一、环境准备

还原快照（xshell完成），开启虚拟机A（只需要一台）

二、服务的管理

•上帝进程：systemd

•Linux系统和服务管理器

–是内核引导之后加载的第一个初始化进程（PID=1）

–负责掌控整个Linux的运行/服务资源组合

•一个更高效的系统&服务管理器

–开机服务并行启动，各系统服务间的精确依赖

–配置目录：/etc/systemd/system/

–服务目录：/lib/systemd/system/

–主要管理工具：systemctl

[root@svr7 ~]# systemctl -t service --all #列出所有的服务

•对于服务的管理

systemctl restart 服务名 #重起服务

systemctl start 服务名 #开启服务

systemctl stop 服务名 #停止服务

systemctl status 服务名 #查看服务当前的状态

systemctl enable 服务名 #设置服务开机自启动

systemctl disable 服务名 #设置服务禁止开机自启动

systemctl is-enabled 服务名 #查看服务是否开机自启

[root@svr7 ~]# yum -y install httpd

[root@svr7 ~]# vim /etc/resolv.conf #清除文件内容

[root@svr7 ~]# killall httpd #杀死手动启

动的httpd

Kill -l 查看kill可以加的选项

[root@svr7 ~]# systemctl restart httpd #重启httpd服务

[root@svr7 ~]# systemctl status httpd #查看服务httpd状态

[root@svr7 ~]# systemctl enable httpd #设置httpd开机自启动

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

[root@svr7 ~]# systemctl is-enabled httpd #查看httpd是否是开机自启动

[root@svr7 ~]# systemctl disable httpd #关闭httpd开机自启动

Removed symlink /etc/systemd/system/multi-user.target.wants/httpd.service.

[root@svr7 ~]# systemctl is-enabled httpd

**管理运行级别**

RHEL6:运行级别 300

0：关机 0个服务

1：单用户模式（基本功能的实现，破解Linux密码） 50个服务

2：多用户字符界面（不支持网络） 80个服务

**3：多用户字符界面（支持网络）服务器默认的运行级别** 100个服务

4：未定义 0个服务

**5：图形界面 300个服务**

6：重起 0个服务

切换运行级别：init 数字

RHEL7：运行模式（运行级别）

字符模式：multi-user.target

图形模式：graphical.target

当前直接切换到字符模式

[root@svr7 /]# systemctl isolate multi-user.target #相当于原来的init 3

当前直接切换到图形模式

[root@svr7 /]# systemctl isolate graphical.target #相当于原来的init 5

查看每次开机默认进入模式

[root@svr7 /]# systemctl get-default

设置永久策略，每次开机自动进入multi-user.target

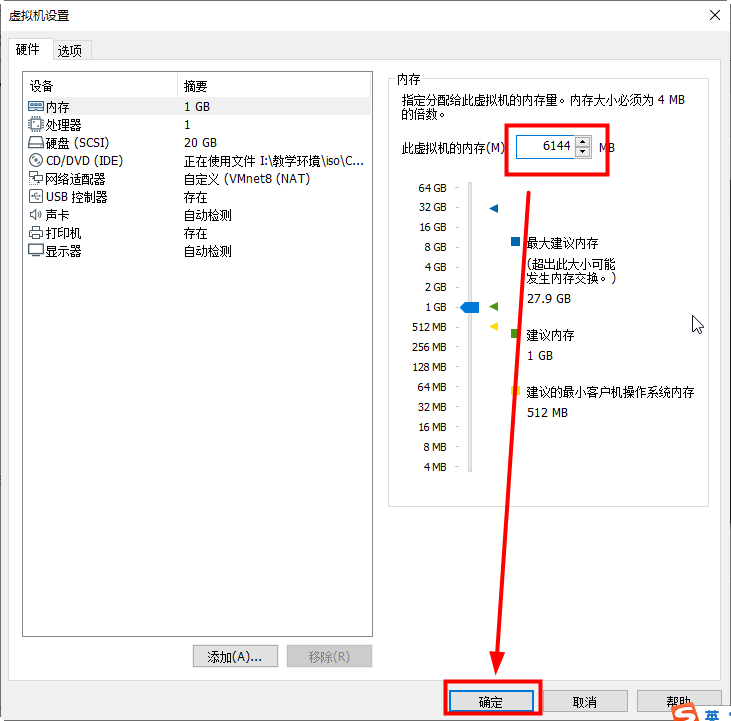
[root@svr7 /]# systemctl set-default multi-user.target

