牛犇

学习资料链接地址：

正式课程笔记：码云

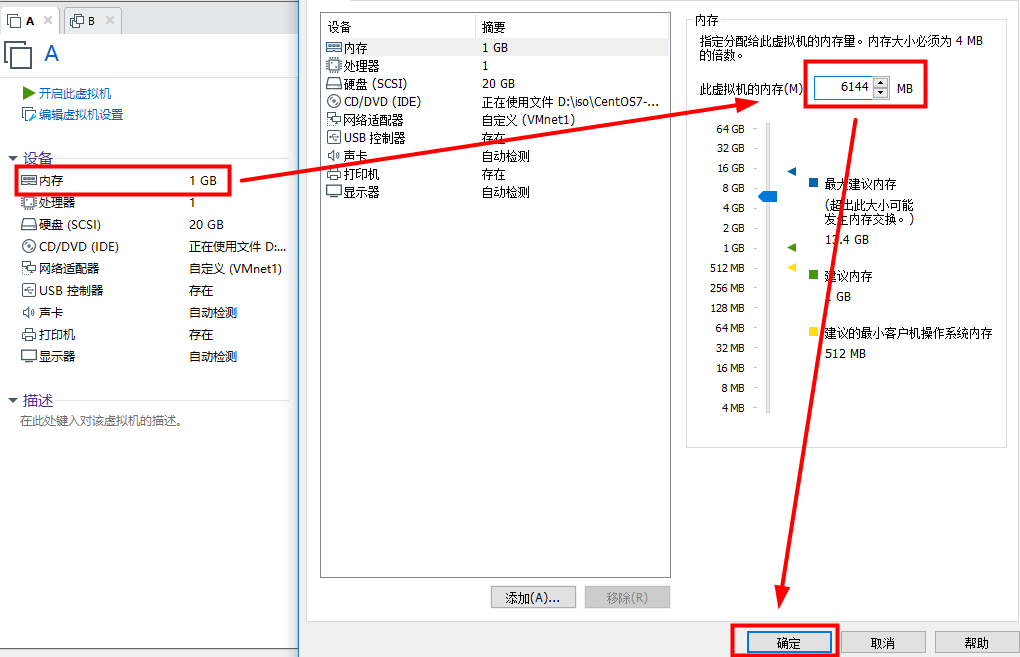
https://gitee.com/niubenlinux/nsd2011

一、环境准备

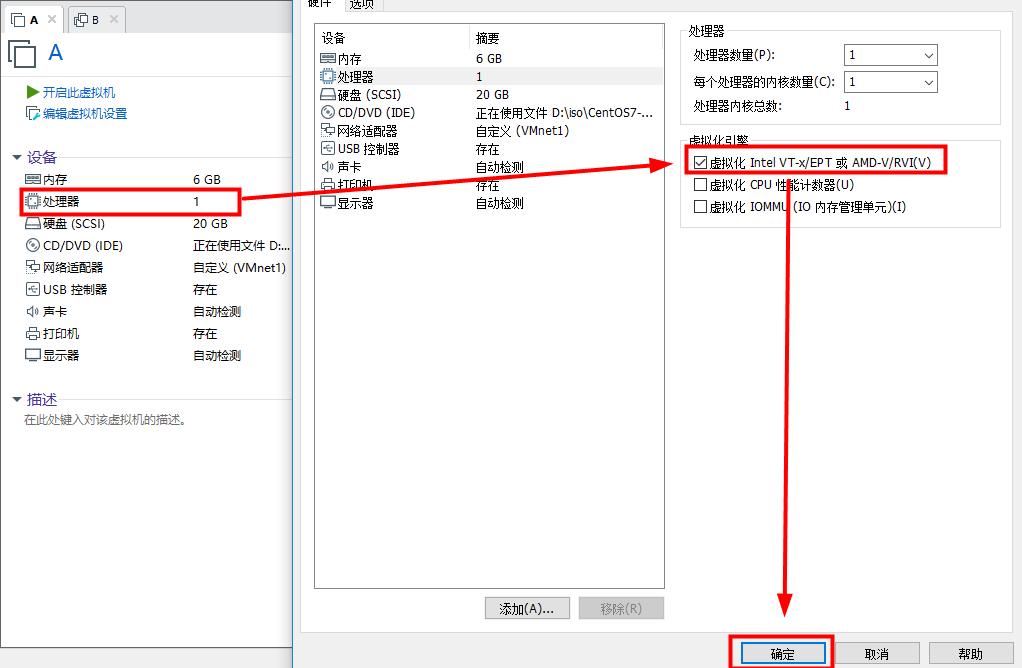
还原快照“网络配置完成”虚拟机A

二、虚拟机A预先设置

* **内存设置6G**



* **CPU设置开启虚拟化功能**



* **添加一块新50G硬盘**



三、虚拟机A扩展根设备的空间

]# lsblk #查看识别的硬盘

]# vgs #查看卷组信息

]# lvs #查看逻辑卷信息

]# vgextend centos /dev/sdb #扩展卷组的空间

Physical volume "/dev/sdb" successfully created.

Volume group "centos" successfully extended

]# vgs

]# lvextend -L 40G /dev/centos/root #扩展空间

]# df -h | grep root

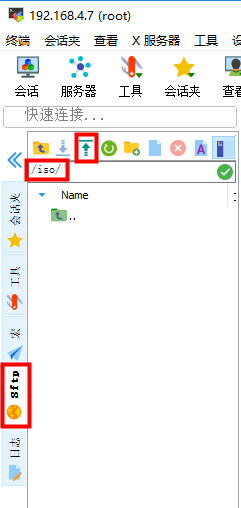
]# blkid /dev/centos/root #查看文件系统类型

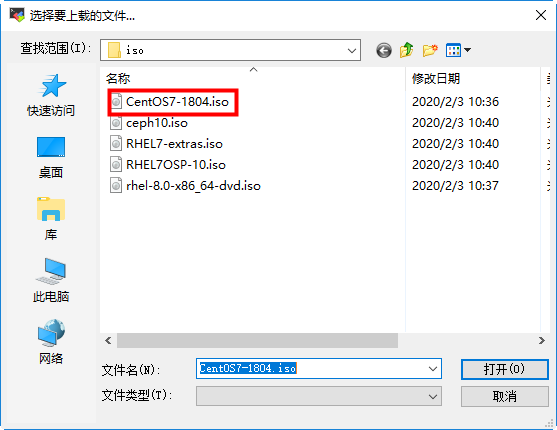
]# xfs\_growfs /dev/centos/root #刷新文件系统

]# df -h | grep root

