KVM 命令使用

目录

一、virsh/qemu-img 命令

1 虚拟机配置路径：/etc/libvirt/qemu

```
root@localhost qemu]# cd /etc/libvirt/qemu
root@localhost qemu]# ls
networks  r-12-VM.xml  s-9-VM.xml  v-10-VM.xml
root@localhost qemu]#
```

2 创建硬盘

qemu-img create /home/kvm/123.img 5G

3 创建虚拟机

virt-install \
--name=Winxp \
--ram 512 \
--vcpus=2 \
-f /home/123.img \
--cdrom /home/acton-systemvm-02062012.qcow2.bz2 \
--graphics vnc,listen=0.0.0.0,port=5980,password='12345678',keymap='en-us' \
--network bridge=cloudbr0 \
--force --autostart

(name 虚拟机名称，ram 分配内存， vcpus 分配 cpu 个数， cdrom guest 系统文件地址，network 网卡桥接名称)

```
[root@localhost qemu]# virt-install \
> --name=winxp \
> --ram 512 \
> --vcpus=2 \
> -f /home/123.img \
> --cdrom /home/acton-systemvm-02062012.qcow2.bz2 \
> --graphics vnc,listen=0.0.0.0,port=5980,password='12345678',keymap='en-us' \
> --network bridge=cloudbr0  \
> --force  --autostart

Starting install...
Creating domain...
Cannot open display:
Run 'virt-viewer --help' to see a full list of available command line options
Domain installation still in progress. You can reconnect to
the console to complete the installation process.
[root@localhost qemu]#
[root@localhost qemu]# virsh
Welcome to virsh, the virtualization interactive terminal.

Type:  'help' for help with commands
       'quit' to quit

virsh # list
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 7     r-12-VM                        running
 9     winxp                          running

virsh #
```

4 管理虚拟机:virsh

```
[root@localhost agent]# virsh
Welcome to virsh, the virtualization interactive terminal.

Type:  'help' for help with commands
       'quit' to quit

virsh #
```

5 显示虚拟机 list 或者 list –all

(list 显示本地活动虚拟机; list –all 显示本地所有的虚拟机（活动的+不活动的）)

```
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 -     i-2-11-VM                      shut off

virsh # list
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running

virsh #
```

6 网卡列表: virsh domiflist i-2-11-VM

```
[root@localhost ~]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 9     Winxp                          running
 10    r-12-VM                        running
 15    test1                          running
 24    i-2-11-VM                      running
 -     v-10-VM                        shut off

[root@localhost ~]# virsh domiflist i-2-11-VM
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet11     bridge     cloudVirBr129 e1000       06:2d:42:00:00:17

[root@localhost ~]#
```

7 网卡状态：virsh domifstat i-2-11-VM vnet11

```
[root@localhost ~]# virsh domiflist i-2-11-VM
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet11     bridge     cloudVirBr129 e1000       06:2d:42:00:00:17

[root@localhost ~]# virsh domifstat i-2-11-VM vnet11
vnet11 rx_bytes 340498
vnet11 rx_packets 4217
vnet11 rx_errs 0
vnet11 rx_drop 0
vnet11 tx_bytes 11602
vnet11 tx_packets 139
vnet11 tx_errs 0
vnet11 tx_drop 0

[root@localhost ~]#
```

8 磁盘列表：virsh domblklist i-2-11-VM

```
[root@localhost ~]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 9     Winxp                          running
 10    r-12-VM                        running
 15    test1                          running
 24    i-2-11-VM                      running
 -     v-10-VM                        shut off

[root@localhost ~]# virsh domblklist i-2-11-VM
Target     Source
------------------------------------------------
hda        /mnt/c209cbf5-0446-32b6-a53f-d4576c39b163/089fac27-5341-4f3b-b44d-a5980d93149f
hdc        -

[root@localhost ~]#
```

9 虚拟机快照

1）查看硬盘路径:

```
    'quit' to quit

virsh # dumpxml jy-VM
<domain type='kvm' id='33'>
  <name>jy-VM</name>
  <uuid>d33a191b-90a7-7399-a5d5-dfe7731b55b6</uuid>
  <memory unit='KiB'>524288</memory>
  <currentMemory unit='KiB'>524288</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.2.0'>hvm</type>
    <boot dev='cdrom'/>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>destroy</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' cache='none'/>
      <source file='/virhost/vmware/jy-test.img'/>
      <target dev='vda' bus='virtio'/>
      <alias name='virtio-disk0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </disk>
```

2）RAW 格式不具备快照功能，需要将 RAW 格式的镜像文件转换为 qcow2 格式

qemu-img convert -f raw -O qcow2 100.img 111.img

```
[root@localhost qemu]# cd /virhost/vmware
[root@localhost vmware]# ls
jy-test.img
[root@localhost vmware]# qemu-img convert -f raw -O qcow2 jy-test.img jy-test-bak.img
[root@localhost vmware]# ls
jy-test-bak.img  jy-test.img
```

3）查看镜像列表: virsh snapshot-list jy-VM

```
[root@localhost vmware]# virsh snapshot-list jy-VM
 Name                 Creation Time             State
------------------------------------------------------------
```

4）创建快照: 首先需要关闭虚拟机，然后按照下面的命令进行快照。最后恢复快照的时候先关机在恢复

virsh snapshot-create-as jy-VM kuaizhao

```
[root@localhost vmware]# virsh snapshot-create-as jy-VM kuaizhao
Domain snapshot kuaizhao created
[root@localhost vmware]#
```

或者: virsh snapshot-create-as --domain jy-VM --name kuaizhao1 --description "URL: jy-VM"

```
[root@localhost vmware]# virsh snapshot-create-as --domain jy-VM --name kuaizhao1 --description "URL: jy-VM"
Domain snapshot kuaizhao1 created
[root@localhost vmware]#
```

```
[root@localhost vmware]# virsh snapshot-list jy-VM
 Name                 Creation Time             State
------------------------------------------------------------
 kuaizhao             2013-11-05 17:50:33 +0800 running
 kuaizhao1            2013-11-05 18:02:47 +0800 running
```

