

虚拟化技术（KVM）

一、KVM网络虚拟化技术

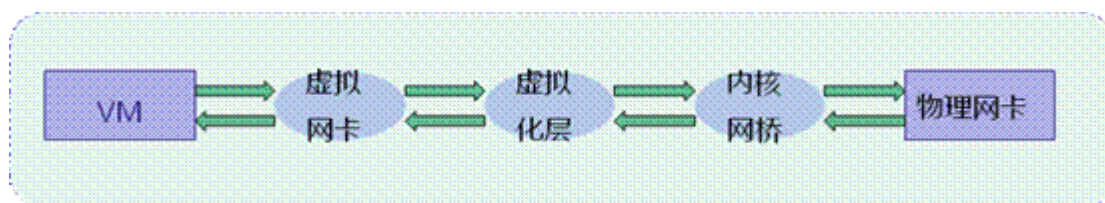
学习目标

- ☐ 能够了解半虚拟化驱动工作原理
- ☐ 能够下载半虚拟化驱动
- ☐ 能够查看主机是否支持半虚拟化驱动
- ☐ 能够安装半虚拟化驱动
- ☐ 能够通过virt-manager为虚拟机添加网卡
- ☐ 能够通过virsh命令为虚拟机添加网卡
- ☐ 能够添加四种虚拟网络交换机

1. 为什么使用半虚拟化驱动？

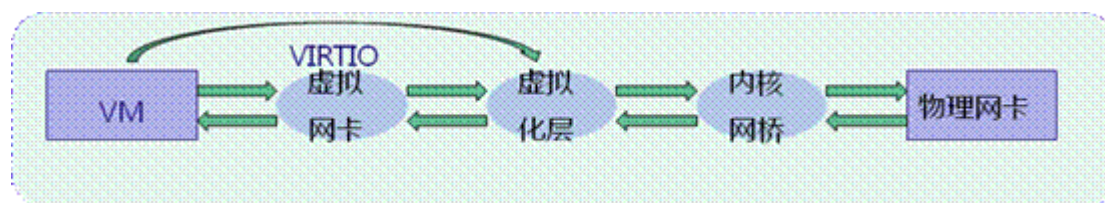
虚拟化层完全模拟出来的网卡

全虚拟化网卡环境中，一个完整的数据包从虚拟机到物理机网卡的路径是：



通过驱动对操作系统进行了修改，即Virtio技术

半虚拟化网卡环境中，虚拟机直接访问虚拟化层，步骤减少：



2.使用半虚拟化驱动方法

http://www.linux-kvm.org/page/WindowsGuestDrivers/Download_Drivers

```
1 [root@localhost ~]# wget https://fedorapeople.org/groups/virt/virtio-win/virtio-  
win.repo -O /etc/yum.repos.d/virtio-win.repo  
2 [root@localhost ~]# yum install virtio-win
```

3.查看是否支持半虚拟化驱动

宿主机查看方法

```
1 [root@localhost ~]#virsh dumpxml smartgo_01 | grep interface -C2
```

虚拟机查看方法

```
1 [root@localhost ~]#virt-what  
2 [root@localhost ~]#ethtool -i eth0
```

4.设置半虚拟化驱动方法

- 使用virt-manager添加
- 修改域的配置文件中
 - virsh edit 域名

5.添加网卡

1.virt-manager

选择虚拟机直接执行添加网卡操作

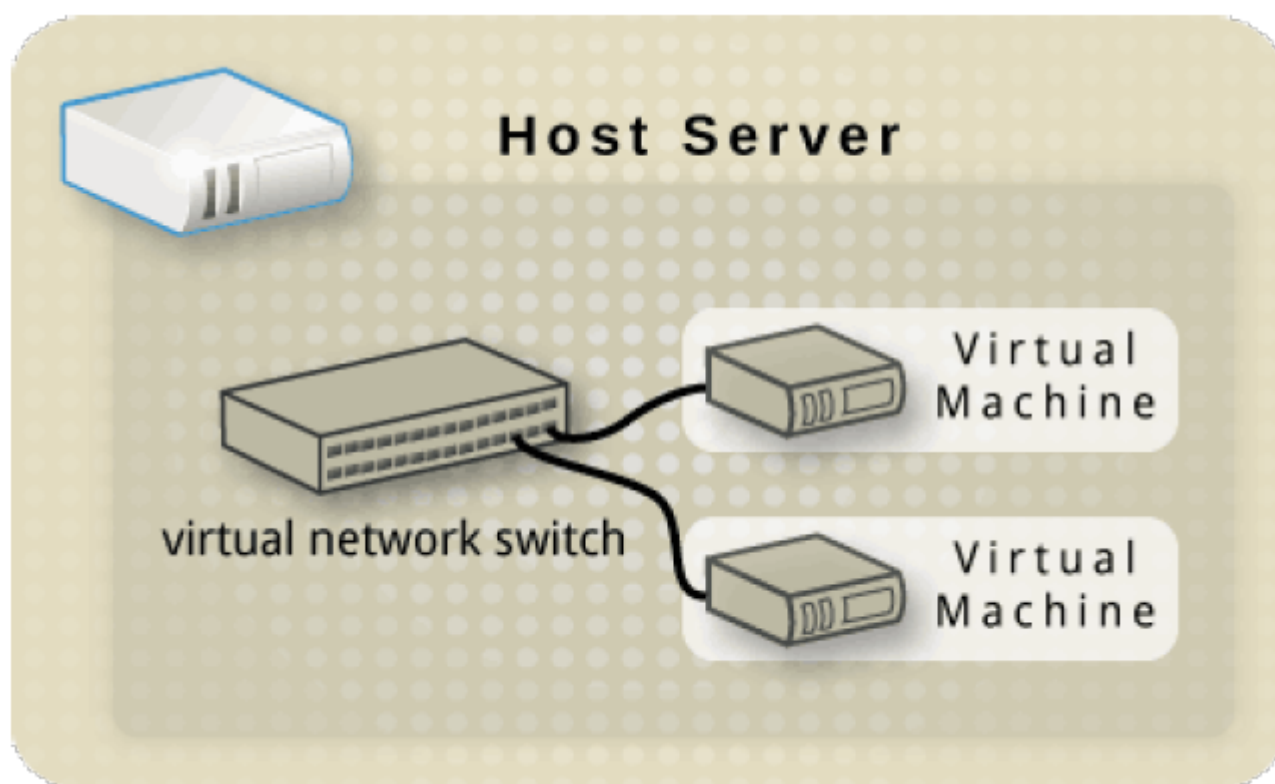
2.virsh命令

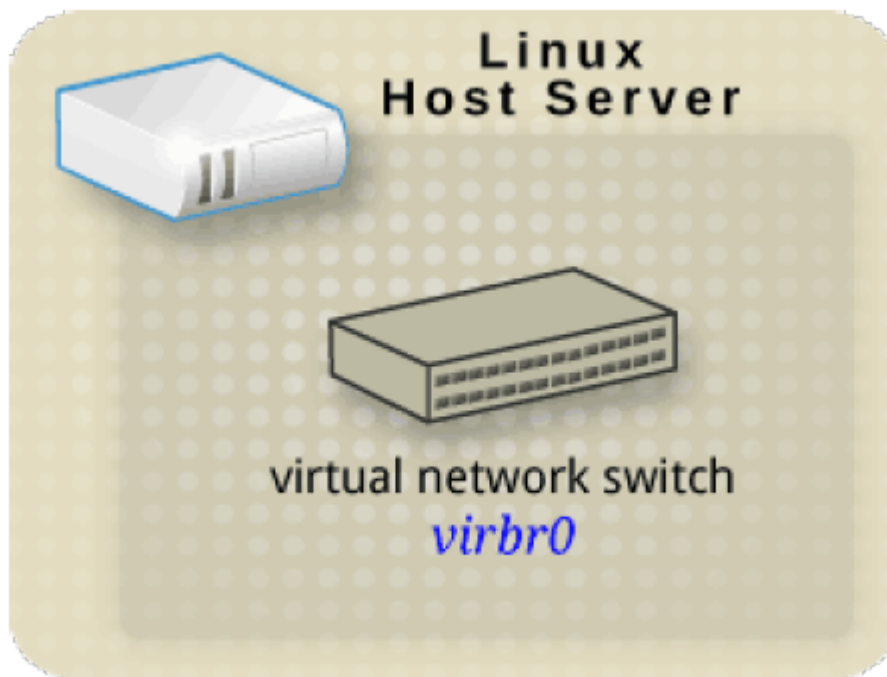
```
1 [root@localhost ~]#virsh list  
2 [root@localhost ~]#virsh domiflist smartgo_01  
3 [root@localhost ~]#virsh attach-interface smartgo_01 \  
4 --type network \  
5 --source default \  
6 --model virtio \  
7 --config
```

虚拟机查看网卡添加情况

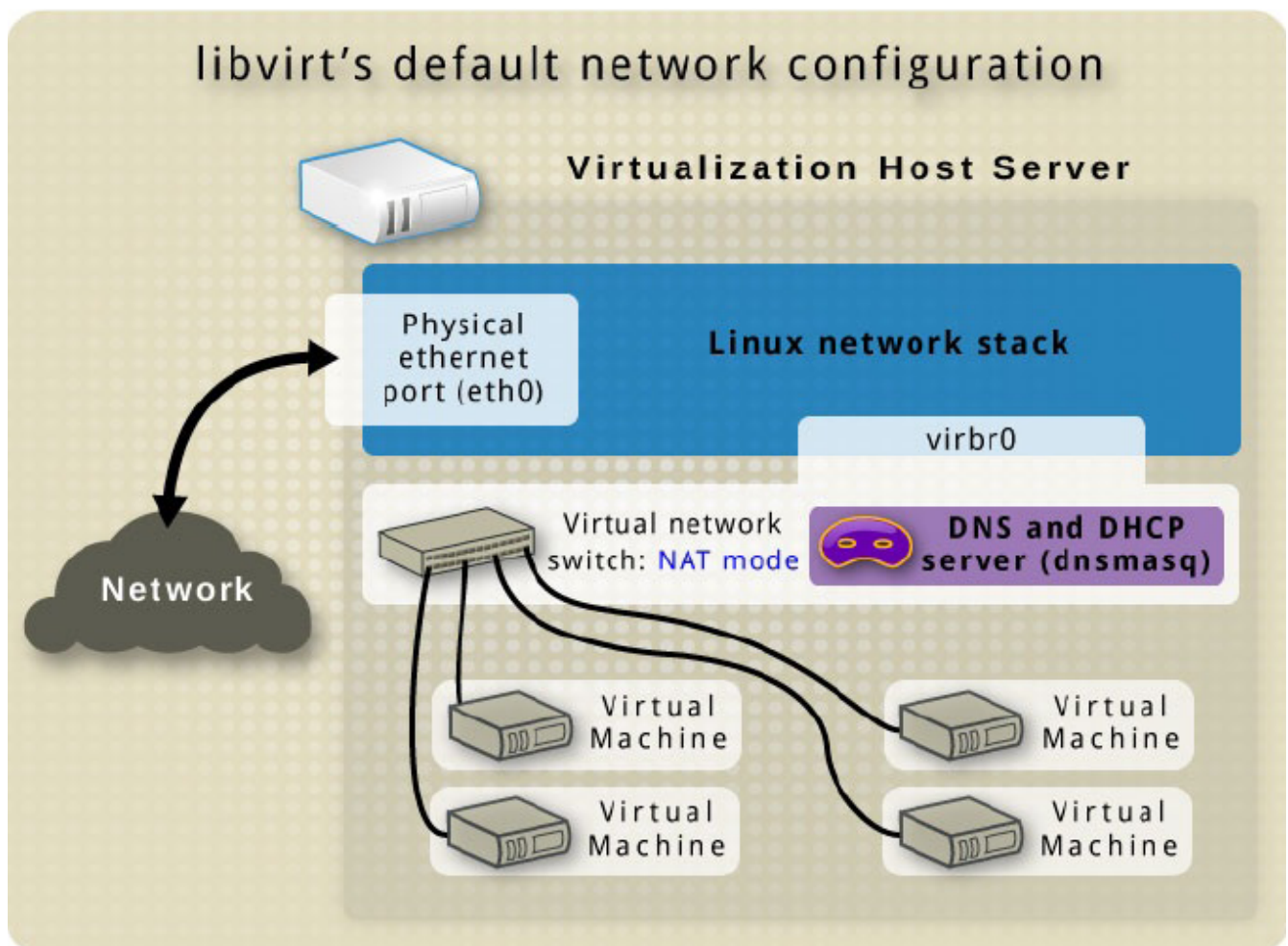
```
1 [root@smartgo_01 ~]#ip a s  
2
```

6.虚拟网络交换机

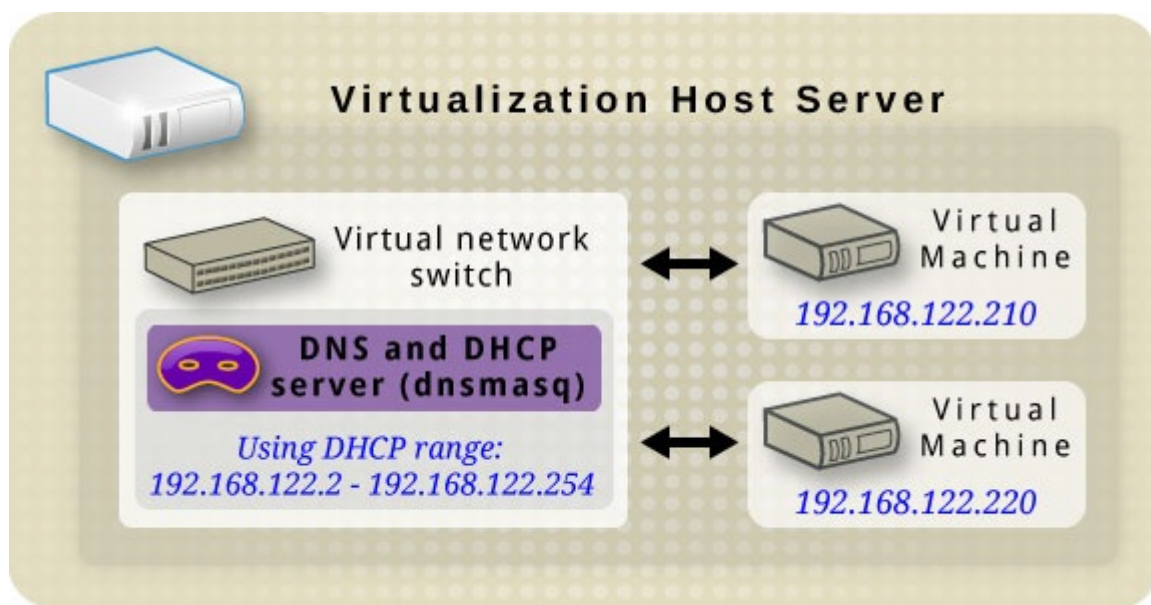
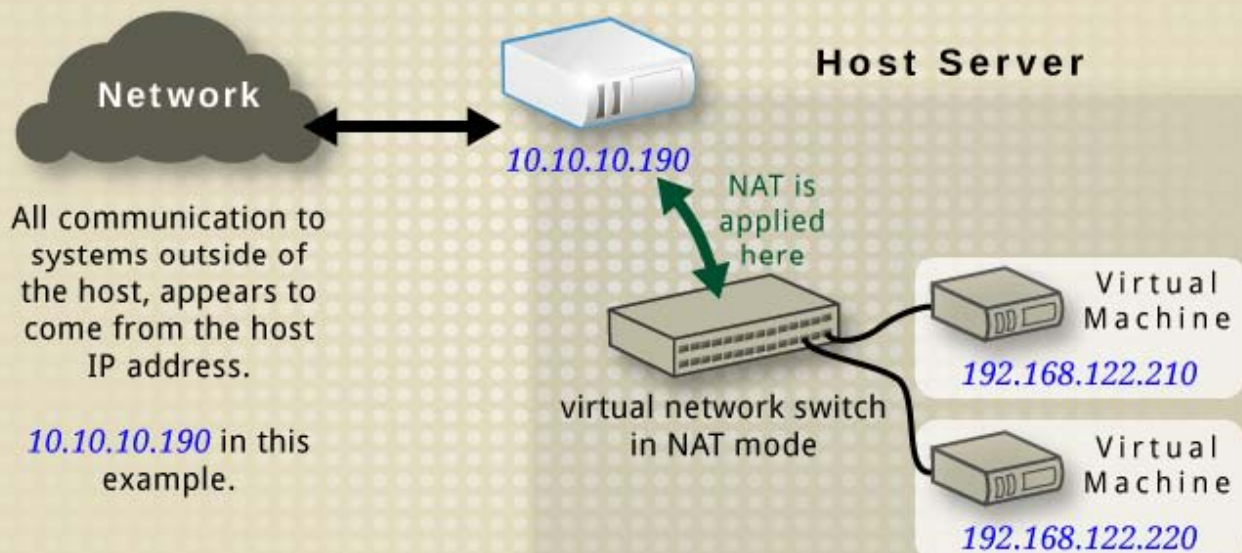