四、真机传递数据到虚拟机A

[root@svr7 ~]# mkdir /iso





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

8.8G /iso

五、虚拟机A部署虚拟化

* 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 |

* 虚拟机A安装部署KVM
  + 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

]# yum -y install qemu-kvm

]# yum -y install libvirt-daemon

]# yum -y install **libvirt-client**

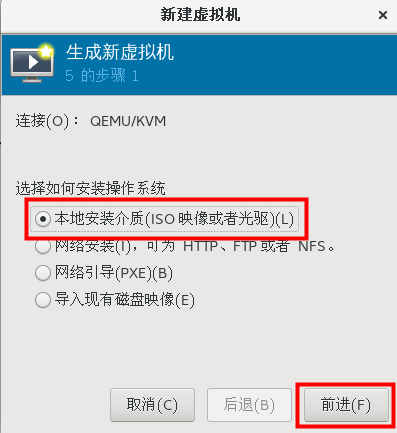
]# yum -y install libvirt-daemon-driver-qemu

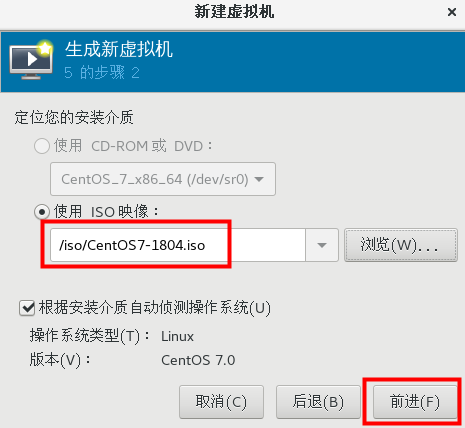
]# yum -y install **virt-manager**

六、在虚拟机A中新建虚拟机

[root@svr7 ~]# virt-manager













七、利用virsh命令管理虚拟机

* 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、Linux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