[root@svr7 /]# reboot

三、虚拟机环境设置

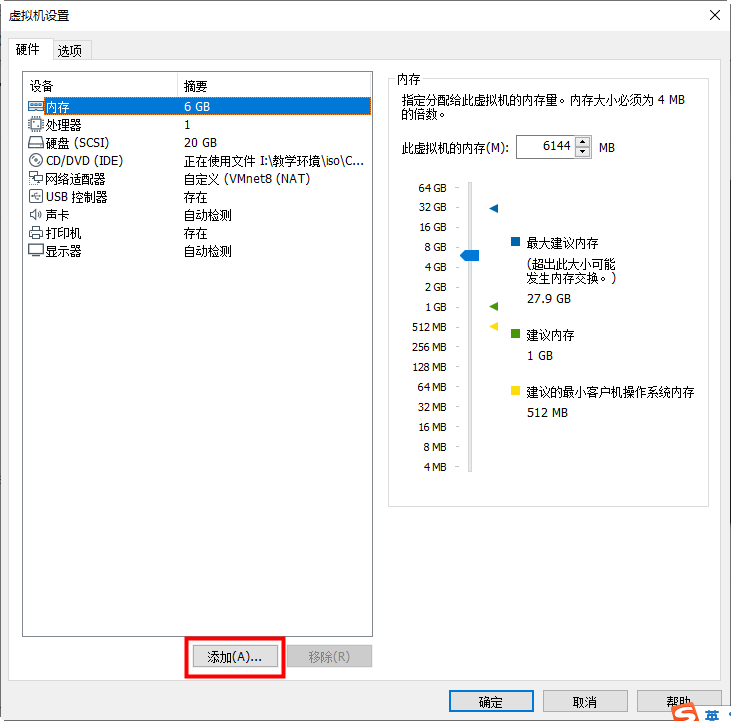
1.内存修改大小6G

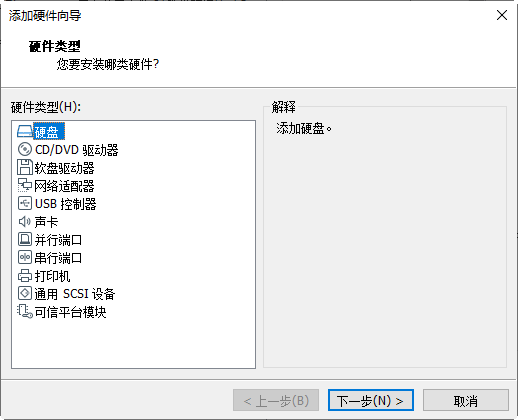




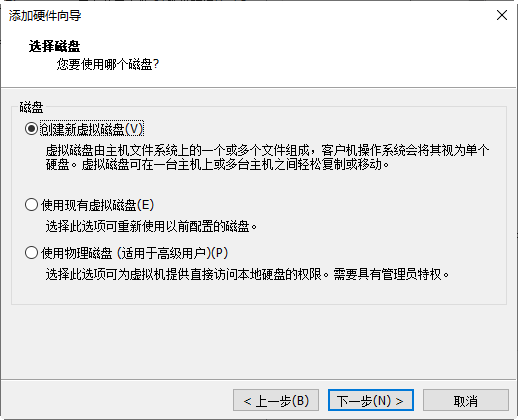
2.添加一块50G的硬盘

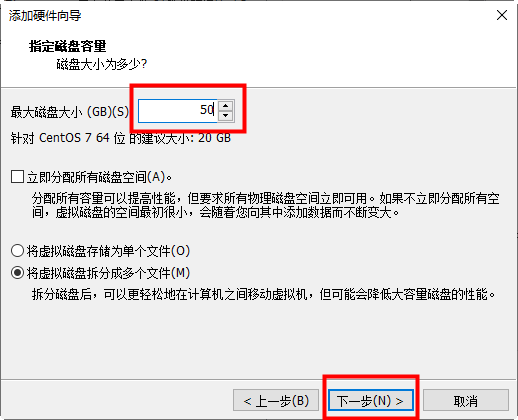




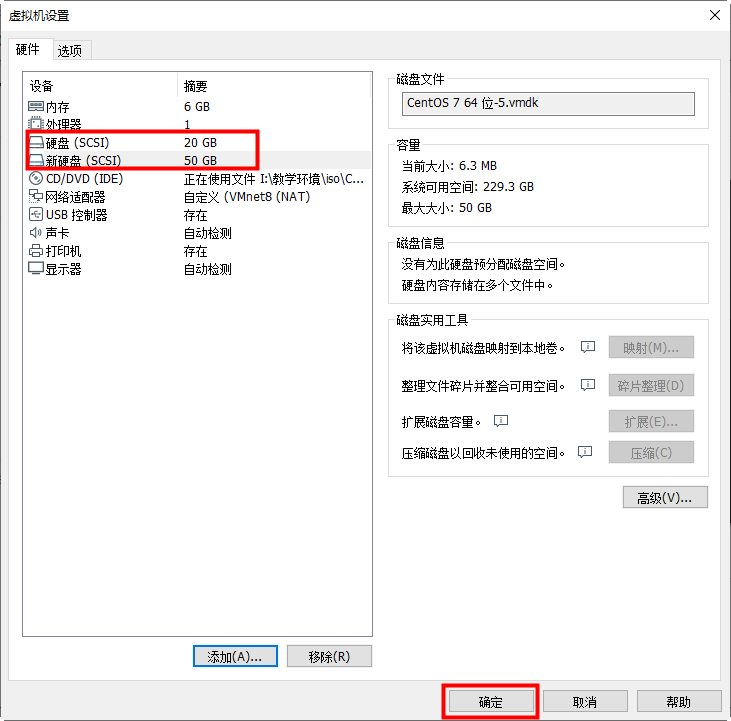






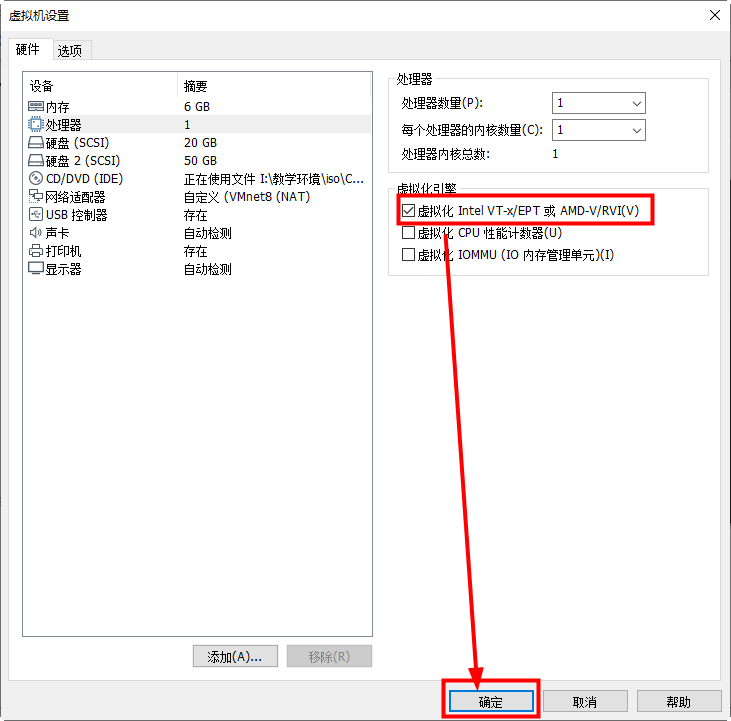






3.CPU开启虚拟化功能





四、虚拟机扩展根分区的空间

情况一：根分区为逻辑卷

[root@svr7 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 20G 0 disk

├─sda1 8:1 0 1G 0 part /boot

└─sda2 8:2 0 19G 0 part

**├─centos-root 253:0 0 17G 0 lvm /**

[root@svr7 ~]# vgs

[root@svr7 ~]# lvs

1.扩展卷组的空间

[root@svr7 ~]# vgextend centos /dev/sdb

[root@svr7 ~]# vgs

2.扩展逻辑卷的空间

[root@svr7 ~]# lvextend -L 40G /dev/centos/root

[root@svr7 ~]# lvs

4.扩展逻辑卷的文件系统

[root@svr7 ~]# df -h

[root@svr7 ~]# xfs\_growfs /dev/centos/root

[root@svr7 ~]# df -h | head -2

文件系统 容量 已用 可用 已用% 挂载点

/dev/mapper/centos-root 40G 3.5G 37G 9% /

[root@svr7 ~]# mkdir /iso

Mount /dev/mapper/centos-root /iso

情况二：根分区为基本分区

[root@svr7 ~]# fdisk /dev/sdb

命令(输入 m 获取帮助)：n

Partition type:

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

e extended

Select (default p): #回车

Using default response p

分区号 (1-4，默认 1)： #回车

起始 扇区 (2048-83886079，默认为 2048)： #回车

将使用默认值 2048

Last 扇区, +扇区 or +size{K,M,G} (2048-83886079，默认为 83886079)： #回车

将使用默认值 83886079

分区 1 已设置为 Linux 类型，大小设为 50 GiB

命令(输入 m 获取帮助)：w #保存并退出

The partition table has been altered!

Calling ioctl() to re-read partition table.