5）查看快照配置: virsh snapshot-current jy-VM

```
[root@localhost vmware]# virsh snapshot-current jy-VM
<domainsnapshot>
  <name>kuaizhao1</name>
  <description>URL: jy-VM</description>
  <state>running</state>
  <parent>
    <name>kuaizhao</name>
  </parent>
  <creationTime>1383645767</creationTime>
  <domain type='kvm'>
    <name>jy-VM</name>
    <uuid>d33a191b-90a7-7399-a5d5-dfe7731b55b6</uuid>
    <memory unit='KiB'>524288</memory>
    <currentMemory unit='KiB'>524288</currentMemory>
    <vcpu placement='static'>1</vcpu>
    <os>
      <type arch='x86_64' machine='rhel6.2.0'>hvm</type>
      <boot dev='cdrom'/>
```

6）恢复快照: virsh snapshot-revert jy-VM kuaizhao

```
[root@localhost vmware]# virsh snapshot-revert jy-VM kuaizhao

[root@localhost vmware]#
```

7）删除快照: virsh snapshot-delete jy-VM kuaizhao1

```
[root@localhost vmware]# virsh snapshot-list jy-VM
 Name                 Creation Time             State
------------------------------------------------------------
 kuaizhao             2013-11-05 17:50:33 +0800 running
 kuaizhao1            2013-11-05 18:02:47 +0800 running

[root@localhost vmware]# virsh snapshot-delete jy-VM kuaizhao1
Domain snapshot kuaizhao1 deleted

[root@localhost vmware]# virsh snapshot-list jy-VM
 Name                 Creation Time             State
------------------------------------------------------------
 kuaizhao             2013-11-05 17:50:33 +0800 running

[root@localhost vmware]#
```

10 虚拟机状态：domstate 虚拟机名称或者 ID 或者 UUID

```
virsh # domstate i-2-11-VM
running

virsh #
```

11 启动虚拟机 start i-2-11-VM

```
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 7     Winxp                          running
 -     i-2-11-VM                      shut off

virsh # start i-2-11-VM
Domain i-2-11-VM started

virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 7     Winxp                          running
 13    i-2-11-VM                      running

virsh #
```

12 自启设置：autostart [--disable]虚拟机名称或者 ID 或者 UUID

```
virsh # dominfo 25
Id:             25
Name:           i-2-11-VM
UUID:           3dc3be86-f1c6-339a-b684-b9334ec5038d
OS Type:        hvm
State:          running
CPU(s):         1
CPU time:       18.2s
Max memory:     524288 kB
Used memory:    23548 kB
Persistent:     yes
Autostart:      disable
Managed save:   no

virsh # autostart 25
Domain 25 marked as autostarted

virsh # dominfo 25
Id:             25
Name:           i-2-11-VM
UUID:           3dc3be86-f1c6-339a-b684-b9334ec5038d
OS Type:        hvm
State:          running
CPU(s):         1
CPU time:       53.1s
Max memory:     524288 kB
Used memory:    23548 kB
Persistent:     yes
Autostart:      enable
Managed save:   no

virsh #
```

```
virsh # dominfo 25
Id:             25
Name:           i-2-11-VM
UUID:           3dc3be86-f1c6-339a-b684-b9334ec5038d
OS Type:        hvm
State:          paused
CPU(s):         1
CPU time:       1194.8s
Max memory:     524288 kB
Used memory:    23548 kB
Persistent:     yes
Autostart:      enable
Managed save:   no

virsh # autostart --disable 25
Domain 25 unmarked as autostarted

virsh # dominfo 25
Id:             25
Name:           i-2-11-VM
UUID:           3dc3be86-f1c6-339a-b684-b9334ec5038d
OS Type:        hvm
State:          paused
CPU(s):         1
CPU time:       1195.9s
Max memory:     524288 kB
Used memory:    23548 kB
Persistent:     yes
Autostart:      disable
Managed save:   no

virsh #
```

13 关闭虚拟机 shutdown i-2-11-VM

```
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 7     Winxp                          running
 11    i-2-11-VM                      running

virsh # shutdown i-2-11-VM
Domain i-2-11-VM is being shutdown

virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 7     Winxp                          running
 -     i-2-11-VM                      shut off
```

14 重启虚拟机 reboot i-2-11-VM

15 强制关闭电源 destroy i-2-11-VM

```
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 15    i-2-11-VM                      running
 16    Winxp                          running

virsh # destroy Winxp
Domain Winxp destroyed

virsh #
virsh #
virsh #
virsh #
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 15    i-2-11-VM                      running
 -     Winxp                          shut off
```

16 从 XML 定义一个虚拟机 define xml 文件

```
virsh # define /root/100.xml
Domain test defined from /root/100.xml

virsh # list
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 9     Winxp                          running
 10    r-12-VM                        running

virsh # start test
Domain test started

virsh # list
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 9     Winxp                          running
 10    r-12-VM                        running
 12    test                           running

virsh #
```

17 移除虚拟机：undefine 虚拟机名称或者 ID 或者 UUID

（使用 undefine 的前提是，虚拟机是关闭的，那我们怎么关闭虚拟机呢，可以使用 destroy，确切的说这个操作就是一脚将服务器的电源踹掉。为什么不用 shutdown 呢，可能是个 bug，apci 这个设备不好好工作。）

```
[root@localhost ~]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 9     Winxp                          running
 10    r-12-VM                        running
 15    test1                          running
 24    i-2-11-VM                      running
 26    test2                          running
 -     v-10-VM                        shut off

[root@localhost ~]# virsh destry test2
error: unknown command: 'destry'
[root@localhost ~]# virsh destroy test2
Domain test2 destroyed

[root@localhost ~]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 9     Winxp                          running
 10    r-12-VM                        running
 15    test1                          running
 24    i-2-11-VM                      running
 -     test2                          shut off
 -     v-10-VM                        shut off

[root@localhost ~]# virsh undefine test2
Domain test2 has been undefined

[root@localhost ~]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 9     Winxp                          running
 10    r-12-VM                        running
 15    test1                          running
 24    i-2-11-VM                      running
 -     v-10-VM                        shut off

[root@localhost ~]#
```

