host-device=HOSTDEV 附加一个物理主机设备到客户机。HOSTDEV是随着libvirt使用的一个节点设备名 (具体设备如 'virsh nodedev-list' 的显示的结果)
- --accelerate KVM或KQEMU内核加速,这个选项是推荐最好加上。如果KVM和KQEMU都支持, KVM加速器优先使用。
- -x EXTRA, --extra-args=EXTRA 当执行从"--location"选项指定位置的客户机安装时, 附加内核命令行参数到安装程序
- --nographics "virt-install" 将默认使用--vnc选项, 使用nographics指定没有控制台被分配给客户机