1. NAT mode (Network Address Translation)



Virtual switch: NAT mode



Host configuration

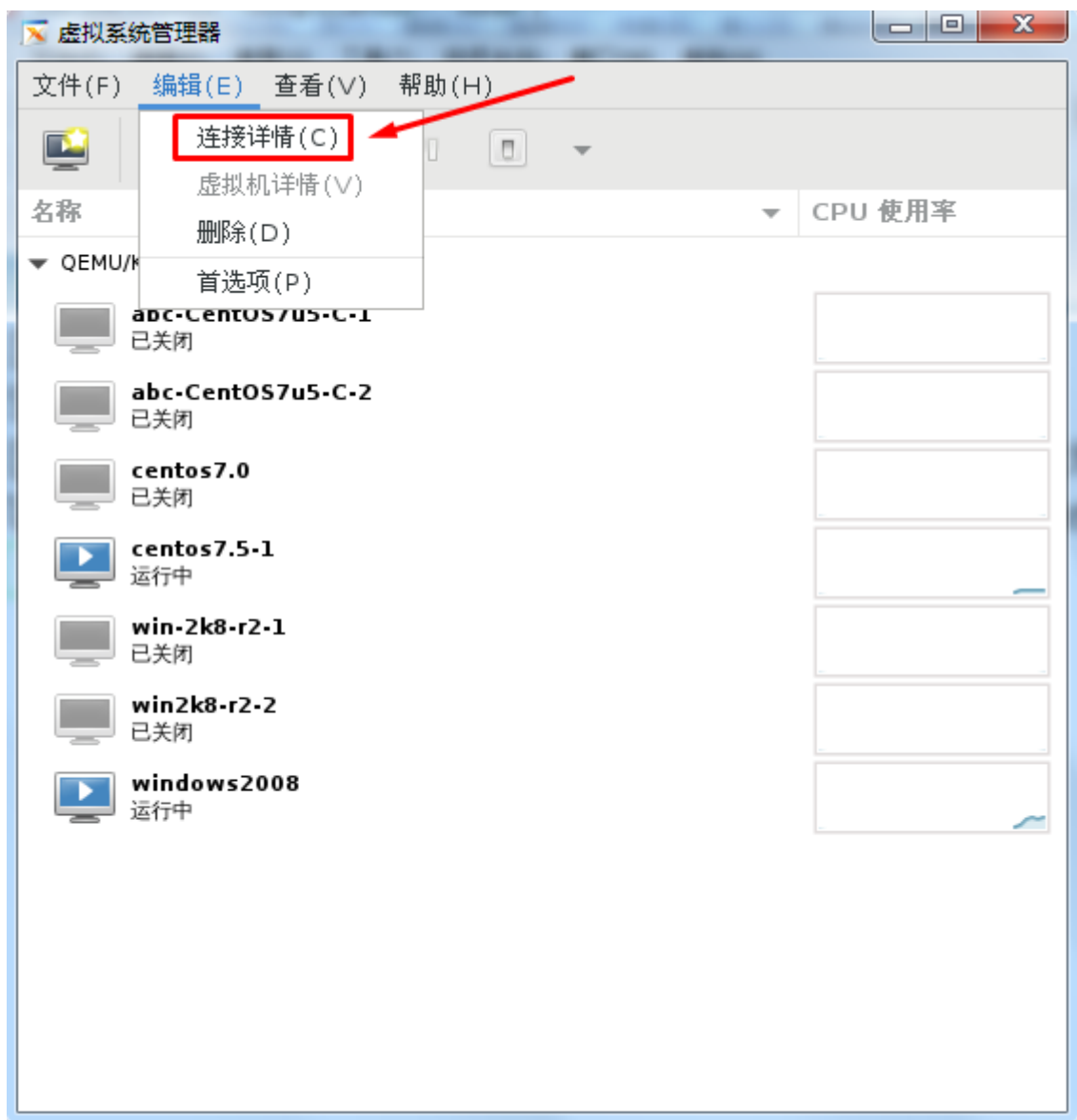
1.1 查看虚拟网络

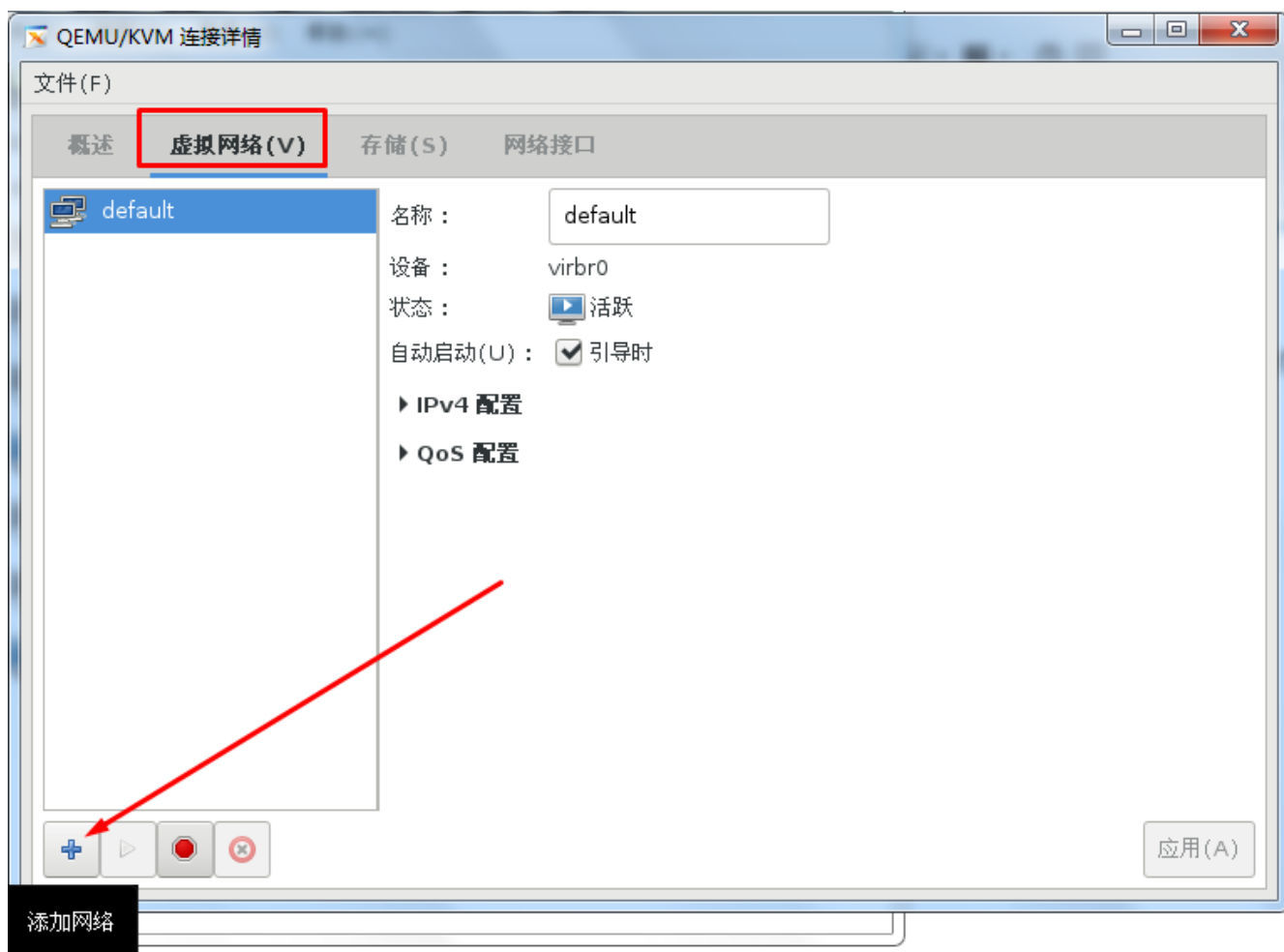

```
1 [root@localhost ~]#virsh net-list --all
2 [root@localhost ~]#virsh net-edit default
3 [root@localhost ~]#ls /etc/libvirt/qemu/networks/autostart/
4 [root@localhost ~]# sysctl -a | grep ip_forward
5 [root@localhost ~]# echo 1 > /proc/sys/net/ipv4/ip_forward #当ip_forward内容为0时使用此命令。
6 [root@localhost ~]#sysctl -p
```

1.2 查看防火墙状态

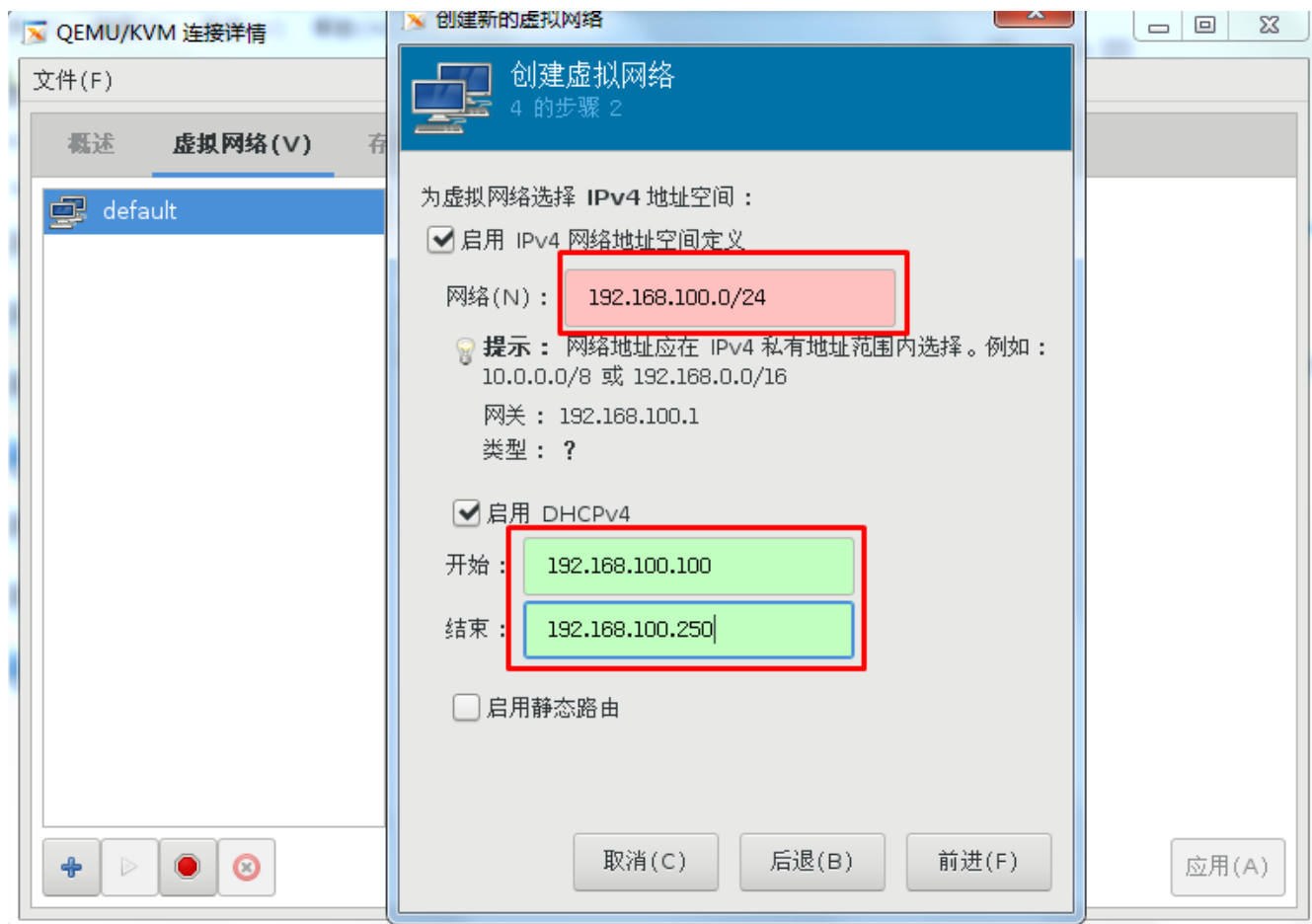
```
[root@tianyun ~]# firewall-cmd --permanent --list-all --zone=external
external
  interfaces:
  sources:
  services: ssh
  ports:
  masquerade: yes
  forward-ports:
  icmp-blocks:
  rich rules:
```