18 挂起一个正在运行的虚拟机，该虚拟机仍旧占用资源:suspend 虚拟机名称或者 ID 或者 UUID

```
virsh # list
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 23    Winxp                          running
 25    i-2-11-VM                      running

virsh # suspend 25
Domain 25 suspended

virsh # list
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 23    Winxp                          running
 25    i-2-11-VM                      paused

virsh #
```

19 从挂起状态恢复一个虚拟机: resume i-2-11-VM

```
virsh # list
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 23    Winxp                          running
 25    i-2-11-VM                      paused

virsh # resume i-2-11-VM
Domain i-2-11-VM resumed

virsh # list
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 23    Winxp                          running
 25    i-2-11-VM                      running

virsh #
```

20 输出客户端 XML 配置文件: dumpxml i-2-11-VM

```
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 15    i-2-11-VM                      running
 18    test                           running

virsh # dumpxml i-2-11-VM
<domain type='kvm' id='15'>
  <name>i-2-11-VM</name>
  <uuid>3dc3be86-f1c6-339a-b684-b9334ec5038d</uuid>
  <description>Other Linux (64-bit)</description>
  <memory unit='KiB'>524288</memory>
  <currentMemory unit='KiB'>524288</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <cputune>
    <shares>2000</shares>
    <period>100000</period>
    <quota>-1</quota>
  </cputune>
  <os>
    <type arch='x86_64' machine='rhel6.2.0'>hvm</type>
    <boot dev='cdrom'/>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' cache='none'/>
```

Default

就绪

21 导出客户端 XML 配置文件 virsh dumpxml 虚拟机名称或者 ID 或者 UUID >xml 文件（可以是相对路径也可以是绝对路径）

```
virsh # ^C
[root@localhost ~]# virsh list
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 9     Winxp                          running
 10    r-12-VM                        running
 13    i-2-11-VM                      paused
 15    test1                          running

[root@localhost ~]# ls
100.xml  12.xml  anaconda-ks.cfg  CentOS-6.3-x86_64-bin-DVD1.iso  install.log  install.log.syslog
[root@localhost ~]# virsh dumpxml test1 >test1.xml
[root@localhost ~]# ls
100.xml  12.xml  anaconda-ks.cfg  CentOS-6.3-x86_64-bin-DVD1.iso  install.log  install.log.syslog  test1.xml
[root@localhost ~]#
```

22 用 xml 文件创建虚拟机

virsh dumpxml node4 >/etc/libvirt/qemu/node6.xml
#导出虚拟机 node6 的硬件配置信息为/etc/libvirt/qemu/node6.xml

[root@target ~]# vim /etc/libvirt/qemu/node6.xml
<domain type='kvm' id='20'>　　#修改 node6 的 id 号
　<name>node6</name>　　#虚拟机 node6 的 name
　<uuid>4b7e91eb-6521-c2c6-cc64-c1ba72707fc7</uuid>　#uuid 必须修改，否则会和 node4 的冲突
　<memory>524288</memory>
　<currentMemory>524288</currentMemory>
　<vcpu>1</vcpu>
　<os>
　　<type arch='x86_64' machine='rhel5.4.0'>hvm</type>
　　<boot dev='network'/>
　</os>
　<features>
　　<acpi/>
　　<apic/>
　　<pae/>
　</features>
　<clock offset='utc'/>
　<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='none'/>
　　　<source file='/virhost/node4.img'/>　　#指定新虚拟机的硬盘文件
　　　<target dev='vda' bus='virtio'/>

```
    </disk>
    <interface type='bridge'>
      <mac address='54:52:00:69:d5:c7'/>
      <source bridge='br0'/>
      <target dev='vnet0'/>
      <model type='virtio'/>
    </interface>
    <interface type='bridge'>
      <mac address='54:52:00:69:d5:d7'/>
      <source bridge='br0'/>
      <target dev='vnet1'/>
      <model type='virtio'/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/4'/>
      <target port='0'/>
    </serial>
    <console type='pty' tty='/dev/pts/4'>
      <source path='/dev/pts/4'/>
      <target port='0'/>
    </console>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='5900' autoport='yes' keymap='en-us'/>
  </devices>
</domain>
```

[root@target ~]# virsh define /etc/libvirt/qemu/node6.xml
#使用虚拟描述文档建立虚拟机，可用 virsh edit node6 修改 node6 的配置文件

[root@target ~]# virsh start node6
#启动虚拟机

为虚拟机开启 vnc
[root@target ~]# virsh edit node4　　#编辑 node4 的配置文件
<graphics type='vnc' port='-1' autoport='yes' listen='127.0.0.1' keymap='en-us'/>
#port='-1' autoport='yes'：port 自动分配，监听回环网络（virt-manager 管理需要 listen='127.0.0.1'）,无密码
改为
<graphics type='vnc' port='5904' autoport='no' listen='0.0.0.0' keymap='en-us' passwd='xiaobai'/>
　#固定 vnc 管理端口 5904，不自动分配，vnc 密码 xiaobai，监听所有网络

远程 vnc 访问地址：192.168.32.40:5904

23 从 XML 配置文件生成客户端并启动新客户端: create /root/12.xml

```
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 15    i-2-11-VM                      running

virsh # create
error: command 'create' requires <file> option
virsh # create /root/12.xml
Domain test created from /root/12.xml

virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 15    i-2-11-VM                      running
 18    test                           running

virsh #
```

