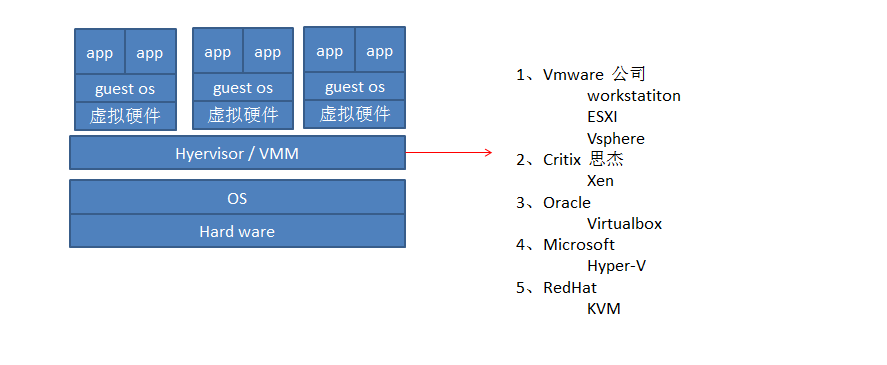
## 虚拟化技术 Virtualization

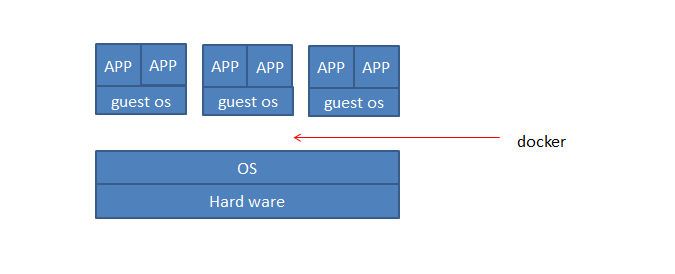
**类型：**

一、根据虚拟化技术实现不同

1、平台虚拟化 platform **virtualization**



2、操作系统虚拟化 os virtualization



二、根据hypervisor的安装方式不同

1、原生虚拟化

直接安装在物理硬件上

Vmware ESXI, Vsphere

Critix Xen

2、寄居虚拟化

依赖于物理机的操作系统

kvm, vmware workstation, Microsoft Hyper-v, Oralce virtualBox

三、根据hypervisor实现虚拟化管理方式不同

软件虚拟化 QEMU

半虚拟化 para-virtulization XEN

需要修改内核

RedHat在kvm虚拟化中研发驱动virtio半虚拟化驱动

Linux类型： 默认是安装的

Windows类型：手动安装virtio半虚拟化驱动

硬盘：IDE

网卡：rtl8139

硬件辅助虚拟化

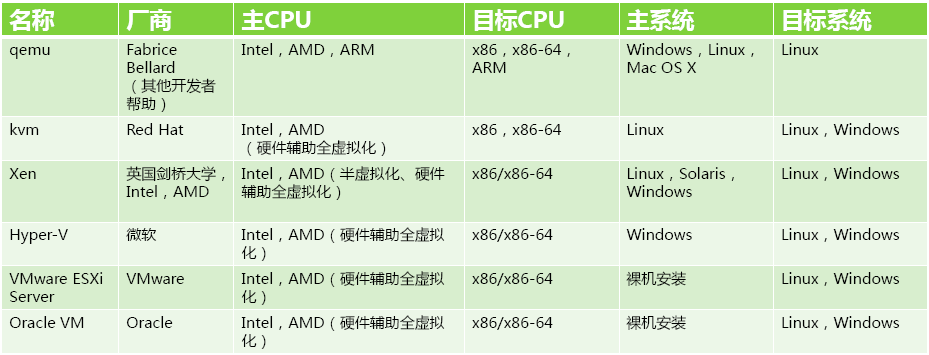
需要物理机CPU(指令集)的支持

kvm, xen, esxi, vsphere

如何确认CPU是否支持硬件辅助虚拟化？

Intel： VT-x

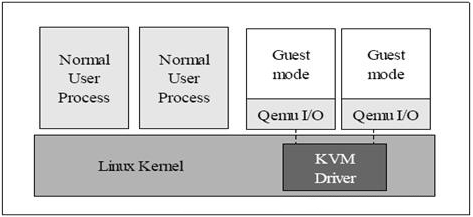
AMD： amd-v



**kvm ------ Kernel-Based Virutal Manager 基于内核虚拟化管理器**

**kvm --- 内核模块**

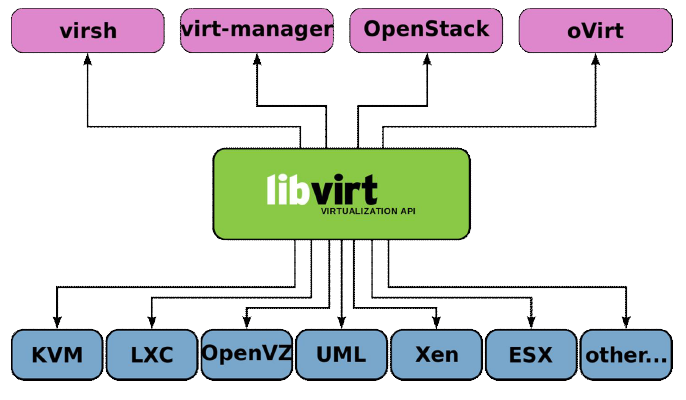
**qemu-kvm 实现对虚拟硬件的模拟功能**



kvm管理工具：

virt-manager, virt-viewer, virsh, virt-install

依赖C函数库 libvirtd 服务



## kvm虚拟化安装

1、确保CPU启用虚拟化功能

[root@localhost ~]# lscpu

虚拟化： VT-x

方法一：软件包

