一个虚拟机至少由两个文件组成，默认存放位置：

/etc/libvirt/qemu                 ##虚拟机配置文件

/var/lib/libvirt/images/        ##虚拟机磁盘文件

**qemu、qumu-kvm**

QEMU是一个主机上的VMM（virtual machine monitor）,通过动态二进制转换来模拟CPU，并提供一系列的硬件模型，使guest os认为自己和硬件直接打交道，其实是同QEMU模拟出来的硬件打交道，QEMU再将这些指令翻译给真正硬件进行操作。通过这种模式，guest os可以和主机上的硬盘，网卡，CPU，CD-ROM，音频设备和USB设备进行交互。但由于所有指令都需要经过QEMU来翻译，因而性能会比较差：

**qemu-kvm**

从前面的介绍可知，KVM负责cpu虚拟化+内存虚拟化，实现了cpu和内存的虚拟化，但kvm并不能模拟其他设备，还必须有个运行在用户空间的工具才行。KVM的开发者选择了比较成熟的开源虚拟化软件QEMU来作为这个工具，QEMU模拟IO设备（网卡，磁盘等），对其进行了修改，最后形成了QEMU-KVM。

在QEMU-KVM中，KVM运行在内核空间，QEMU运行在用户空间，实际模拟创建、管理各种虚拟硬件，QEMU将KVM整合了进来，通过/ioctl 调用 /dev/kvm，从而将CPU指令的部分交给内核模块来做，KVM实现了CPU和内存的虚拟化，但KVM不能虚拟其他硬件设备，因此qemu还有模拟IO设备（磁盘，网卡，显卡等）的作用，KVM加上QEMU后就是完整意义上的服务器虚拟化。

综上所述，QEMU-KVM具有两大作用：

1. 提供对cpu，内存（KVM负责），IO设备（QEMU负责）的虚拟
2. 对各种虚拟设备的创建，调用进行管理（QEMU负责）

这个方案中，QEMU模拟其他的硬件，如Network, Disk，同样会影响这些设备的性能。于是又产生了pass through半虚拟化设备virtio\_blk, virtio\_net，提高设备性能。

**四、qemu下使用的几种镜像格式。**

1、raw

raw格式是最简单，什么都没有，所以叫raw格式。连头文件都没有，就是一个直接给虚拟机进行读写的文件。raw不支持动态增长空间，必须一开始就指定空间大小。所以相当的耗费磁盘空间。但是对于支持稀疏文件的文件系统（如ext4）而言，这方面并不突出。ext4下默认创建的文件就是稀疏文件，所以不要做什么额外的工作。

du -sh 文件名 可以查看文件的实际大小。也就是说，不管磁盘空间有多大，运行下面的指令没有任何问题：qemu-img create -f raw test.img 10000G

raw镜像格式是虚拟机种I/O性能最好的一种格式，大家在使用时都会和raw进行参照，性能越接近raw的越好。但是raw没有任何其他功能。对于稀疏文件的出现，像qcow这一类的运行时分配空间的镜像就没有任何优势了

2、cow

cow格式和raw一样简单，也是创建时分配所有空间，但cow有一个bitmap表记录当前哪些扇区被使用，所以cow可以使用增量镜像，也就是说可以对其做外部快照。但cow也没有其他功能，其特点就是简单。

3、qcow

qcow在cow的基础上增加了动态增加文件大小的功能，并且支持加密，压缩。qcow通过2级索引表来管理整个镜像的空间分配，其中第二级的索引用了内存cache技术，需要查找动作，这方面导致性能的损失。qcow现在基本不用，一方面其优化和功能没有qcow2好，另一方面，读写性能又没有cow和raw好

4、qcow2

qcow2是集各种技术为一体的超级镜像格式，支持内部快照，加密，压缩等一系列功能，访问性能也在不断提高。但qcow2的问题就是过于臃肿，把什么功能都集于一身。

**五、qcow2与raw的区别和联系**

。

qcow2 镜像格式是 QEMU 模拟器支持的一种磁盘镜像。它也是可以用一个文件的形式来表示一块固定大小的块设备磁盘。与普通的 raw 格式的镜像相比，有以下特性：

1. 更小的空间占用，即使文件系统不支持空洞(holes)；
2. 支持写时拷贝（COW, copy-on-write），镜像文件只反映底层磁盘的变化；
3. 支持快照（snapshot），镜像文件能够包含多个快照的历史；
4. 可选择基于 zlib 的压缩方式
5. 可以选择 AES 加密

**Copy-on-Write 镜像文件**

qcow2 镜像可以用来保存另一个镜像文件的变化，它并不去修改原始镜像文件，只记录与原始镜像文件的不同即可，这种镜像文件就叫做 copy-on-write 镜像。虽然是一个单独的文件，但它的大部分的数据都来自原始镜像，只有跟原始镜像文件相比有变化的 cluster 才会被记录下来。

这很容易去实现，在头部信息中记录原始文件路径即可。当需要从一个 copy-on-write 镜像文件中读取一个 cluster 的时候，首先检查这块区域是否已经在该镜像文件中被分配，如果没有就从原始文件读取。

**快照**

快照有些类似 Copy-On-Write 文件，但区别是快照是一个可写的。快照就是原始文件本身（内部快照）。它既包含做快照之前的原始文件部分，它本身也包含可写的部分。