24 显示客户端 ID: domid 虚拟机名称或者 UUID

```
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 22    i-2-11-VM                      running

virsh # domid i-2-11-VM
22

virsh #
```

25 显示客户端 UUID: domuuid 虚拟机名称或者 ID

```
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 22    i-2-11-VM                      running

virsh # domid i-2-11-VM
22

virsh # domuuid i-2-11-VM
3dc3be86-f1c6-339a-b684-b9334ec5038d

virsh #
```

26 显示客户端信息: dominfo 虚拟机名称或者 ID 或者 UUID

```
virsh # dominfo i-2-11-VM
Id:             22
Name:           i-2-11-VM
UUID:           3dc3be86-f1c6-339a-b684-b9334ec5038d
OS Type:        hvm
State:          running
CPU(s):         1
CPU time:       15.7s
Max memory:     524288 kB
Used memory:    524288 kB
Persistent:     yes
Autostart:      disable
Managed save:   no

virsh #
```

27 显示客户端名称: domname ID 或者 UUID

```
virsh # domid i-2-11-VM
22

virsh # domuuid i-2-11-VM
3dc3be86-f1c6-339a-b684-b9334ec5038d

virsh # dominfo i-2-11-VM
Id:             22
Name:           i-2-11-VM
UUID:           3dc3be86-f1c6-339a-b684-b9334ec5038d
OS Type:        hvm
State:          running
CPU(s):         1
CPU time:       15.7s
Max memory:     524288 kB
Used memory:    524288 kB
Persistent:     yes
Autostart:      disable
Managed save:   no

virsh # domname 22
i-2-11-VM

virsh # domname 3dc3be86-f1c6-339a-b684-b9334ec5038d
i-2-11-VM

virsh #
```

28 编辑虚拟机配置

```
virsh # list
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 23    Winxp                          running
 25    i-2-11-VM                      running

virsh # edit i-2-11-VM

<domain type='kvm'>
  <name>i-2-11-VM</name>
  <uuid>3dc3be86-f1c6-339a-b684-b9334ec5038d</uuid>
  <description>Other Linux (64-bit)</description>
  <memory unit='KiB'>524288</memory>
  <currentMemory unit='KiB'>524288</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.2.0'>hvm</type>
    <boot dev='cdrom'/>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' cache='none'/>
      <source file='/mnt/c209cbf5-0446-32b6-a53f-d4576c39b163/089fac27-5341-4f3b-b44d-a5980d93149f'/>
```

29 显示 VNC 端口号：vncdisplay 虚拟机名称或者 ID

```
virsh # list
 Id    Name                           State
----------------------------------------------------
 1     v-10-VM                        running
 2     s-9-VM                         running
 3     r-12-VM                        running
 23    Winxp                          running
 25    i-2-11-VM                      running

virsh # vncdisplay i-2-11-VM
:3

virsh #
```

30 下表提供所有 virsh 命令行选项的快速参考。

命令 Description
help 打印基本帮助信息。
list 列出所有客户端。
dumpxml 输出客户端 XML 配置文件。
create 从 XML 配置文件生成客户端并启动新客户端。
start 启动未激活的客户端。
destroy 强制客户端停止。
define 为客户端输出 XML 配置文件。
domid 显示客户端 ID。
domuuid 显示客户端 UUID。
dominfo 显示客户端信息。

domname 显示客户端名称。
domstate 显示客户端状态。
quit 退出这个互动终端。
reboot 重新启动客户端。
restore 恢复以前保存在文件中的客户端。
resume 恢复暂停的客户端。
save 将客户端当前状态保存到某个文件中。
shutdown 关闭某个域。
suspend 暂停客户端。
undefine 删除与客户端关联的所有文件。
migrate 将客户端迁移到另一台主机中。

使用以下 virsh 命令管理客户端及管理程序资源：
命令 Description
setmem 为客户端设定分配的内存。
setmaxmem 为管理程序设定内存上限。
setvcpus 修改为客户端分配的虚拟 CPU 数目。
vcpuinfo 显示客户端的虚拟 CPU 信息。
vcpupin 控制客户端的虚拟 CPU 亲和性。
domblkstat 显示正在运行的客户端的块设备统计。
domifstat 显示正在运行的客户端的网络接口统计。

attach-device 使用 XML 文件中的设备定义在客户端中添加设备。
attach-disk 在客户端中附加新磁盘设备。
attach-interface 在客户端中附加新网络接口。
detach-device 从客户端中分离设备，使用同样的 XML 描述作为命令 attach-device。
detach-disk 从客户端中分离磁盘设备。
detach-interface 从客户端中分离网络接口。

31 存储池和存储卷的管理

1.创建 KVM 主机存储池

1).创建基于文件夹（目录）的存储池
virsh pool-define-as vmware_pool --type dir --target /virhost/vmware #定义存储池 vmware_pool 或
virsh pool-create-as --name vmware_pool --type dir --target /virhost/vmware
#创建存储池 vmware_pool，类型为文件目录,/virhost/vmware，与 pool-define-as 结果一样

2).创建基于文件系统的存储池
virsh pool-define-as --name vmware_pool --type fs --source-dev

/dev/vg_target/LogVol02 --source-format ext4 --target /virhost/vmware
或
virsh pool-create-as --name vmware_pool --type fs --source-dev /dev/vg_target/LogVol02 --source-format ext4 --target /virhost/vmware