**/etc/libvirt/qemu/**：xml配置文件默认存放路径

* + **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

**/var/lib/libvirt/images**：磁盘镜像文件默认存放

]# ls /var/lib/libvirt/images/nsd2010.qcow2

九、命令行手动克隆虚拟机

* **建立新的磁盘镜像文件**

]# virsh destroy nsd2011 #强制关闭虚拟机

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

]# ls

]# cp nsd2011.qcow2 stu01.qcow2

]# ls

* **建立新的xml配置文件**

]# cd /etc/libvirt/qemu/

]# ls

]# cp nsd2011.xml stu01.xml

]# ls

]# vim stu01.xml

虚拟机名字： <name>**stu01**</name>

虚拟机的UUID：<uuid>76…d8</uuid> **#整行删除**

磁盘镜像文件: <source file='……/**stu01**.qcow2'/>

网卡的MAC地址: <mac …..52….ed'/> **#整行删除**

* **导入新的xml配置文件**

]# virsh define /etc/libvirt/qemu/stu01.xml

]# virsh list --all

]# virsh start stu01 #开启stu01虚拟机

]# virsh list --all

常见问题：

1.UUID没有修改，导入不成功

]# virsh define /etc/libvirt/qemu/stu01.xml

错误：从 /etc/libvirt/qemu/stu01.xml 定义域失败

错误：操作失败: 已使用 uuid nsd2010 定义域名 '768f812c-0c5f-4e6b-9c07-accf28407cd8'

2.磁盘镜像文件修改错误，导入成功，无法启动虚拟机

]# virsh start stu01

错误：开始域 stu01 失败

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

3.网卡mac修改错误，导入成功启动虚拟机成功，网络通信失败

再次练习：命令行手动克隆虚拟机

* **建立新的磁盘镜像文件**

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

]# cp nsd2010.qcow2 stu02.qcow2

]# ls

* **建立新的xml配置文件**

]# cd /etc/libvirt/qemu/

]# cp nsd2010.xml stu02.xml

]# vim stu02.xml

虚拟机名字： <name>**stu02**</name>

虚拟机的UUID：<uuid>76…d8</uuid> **#整行删除**

磁盘镜像文件: <source file='……/**stu02**.qcow2'/>

网卡的MAC地址: <mac …..52….ed'/> **#整行删除**

* **导入新的xml配置文件**

]# virsh define /etc/libvirt/qemu/stu02.xml

]# virsh list --all

]# virsh start stu02 #开启stu02虚拟机

]# virsh list --all

* **命令行删除虚拟机**

]# virsh destroy stu02 #强制关闭虚拟机stu02

]# virsh undefine stu02 #取消stu02虚拟机定义

域 stu02 已经被取消定义

]# virsh list --all

]# ls /var/lib/libvirt/images/

]# rm -rf /var/lib/libvirt/images/stu02.qcow2

]# ls /var/lib/libvirt/images/

再次练习：命令行手动克隆虚拟机

* **建立新的磁盘镜像文件**

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

]# cp nsd2010.qcow2 dc01.qcow2

]# ls

* **建立新的xml配置文件（复制、修改、导入）**

]# **virsh edit nsd2010**

虚拟机名字： <name>**dc01**</name>

虚拟机的UUID：<uuid>76…d8</uuid> **#整行删除**

磁盘镜像文件: <source file='……/**dc01**.qcow2'/>

网卡的MAC地址: <mac …..52….ed'/> **#整行删除**

]# virsh list --all

十、磁盘镜像文件的快速产生

* 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 |

* **C**opy **O**n **W**rite，写时复制
  + 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)
  + 前端盘至少与后端盘一样大小，可以前端盘比后端盘大，不可以前端盘比后端盘小
* 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