# yum install qemu-kvm qemu-img libvirt virt-install libvirt-python virt-manager python-virtinst libvirt-client

qemu-kvm：提供kvm虚拟化资源管理

libvirt： C函数库，确保kvm管理工具可正常使用

qemu-img：用于管理虚拟磁盘

virt-install： 用于创建kvm虚拟机的命令行工具

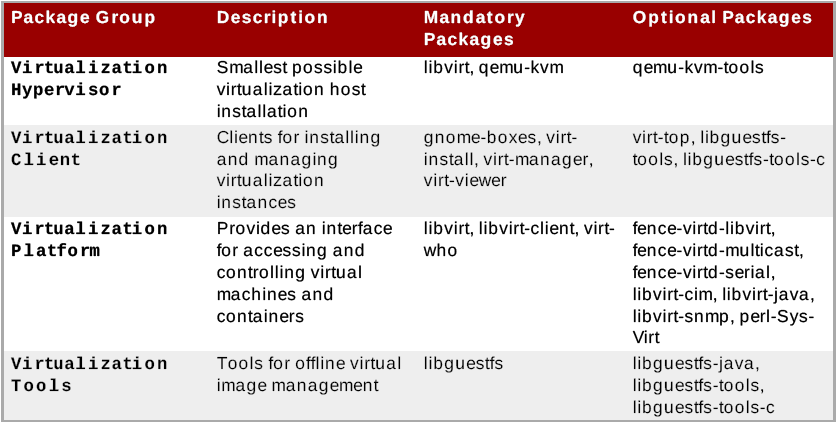
virt-manager：提供图形化工具

libvirt-python：python连接libvirt的驱动

python-virtinst：python连接libvirt的模块

libvirt-client：提供libvirt的客户端工具

方法二：安装软件包组



3、确保kvm安装成功

1) 查看kvm的内核模块

[root@localhost ~]# lsmod | grep kvm

kvm\_intel 162153 0

kvm 525259 1 kvm\_intel

[root@localhost ~]#

2) libvirtd服务

centos 7：libvirtd服务启动

[root@localhost ~]# systemctl is-active libvirtd

active

[root@localhost ~]# systemctl is-enabled libvirtd

enabled

centos 6：libvirtd需要手动启动

# service libvirtd start

# chkconfig libvirtd on

**kvm虚拟化安装后系统的变化**

1、默认default网络

工作在NAT模式

自带dhcp服务 192.168.122.0/24

virbr0 ---- 192.168.122.1 作为所有default网络中的kvm虚拟机的网关

2、路由转发功能会自动打开

[root@localhost ~]# cat /proc/sys/net/ipv4/ip\_forward

1

3、在iptables防火墙的NAT表中会自动SNAT转换规则

[root@localhost ~]# iptables -t nat -nL

Chain POSTROUTING (policy ACCEPT)