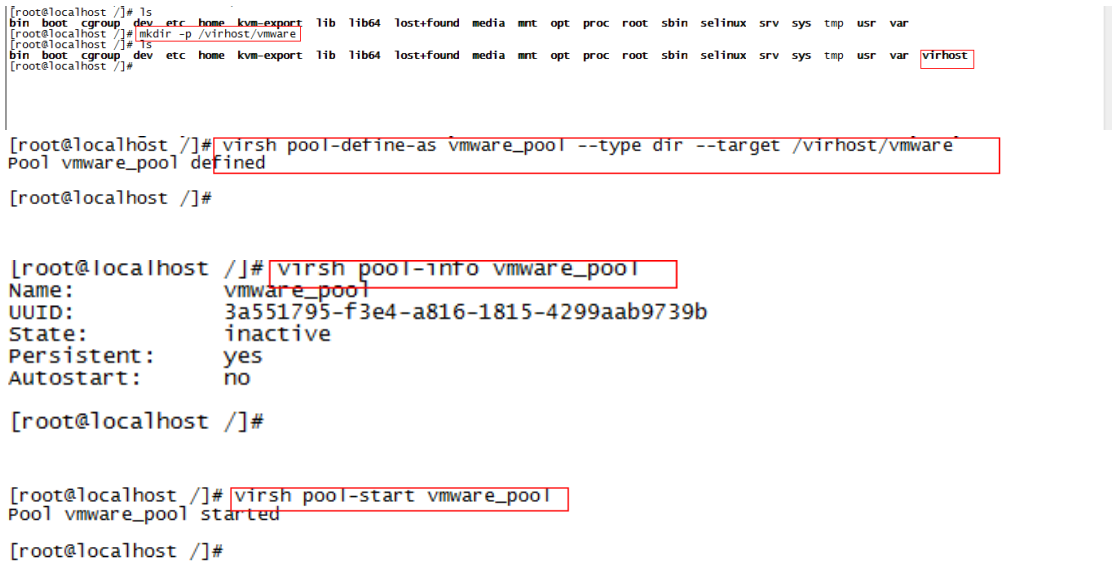
3).查看存储池信息
virsh pool-info vmware_pool #查看存储域（池）

4).启动存储池
virsh pool-start vmware_pool #启动存储池
virsh pool-list

5)销毁存储域,取消存储池
virsh pool-destroy vmware_pool #销毁存储池
virsh pool-list --all
virsh pool-undefine vmware_pool #取消存储池的定义
virsh pool-list --all

2.创建了存储池后，就可以创建一个卷，这个卷是用来做虚拟机的硬盘
virsh vol-create-as --pool vmware_pool --name node6.img --capacity 10G --allocation 1G --format qcow2#创建卷 node6.img,所在存储池为 vmware_pool，容量 10G，初始分配 1G，文件格式类型 qcow2
virsh vol-info /virhost/vmware/node6.img #查看卷信息名称： node6.img 类型： 文件容量： 10.00 GB 分配： 136.00 KB

3.在存储卷上安装虚拟主机
virt-install --connect qemu:///system \-n node7 \-r 512 \-f /virhost/vmware/node7.img \--vnc \--os-type=linux \--os-variant=rhel6 \--vcpus=1 \--network bridge=br0 \-c /mnt/rhel-server-6.0-x86_64-dvd.iso



```
[root@localhost /]# virsh pool-list
Name                 State      Autostart
-----------------------------------------
5248833d-e4f8-4b59-ac96-fa9d0cb4612e active     no
c209cbf5-0446-32b6-a53f-d4576c39b163 active     no
vmware_pool          active     no

[root@localhost /]#
```

```
[root@localhost /]# virsh vol-create-as --pool vmware_pool --name jy-test.img --capacity 10G --allocation 1G --format qcow2
Vol jy-test.img created

[root@localhost /]#
```

```
[root@localhost /]# virsh vol-info /virhost/vmware/jy-test.img
Name:           jy-test.img
Type:           file
Capacity:       10.00 GB
Allocation:     136.00 KB

[root@localhost /]#
```

```
[root@localhost /]# cd /home/
[root@localhost home]# ls
123.img  acton-systemvm-02062012.qcow2.bz2  CloudStack-oss-3.0.6  CloudStack-oss-3.0.6-20130807.tar.gz2  lost+found
[root@localhost home]#
```

```
[root@localhost home]# virt-install --connect qemu:///system \-n jy-VM \-r 512 \-f /virhost/vmware/jy-test.img \--vnc \--os-type=linux \--os-variant=rhel6 \--vcpus=1 \-
-network bridge=cloudbr0 \-c /home/acton-systemvm-02062012.qcow2.bz2

starting install...
Creating domain...                                                                                                  |    0 B     00:00
Cannot open display:
Run 'virt-viewer --help' to see a full list of available command line options
Domain installation still in progress. You can reconnect to
the console to complete the installation process.
[root@localhost home]#
```

```
[root@localhost home]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 5     s-13-VM                        running
 6     v-14-VM                        running
 9     Winxp                          running
 10    r-12-VM                        running
 15    test1                          running
 31    i-2-11-VM                      running
 32    node7                          running
 33    jy-VM                          running
 -     v-10-VM                        shut off

[root@localhost home]#
```

```
[root@localhost home]# virsh domblklist jy-VM
Target     Source
------------------------------------------------
vda        /virhost/vmware/jy-test.img
hdc        /home/acton-systemvm-02062012.qcow2.bz2

[root@localhost home]#
```

二、KVM 虚拟机安装

备注

安装 virtsh virt-install 管理工具

- yum install libvirt-client python-virtinst

1、虚拟机安装

1）建立虚拟机磁盘镜像文件

qcow2 格式是 kvm 支持的标准格式，raw 格式为虚拟磁盘文件通用格式。有测试数据表明 raw 格式的 I/O 性能略高于 qcow2 格式，但是在加密，容量，快照方面

qcow2 格式有优势

1. qemu-img **create** -f qcow2 test.qcow2 20G //建立 qcow2 格式磁盘文件
2. qemu-img **create** -f raw test.raw 20G //建立 raw 格式磁盘文件
3. qemu-img info test.qcow2 //查看已经创建的虚拟磁盘文件

2）建立虚拟机示例

创建 1G 内存、2 核 CPU、单网卡，磁盘和网络驱动使用 virtio 的 linux 虚拟机

1. virt-install --name rhel5.4 \
2. --boot network,cdrom,menu=on \
3. --ram 1024 --vcpus=2 \
4. --os-variant=rhel5.4 \
5. --accelerate \
6. --cdrom=/troodon/ISO/CentOS-6.2-x86_64-bin-DVD1.iso \
7. --disk path=/troodon/KVM/test01/test01.img,size=30,format=qcow2,bus=virtio \
8. --bridge=br0,model=virtio --mac=54:52:00:01:79:e9 \
9. --vnc --vncport=5991 --vnclisten=0.0.0.0.