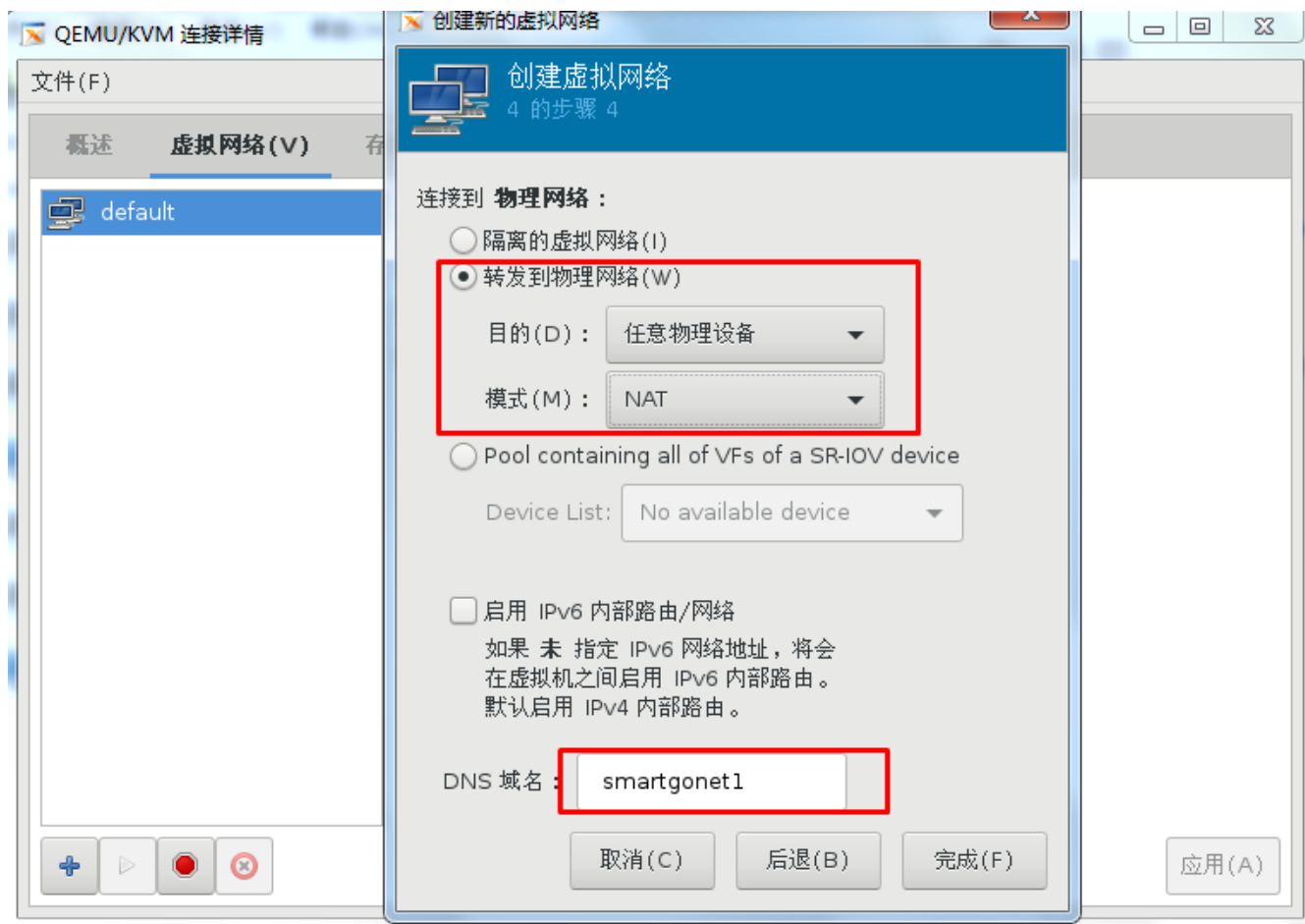
1.3创建虚拟网络（虚拟网络交换机）

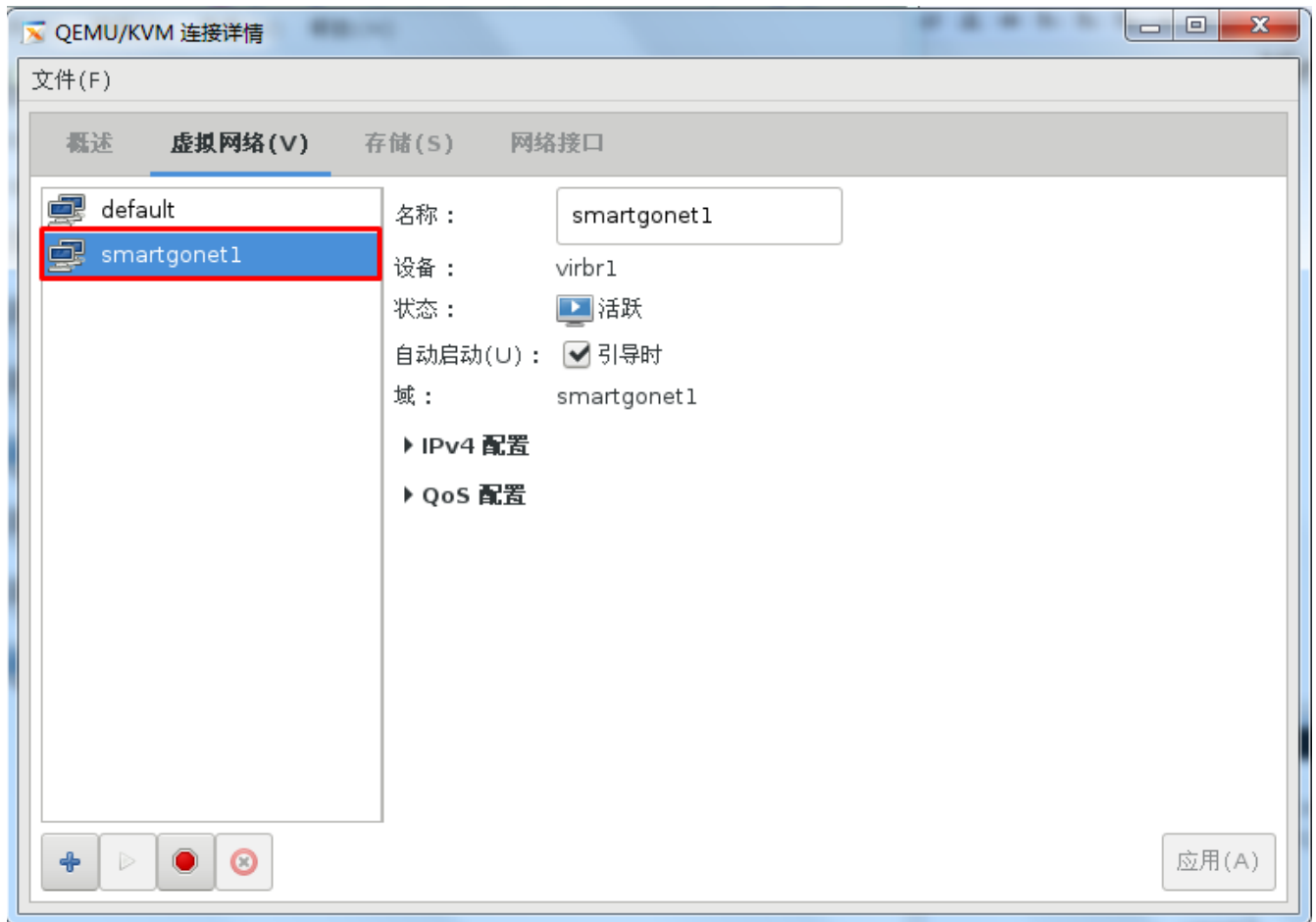










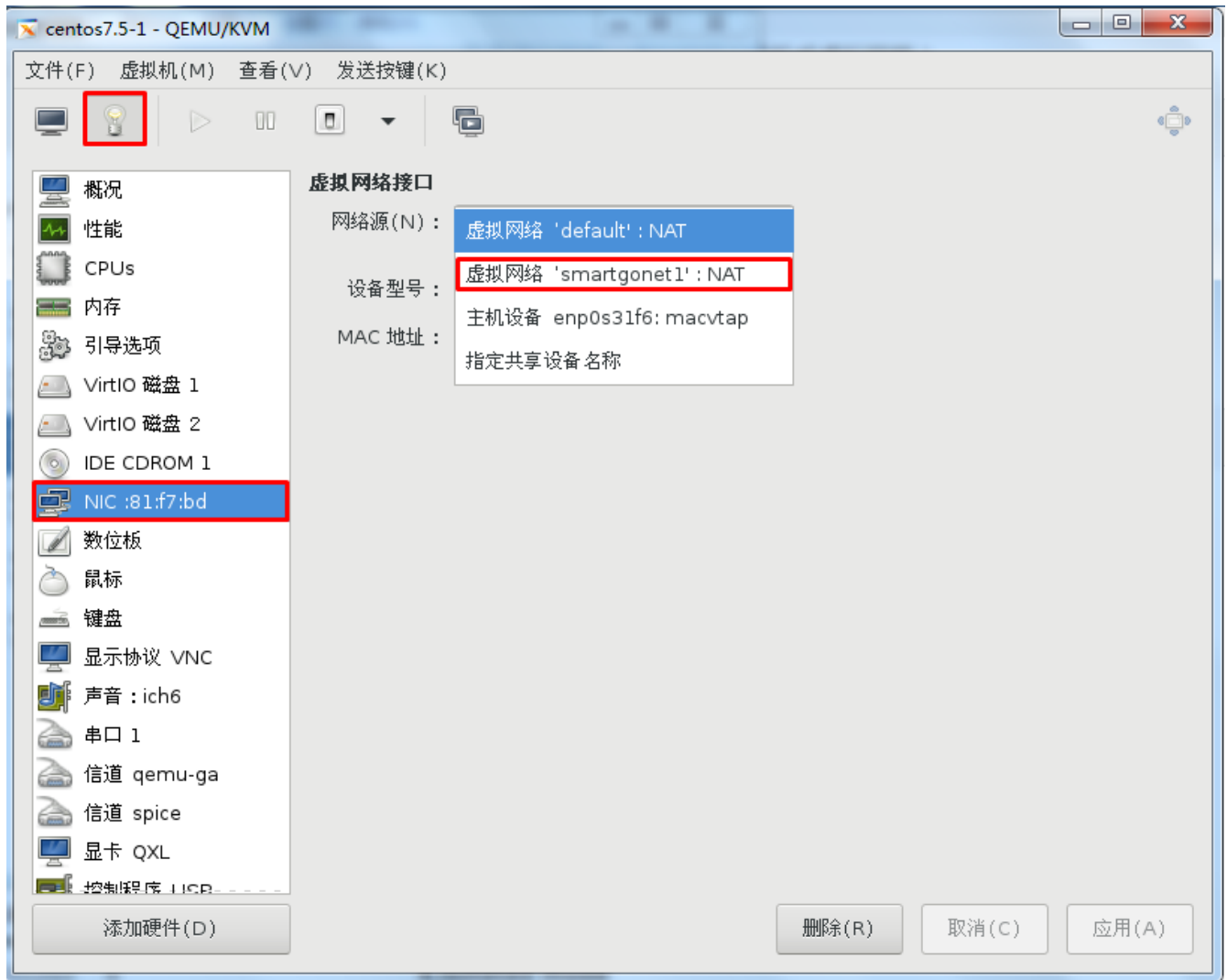


1.4 查看虚拟网络（虚拟网络交换机或虚拟网桥）

```
1 | [root@localhost ~]#brctl show
```

Guest virtual machine configuration

1.1 virt-manager



1.2 virsh命令行修改配置文件

```
1 [root@localhost ~]#virsh edit centos7.5-1
2 <interface type='network'>
3     <mac address='52:54:00:xx:xx:xx' />
4     <source network='smartgonet1' />
5     <model type='virtio' />
6     <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
7 </interface>
```

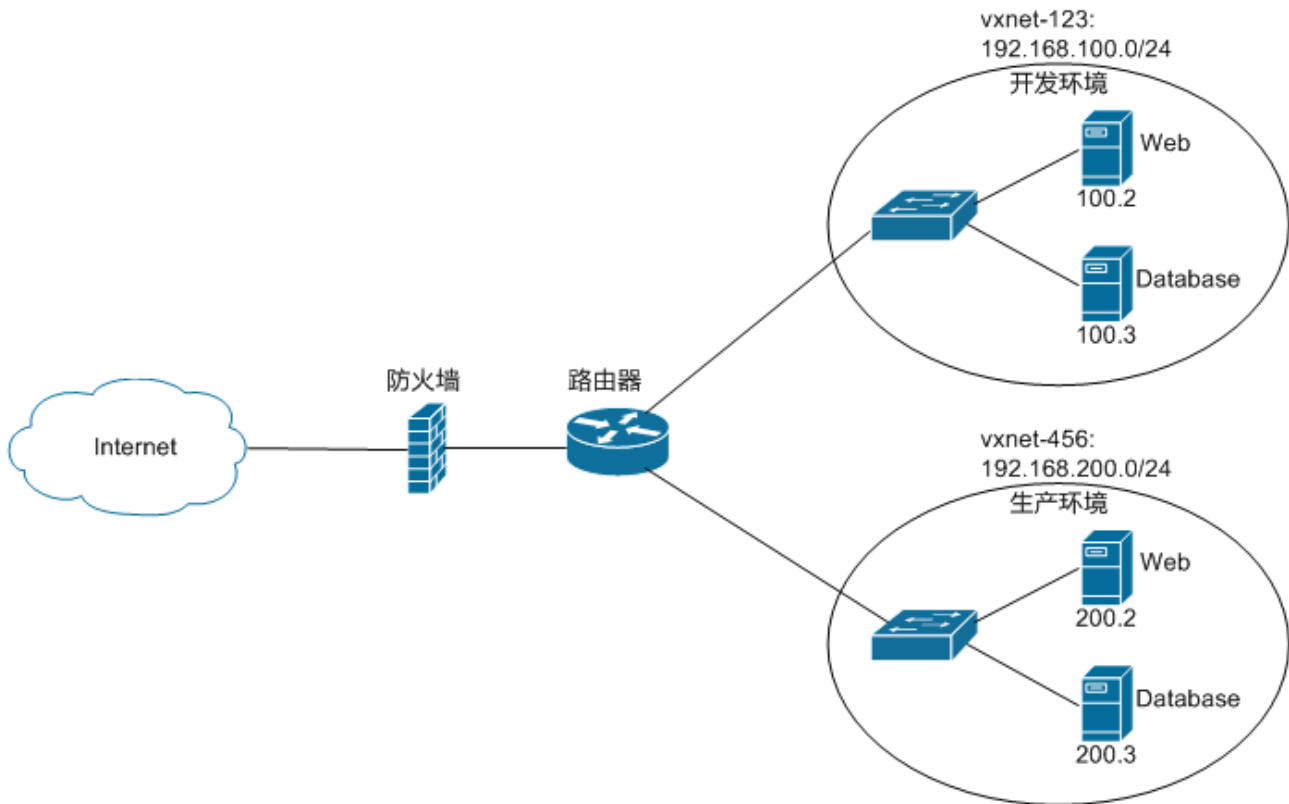
1.3 通过配置文件创建网络

```
1 [root@localhost ~]#cp -rf /etc/libvirt/qemu/networks/default.xml
  /etc/libvirt/qemu/networks/smartgonet2.xml
2 [root@localhost ~]#uuidgen #生成UUID
3 [root@localhost ~]#openssl rand -hex 6 |sed -r 's/(.)/\1:/g;s/.$// ' # 生成MAC地址
4 [root@localhost ~]#vim /etc/libvirt/qemu/networks/smartgonet2.xml
```

```

1 [root@localhost ~]#virsh net-define smartgonet2.xml  #从配置文件中生成网络
2 [root@localhost ~]#virsh net-start smartgonet2
3 [root@localhost ~]#virsh net-autostart smartgonet2 #设置开机自启动
4 [root@localhost ~]#virsh net-list --all
5 [root@localhost ~]#ip a s