target prot opt source destination

RETURN all -- 192.168.122.0/24 224.0.0.0/24

RETURN all -- 192.168.122.0/24 255.255.255.255

MASQUERADE tcp -- 192.168.122.0/24 !192.168.122.0/24 masq ports: 1024-65535

MASQUERADE udp -- 192.168.122.0/24 !192.168.122.0/24 masq ports: 1024-65535

MASQUERADE all -- 192.168.122.0/24 !192.168.122.0/24

## kvm虚拟机安装

1、virt-manager图形化管理工具

2、virt-install 命令行工具

示例01：使用本地镜像安装centos 6.6

[root@localhost ~]# virt-install \

> --name=pc-centos6.6 \

> --graphics vnc,listen=0.0.0.0,port=5920,keymap=en\_us \

> --memory 512,maxmemory=1024 \

> --vcpus 1,maxvcpus=2 \

> --disk path=/var/lib/libvirt/images/pc\_centos6.6.img,size=8,format=qcow2 \

> --bridge=virbr0 \

> --cdrom=/var/ftp/linux/upload/iso/centos6u6.iso

示例02：安装windows server 2008，将网卡、硬盘的驱动指定为virtio

[root@localhost ~]# virt-install \

> --name=win2k8 \

> --memory 1024,maxmemory=2048 \

> --vcpus 1,maxvcpus=2 \

> --cdrom=/var/ftp/linux/upload/iso/cn\_windows\_server\_2008\_r2.iso \

> --disk path=/var/lib/libvirt/images/win2k8.img,size=20,format=qcow2,bus=virtio \

> --network bridge=virbr0,model=virtio \

> --disk path=/var/ftp/linux/upload/iso/virtio-win\_amd64.vfd,device=floppy \

> --vnc \

> --vnclisten=0.0.0.0 \

> --vncport=5940

## kvm虚拟机的连接方式

1、ssh

2、virt-viewer命令

[root@localhost ~]# virsh list

[root@localhost ~]# virsh list --all

[root@localhost ~]# virt-viewer pc\_test &

3、vnc连接

[root@localhost ~]# vncviewer 127.0.0.1:5920

4、console的连接方式

# virsh console <虚拟机名称>

centos 6：

编辑/boot/grub/grub.conf文件，在kernel行的末尾添加console=ttyS0，重启系统

[root@localhost ~]# virsh console pc-centos6.6

退出console连接

ctrl + ]

centos 7：

1. 编辑文件/etc/sysconfig/grub

在GRUB\_CMD\_LINELINUX行末尾添加console=ttyS0

2. 重新生成grub配置文件

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

## kvm虚拟机的相关文件

一台kvm虚拟机 = 配置文件 + 磁盘镜像文件

配置文件：/etc/libvirt/qemu

[root@localhost qemu]# ls /etc/libvirt/qemu

编辑xml的配置文件

# virsh edit <kvm虚拟机名称>

磁盘镜像文件：

[root@localhost ~]# ls /var/lib/libvirt/images/

kvm相关的其他文件：

1、/etc/libvirt/qemu/autostart目录

存放开机自启动的kvm配置文件的软链接

[root@localhost ~]# virsh autostart centos7-1

[root@localhost ~]# virsh autostart win2k8

[root@localhost ~]# ls -l /etc/libvirt/qemu/autostart/

总用量 0

lrwxrwxrwx 1 root root 31 9月 19 11:51 centos7-1.xml -> /etc/libvirt/qemu/centos7-1.xml