创建 8G 内存、4 核 CPU、双网卡，磁盘和网络驱动使用 virtio 的 linux 虚拟机

1. virt-install --name rhel5.2 \
2. --boot network,cdrom,menu=on \
3. --ram 8192 --vcpus=4 \
4. --os-variant=rhel6 \
5. --cdrom=/troodon/ISO/CentOS-6.2-x86_64-bin-DVD1.iso \
6. --disk path=/troodon/KVM/test01/test01.img,format=qcow2,bus=virtio \
7. --bridge=br0,model=virtio --mac=54:52:00:0b:8b:79 \
8. --bridge=br1,model=virtio --mac=54:52:00:7e:8e:cd \
9. --vnc --vncport=5991 --vnclisten=0.0.0.0

创建 1G 内存、1 核 CPU、单网卡，磁盘和网络驱动使用 virtio 的 windows 虚拟机

1. virt-install --name win2k3 \
2. --ram 1024 --vcpus=1 \
3. --os-variant=win2k3 \
4. --accelerate \
5. --cdrom=/troodon/ISO/Windows/2003/win2k3sp1.iso \
6. --disk path=/troodon/KVM/win2k3/win2k3.img,size=5,format=qcow2,bus=virtio \
7. --bridge=br0,model=virtio \
8. --vnc --vncport=5991 --vnclisten=0.0.0.0

一些解释:

virtio 驱动对于磁盘和网络 I/O 性能有很大提升（默认 qemu 驱动），linux 虚拟机安装完成后自动加载 virtio 驱动，windows 虚拟机需要下载 virtio 驱动后安装
--mac 参数指定网卡 mac 地址，如果不指定系统会自动分配
虚拟机配置文件存储在 /etc/libvirt/qemu 目录，配置文件为 XML 格式
开机自启动的虚拟机配置可以通过链接放到 /etc/libvirt/qemu/autostart 目录，配置文件为 XML 格式

3）虚拟机安装指令 virt-install 简介

1. --name 指定虚拟机名称，virsh 操作指定虚拟机时所需要的参数，不可以重复。
2. --ram 分配内存大小，安装完成后可以用 virsh 调整。
3. --vcpus 分配 CPU 核心数，最大与实体机 CPU 核心数相同，安装完成后也可以用 virsh 调整。
4. --disk 指定虚拟机镜像， size 指定分配大小单位为 G。
5. --network 网络类型，此处用的是默认，一般用的应该是 bridge 桥接。
6. --os-variant 指定操作系统类型，此处使用的是标准 Linux 2.6，其他的可以通过 man virt-install 详细查看。
7. --accelerate 加速，具体什么原理还不太清楚。
8. --cdrom 指定安装镜像所在。
9. --vnc 启用 VNC 远程管理，一般安装系统都要启用。
10. --vncport 指定 VNC 监控端口，默认端口为 5900，端口不能重复。
11. --vnclisten 指定 VNC 绑定 IP，默认绑定 127.0.0.1，这里将其改为 0.0.0.0 以便可以通过外部连接。

4）虚拟机管理指令 virsh 简介

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 start domain-**name** 启动指定虚拟机
4. virsh shutdown domain-**name** 停止指定虚拟机
5. virsh reboot domain-**name** 重新启动指定虚拟机
6. virsh autostart domain-**name** 指定虚拟机开机自动启动

三、qemu-img 命令详解

qemu-img 是 QEMU 的磁盘管理工具，在 qemu-kvm 源码编译后就会默认编译好 qemu-img 这个二进制文件。qemu-img 也是 QEMU/KVM 使用过程中一个比较重要的工具，本节对其用法和实践使用方法进行介绍。

qemu-img 工具的命令行基本用法如下：
qemu-img *command* [*command options*]
它支持的命令分为如下几种：

（1）check [-f *fmt*] *filename*
对磁盘镜像文件进行一致性检查，查找镜像文件中的错误，目前仅支持对“qcow2”、“qed”、“vdi”格式文件的检查。参数-f *fmt* 是指定文件的格式，如果不指定格式 qemu-img 会自动检测，*filename* 是磁盘镜像文件的名称（包括路径）。
如下命令行演示了 qemu-img 的 check 命令的使用方法。
[root@jay-linux kvm_demo]# qemu-img check rhel6u3.qcow2
No errors were found on the image.

（2）create [-f *fmt*] [-o *options*] *filename* [*size*]
创建一个格式为 *fmt* 大小为 *size* 文件名为 *filename* 的镜像文件。根据文件格式 *fmt* 的不同，还可以添加一个或多个选项（*options*）来附加对该文件的各种功能设置，可以使用“-o ?”来查询某种格式文件支持那些选项，在“-o”选项中各个选项用逗号来分隔。
size 选项用于指定镜像文件的大小，其默认单位是字节（bytes），也可以支持 k（或 K）、M、G、T 来分别表示 KB、MB、GB、TB 大小。另外，镜像文件的大小（size）也并非必须写在命令的最后，它也可以被写在“-o”选项中作为其中一个选项。
对 create 命令的演示如下所示，其中包括查询 qcow2 格式支持的选项、创建有 backing_file 的 qcow2 格式的镜像文件、创建没有 backing_file 的 10GB 大小的 qcow2 格式的镜像文件。
[root@jay-linux kvm_demo]# **qemu-img create -f qcow2 -o ? temp.qcow**
Supported options:
size Virtual disk size
compat Compatibility level (0.10 or 1.1)
backing_file File name of a base image
backing_fmt Image format of the base image
encryption Encrypt the image
cluster_size qcow2 cluster size
preallocation Preallocation mode (allowed values: off, metadata)

