# 生命周期管理

## 创建虚拟机

# virsh create <domain.xml>

## 关闭虚拟机

# virsh shutdown <id>|<name>|<uuid>| [-mode <agent>|<acpi>]

注：支持两种关闭虚机的方式：acpi模式和qga模式。

## 强制关闭虚机

# virsh destroy <id>|<name>|<uuid>

## 重启虚拟机

# virsh reboot <id>|<name>|<uuid> [-mode <acpi>|<agent>]

注：支持两种模式重启虚机：acpi模式和qga模式。

## 复位虚拟机

# virsh reset <id>|<name>|<uuid>

## 暂停虚拟机

# virsh suspend <id>|<name>|<uuid>

## 恢复暂停虚拟机

# virsh resume <id>|<name>|<uuid>

## 创建/恢复虚机快照

### 带内存快照

#### 创建raw格式快照

首先，用virsh save命令保存内存状态，保存完成后虚机将自动关机。

# virsh save <id>|<name>|<uuid> /path/file.save

然后通过拷贝的方式保存磁盘状态，virsh save时虚拟机所在磁盘将作为快照恢复后的运行节点。

注：快照与克隆的唯一区别在于快照只读不写，克隆可读写！

#### 恢复raw格式快照

# virsh restore /path/file.save

#### 创建qcow2格式快照

创建之前需要使虚拟机处于活动状态（raw格式则是处于关闭状态）。

管理层负责存储节点可用，将快照运行磁盘文件路径、磁盘备份文件路径、内存快照文件路径传给libvirt，所有的工作由libvirt和qemu完成。管理层负责删除已经没有作用的文件lv。

# virsh snapshot-create-as <id>|<name>|<uuid> [snapshot\_name]

--diskspec vda,file=[vda\_running\_file],snapfile=[vda\_snapshot\_file]

--diskspec vdb,file=[vdb\_running\_file],snapfile=[vdb\_snapshot\_file]

--memspec file=[memory\_file]

--external-opt

--cow

--no-metadata

说明：

--diskspec\_file：快照后虚拟机运行节点（快照运行树结构中的叶子节点）

--diskspec\_snapfile：快照后备份节点

--memspec：内存快照文件

--external-opt：说明要创建的为外置快照

--cow：说明创建cow快照

--no-metadata：创建快照时不写入libvirt数据库，否则不能迁移

如果管理层事先已经创建好qcow2格式磁盘文件，需要加上—reuse-external参数。

#### 恢复qcow2格式快照

快照恢复之前需要关闭虚机，然后执行恢复操作：

管理层将虚拟机运行磁盘文件路径、创建快照时的快照备份文件路径、内存快照文件路径传给libvirt。管理层负责删除已经没用的文件lv。

# virsh snapshot-revert [vm\_name] [snapshot\_name]

–memspec file=[memory\_file]

--diskspec vda,backfile=[vda\_back\_file],snapshot=external,runfile=[vda\_run\_file]

--diskspec vdb,backfile=[vdb\_back\_file],snapshot=external,runfile=[vdb\_run\_file]

--external

说明：

--memspec：内存快照文件

--diskspec backfile：创建快照时创建的备份节点

--diskspec runfile：恢复快照后虚拟机运行的节点

--external：说明快照类型为外置快照

注意：diskspec runfile参数长度不能超过虚拟机配置中disk路径长度。

### 不带内存快照

raw格式虚拟机不支持不带内存的快照。

#### 创建qcow2格式快照

# virsh snapshot-create-as <id>|<name>|<uuid> [snapshot\_name] –disk-only

--diskspec vda,file=[vda\_running\_file],snapfile=[vda\_snapshot\_file]

--diskspec vdb,file=[vdb\_running\_file],snapfile=[vdb\_snapshot\_file]

--no-metadata

说明：

--disk-only：只对磁盘做快照

--diskspec\_file：快照后虚拟机运行节点

--diskspec\_snapfile：快照后备份节点

--no-metadata：创建快照时不写入libvirt数据库，否则不能迁移

如果管理层事先已经创建好qcow2格式磁盘文件，需要加上—reuse-external参数。

如果只想拉一个叶子节点，--diskspec后只加file参数，不需要加snapfile参数。

注意：只支持对非系统盘进行不带内存的快照，否则可能会导致虚拟机系统异常。

#### 恢复qcow2格式快照

修改虚拟机的配置文件中磁盘路径为快照时备份磁盘路径，用virsh create启动虚拟机。

如果用virsh start启动虚拟机，需要重新对该虚拟机执行一次virsh define。

## 虚拟机迁移

先调用（获取磁盘设备名称）：

# virsh qemu-monitor-command <id>|<name>|<uuid> -hmp “migrate\_set\_capability auto-converge on”

再调用：

# virsh migrate <id>|<name>|<uuid> qemu+tcp://<des-ip>/system tcp://<des-ip> --live

说明：虚机迁移只是把虚拟机迁移到别的地方，但是存储本身不动。

## 虚拟机存储迁移

# virsh blockcopy <id>|<name>|<uuid> path dest –shallow –wait –reuse-external –pivot –verbose

该命令是针对一个磁盘的，若虚拟机有多个磁盘需要存储迁移，则执行多条命令

说明：

–shallow：若当前节点有backing\_file，则只迁移当前节点，即叶子节点

–pivot：迁移完成后，虚拟机切换为dest磁盘

–reuse-external：dest是已有文件

–verbose：动态显示迁移完成比率

如果虚拟机磁盘有父节点，需要按照下面的步骤执行迁移：

环境：storageA作为源存储，storageB作为目的存储，虚拟机名为win7，磁盘为vda\_leaf\_qcow2，父节点为vda\_root\_qcow2。

迁移父节点：

dd storage/vda\_root.qcow2 storageB/vda\_mig\_root.qcow2

准备叶子节点（目的节点的叶子节点，不涉及源节点）：

qemu-img create –f qcow2 –F qcow2 –b storage vda\_mig\_root.qcow2 stoprageB storage\_min\_leaf.qcow2

存储迁移：

virsh blockcopy win7 vda storage vda\_mig\_leaf.qcow2 –shallow –wait –resume-external –pivot

删除原磁盘（删除命令根据不同存储而不同，在此以fs为例）：

rm –f storageA/vda\_root.qcow2

rm –f storage/vda\_leaf.qcow2

注：执行存储迁移前，该虚拟机必须为未定义状态。即如果该虚机已经执行过virsh define操作，需要先执行virsh undefine操作。

## 虚拟机整机迁移