正在同步磁盘。

[root@svr7 ~]# lsblk

[root@svr7 ~]# mkfs.xfs /dev/sdb1 #格式化文件系统

[root@svr7 ~]# mkdir /iso

[root@svr7 ~]# mount /dev/sdb1 /iso

[root@svr7 ~]# vim /etc/fstab

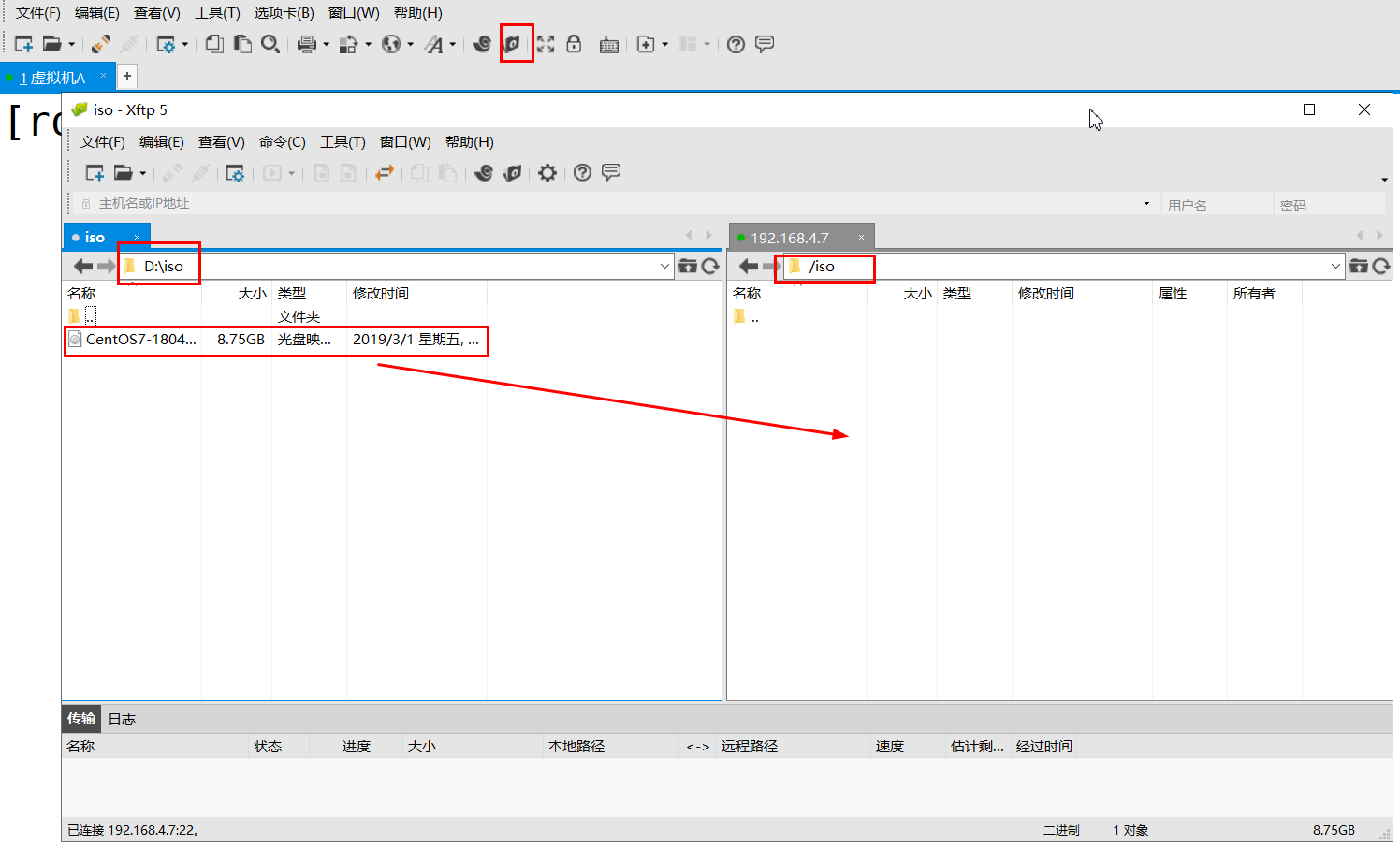
/dev/sdb1 /iso xfs defaults 0 0

[root@svr7 ~]# umount /iso #卸载设备

[root@svr7 ~]# df -h

[root@svr7 ~]# mount -a #检测/etc/fstab是否书写正确

[root@svr7 ~]# df -h



[root@svr7 iso]# du -sh /iso/

8.8G /iso/

五、安装Linux平台虚拟化软件

•virtualization 资源管理

–x个物理资源 --> y个逻辑资源

–实现程度：完全、部分、硬件辅助（CPU）

•虚拟化主要厂商及产品

|  |  |
| --- | --- |
| **系  列** | **PC/服务器版代表** |
| VMware | VMware Workstation、vSphere |