```



2.Bridged mode

所有的虚拟机和宿主机在同一个子网内，桥工作在OSI的2层，不需要ip_forward

Host configuration

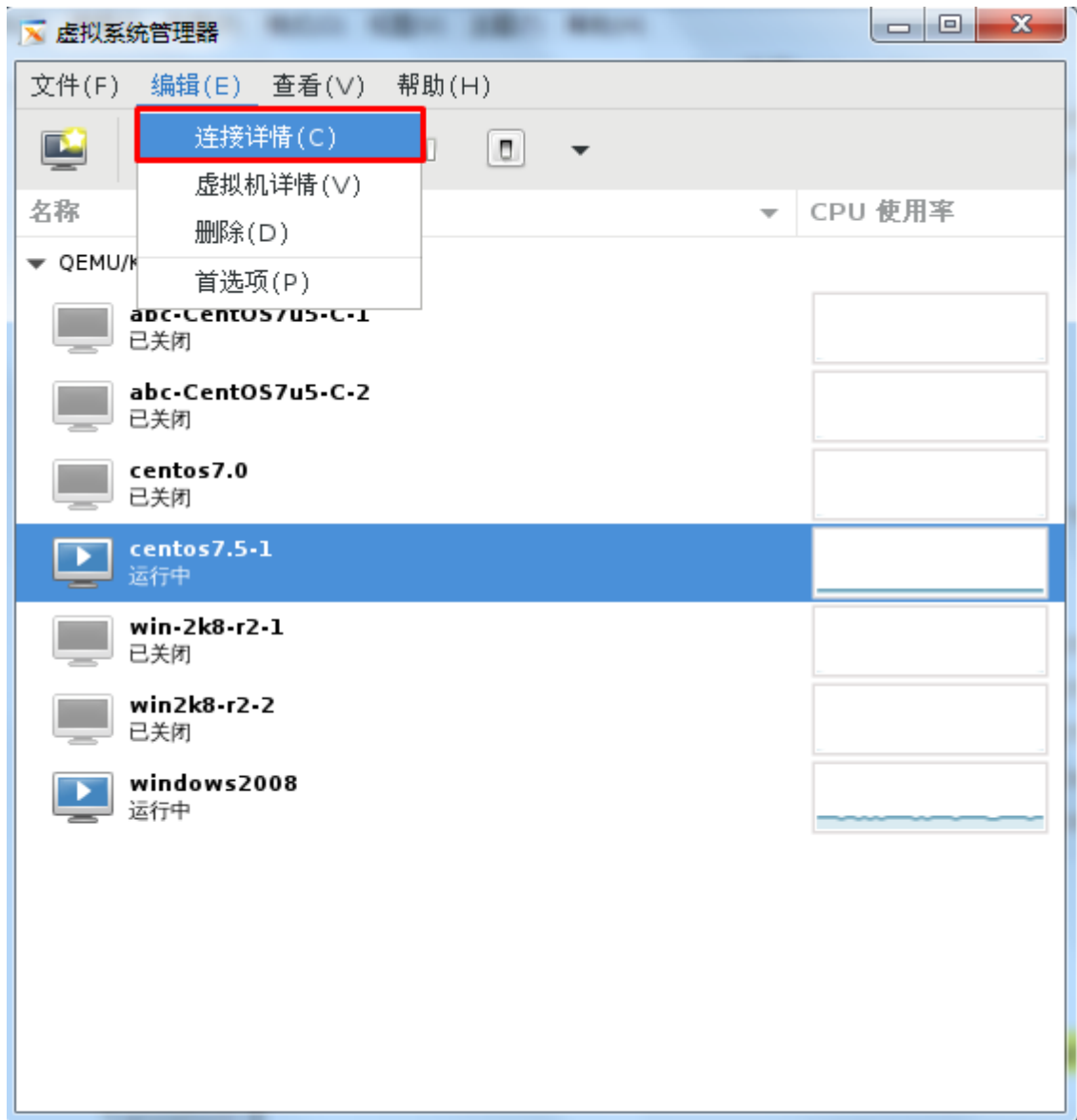
```

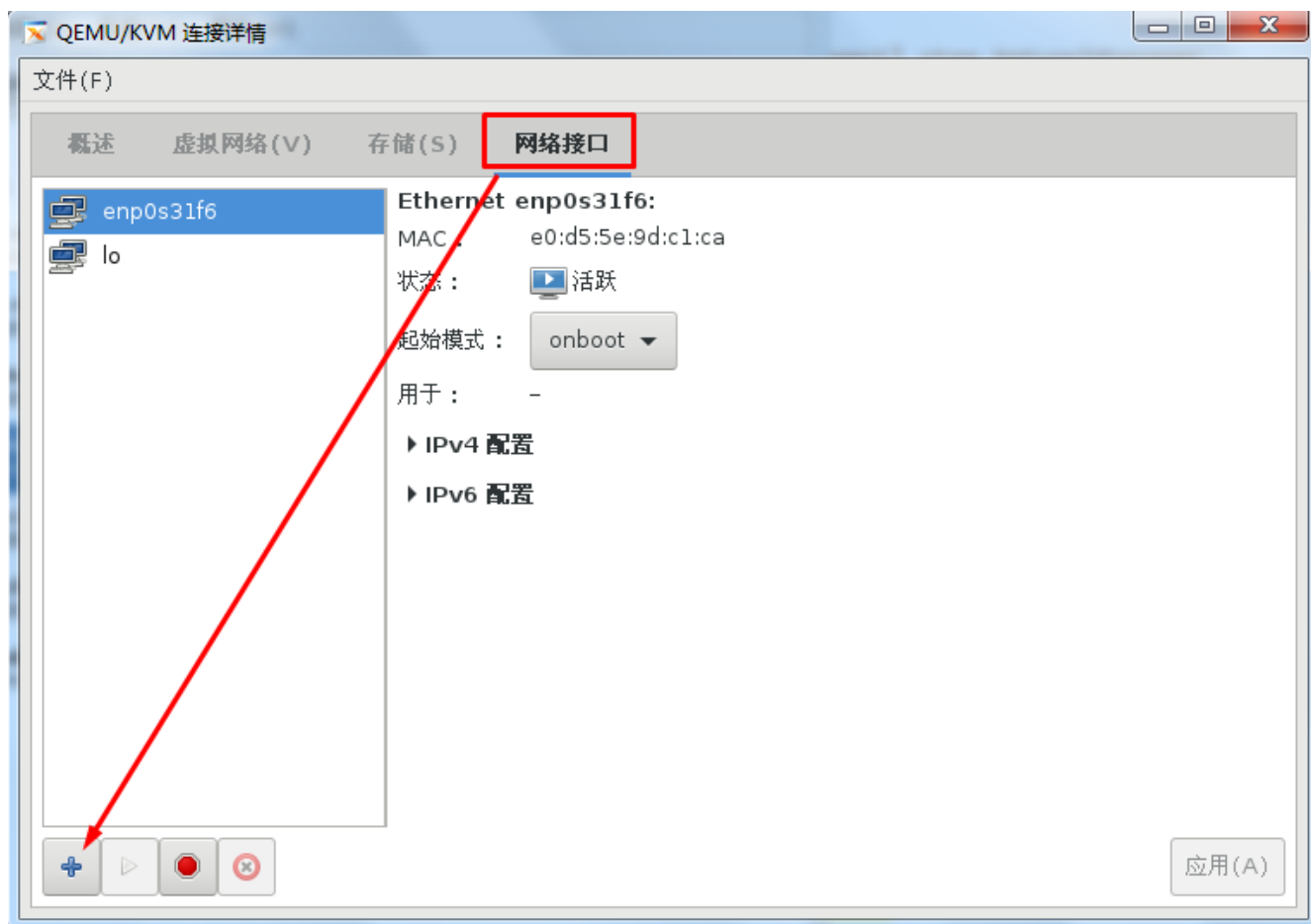
1 1. 停用NetworkManager
2 [root@localhost ~]# systemctl stop NetworkManager
3 [root@localhost ~]# systemctl disable NetworkManager
4
5 2. 创建桥
6 方法一：virt-manager
7
8 方法二：配置文件
9
10 方法三：virsh
11 [root@localhost ~]# virsh iface-bridge eth0 br0
12

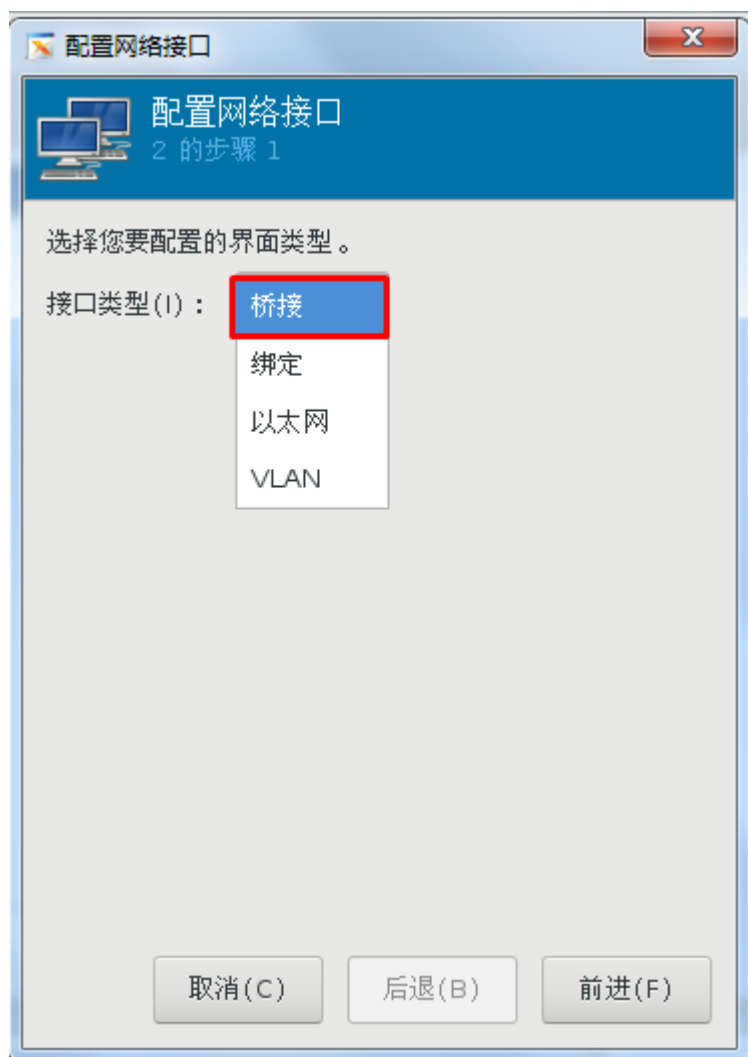
```

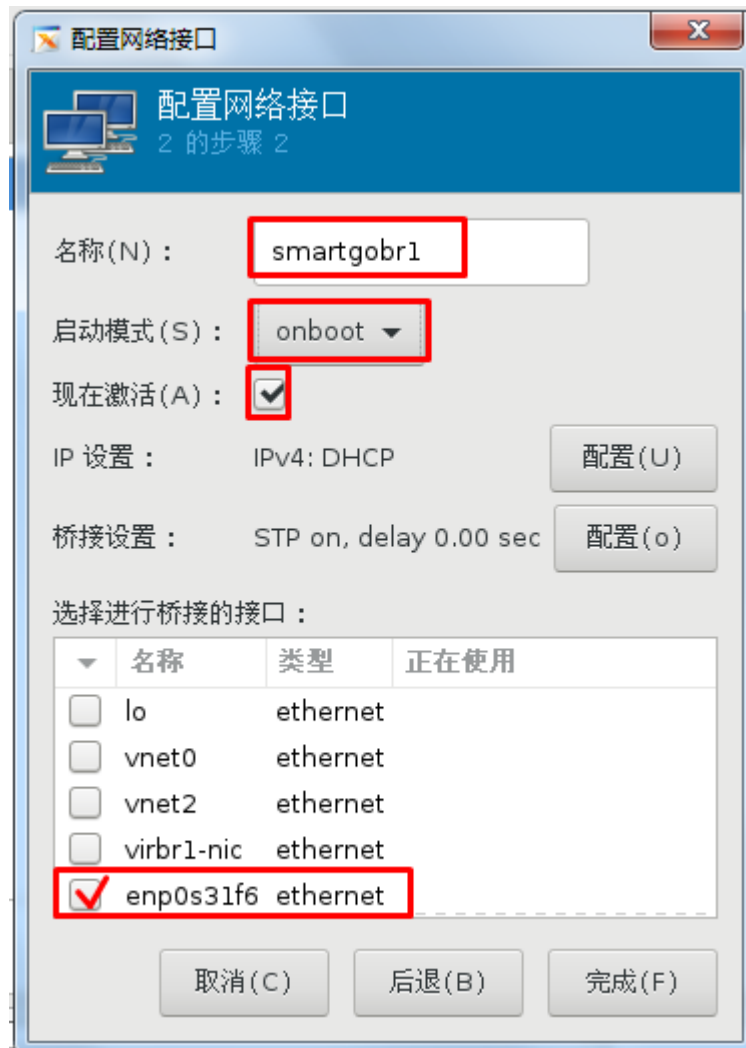
```
13 方法四：brctl (临时)
14 [root@localhost ~]# brctl addbr br1
15 [root@localhost ~]# brctl addif br1 eth1
16 [root@localhost ~]# brctl stp br1 on
17 [root@localhost ~]# ifconfig eth1 0
18 [root@localhost ~]# ifconfig br1 1.1.1.1/24 up
```

方法一具体步骤：









Guest virtual machine configuration

```
1 1. 客户机配置文件 <offline>
2 [root@localhost ~]# virsh edit centos7.5-1
3 <interface type='bridge'>
4 <source bridge='br1' />
5 </interface>
6
7 2. 查看是否生效
8 [root@localhost ~]# brctl show
```

参考：手动配置bridge

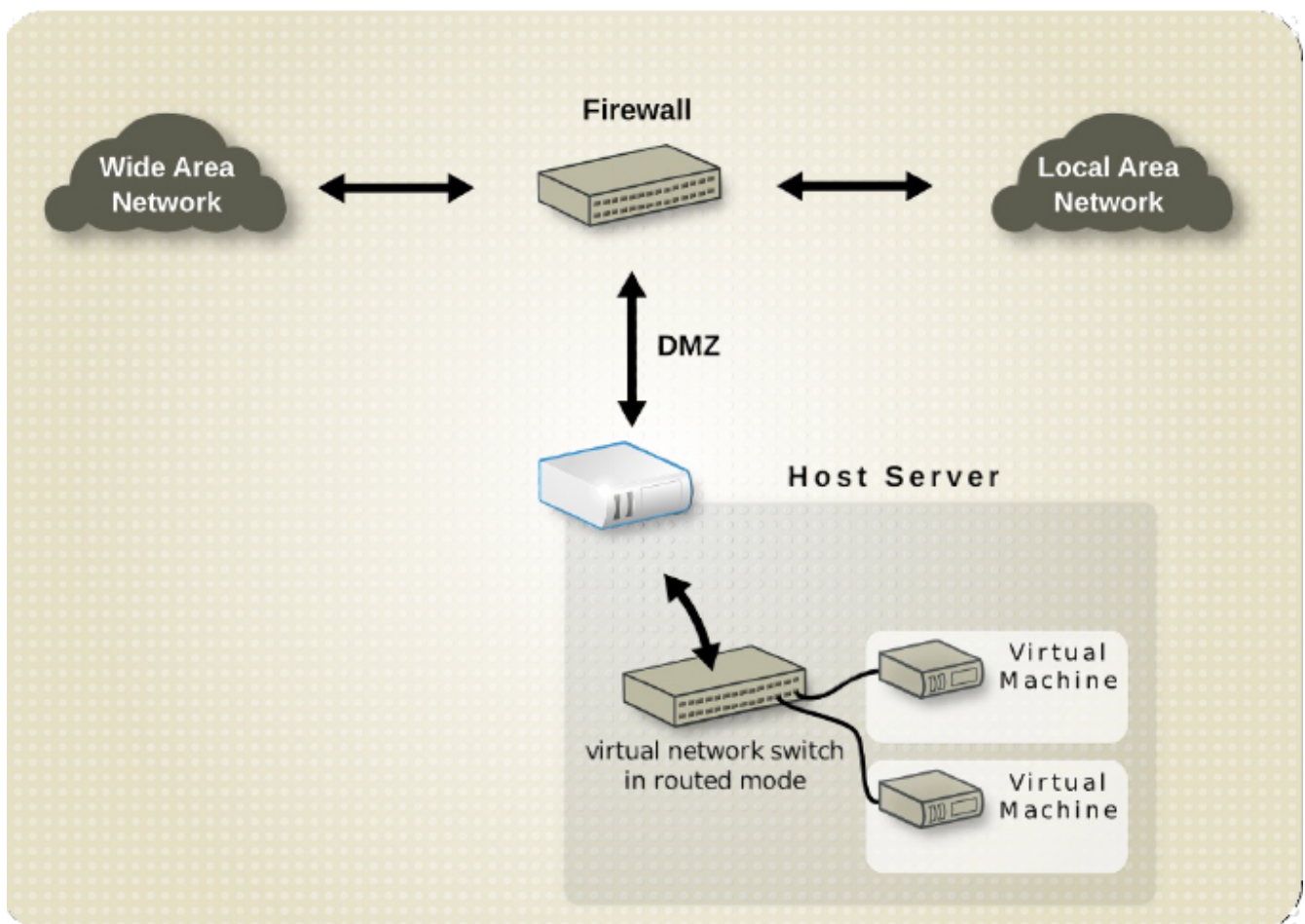
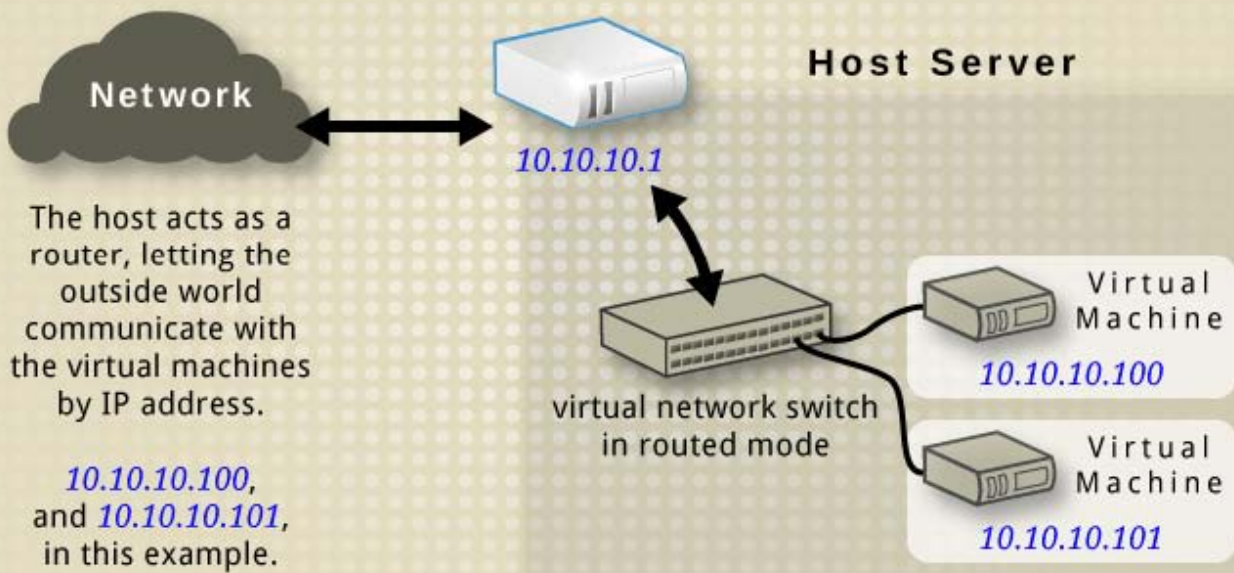
```
1
2 创建桥
3 [root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-br0
```

```
4      DEVICE=br0
5      TYPE=Bridge
6      BOOTPROTO=none
7      IPADDR=192.168.0.230
8      PREFIX=24
9      GATEWAY=192.168.1.254
10     DNS1=8.8.8.8
11     ONBOOT=yes
12     DELAY=0
13     STP=on
14
15     DHCP:
16     DEVICE=br0
17     ONBOOT=yes
18     TYPE=Bridge
19     BOOTPROTO=dhcp
20     STP=on
21     DELAY=0
22
23
24     将物理接口桥接到br0
25     [root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
26     DEVICE=eth0
27     ONBOOT=yes
28     BRIDGE=br0
29
```