[root@jay-linux kvm_demo]# **qemu-img create -f qcow2 -b rhel6u3.img rhel6u3.qcow2**
Formatting ‘rhel6u3.qcow2’, fmt=qcow2 size=8589934592 backing_file=’rhel6u3.img’ encryption=off cluster_size=65536

[root@jay-linux kvm_demo]# **qemu-img create -f qcow2 -o backing_file=rhel6u3.img rhel6u3-1.qcow2**
Formatting ‘rhel6u3-1.qcow2’, fmt=qcow2 size=8589934592 backing_file=’rhel6u3.img’ encryption=off cluster_size=65536

[root@jay-linux kvm_demo]# **qemu-img create -f qcow2 -o backing_file=rhel6u3.img,size=20G rhel6u3-2.qcow2**
Formatting 'rhel6u3-2.qcow2', fmt=qcow2 size=21474836480 backing_file='rhel6u3.img' encryption=off cluster_size=65536

[root@jay-linux kvm_demo]# **qemu-img create -f qcow2 ubuntu.qcow2 10G**
Formatting 'ubuntu.qcow2', fmt=qcow2 size=10737418240 encryption=off cluster_size=65536

（3）commit [-f *fmt*] *filename*
提交 *filename* 文件中的更改到后端支持镜像文件（创建时通过 backing_file 指定的）中去。

（4）convert [-c] [-f *fmt*] [-O *output_fmt*] [-o *options*] *filename* [*filename2* [...]] *output_filename*
将 *fmt* 格式的 *filename* 镜像文件根据 *options* 选项转换为格式为 *output_fmt* 的名为 *output_filename* 的镜像文件。它支持不同格式的镜像文件之间的转换，比如可以用 VMware 用的 vmdk 格式文件转换为 qcow2 文件，这对从其他虚拟化方案转移到 KVM 上的用户非常有用。
其中，"-c"参数是对输出的镜像文件进行压缩，不过只有 qcow2 和 qcow 格式的镜像文件才支持压缩。同样可以使用"-o *options*"来指定各种选项，如：后端镜像、文件大小、是否加密等等。使用 backing_file 选项来指定后端镜像，让生成的文件是 copy-on-write 的增量文件。
下面的命令行演示了两个转换：将 VMware 的 vmdk 格式镜像转换为 KVM 可以使用的 qcow2 镜像，将一个 raw 镜像文件转化为 qcow2 格式的镜像。
[root@jay-linux kvm_demo]# **qemu-img convert my-vmware.vmdk my-kvm.img**
（此处并无实际存在 vmdk 文件，仅演示其命令行操作）

[root@jay-linux kvm_demo]# **qemu-img convert -O qcow2 rhel6u3.img rhel6u3-a.img**

（5）info [-f *fmt*] *filename*
展示 *filename* 镜像文件的信息。如果文件是使用稀疏文件的存储方式，也会显示出它的本来分配的大小以及实际已占用的磁盘空间大小。如果文件中存放有客户机快照，快照的信息也会被显示出来。下面的命令行演示了前面进行文件转换的输入、输出文件的信息。
[root@jay-linux kvm_demo]# **qemu-img info rhel6u3.img**
image: rhel6u3.img
file format: **raw**
virtual size: 8.0G (8589934592 bytes)
disk size: 8.0G
[root@jay-linux kvm_demo]# **qemu-img info rhel6u3-a.img**
image: rhel6u3-a.img
file format: **qcow2**

virtual size: 8.0G (8589934592 bytes)
disk size: 6.8G
cluster_size: 65536

（6）snapshot [-l | -a *snapshot* | -c *snapshot* | -d *snapshot*] *filename*
“-l” 选项是查询并列出镜像文件中的所有快照，“-a *snapshot*”是让镜像文件使用某个快照，“-c *snapshot*”是创建一个快照，“-d”是删除一个快照。

（7）rebase [-f *fmt*] [-t *cache*] [-p] [-u] -b *backing_file* [-F *backing_fmt*] *filename*
改变镜像文件的后端镜像文件，只有 qcow2 和 qed 格式支持 rebase 命令。使用“-b *backing_file*”中指定的文件作为后端镜像，后端镜像也被转化为“-F *backing_fmt*”中指定的后端镜像格式。
它可以工作于两种模式之下，一种是安全模式（Safe Mode）也是默认的模式，qemu-img 会去比较原来的后端镜像与现在的后端镜像的不同进行合理的处理；另一种是非安全模式（Unsafe Mode），是通过“-u”参数来指定的，这种模式主要用于将后端镜像进行了重命名或者移动了位置之后对前端镜像文件的修复处理，由用户去保证后端镜像的一致性。

（8）resize *filename* [+ | -]*size*
改变镜像文件的大小，使其不同于创建之时的大小。“+”和“-”分别表示增加和减少镜像文件的大小，而 size 也是支持 K、M、G、T 等单位的使用。缩小镜像的大小之前，需要在客户机中保证里面的文件系统有空余空间，否则会数据丢失，另外，qcow2 格式文件不支持缩小镜像的操作。在增加了镜像文件大小后，也需启动客户机到里面去应用“fdisk”、“parted”等分区工具进行相应的操作才能真正让客户机使用到增加后的镜像空间。不过使用 resize 命令时需要小心（最好做好备份），如果失败的话，可能会导致镜像文件无法正常使用而造成数据丢失。
如下命令行演示了两个镜像的大小改变：将一个 8GB 的 qcow2 镜像增加 2GB 的空间，也将一个 8GB 大小的 raw 镜像减少 1GB 空间。
[root@jay-linux kvm_demo]# **qemu-img resize rhel6u3-a.img +2G**
Image resized.
[root@jay-linux kvm_demo]# **qemu-img info rhel6u3-a.img**
image: rhel6u3-a.img
file format: qcow2
virtual size: **10G** (10737418240 bytes)
disk size: 6.8G
cluster_size: 65536