| Microsoft | VirtualPC、Hyper-V |
| RedHat | KVM、RHEV |
| Citrix | Xen |
| Oracle | Oracle VM VirtualBox |

•安装kvm组件 ，主要软件包

–qemu-kvm:为 kvm 提供底层仿真支持

–libvirt-daemon:libvirtd 守护进程，管理虚拟机

–libvirt-client:用户端软件，提供客户端管理命令

–libvirt-daemon-driver-qemu:libvirtd 连接 qemu 的驱动

–virt-manager:图形管理工具

[root@svr7 ~]# yum -y install qemu-kvm

[root@svr7 ~]# yum -y install libvirt-daemon

[root@svr7 ~]# yum -y install **libvirt-client**

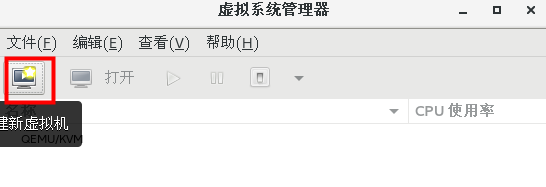
[root@svr7 ~]# yum -y install libvirt-daemon-driver-qemu

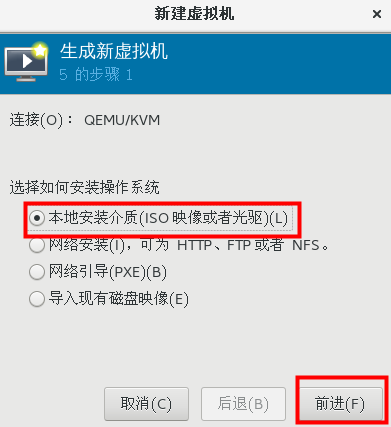
[root@svr7 ~]# yum -y install **virt-manager**