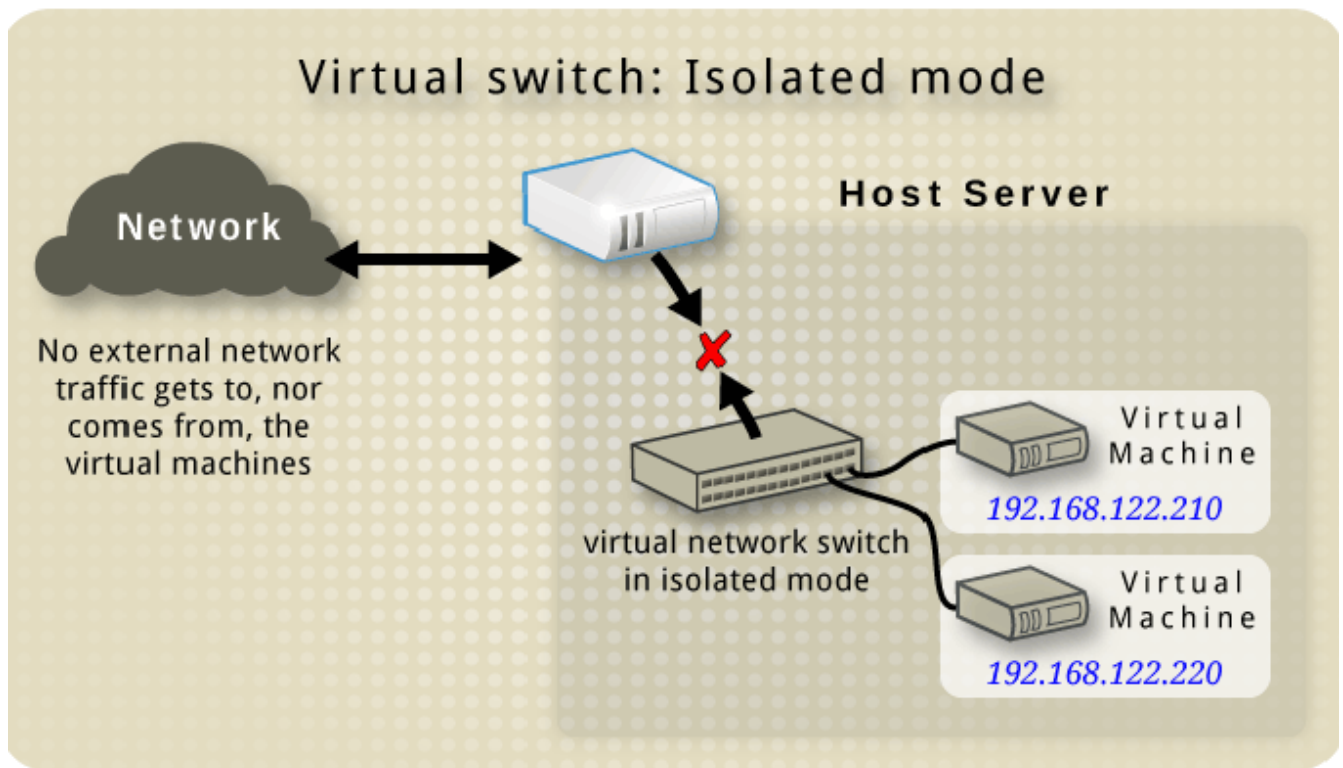
3.Routed mode

当使用路由模式时，所有的虚拟机在他们自己的子网中，通过虚拟交换机路由。路由模式在OSI网络模型中的第三层运行。

Virtual switch: Routed mode



4. Isolated mode



二、KVM存储虚拟化配置

学习目标

- ☐ 能够通过virt-manager添加硬盘
- ☐ 能够通过virsh添加硬盘
- ☐ 能够通过XML文件添加硬盘
- ☐ 能够了解本地存储的作用
- ☐ 能够设置本地存储
- ☐ 能够了解网络存储的作用
- ☐ 能够设置网络存储

1.热添加硬盘

方法一：

```
1  
2 [root@localhost ~]# virsh list
```

```

3 [root@localhost ~]# virsh --help |grep list
4
5 [root@localhost ~]# virsh domblklist centos7.5-1
6
7 [root@localhost ~]# qemu-img create -f qcow2 /disk1.img 1G
8
9
10 [root@localhost ~]# virsh --help | grep attach
11
12 [root@localhost ~]# virsh attach-disk centos7.5-1 --source /disk1.img --target vdb --
13 cache writeback --subdriver qcow2
14
15 [root@localhost ~]# virsh domblklist centos7.5-1
16
17 [root@localhost ~]# virsh attach-disk centos7.5-1 --source /disk2.img --target vdc --
18 cache writeback --subdriver qcow2 --persistent

```

方法二：基于xml文件

```

1 [root@localhost ~]# qemu-img create -f qcow2 /disk3.img 1G
2
3 [root@localhost ~]# vim temp.xml
4 <disk type='file' device='disk'>
5     <driver name='qemu' type='qcow2' cache='writeback' />
6     <source file='/disk3.img' />
7     <target dev='vdc' bus='virtio' />
8 </disk>
9
10 [root@localhost ~]# virsh attach-device centos7.5-1 temp.xml --persistent
11
12 [root@localhost ~]# virsh domblklist centos7.5-1
13

```

强制删除：

```

1 [root@localhost ~]# virsh detach-disk centos7.5-1 vdb
2
3 [root@localhost ~]# virsh domblklist centos7.5-1

```

2.存储池管理

- 本地存储池
- 网络存储池

2.1 本地存储

Directory-based Disk-based

Partition-based LVM-based

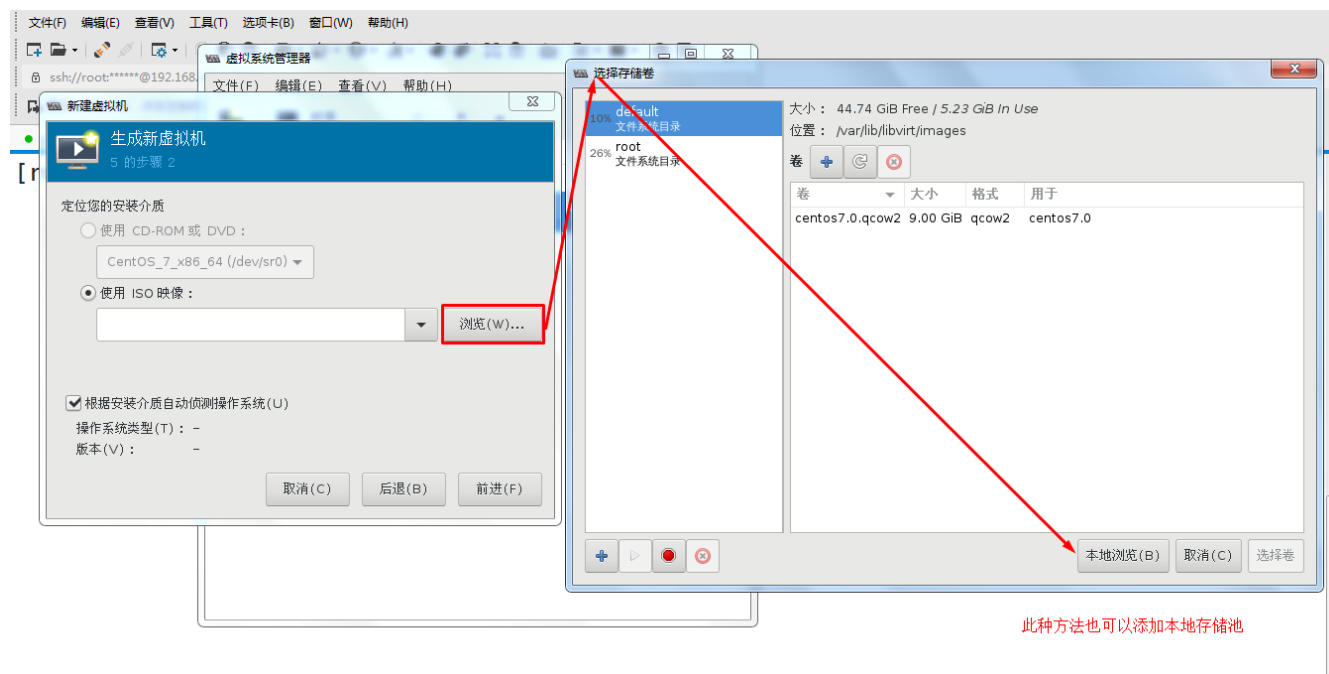
2.2 本地存储查看方法

```
1 [root@localhost ~]# virsh pool-list --all
2
3 [root@localhost ~]# ls /etc/libvirt/storage/
4
5 [root@localhost ~]# ls /etc/libvirt/storage/autostart/
6
7 [root@localhost ~]# virsh pool-edit default
8
9 [root@localhost ~]# ls /var/lib/libvirt/images/
```