lrwxrwxrwx 1 root root 28 9月 19 11:52 win2k8.xml -> /etc/libvirt/qemu/win2k8.xml

2、/etc/libvirt/qemu/networks

存放网络的配置文件

3、/etc/libvirt/qemu/storage

存放存储池相关的配置文件

**删除虚拟机**

[root@localhost ~]# virsh destroy centos6-1

[root@localhost ~]# virsh undefine centos6-1

## kvm cpu热添加

前提：设置最大CPU个数

仅针对 centos 7以上的系统

centos 7：

1、查看当前kvm主机的信息

[root@localhost ~]# virsh dominfo centos7-1

2、调整CPU

[root@localhost ~]# virsh setvcpus centos7-1 2 --live

window：

[root@localhost ~]# virsh setvcpus win2k8 2 --live

## kvm 内存气球技术

在线调整内存大小

KVM内存气球技术可以对虚拟机使用的内存按需调节，从而提高内存的利用率。

前提：最大内存量

确认虚拟机支持内存气球的驱动

[root@localhost ~]# virsh dumpxml win2k8 | grep -C 2 memballoon

centos 6：

1、查看当前内存气球大小

[root@localhost ~]# virsh qemu-monitor-command pc-centos6.6 --hmp info balloon

balloon: actual=512

2、调整内存气球

[root@localhost ~]# virsh qemu-monitor-command pc-centos6.6 --hmp balloon 900

[root@localhost ~]# virsh qemu-monitor-command pc-centos6.6 --hmp info balloon

balloon: actual=900

windows

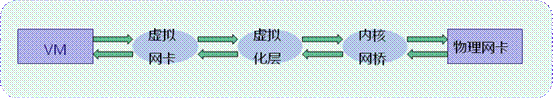
1) 安装virtio内存气球驱动

2) 调整内存

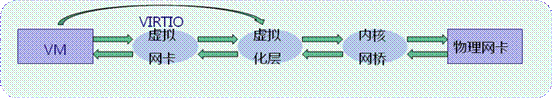
## kvm网络管理

网卡驱动：

全虚拟化驱动



半虚拟化驱动



1、通过命令行添加网卡

[root@localhost ~]# virsh attach-interface centos7.2-1 --type network --source default --model virtio --persistent

--live：立即生效，临时

--persistent：立即生效、永久保存

--config：配置会写入硬盘, 需要重启kvm主机才会生效

2、通过命令删除网卡

[root@localhost ~]# virsh detach-interface centos7-1 --type network --mac 52:54:00:37:a3:64 --persistent

**kvm网络模式**

1、NAT模式

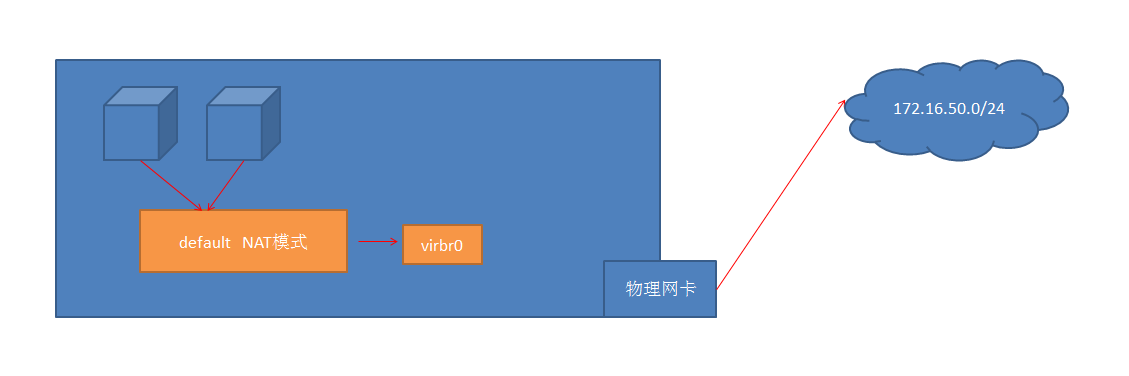
2、桥接模式

3、隔离模式

4、路由模式

**1、NAT模式**

所有连接到NAT模式的网络中的kvm主机，会通过物理机的SNAT转换规则，实现对外部网络的访问



发布服务

[root@localhost ~]# iptables -t nat -A PREROUTING -d 172.16.8.252 -p tcp --dport 80 -j DNAT --to-destination 10.1.1.170:80