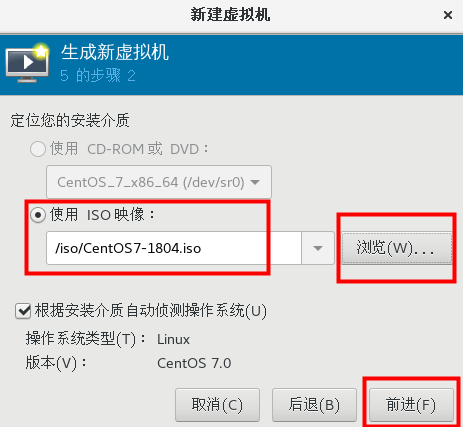
[root@svr7 ~]# virt-manager

六、利用kvm新建虚拟机，安装操作系统

[root@svr7 ~]# virt-manager



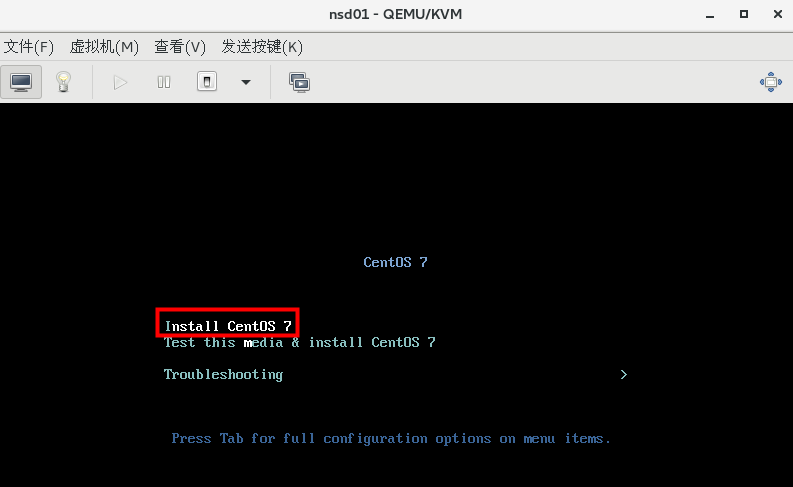




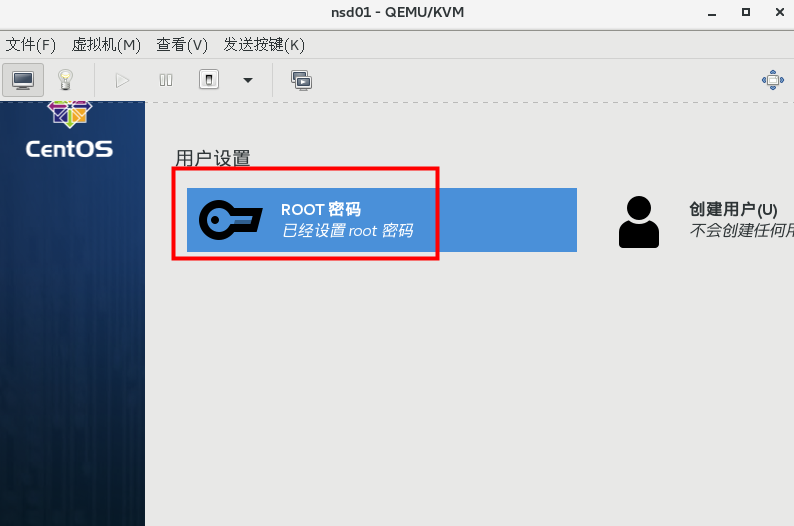












七、管理kvm虚拟机的命令

•提供管理各虚拟机的命令接口

–支持交互模式，查看/创建/停止/关闭 .. ..

–格式：virsh  控制指令  [虚拟机名称]  [参数]

•查看KVM节点（服务器）信息

–virsh  nodeinfo

•列出虚拟机

–virsh  list  [--all]