2.3 本地存储添加方法

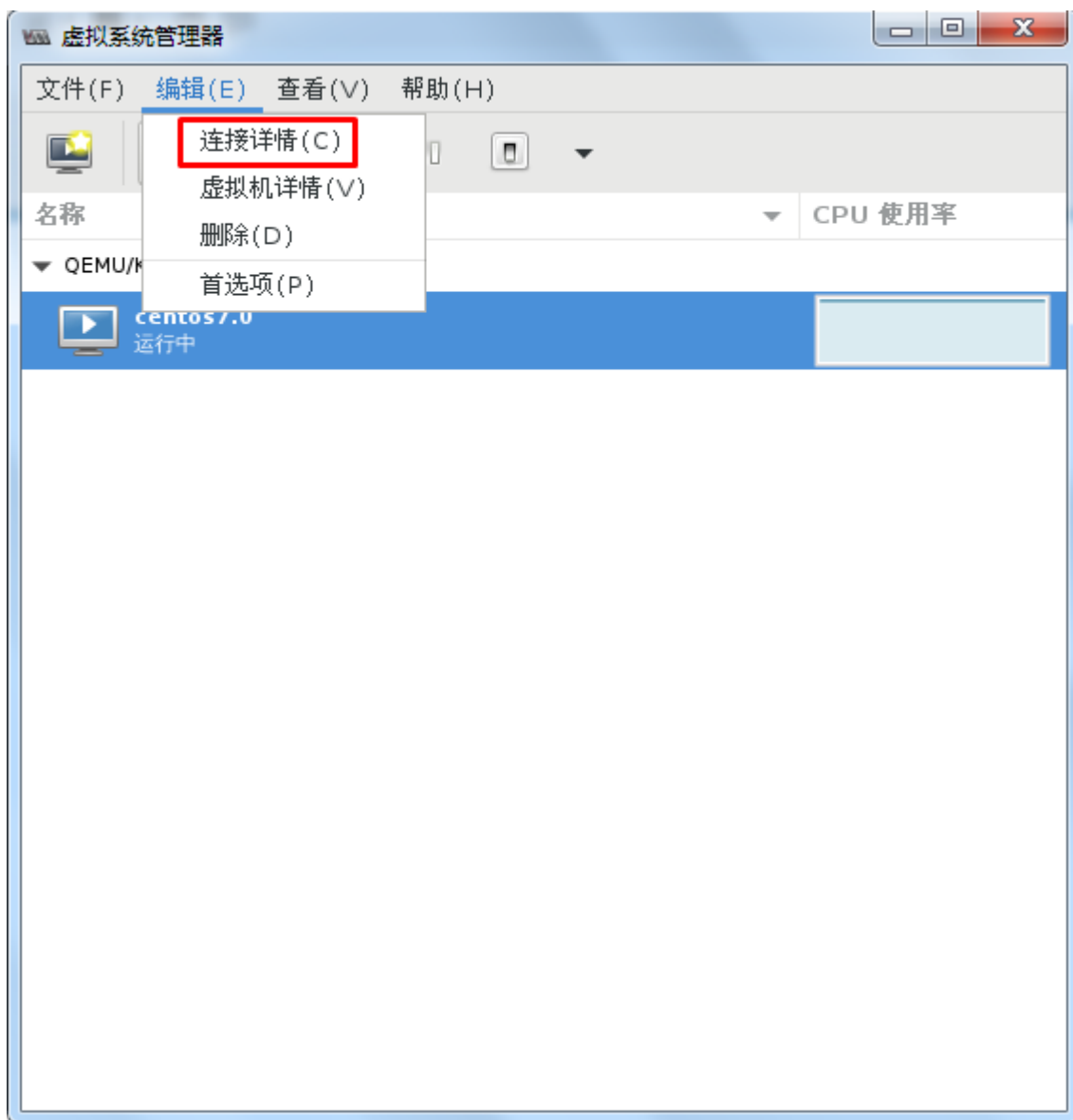
方法一

选择文件时，文件所有的目录会被自动添加为存储池

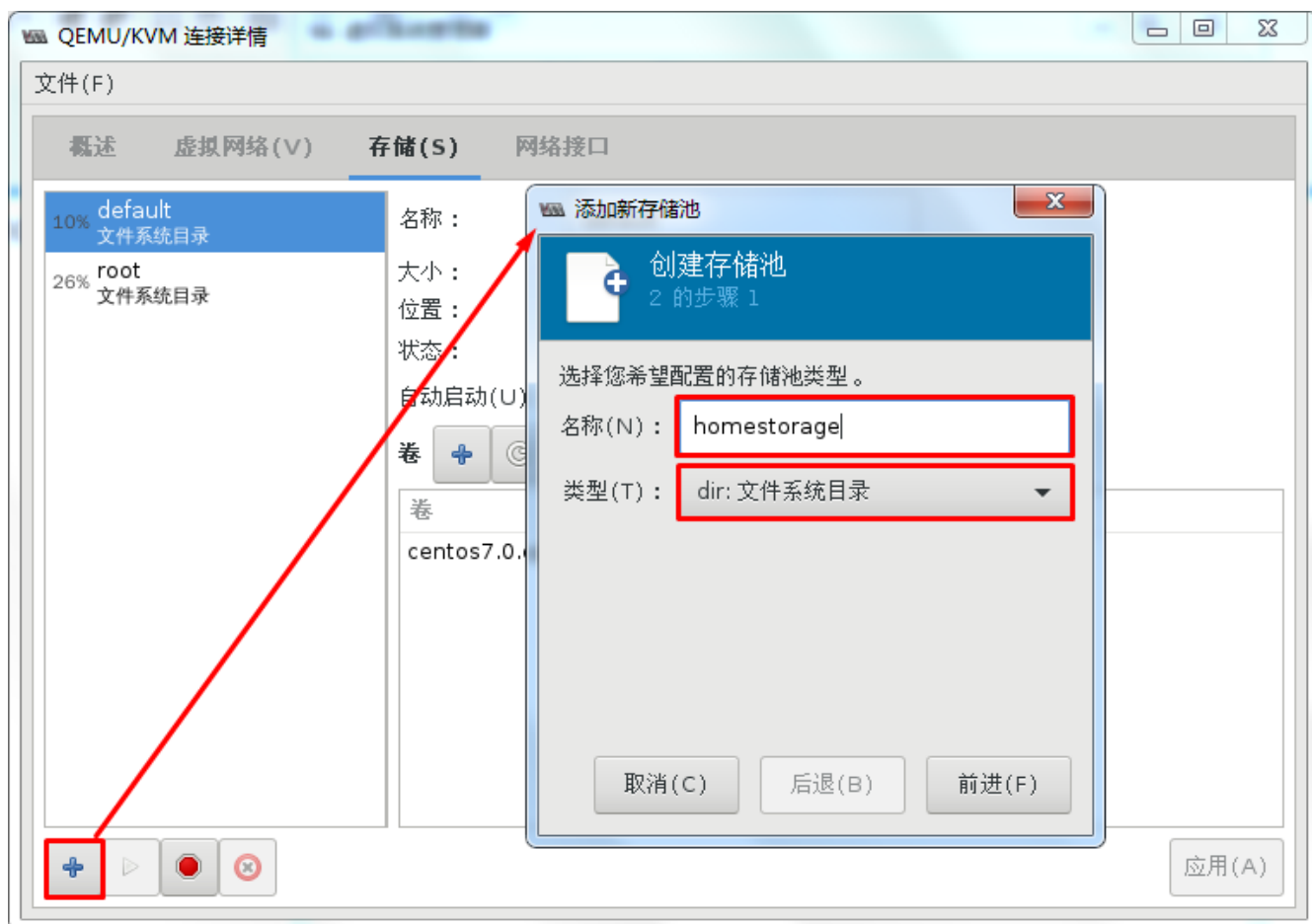


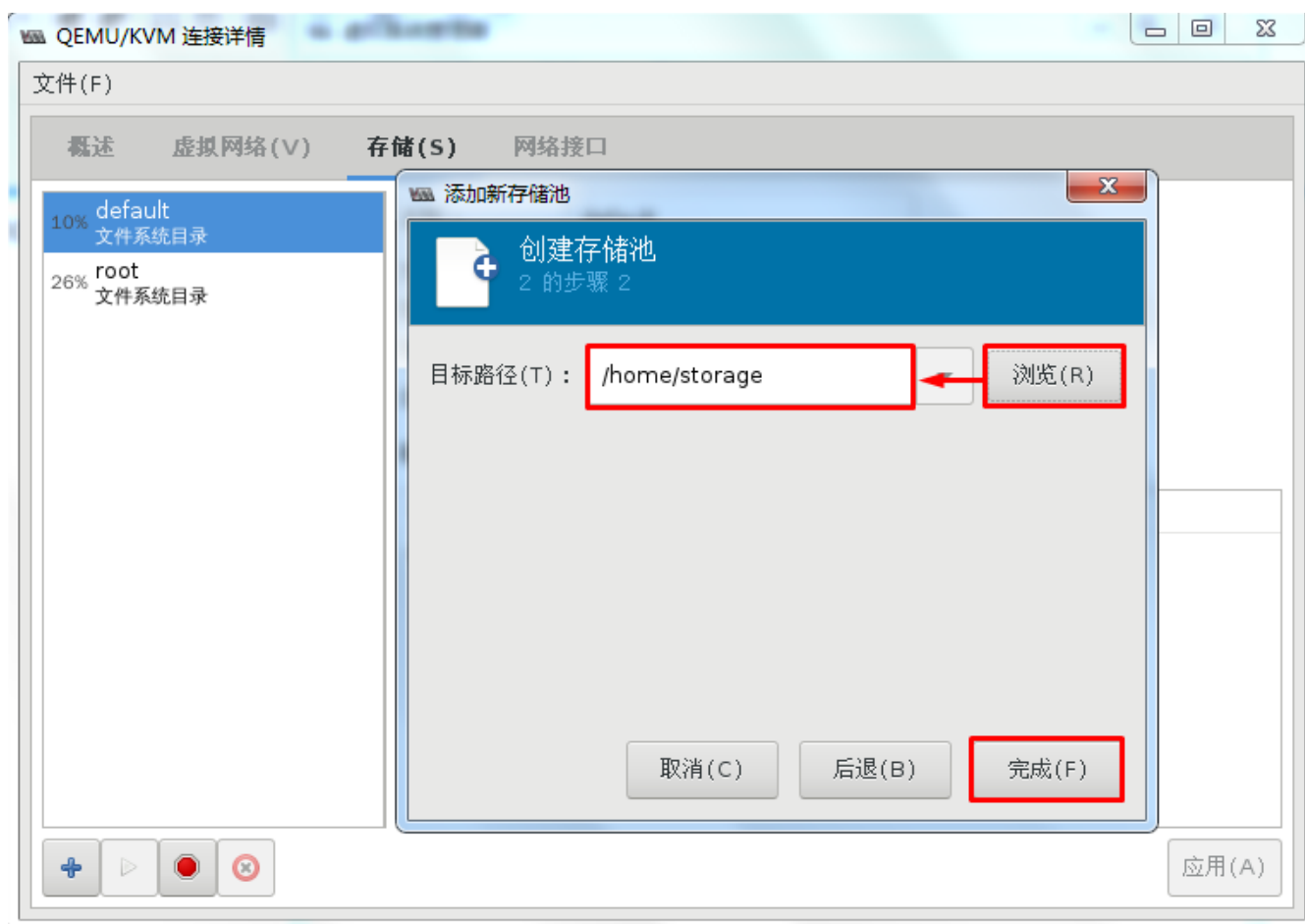
方法二

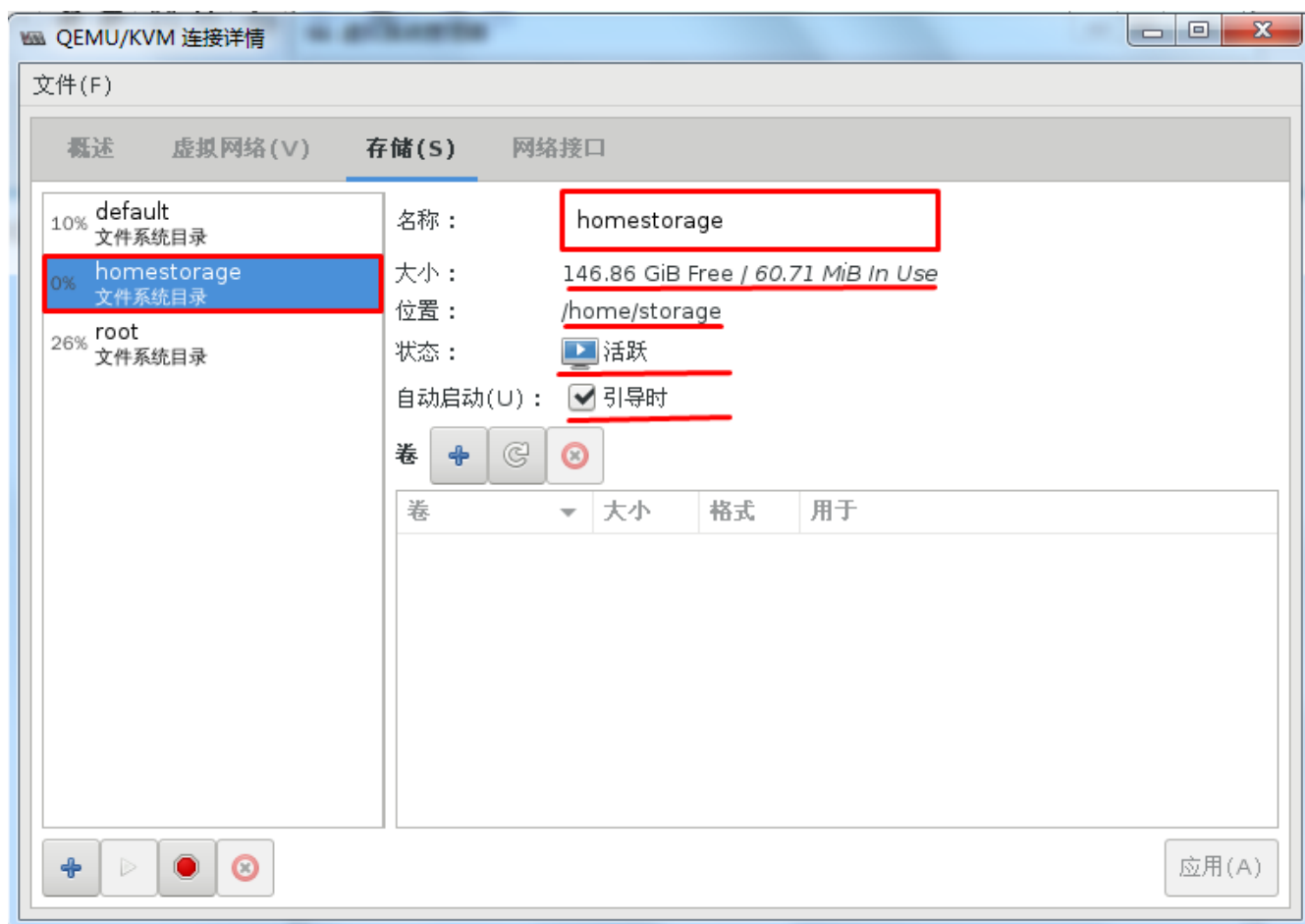
```
1 [root@localhost ~]# mkdir /home/storage
```

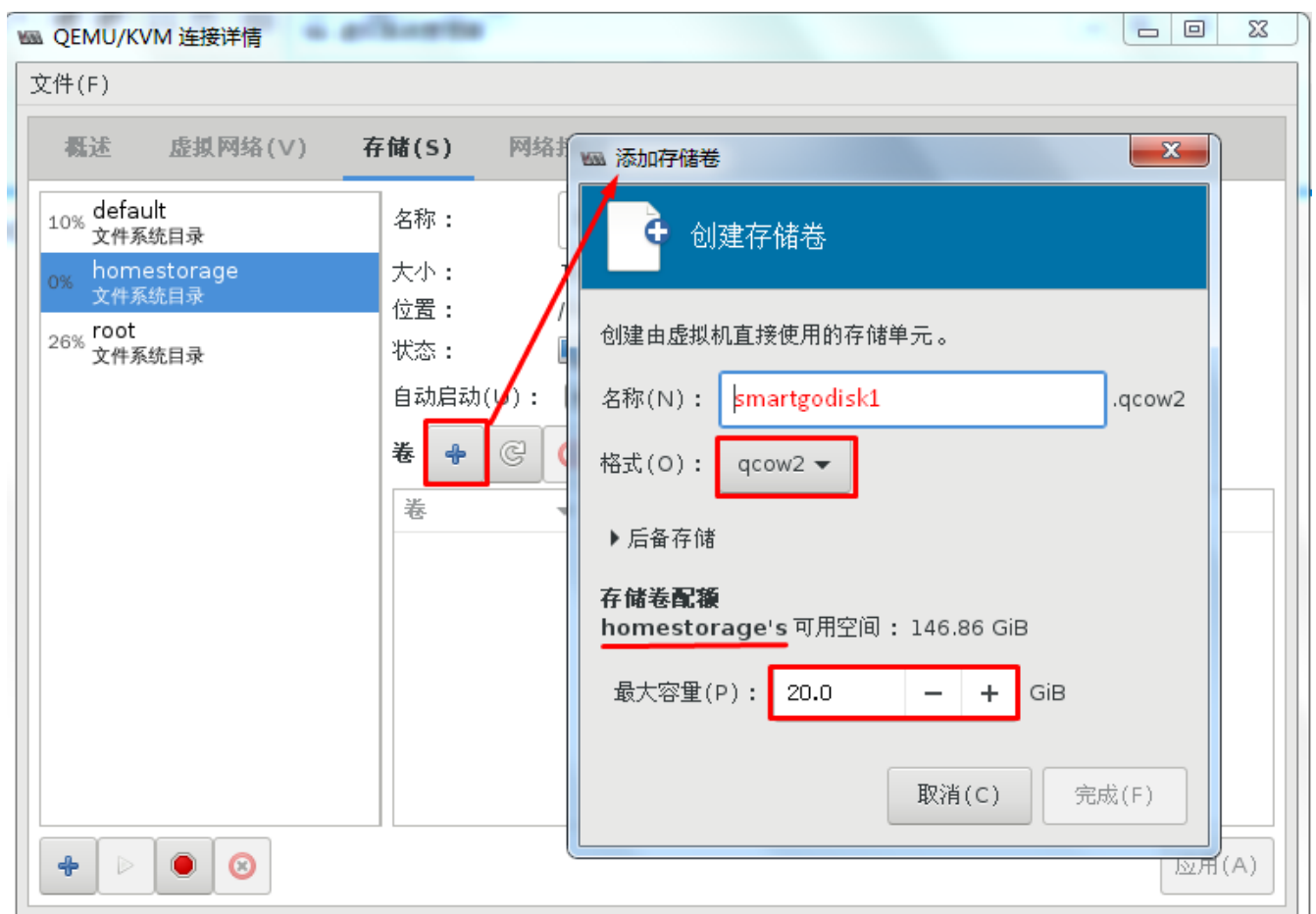
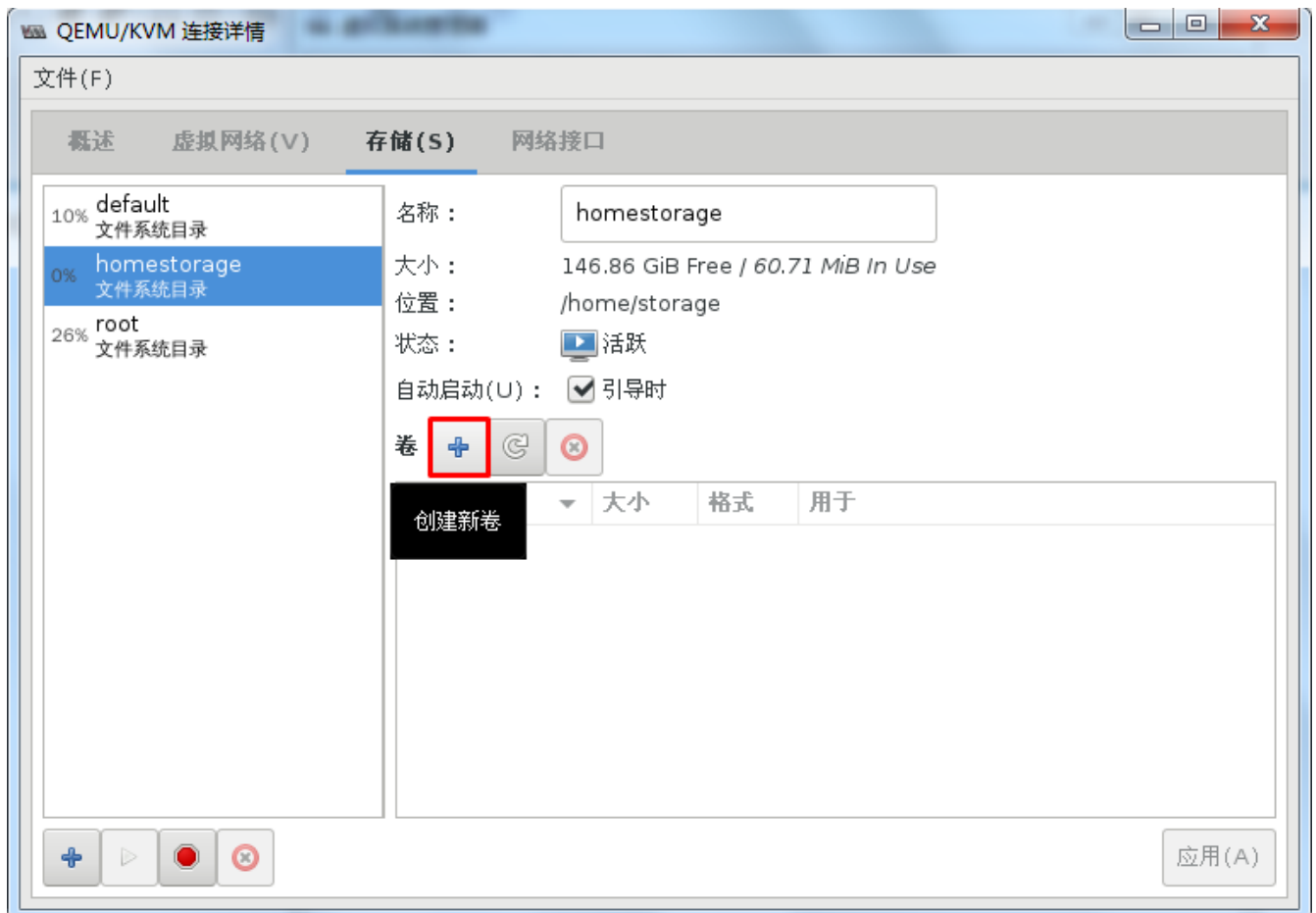


