

kvm虚拟机命令行管理方式

本文介绍kvm虚拟机的命令行管理方式，包括kvm及libvirt的安装，虚拟机镜像的创建，虚拟机的创建及管理。

1. 虚拟机磁盘的创建

创建一个全新的kvm虚拟机，第一步应该是创建需要使用的虚拟机磁盘，qemu-img命令用于创建虚拟机磁盘：

a. 创建一个空间为40G的虚拟机磁盘文件disk.img，镜像格式为qcow2:

```
qemu-img create -f qcow2 ./disk.img 40G
```

注意，由于是松散文件，此时创建后的磁盘文件体积暂时还很小。

b. 基于已有的虚拟磁盘文件base.img来创建虚拟磁盘disk.img，相当于“克隆”的功能：

```
qemu-img create -b base.img ./disk.img -f qcow2 ./disk.img 40G
```

注意，此处使用-b参数来指定已有的磁盘文件，使用该方式生成的虚拟机磁盘文件虽然体积小，但已具有磁盘文件base.img的全部功能，且此时disk.img文件仅会存储与基础镜像的差异，是大规模部署相同业务时的首选方案。但在进行虚拟机迁移的时候，千万不要忘记将base.img文件也一并移走。

2. 编写虚拟机配置文件

kvm虚拟机可以使用命令行的方式启动，但是需要带着众多参数，记忆麻烦，管理也困难，因此libvirt这个服务在管理kvm虚拟机时，使用了xml文件来存储虚拟机的参数配置，一个标准的kvm虚拟机配置文件如下：

```

<domain type='kvm'>
  <name>origin</name>
  <memory unit='KiB'>4194304</memory>
  <currentMemory unit='KiB'>4194304</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <os>
    <type arch='x86_64'>hvm</type>
    <boot dev='cdrom'>
    <boot dev='hd'>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='localtime'>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' cache='writeback'>
      <source file='/opt/dyn_vm/vmdk/origin.img'>
      <target dev='vda' bus='virtio'>
    </disk>
    <disk type='file' device='cdrom'>
      <driver name='qemu' type='raw'>
      <source file='/opt/dyn_vm/iso/ubuntu-10.04.4-alternate-amd64.iso'>
      <target dev='hdc' bus='ide'>
      <readonly/>
    </disk>
    <interface type='network'>
      <source network='default'>
      <model type='virtio'>
    </interface>
    <input type='tablet' bus='usb'>
    <input type='mouse' bus='ps2'>
    <graphics type='vnc' port='5955' autoport='no' listen='0.0.0.0'>
    </graphics>
  </devices>
</domain>

```

以上的配置文件适合全新安装一个kvm虚拟机，该文件需要修改的部分有以下的部分：

a. `<name>origin</name>`

此处标志着虚拟机的名称，该名称不能与当前系统中已有的虚拟机名称冲突

b. `<memory unit='KiB'>4194304</memory>`

`<currentMemory unit='KiB'>4194304</currentMemory>`

此处配置虚拟机的内存，分为两个参数，memory元素指定了虚拟机启动时占用的最大内存，currentMemory元素指定了虚拟机系统正常运行时占用的内存，参数unit指定了单位。两个参数可以设置为同一个值。

c. `<os>`

`<type arch='x86_64'>hvm</type>`

`<boot dev='cdrom'/>`

`<boot dev='hd'/>`

`</os>`

type指虚拟机的类型，hvm表示全虚拟化，一般不需要修改此处。

boot元素指定了启动虚拟机的设备及设备的启动顺序，示例中的配置指定了两个可用于启动的设备：cdrom和磁盘，由于需要全新安装一个操作系统，所以将cdrom放到了hd的前面，可以让虚拟机启动后先从光驱启动。

d. `<emulator>/usr/libexec/qemu-kvm</emulator>`

此处指定了启动虚拟机使用的模拟器位置，确认此处配置的路径与系统中的qemu-kvm路径一致即可，在redhat系列的操作系统中使用yum来安装kvm的话，该选项一般不需要修改。

e. `<disk type='file' device='disk'>`

`<driver name='qemu' type='qcow2' cache='writeback'/>`

`<source file='/opt/dyn_vm/vmdk/origin.img'/>`

`<target dev='vda' bus='virtio'/>`

`</disk>`

此处配置虚拟机使用的硬盘，需要注意已创建的虚拟机磁盘的格式与type参数指定的类型一致，source元素的file参数指定的路径即为已创建的虚拟机磁盘的路径。

该配置项中的cache参数可修改为none、writethrough和writeback，三种参数各有利弊，但对于虚拟机中仅做少量文件写入的计算型业务来说，推荐使用writeback的模式，可以在少量数据写入时利用缓存达到较高的速度。

该配置中的target元素指定了使用virtio驱动，可以提高虚拟机磁盘的性能，但是需要注意的是，非server版的windows系统对该驱动并非原生支持，当要创建全新的windows虚拟机时，可以先去掉该选项，或是再创建一个floppy设备来放置virtio驱动。

f. `<disk type='file' device='cdrom'>`

`<driver name='qemu' type='raw'/>`

`<source file='/opt/dyn_vm/iso/ubuntu-10.04.4-alternate-amd64.iso'/>`

`<target dev='hdc' bus='ide'/>`

`<readonly'/>`

`</disk>`

此处是对cdrom设备的配置，仅需要修改光盘镜像的路径即可。

g. `<interface type='network'>`

`<source network='default'/>`

`<model type='virtio'/>`

`</interface>`

该配置指定了虚拟机的网卡信息，libvirt默认提供了一个名为default的虚拟网络，创建的虚拟机使用该网络以NAT方式连接外网。在不使用桥接的情况下，该default虚拟网络可以支持使用，不必进行修改。

h. `<graphics type='vnc' port='5955' autoport='no' listen='0.0.0.0'>`

`</graphics>`

该配置为虚拟机使用的图形设备的信息，此处选择使用vnc作为外部连接虚拟机的图形方式，端口为5955，autoport参数可配置是否每次启动虚拟机时自动生成一个端口，listen参数指定了只能由哪些IP段的机器连接该vnc。

3. 定义虚拟机
虚拟机的配置文件编写完成后，可以使用以下命令来定义该虚拟机：
`virsh define origin.xml`
4. 启动虚拟机
可以使用“start + 虚拟机名称”命令来启动已定义的虚拟机：
`virsh start origin`
5. 查看虚拟机的vnc端口
使用vncdisplay来查看虚拟机使用vnc端口
`virsh vncdisplay origin`
6. 关闭虚拟机
最正常的关闭虚拟机的方式就是在虚拟机系统中执行关机命令来关闭系统，如果虚拟机系统支持acpi，还可以使用以下的命令在外部优雅地关闭该虚拟机：
`virsh shutdown origin`
如果虚拟机已死机，或是该虚拟机不再重要，需要删除，则可以使用以下的命令来直接对虚拟机造成“拔电源”的效果：
`virsh destroy origin`
7. 删除虚拟机
如果需要删除虚拟机，则需要执行undefine命令来取消该虚拟机的定义：
`virsh undefine origin`