•查看指定虚拟机的信息

–virsh  dominfo  虚拟机名称

•运行|重启|关闭指定的虚拟机

–virsh  start或reboot或shutdown  虚拟机名称

•强制关闭指定的虚拟机

–virsh  destroy  虚拟机名称

•将指定的虚拟机设为开机自动运行

–virsh  autostart  [--disable]  虚拟机名称

八、kvm虚拟机的组成

•KVM虚拟机的组成

–xml配置文件：定义虚拟机的名称、UUID、CPU、内存、虚拟磁盘、网卡等各种参数设置

**默认存放路径：/etc/libvirt/qemu/**

–磁盘镜像文件：保存虚拟机的操作系统及文档数据，镜像路径取决于xml配置文件中的定义

**默认存放路径： /var/lib/libvirt/images/**

九、手动克隆kvm虚拟机

1.建立新的磁盘镜像文件

[root@svr7 ~]# virsh destroy nsd01 #关闭虚拟机nsd01

[root@svr7 ~]# cd /var/lib/libvirt/images/ #切换到磁盘文件路径

[root@svr7 images]# cp nsd01.qcow2 stu01.qcow2

[root@svr7 images]# ls

nsd01.qcow2 stu01.qcow2

[root@svr7 images]#

2.建立新的xml配置文件

[root@svr7 images]# cd /etc/libvirt/qemu/

[root@svr7 qemu]# cp nsd01.xml stu01.xml

[root@svr7 qemu]# ls

[root@svr7 qemu]# vim stu01.xml

1)虚拟机的名字：<name>**stu01**</name>

2)虚拟机的UUID：<uuid>6516.......</uuid> **#整行删除**

3)修改虚拟机的磁盘镜像文件：

<source file='/var/lib/libvirt/images/**stu01.qcow2**'/>

4)虚拟机的网卡MAC地址：<mac address='5......:cf'/> **#整行删除**

3.导入虚拟机xml配置文件信息

[root@svr7 qemu]# virsh list --all

[root@svr7 qemu]# virsh define /etc/libvirt/qemu/stu01.xml

定义域 stu01（从 /etc/libvirt/qemu/stu01.xml）

[root@svr7 qemu]# virsh list --all

[root@svr7 qemu]# virsh start stu01

域 stu01 已开始

[root@svr7 qemu]# virt-manager

常见错误：UUID错误

[root@svr7 /]# cd /var/lib/libvirt/images/

[root@svr7 images]# cp nsd01.qcow2 stu02.qcow2

[root@svr7 images]# cd /etc/libvirt/qemu/

[root@svr7 qemu]# cp nsd01.xml stu02.xml

[root@svr7 qemu]# vim stu02.xml

[root@svr7 qemu]# virsh define /etc/libvirt/qemu/stu02.xml

错误：从 /etc/libvirt/qemu/stu02.xml 定义域失败

错误：操作失败: 已使用 uuid nsd01 定义域名 '5516435a-4905-44a5-a4ef-3971a6a1ea6e'

常见错误：磁盘文件有误

[root@svr7 qemu]# cd /var/lib/libvirt/images/

[root@svr7 images]# cp nsd01.qcow2 stu03.qcow2

[root@svr7 images]# cd /etc/libvirt/qemu/

[root@svr7 qemu]# cp nsd01.xml stu03.xml

[root@svr7 qemu]# vim stu03.xml

[root@svr7 qemu]# virsh define /etc/libvirt/qemu/stu03.xml

定义域 stu03（从 /etc/libvirt/qemu/stu03.xml）

[root@svr7 qemu]# virsh start stu03

错误：开始域 stu03 失败

错误：Cannot access storage file '/var/lib/libvirt/images/nsd03.qcow2' (as uid:107, gid:107): 没有那个文件或目录