作业：

1) httpd, nginx, tomcat, mysql

1、通过命令行工具创建网络

1) 准备一个网络的配置文件

[root@localhost qemu]# cd /etc/libvirt/qemu/networks/

[root@localhost networks]# cp default.xml nat\_test.xml

[root@localhost networks]# cat nat\_test.xml

<network>

<name>nat\_test</name>

<uuid>b6c777aa-a9ba-4e70-a46a-821b16baac80</uuid>

<forward mode='nat'/>

<bridge name='virbr1' stp='on' delay='0'/>

<ip address='10.1.1.1' netmask='255.255.255.0'>

<dhcp>

<range start='10.1.1.2' end='10.1.1.254'/>

</dhcp>

</ip>

</network>

2) 定义网络

[root@localhost networks]# virsh net-define nat\_test.xml

[root@localhost networks]# virsh net-start nat\_test

[root@localhost networks]# virsh net-autostart nat\_

3) 查看

[root@localhost networks]# virsh net-list

名称 状态 自动开始 持久

----------------------------------------------------------

default 活动 是 是

nat\_test 活动 是 是

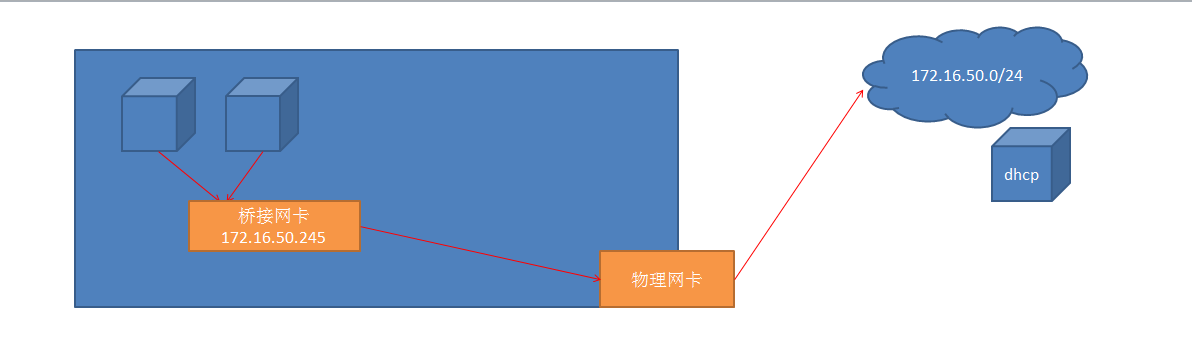
删除网络

[root@localhost ~]# virsh net-destroy nat\_test

[root@localhost ~]# virsh net-undefine nat\_test

**2、桥接模式bridge**

适用于跨物理机的kvm主机通信



创建桥接网卡

注意：

1、NetworkManager服务关闭 [nmcli]

2、物理网卡手工配置IP参数

方法一：virt-manager

方法二：virsh

[root@localhost ~]# virsh iface-bridge eth0 br0

使用附加设备 br0 生成桥接 eth0 失败

已启动桥接接口 br0

[root@localhost ~]# ip addr show

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast master br0 state UP qlen 1000