**qcow2 的其他特性**

qcow2 支持压缩，它允许每个簇（cluster）单独使用 zlib 压缩。它也支持使用 128 位的 AES 密钥进行加密。

**创建 qcow2 和 raw 文件以及两种镜像的对比**

使用 QEMU 软件包自带的 qemu-img 软件创建 qcow2 文件

***4. 创建 qcow2 和 raw 文件***

**qemu-img create -f qcow2 test.qcow2 10G**

Formatting 'test.qcow2', fmt=qcow2 size=10737418240 encryption=off cluster\_size=65536 lazy\_refcounts=off

**qemu-img create -f raw test.raw 10G**

Formatting 'test.raw', fmt=raw size=10737418240

***5. qcow2 和 raw 文件占用空间情况对比***

从对比中可以看出 qcow 格式的镜像文件大小位 197120 字节，占用空间为 200K，占用了 200 块磁盘空间。而 raw 格式的文件则没有占用磁盘空间，它是一个空洞文件

**Raw 格式与 qcow2 转化**

QEMU 软件包里面提供的 qemu-img 工具可用于 image 镜像一些常用操作

将 raw 格式转化为 qcow2 格式的文件命令如下：

qemu-img convert -f raw -O qcow2 test.raw test.raw.qcow2

[qiaoliyong@localhost kimchi]$ ll -sh test.\*

200K -rw-r--r-- 1 qiaoliyong qiaoliyong 193K 5 月   6 10:29 test.qcow2

   0 -rw-r--r-- 1 qiaoliyong qiaoliyong  10G 5 月   6 10:28 test.raw

200K -rw-r--r-- 1 qiaoliyong qiaoliyong 193K 5 月   6 10:44 test.raw.qcow2

**两种格式文件的性能比较：**

**表1、使用 ide 作为虚拟磁盘的驱动的三种镜像格式性能对比**

表二、**使用 virtio 作为虚拟磁盘的驱动的三种镜像格式性能对比**

**小结：**

本文着重介绍了 QEMU 虚拟机使用的镜像文件 qcow2 的格式以及特性，并与 raw 格式镜像做了对比。qcow2 格式的文件虽然在性能上比rRaw 格式的有一些损失（主要体现在对于文件增量上，qcow2 格式的文件为了分配 cluster 多花费了一些时间），但是 qcow2 格式的镜像比 Raw 格式文件更小，只有在虚拟机实际占用了磁盘空间时，其文件才会增长，能方便的减少迁移花费的流量，更适用于云计算系统，同时，它还具有加密，压缩，以及快照等 raw 格式不具有的功能。

1）virsh list 列出当前虚拟机列表，不包括未启动的

2）virsh list --all 列出所有虚拟机，包括所有已经定义的虚拟机

3）virsh destroy vm-name 关闭虚拟机

4）virsh start vm-name 启动虚拟机

5）virsh edit vm-name 编辑虚拟机xml文件

6）virsh undefine vm-name 删除虚拟机

7）virsh shutdown vm-name 停止虚拟机

8）virsh reboot vm-name 重启虚拟机

9）virsh autostart vm-name 虚拟机随宿主机启动

**KVM快照：**

有时需要对虚拟机做快照，防止损坏了可以正常恢复，具体操作如下:

kvm快照，分两种：

1种是lvm快照，如果分区是lvm，可以利用lvm进行kvm的快照备份

2种由于raw格式不支持镜像，所以需要将格式转换为qcow2才可以创建快照。

方法2具体操作如下：

kvm虚拟机默认使用raw格式的镜像格式，性能最好，速度最快，它的缺点就是不支持一些新的功能，如支持镜像,zlib磁盘压缩,AES加密等。

要使用镜像功能，磁盘格式必须为qcow2。下面开始kvm虚拟机快照备份的过程。

virt-install管理

virt-manager命令打开虚拟机管理界面

virsh start centos7.0

virsh list --all 列出所有的虚拟机

virsh dominfo centos7.0 显示虚拟机信息

virt-top 显示虚拟机内存和cpu的使用情况

virsh shutdown centos7.0 关闭虚拟机（shutodwn）

virsh undefine centos7.0 删除虚拟机

virsh destroy centos7.0 强行终止虚拟机

systemctl restart libvirtd.service     ##重启虚拟机服务

#virsh snapshot-create-as --domain centos7.0 --name centos7\_snp\_ok 生成快照

#qemu-img create -f qcow2 -b centos7.0.qcow2 centos7.0\_node1.qcow2 生成快照

qemu-img create创建 -f格式 qcow2 -b 源文件 镜像文件

查看磁盘格式:

#cd /var/lib/libvirt/images/ 进入

#qemu-img info centos7.0.qcow2 查看

# qemu-img convert -fraw -O qcow2 test-script.img test-script.qcow2 转换

-f  源镜像的格式

-O 目标镜像的格式

virsh snapshot-current centos7.0 可以看到为当前最新的快照版本

**快照配置文件在/var/lib/libvirt/qemu/snapshot/虚拟机名称下**

#qemu-img convert -f raw CentOS6u5.raw -O qcow2 CentOS6u5.raw.qcow2 快照一定需要qcow2格式 才行

# virsh snapshot-list centos7.0 查看快照

# virsh snapshot-revert centos7.0 centos7\_snp\_ok 恢复快照 恢复虚拟机快照必须关闭虚拟机。

virshsnapshot-revert test-script 1419995560 确认恢复时间的快照

virsh snapshot-delete centos7.0 centos7\_snp\_ok 删除快照

可以通过修改虚拟机配置文件中的磁盘路径，或移动虚拟机磁盘文件至配置文件中的路径，恢复虚拟机

命令示例：

virsh create jinx.xml 通过配置文件启动虚拟机

virsh define Jinx.xml 通过配置文件添加虚拟机

virsh undefine Jinx 删除虚拟机配置文件