[root@svr7 qemu]# vim /etc/libvirt/qemu/stu03.xml

磁盘文件路径

<source file='/var/lib/libvirt/images/**stu03.qcow2**'/>

[root@svr7 qemu]# virsh define /etc/libvirt/qemu/stu03.xml

定义域 stu03（从 /etc/libvirt/qemu/stu03.xml）

十、手动克隆kvm虚拟机（virsh edit 虚拟机名字）

1.建立新的磁盘镜像文件

[root@svr7 ~]# virsh destroy nsd01 #关闭虚拟机nsd01

[root@svr7 ~]# cd /var/lib/libvirt/images/ #切换到磁盘文件路径

[root@svr7 images]# cp nsd01.qcow2 stu05.qcow2

[root@svr7 images]# ls

2.建立新的xml配置文件

[root@svr7 /]# virsh edit nsd01 #编辑并且导入

编辑了域 stu05 XML 配置。

1)虚拟机的名字：<name>**stu05**</name>

2)虚拟机的UUID：<uuid>6516.......</uuid> **#整行删除**

3)修改虚拟机的磁盘镜像文件：

<source file='/var/lib/libvirt/images/**stu05.qcow2**'/>

4)虚拟机的网卡MAC地址：<mac address='5......:cf'/> **#整行删除**

[root@svr7 /]# virsh list --all

十一、手动删除kvm虚拟机

[root@svr7 /]# virsh undefine stu05

域 stu05 已经被取消定义

[root@svr7 /]# virsh list --all

[root@svr7 /]# ls /var/lib/libvirt/images/

[root@svr7 /]# rm -rf /var/lib/libvirt/images/stu05.qcow2

•虚拟机的磁盘镜像文件格式

|  |  |  |
| --- | --- | --- |
| **特点\类型** | **RAW** | **QCOW2** |
| KVM默认 | 否 | 是 |
| I/O效率 | 高 | 较高 |
| 占用空间 | 大 | 小 |
| 压缩 | 不支持 | 支持 |
| 后端盘复用 | 不支持 | 支持 |
| 快照 | 不支持 | 支持 |

十二、**COW技术原理**

•Copy On Write，写时复制

–前端盘直接映射原始盘（后端盘）的数据内容

–原始盘（后端盘）内容不变，并且不能修改原始盘内容，否则所有前端盘无法使用

–对前端盘的修改不回写到原始盘（后端盘）

1.快速产生磁盘镜像文件

•qemu-img 通过 -b 选项复用指定后端盘

–qemu-img  create  -f  qcow2  -b  后端盘  前端盘

]# cd /var/lib/libvirt/images/

]# qemu-img create -f qcow2 -b nsd01.qcow2 dc01.qcow2

Formatting 'dc01.qcow2', fmt=qcow2 size=9663676416 backing\_file='nsd01.qcow2' encryption=off cluster\_size=65536 lazy\_refcounts=off

]# du -sh dc01.qcow2

]# qemu-img info dc01.qcow2 #查看前端盘信息

virtual size: 9.0G (9663676416 bytes) #虚拟大小

disk size: 196K #真正占用磁盘空间的大小

backing file: nsd01.qcow2 #指明后端盘

2.建立新的xml配置文件

[root@svr7 images]# cd /etc/libvirt/qemu/

[root@svr7 qemu]# cp nsd01.xml dc01.xml

[root@svr7 qemu]# vim dc01.xml

1)虚拟机的名字：<name>**dc01**</name>

2)虚拟机的UUID：<uuid>6516.......</uuid> **#整行删除**

3)修改虚拟机的磁盘镜像文件：

<source file='/var/lib/libvirt/images/**dc01.qcow2**'/>

4)虚拟机的网卡MAC地址：<mac address='5......:cf'/> **#整行删除**

3.导入虚拟机xml配置文件信息

[root@svr7 qemu]# virsh list --all

[root@svr7 qemu]# virsh define /etc/libvirt/qemu/dc01.xml

[root@svr7 qemu]# virsh start dc01

十三、离线访问虚拟机

•使用 guestmount 工具

–支持离线挂载 raw、qcow2 格式虚拟机磁盘

–可以在虚拟机关机的情况下，直接修改磁盘中的文档

–方便对虚拟机定制、修复、脚本维护

   !!! 需要注意 SELinux 机制的影响

•基本用法

guestmount  -a  虚拟机磁盘路径  -i  /挂载点

1.查询仓库中那个软件包产生该程序

[root@svr7 /]# yum provides \*/guestmount

[root@svr7 /]# yum -y install libguestfs-tools-c

]# guestmount -a /var/lib/libvirt/images/nsd01.qcow2 -i /mnt

fuse: mountpoint is not empty

fuse: if you are sure this is safe, **use the 'nonempty' mount option**

libguestfs: error: fuse\_mount: /mnt: 成功

]# guestmount -o nonempty -a /var/lib/libvirt/images/nsd01.qcow2 -i /mnt

]# ls /mnt/

]# umount /mnt/

]# ls /mnt/

[root@svr7 /]# vim /mnt/etc/selinux/config

[root@svr7 /]# vim /mnt/etc/shadow

root:::0:99999:7:::