[root@jay-linux kvm_demo]# **qemu-img resize rhel6u3-b.img -1G**
Image resized.
[root@jay-linux kvm_demo]# **qemu-img info rhel6u3-b.img**
image: rhel6u3-b.img
file format: raw
virtual size: **7.0G** (7516192768 bytes)
disk size: 6.5G

四、virsh 命令补充

1、使用方法

虚拟机配置路径：/etc/libvirt/qemu

创建硬盘：
#qemu-img create /home/kvm/123.img 5G

创建虚拟机：
virt-install \
--name=Winxp \
--ram 512 \
--vcpus=2 \
-f /home/kvm/123.img \
--cdrom /home/centos64.iso \
--graphics vnc,listen=0.0.0.0,port=5980,password='12345678',keymap='en-us' \
--network bridge=br0 \
--force --autostart

(name 虚拟机名称，ram 分配内存， vcpus 分配 cpu 个数， cdrom guest 系统文件地址，network 网卡桥接名称)

恢复虚拟机
virsh# define /etc/libvirt/qemu/winxp.xml

virsh 进行管理虚拟机
virsh# list --all # 显示所有虚拟机 --all 显示全部

启动虚拟机
#virsh start Winxp

关闭虚拟机
#virsh shutdown Winxp

强制关机
#virsh destroy Winxp

移除虚拟机

#virsh undefine Winxp

显示 vnc 端口
#virsh vncdisplay 2

动态查询 kvm 使用资源
#top -d 1 | grep kvm

查询 kvm 进程
ps -aux | grep kvm

开机自动启动虚拟机
#virsh autostart Winxp（虚拟机名）

克隆 KVM 虚拟机
virt-clone -o Winxp -n winxpclong -f /home/kvm/winxpclong.img
(-o 原始客体的名称 -n 新客户端的名称 -f 作为新客户端磁盘映像的新文件)

导出虚拟机 Winxp 的硬件配置信息为/etc/libvirt/qemu/Winxpbak.xml
#virsh dumpxml Winxp >/etc/libvirt/qemu/Winxpbak.xml

编辑虚拟机配置
#virsh edit Winxp

2、virsh 命令管理虚拟机

```
$ sudo virsh -c qemu:///system list
 Id Name                 State
----------------------------------
  1 Ubuntu               running
  2 Ubuntu-Server        running
```

virsh

显示虚拟机列表：
virsh # list --all

启动虚拟机：
virsh # start [name]

关闭虚拟机：
virsh # shutdown [name]

重启虚拟机：
virsh # reboot [name]

例 29.1. virsh
virsh # list --all

Id	Name	State
-	CentOS6.4	shut off
-	FreeBSD	shut off
-	Test	shut off
-	Ubuntu	shut off
-	www	shut off

virsh # start Ubuntu
Domain Ubuntu started

virsh # list --all

Id	Name	State
1	Ubuntu	running
-	CentOS6.4	shut off
-	FreeBSD	shut off
-	Test	shut off
-	www	shut off

virsh # quit

六、qemu-kvm 命令行参数

http://qemu.weilnetz.de/qemu-doc.html
http://wiki.qemu.org/download/qemu-doc.html
http://wiki.gentoo.org/wiki/QEMU/Options
http://wiki.libvirt.org/page/QEMUSwitchToLibvirt

1、cpu 相关参数

-cpu：指定 cpu 模型，默认的为 qemu64，可以通过“-cpu ？”查询当前支持的 cpu 模型

-smp：设置虚拟机的 vcpu 个数。后面还可以加 cores threads socke.

2、内存相关参数
-m:设置虚拟机内存大小，默认单位为 MB。
-mem-path patch:指定从 path 路径表示的临时文件中为 guest 分配内存。
-mem-prealloc:启动时即分配全部内存，而不是根据 guest 请求动态分配，与 -mem-path 参数配合使用。
-balloon：开全内存 balloon 功能，俗称内存气球。
3.磁盘相关参数
-hda、-hdb 和 cdrom 等：设置虚拟机的 IDE 磁盘和光盘设置。
-driver：配置驱动器。
-boot：设置虚拟机的启动选项
4.网络相关参数
-net nic:为虚拟机创建一个 nic 网卡
-net user:让虚拟机使用不需要管理权限的用户模式网络(user mode network).
-net tap:使用 host 的 tap 网络接口来帮助 guest 建立网络。
-net none:不配置任何网络设备。
5.图形显示参数
-sdl:使用 sdl 方式显示客户机。
-vnc：使用 vnc 方式显示客户机。
-vga：设置虚拟机中的 vga 显卡类型，默认为“-vga cirrus”.
-nographic：关闭 qemu 的图形化界面输出。
6.其他常用参数
-h:显示帮助手册
#qemu-system-x86_64 -h：会显示所有参数
-noreboot:guest 执行 reboot 操作时，系统关闭后退出 qemu-kvm，而不会再启动虚拟机。
-no-shutdown:虚拟机 shutdown 后，系统关闭后，不退出 qemu-kvm 进程，保持这个进程存在，他的 monitor 仍然可以用。
-loadvm:加载快照状态，与 monitor 中的“loadvm”命令类似
-nodefaults:不创建默认的设备。默认会创建一些显卡、串口、控制台等设备
-readconfig:从文件中读虚拟机设备的配置信息。
-writeconfig：将虚拟机的配置信息写到文件中。
-nodedefconfig:不加载默认的配置文件。默认会加载/use/local/share/qemu 下的文件。
-no-user-config:不加载用户自定义的配置文件。
libvirt 透传命令到 qemu。如透传-s 到 qemu，xml 格式如下：

```
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  ......
<qemu:commandline>
    <qemu:arg value='-s'/>
</qemu:commandline>
</domain>
```