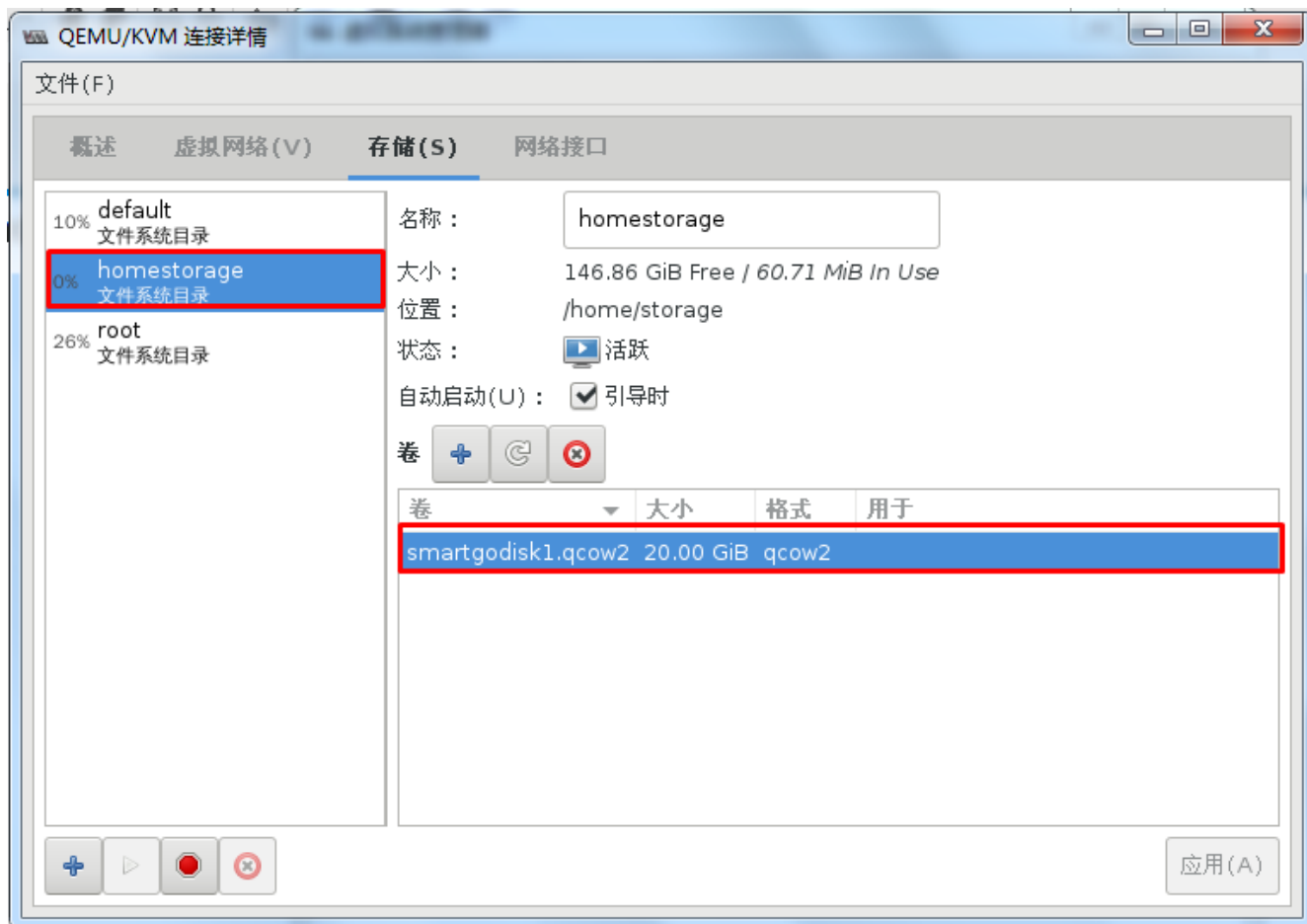










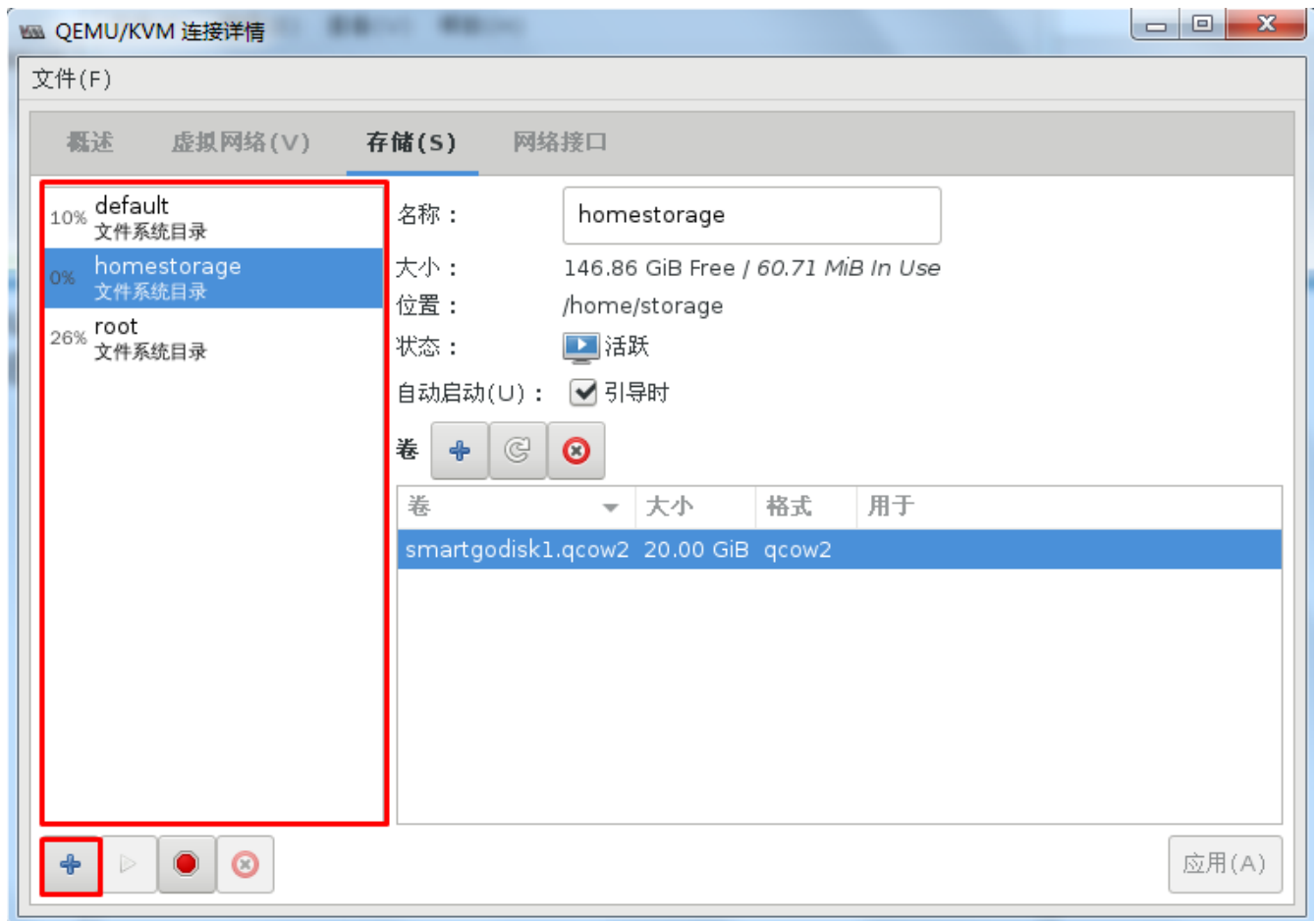


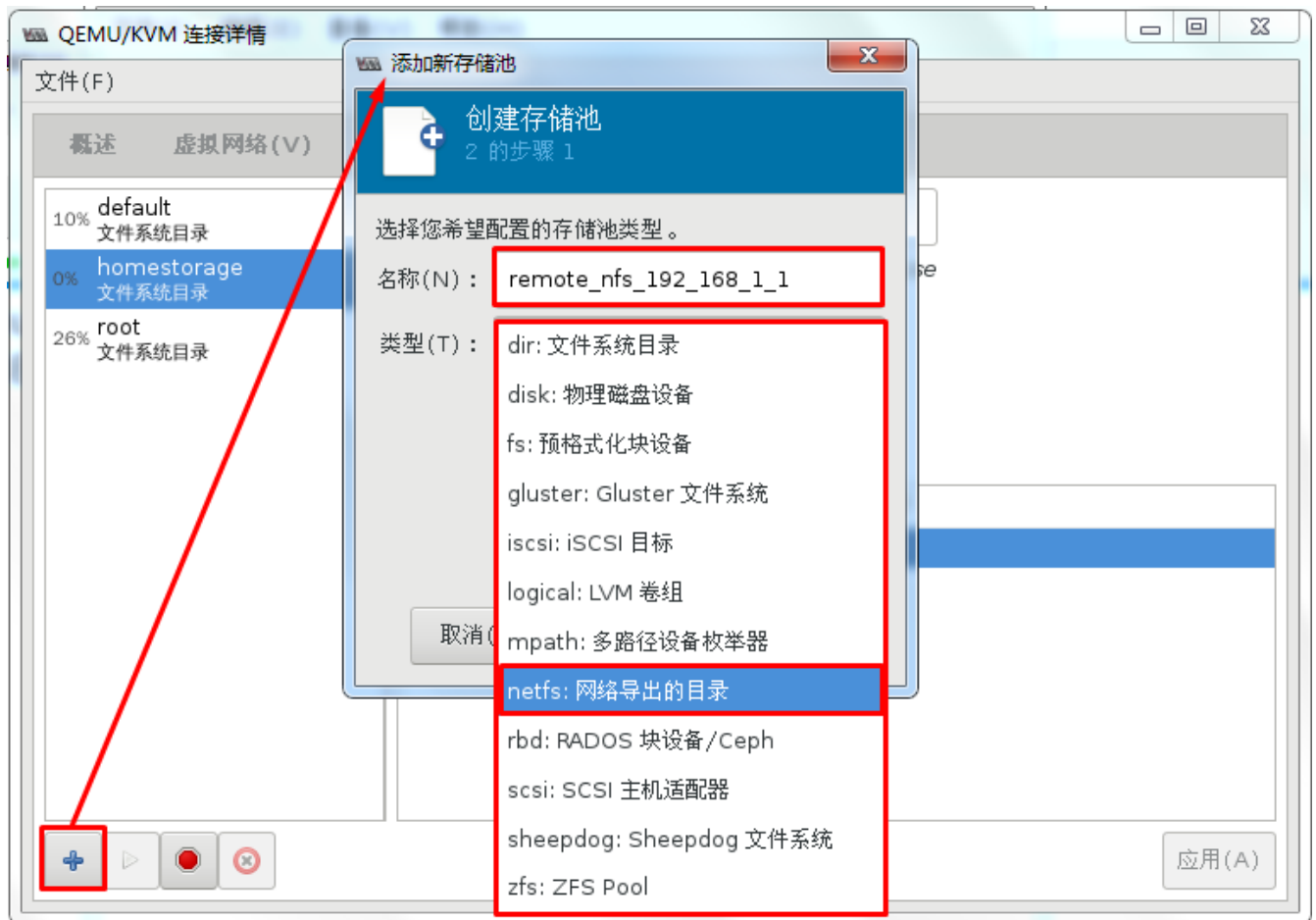
2.4网络存储池

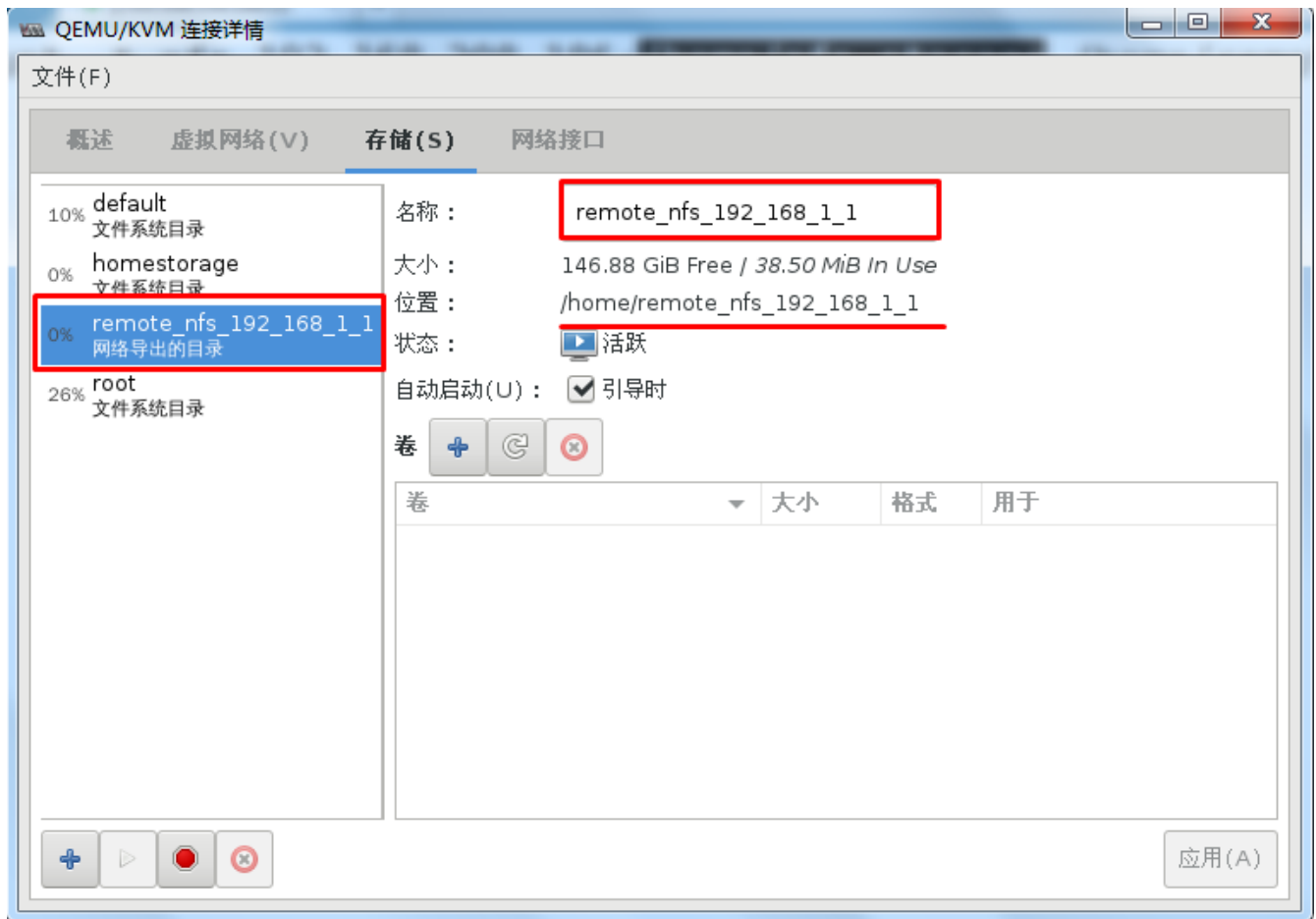
NFS-based GlusterFS-based Ceph-based iSCSI-based GFS-based SAN-based

方法一

```
1 使用NFS-based实现存储池
2
3 存储端: 192.168.1.1
4 [root@install ~]# mkdir /home/nfs-storage
5 [root@install ~]# vim /etc/exports
6 /home/nfs-storage 192.168.1.0/24(rw,sync,no_root_squash)
7
8
9 KVM主机端: 192.168.1.2
10 [root@localhost ~]# showmount -e 192.168.1.1
11 Export list for 192.168.1.1:
12 /home/nfs-storage 192.168.1.0/24
13 [root@localhost ~]# mkdir /home/remote_nfs_192_168_1_1
```







验证：

```
1 [root@localhost ~]# virsh pool-list --all
2 [root@localhost ~]# virsh pool-edit remote_nfs_192_168_1_1
```

方法二

```
1 virsh pool-define-as
2 pool-create-as name --print-xml type [source-host] [source-path] [source-dev] [source-
  name] [<target>] [--source-format format]
3
4 [root@localhost ~]# mkdir /home/remote_nfs_192_168_1_1 #如与方法一冲突，可考虑换目录名
5 [root@localhost ~]# virsh pool-define-as my-pool --source-format nfs --type netfs --
  source-host 192.168.1.1 --source-path /home/nfs-storage --target
  /home/remote_nfs_192_168_1_1
6 [root@localhost ~]# virsh pool-autostart my-pool #设置自启动
7 [root@localhost ~]# virsh pool-start my-pool #设置启动
8
```

my-pool 本地存储池名称

--target 本地挂载目录

安装虚拟机到存储池my-pool:

```
1 [root@localhost ~]# virt-install \
2     --vnc \
3     --name=smartgohost \
4     --ram=512 \
5     --arch=x86_64 \
6     --vcpus=1 \
7     --os-type=linux \
8     --os-variant=rhel7 \
9     --hvm \
10    --accelerate \
11    --disk path=/home/remote_nfs_192_168_1_1/smartgohost.img,size=8,format=qcow2 \
12    --bridge=virbr0 \
13    --cdrom=/root/centos7.5.iso
```