link/ether 1c:87:2c:46:c4:1b brd ff:ff:ff:ff:ff:ff

53: br0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP qlen 1000

link/ether 1c:87:2c:46:c4:1b brd ff:ff:ff:ff:ff:ff

inet 172.16.50.127/24 brd 172.16.50.255 scope global br0

valid\_lft forever preferred\_lft forever

inet6 fe80::1e87:2cff:fe46:c41b/64 scope link

valid\_lft forever preferred\_lft forever

**删除桥接网卡**

1、禁用桥接网卡

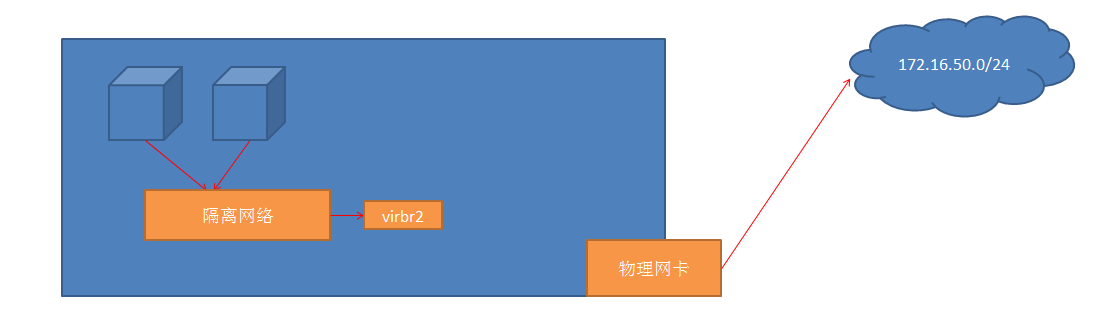
# ifconfig br0 down

2、删除桥接网卡

# brctl delbr br0

3、删除桥接网卡配置文件，重新配置物理网卡配置文件

**3、隔离模式 Isolated**



## kvm存储管理

一、磁盘管理

1、通过命令行工具热添加硬盘

[root@localhost ~]# virsh domblklist centos6-1

目标 源

------------------------------------------------

vda /var/lib/libvirt/images/centos6-1.qcow2

[root@localhost ~]# qemu-img create -f qcow2 /var/lib/libvirt/images/disk01.img 2G

[root@localhost ~]# virsh attach-disk centos6-1 --source /var/lib/libvirt/images/disk01.img --target vdb --cache writeback --subdriver qcow2 --persistent

--cache：磁盘的缓存模式

writethrough 通写

硬盘缓存只要收到数据会立即同步给硬盘

writeback 回写

缓存发现数据需要被置换，才会同步给硬盘

2、删除硬盘

[root@localhost ~]# virsh detach-disk centos6-1 vdc --persistent

二、存储池 storage pool

存储kvm主机磁盘镜像的位置

类型：

基于本地目录

基于共享存储

1、查看存储池

[root@localhost ~]# virsh pool-list

[root@localhost ~]# virsh pool-info default

示例：创建基于NFS共享存储的存储池

1、配置nfs服务器共享目录

[root@nfs\_server ~]# mkdir /data

[root@nfs\_server ~]# chmod o+w /data/

[root@nfs\_server ~]# cat /etc/exports

/data 172.16.8.252(rw,sync,no\_root\_squash)

[root@nfs\_server ~]# exportfs -rav

exporting 172.16.8.252:/data

[root@nfs\_server ~]# systemctl restart nfs-server

2、测试挂载nfs服务器

[root@localhost ~]# mount -t nfs 172.16.50.127:/data /mnt/

3、创建存储池

virt-manager

4、使用nfs存储池

[root@localhost ~]# virt-install --name=pc\_centos\_6 --memory=1024 --vcpus=1 --graphics vnc,listen=0.0.0.0,port=5990,keymap=en\_us --disk path=/test/pc\_centos\_6.img,format=qcow2,size=8 --bridge=virbr0 --location=ftp://172.16.8.100/rhel6.4 --extra-args="ks=ftp://172.16.8.100/rhel6.4.ks"