命令行手动克隆虚拟机

* **建立新的磁盘镜像文件**

]# virsh list --all

]# virsh destroy nsd2010

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

]# qemu-img create -f qcow2 -b nsd2010.qcow2 haha.qcow2

]# qemu-img info haha.qcow2 #查看前端盘信息

virtual size: 9.0G #虚拟的大小

disk size: 196K #真正占用真机的磁盘空间

backing file: nsd2010.qcow2 #后端盘

* **建立新的xml配置文件（复制、修改、导入）**

]# **virsh edit nsd2010**

虚拟机名字： <name>**haha**</name>

虚拟机的UUID：<uuid>76…d8</uuid> **#整行删除**

磁盘镜像文件: <source file='……/**haha**.qcow2'/>

网卡的MAC地址: <mac …..52….ed'/> **#整行删除**

]# virsh list --all

十一、离开访问虚拟机磁盘镜像文件

* 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

* **修改SELinux运行模式**

]# getenforce #查看运行模式

]# setenforce 0 #修改当前运行模式

]# getenforce

]# vim /etc/selinux/config #永久修改

SELINUX=permissive

[root@svr7 ~]# guestmount

bash: guestmount: 未找到命令...

]# yum provides \*/guestmount #查询仓库中由谁产生该程序

]# yum -y install libguestfs-tools-c

]# rpm -q libguestfs-tools-c

]# guestmount -a /var/lib/libvirt/images/nsd2010.qcow2 -i /mnt

常见问题：

]# guestmount -a /var/lib/libvirt/images/nsd2010.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/nsd2010.qcow2 -i /mnt

十二、服务的管理

用户---》systemd---》服务

•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（与手动启动有冲突）

**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

]# yum -y install httpd

]# > /etc/resolv.conf #清除文件内容

]# killall httpd #杀死手动启动的httpd

]# systemctl restart httpd #重启httpd服务

]# systemctl status httpd #查看服务httpd状态

]# 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.

]# systemctl is-enabled httpd #查看httpd是否是开机自启动

]# systemctl disable httpd #关闭httpd开机自启动

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

]# systemctl is-enabled httpd #查看httpd是否是开机自启动

**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

]# systemctl isolate multi-user.target #相当于原来的init 3

当前直接切换到图形模式

]# 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

十二、VDO 了解内容

•Virtual Data Optimizer（虚拟数据优化器）

–一个内核模块，目的是通过重删减少磁盘的空间占用，以及减少复制带宽

–VDO是基于块设备层之上的，也就是在原设备基础上映射出mapper虚拟设备，然后直接使用即可

•重复数据删除

–输入的数据会判断是不是冗余数据

–判断为重复数据的部分不会被写入，然后对源数据进行更新，直接指向原始已经存储的数据块即可

•压缩

–对每个单独的数据块进行处理

[root@svr7 ~]# yum  -y  install  vdo  kmod-kvdo    #所需软件包

•制作VDO卷

•vdo基本操作：参考man vdo 全文查找/example

–vdo  create  --name=VDO卷名称  --device=设备路径 --vdoLogicalSize=逻辑大小

–vdo  list

–vdo  status  -n  VDO卷名称

–vdo  remove  -n  VDO卷名称

–vdostatus  [--human-readable] [/dev/mapper/VDO卷名称]

•VDO卷的格式化加速（跳过去重分析）：

–mkfs.xfs  –K   /dev/mapper/VDO卷名称

–mkfs.ext4  -E  nodiscard  /dev/mapper/VDO卷名称

前提制作VDO需要2G以上的内存

[root@nb ~]# vdo create --name=vdo0 --device=/dev/sdc --vdoLogicalSize=200G

[root@nb ~]# mkfs.xfs -K /dev/mapper/vdo0

[root@nb ~]# mkdir /nsd01

[root@nb ~]# mount /dev/mapper/vdo0 /nsd01

[root@nb ~]# df -h

[root@nb ~]# vdostats --hum /dev/mapper/vdo0 #查看vdo设备详细信息

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

/dev/mapper/vdo0  /nsd01  xfs  defaults,\_netdev 0 0