验证

```
1 [root@localhost ~]# ls /etc/libvirt/storage/
2 [root@localhost ~]# ls /etc/libvirt/storage/autostart/
```

方法三

基于XML添加存储池

virsh pool-define *.xml

三、镜像管理

学习目标

- ☐ 能够了解KVM虚拟机支持的镜像格式
- ☐ 能够使用qemu-img实现镜像创建
- ☐ 能够使用qemu-img实现镜像查看
- ☐ 能够使用qemu-img实现镜像格式转换
- ☐ 能够了解后备镜像作用
- ☐ 能够了解增量镜像作用
- ☐ 能够基于后备镜像制作增量镜像
- ☐ 能够使用增量镜像创建KVM虚拟机
- ☐ 能够使用guestfish实现镜像管理

1) QEMU支持的镜像格式

```
1 [root@localhost ~]# qemu-img -h
```

vmdk VMware镜像格式 vdi VirtualBox镜像格式 raw 简单的二进制镜像文件，一次性占用分配的磁盘空间，支持**稀疏**文件。qcow2 copy on write 写时复制技术 第二代的QEMU写时复制格式，QEMU推荐的镜像格式，支持稀疏文件，支持AES加密，支持基于zlib的压缩，支持snapshot，支持后备镜像

稀疏文件就是在文件中留有很多空余空间，留备将来插入数据使用。如果这些空余空间被ASCII码的NULL字符占据，并且这些空间相当大，那么，这个文件就被称为稀疏文件，而且，并不分配相应的磁盘块。这样，会产生一个问题，文件已被创建了，但相应的磁盘空间并未被分配，只有在有真正的数据插入进来时，才会被分配磁盘块，如果这时文件系统被占满了，那么对该文件的写操作就会失败。为防止这种情况，有两种办法：不产生稀疏文件或为稀疏文件留够空间。在计算机科学方面，稀疏文件是文件系统中的一种文件存储方式，在创建一个文件的时候，就预先分配了文件需要的连续存储空间，其空间内部大多都还未被数据填充现在有很多文件系统都支持稀疏文件，包括大部分的Unix和NTFS。稀疏文件被普遍用来磁盘镜像，数据库快照，日志文件，还有其他科学运用上。

2)QEMU磁盘镜像管理

2.1 创建磁盘镜像

```
1 [root@localhost ~]#qemu-img create disk1.raw 10G #raw格式
2
3 [root@localhost ~]#qemu-img create disk2.qcow2 -f qcow2 10G #qcow2格式
```

2.2 查看磁盘镜像

```
1 [root@localhost ~]#qemu-img info disk1.raw
2
3 [root@localhost ~]#qemu-img info disk2.qcow2
```

```
1 [root@localhost ~]#ll -h disk*
2
3 [root@localhost ~]#du -h disk*
```

2.3 磁盘镜像格式转换

```
1 [root@localhost ~]#qemu-img info disk1.raw
2 [root@localhost ~]#qemu-img convert -p -f raw -O qcow2 disk1.raw disknew1.qcow2
```

```
1 [root@localhost ~]#qemu-img info disk2.qcow2
2 [root@localhost ~]#qemu-img convert -p -f qcow2 -O raw disk2.qcow2 disknew2.raw
```

2.4 磁盘镜像快照管理 (qcow2)

```
1 [root@localhost ~]# qemu-img snapshot -l centos7.qcow2
2 [root@localhost ~]# qemu-img snapshot -c sp1 centos7.qcow2
3 [root@localhost ~]# qemu-img snapshot -a sp1 centos7.qcow2
4 [root@localhost ~]# qemu-img snapshot -d sp1 centos7.qcow2
```

2.5 基于后备镜像转换为增量镜像

多个虚拟机使用的增量镜像（前端镜像）都指向同一个后备镜像，默认从后备镜像读，写则到自己的增量镜像。

后备镜像可以是raw或qcow2，增量镜像只能是qcow2

使用增量镜像的优势：

- a. 节省磁盘空间
- b. 瞬间快速创建虚拟机

```
1 [root@localhost ~]# qemu-img create -f qcow2 -b centos7.back.img centos7.front.img
2 [root@localhost ~]# qemu-img info centos7.front.img
```

2.6 转换增量镜像为普通镜像

```
1 [root@localhost ~]# qemu-img convert -f qcow2 -o qcow2 centos7.front.img
  centos7.back.new.img
2 [root@localhost ~]# qemu-img info centos7.back.new.img
```

作业：利用后备镜像完成多虚拟机创建

3) guestfish 磁盘镜像管理

```
1 [root@localhost ~]# yum -y install libguestfs-tools # linux
2 [root@localhost ~]# yum -y install libguestfs-winsupport # windows
```

```
1 [root@localhost ~]# virt-inspector centos7.img
```

```
1 [root@localhost ~]# virt-ls -a centos7.qcow2 /root
2 [root@localhost ~]# virt-df -a centos7.qcow2
3 [root@localhost ~]# virt-cat -a centos7.qcow2 /etc/hosts
4 [root@localhost ~]# virt-copy-in /etc/hosts -a centos7.qcow2 /tmp
5 [root@localhost ~]# virt-copy-out -a centos7.qcow2 /tmp/hosts /home
```

四、KVM虚拟机管理

学习目标

- ☐ 能够启动KVM虚拟机
- ☐ 能够暂停KVM虚拟机
- ☐ 能够关闭KVM虚拟机

```
1 [root@localhost ~]# virt-top #查看虚拟机使用本地资源情况
```



```
1 [root@localhost ~]# virsh domstate centos7    #被克隆域必须为关闭或暂停状态
2 [root@localhost ~]# virt-clone -o centos7 -n centos7_new -f
   /home/remote_nfs_192_1_1/centos7_new.img
```

-o为原始客户机名称

-n 新客户机名称

-f 新客户机的磁盘像文件

```
1 [root@localhost ~]# virsh start centos7
```

```
1 [root@localhost ~]# virsh destroy centos7
```

```
1 [root@localhost ~]# virsh shutdown centos7
```

```
1 [root@localhost ~]# virsh domstate centos7
```

```
1 [root@localhost ~]# virsh snapshot-create centos7
2
3 [root@localhost ~]# virsh snapshot-create-as centos7 snapshotname
4 [root@localhost ~]# virsh snapshot-list centos7
5 [root@localhost ~]# virsh snapshot-revert centos7 snapshotname
6 [root@localhost ~]# virsh snapshot-current centos7
```

snapshotname为快照名

五、批量管理虚拟机

学习目标

- ☐ 能够为批量管理KVM虚拟机创建虚拟机模板
- ☐ 能够编写KVM虚拟机批量管理脚本
- ☐ 能够通过虚拟机批量管理脚本实现虚拟机创建等

1.Linux主机模板

1.安装系统 镜像格式: qcow2 磁盘size: 20G 磁盘bus: virtio 网卡model: virtio

2.yum install bash-completion

3.yum update

4.ntp

5.yum configure

6.firewalld/iptables/selinux

systemctl disable firewalld iptables

sed -ri '/^SELINUX=/cSELINUX=disabled' /etc/selinux/config

7.配置本地console连接

vim /etc/sysconfig/grub

GRUB_CMDLINE_LINUX="..... console=ttyS0"

grub2-mkconfig -o /boot/grub2/grub.cfg

8.ssh连接慢的问题

vim /etc/ssh/sshd_config

UseDNS no GSSAPIAuthentication no

```
1 | [root@localhost ~]# chatter +i centos7.qcow2
```

不要使用该镜像启动系统

2.windows主机模板

1.安装系统 镜像格式: qcow2 磁盘size: 50G 磁盘bus: virtio 网卡model: virtio

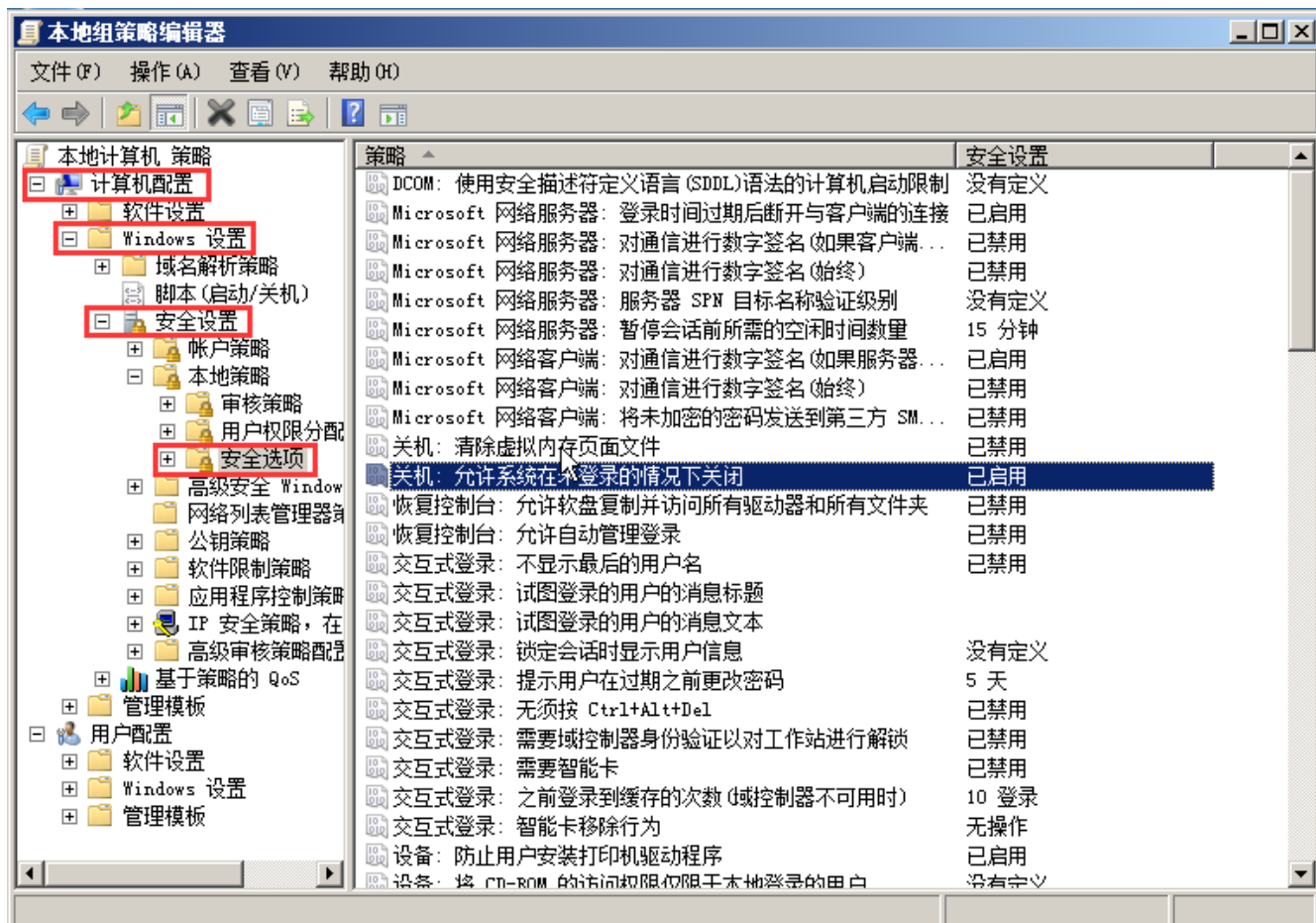
2.防火墙 3389/tcp icmp

3.远程桌面配置

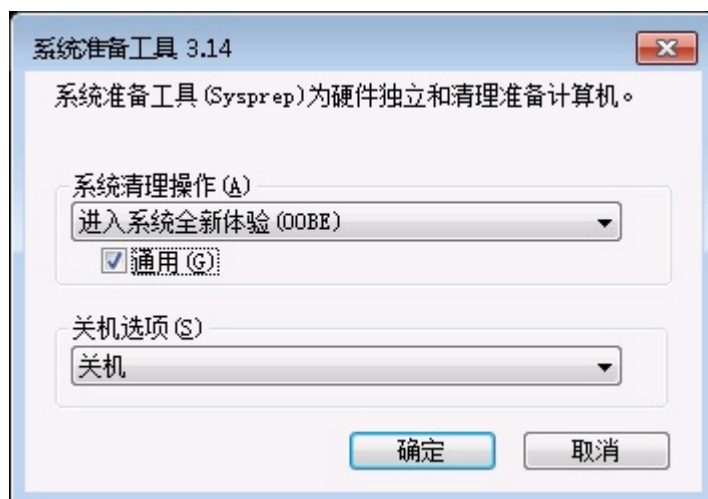
4.取消登录任务配置向导

5.添加SNMP服务 服务器管理器 ——> 功能 ——> 添加功能 ——> SNMP服务

6.配置允许系统在未登录情况下关闭



7. 通过系统准备工具移除Windows个性化信息 完成初始化封装后，通过该镜像部署的虚拟机在首次启动时将自动更新SID、主机名及配置。



C:/windows/system32/sysprep/

```
1 [root@localhost ~]#chattr +i win2k8r2.img
```

不要使用该镜像启动系统

3.编写虚拟机批量管理脚本（shell）

步骤 1.准备后端镜像模板 centos7.img 2.创建虚拟机配置文件（name,uuid,disk,mac） 3.创建虚拟机的磁盘文件（增量镜像）

```
1 1.创建虚拟机的磁盘文件
2 [root@localhost ~]# qemu-img create -f qcow2 -b centos7.back.img centos7.front.img
3
4 2.创建虚拟机的配置文件
5 [root@localhost ~]# cp /etc/libvirt/qemu/centos7.5-1.xml centos7.xml
6
7 <name>域名称</name>
8 <uuid>域UUID</uuid>
9 <source file='域磁盘镜像文件' />
10 <mac address='域网卡MAC地址' />
11
12
13 3.define虚拟机&&启动虚拟机
14 [root@localhost ~]# virsh define centos7.xml
15 [root@localhost ~]# virsh start centos7
```

```
1 [root@localhost ~]# vm_uuid=$(uuidgen); echo $vm_uuid
2 [root@localhost ~]# vm_mac="52:54:$(dd if=/dev/urandom count=1 2>/dev/null | md5sum |
sed -r 's/^(..)(..)(..)(..).*$/\1:\2:\3:\4/')"; echo $vm_mac
```

以下脚本示例

```
1 [root@localhost ~]#vim /home/vm/virt.sh
2 #!/bin/bash
3 # KVM虚拟机管理工具
4 #
5
6 workdir=/home
7 xml=/etc/libvirt/qemu
8 img=/var/lib/libvirt/images
9 vmxmlroot=$workdir/vmm
10 vmdiskroot=$workdir/vmm
11
12 menu() {
13     cat <<- EOF
14     -----
15     A.部署CentOS7u6系统
```

```

16      B.部署win2k8r2系统
17      -----
18      EOF
19  }
20  menu
21
22  read -p "请选择需要安装的操作系统:" choose
23
24
25  case $choose in
26      A|a)
27          read -p "请输入您想使用的域名称:" name
28
29          vmname=${name}
30          vmuuid=$(uuidgen)
31          vmdisk=${name}.qcow2
32          vmmac="52:54:$(dd if=/dev/urandom count=1 2>/dev/null | md5sum | sed -r
's/^(..)(..)(..)(..).*$/\1:\2:\3:\4/')"
33
34          cp $vmxmlroot/centos7u6/centos7u6.xml $xml/${vmname}.xml
35          qemu-img create -f qcow2 -b $vmdiskroot/centos7u6/centos7.6.qcow2
$img/${vmdisk} &> /dev/null
36          sed -ri "s#vmname#${vmname}#" $xml/${vmname}.xml
37          sed -ri "s#vmuuid#${vmuuid}#" $xml/${vmname}.xml
38          sed -ri "s#vmdisk#$img/${vmdisk}#" $xml/${vmname}.xml
39          sed -ri "s#vmmac#${vmmac}#" $xml/${vmname}.xml
40          virsh define $xml/${vmname}.xml
41          ;;
42
43      B|b)
44
45          echo "win2k8r2已安装完成"
46          ;;
47      *)
48          echo "请按菜单选择 ( A | B )"
49  esac
50  virt-manager

```

六、KVM虚拟化管理平台

学习目标

- ☐ 能够了解批量管理虚拟机平台
- 公有云
 - AWS、Aliyun等等
- 私有云
 - OpenStack、VM EXSi等

七、扩展阅读

1.X86特权级

<https://blog.csdn.net/farmwang/article/details/50094959>