通过命令行方式管理存储池

1、查看存储池

[root@localhost ~]# virsh pool-list

2、创建存储池

[root@localhost storage]# pwd

/etc/libvirt/storage

[root@localhost storage]# cat test.xml

<pool type='dir'>

<name>test</name>

<uuid>2601ee89-bd11-497e-869d-8af8cdf2a702</uuid>

<target>

<path>/data</path>

</target>

</pool>

[root@localhost storage]# virsh pool-define test.xml

[root@localhost storage]# virsh pool-start test

[root@localhost storage]# virsh pool-autostart test

3、删除存储池

[root@localhost storage]# virsh pool-destroy test

[root@localhost storage]# virsh pool-undefine test

## kvm虚拟机迁移

kvm虚拟机迁移migrate

离线迁移 offline-migrate

1) kvm关机

2) 拷贝xml配置文件，磁盘镜像文件

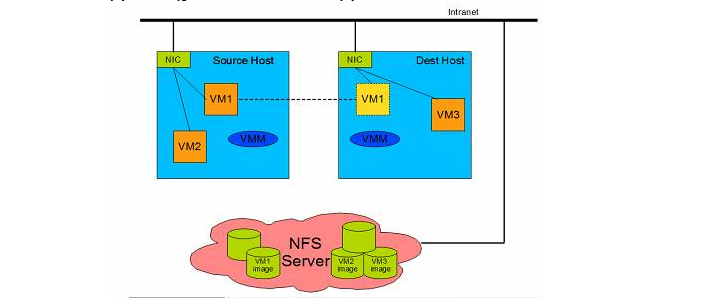
3) virsh define <xml文件>

在线迁移 online-migrate

虚拟机磁盘映像文件存放在共享存储系统(NFS, iSCSI)

建议两台物理机使用有同样的目录结构挂载NFS存储

主机名称解析



示例：在线迁移

1、添加主机名解析

[root@node02 ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

172.16.8.252 node01

172.16.50.138 node02

2、在另一台服务器使用相同的目录结构挂载nfs

[root@node02 ~]# mount -t nfs 172.16.50.127:/data /test/

[root@node02 ~]# ls /test/

pc\_centos\_6.img

3、迁移主机

[root@node01 ~]# virsh migrate pc\_centos\_6 --live qemu+ssh://172.16.50.138/system --unsafe

4、验证结果

## qemu-img工具的使用

管理磁盘镜像文件

1、创建磁盘镜像文件

格式：

raw

一次性分配所有磁盘空间

qcow2

稀疏文件格式

快照snapshot

后端镜像、差量镜像

加密

压缩

1) 创建raw的磁盘文件

[root@localhost ~]# mkdir /data

[root@localhost ~]# qemu-img create /data/disk01.img 2G

Formatting '/data/disk01.img', fmt=raw size=2147483648

[root@localhost ~]# qemu-img info /data/disk01.img

image: /data/disk01.img

file format: raw

virtual size: 2.0G (2147483648 bytes)

disk size: 0

2) 创建qcow2的磁盘文件

[root@localhost ~]# qemu-img create -f qcow2 /data/disk02.img 2G

2、快照管理

镜像快照管理

1) 关机状态

2) qcow2格式

# qemu-img snapshot -l <磁盘映像文件名称> //查看快照

# qemu-img snapshot -c <快照名称> <磁盘映像文件名称> //创建快照

# qemu-img snapshot -a <快照名称> <磁盘映像文件名称> //恢复快照

# qemu-img snapshot -d <快照名称> <磁盘映像文件名称> //删除快照

3、后端镜像

虚拟机模板，快速创建虚拟机

[root@localhost data]# qemu-img create -f qcow2 -b disk02.img disk03.img

示例：快速创建虚拟机

1、创建差量镜像

[root@localhost ~]# qemu-img create -f qcow2 -b /var/lib/libvirt/images/centos6-3.qcow2 /data/test.img

[root@localhost ~]# qemu-img info /data/test.img

image: /data/test.img

file format: qcow2

virtual size: 8.0G (8589934592 bytes)

disk size: 196K

cluster\_size: 65536

backing file: /var/lib/libvirt/images/centos6-3.qcow2

Format specific information:

compat: 1.1

lazy refcounts: false

2、准备kvm的配置文件

[root@localhost qemu]# cp centos6-3.xml test.xml

1、虚拟机名称

2、uuid

3、disk文件位置

4、mac地址

5、vnc端口

3、创建虚拟机

[root@localhost qemu]# virsh define test.xml

批量创建虚拟机 --- centos 7.4

1) vim-enhanced lftp ntpdate net-tools bash-completion

2) yum源

3) 网卡配置文件

自行编写

## 其他操作

1) virt-top //显示所有开机状态的虚拟机的CPU、内存使用率

2) virt-what //查看虚拟化类型

3) virt-clone //克隆虚拟机

关机状态

[root@localhost ~]# virt-clone -o centos7.4\_2 -n cenots7.4\_2\_new -f /var/lib/libvirt/images/centos7.4\_2\_new.img

4) snapshot

[root@localhost ~]# virsh snapshot-list centos7.4\_1

[root@localhost ~]# virsh snapshot-create-as centos7.4\_1 --name clean\_os