## KVM 简介

KVM 全称是 基于内核的虚拟机（Kernel-based Virtual Machine），它是Linux 的一个内核模块，该内核模块使得 Linux 变成了一个 Hypervisor。

KVM是以色列初创公司Qumranet开发，2008年9月RedHat公司收购了Qumranet。KVM是完全开源的，RedHat基于KVM的虚拟化解决方案叫做RHEV。

KVM在Linux操作系统里面以进程的形式出现，由标准的Linux调度程序进行调度，这使得KVM能够使用Linux内核的已有功能，只有一个KVM内核模块还不能实现虚拟化的全部功能，就好比操作系统只有内核还不能成为一个完整的操作系统一样。

QEMU是一个开源的虚拟化软件，纯软件，可以虚拟化所以的硬件，性能不强KVM基于QEMU开发了一个能够运行在用户空间的工具QEMU-KVM。

磁盘、网络设备等都是通过QEMU-KVM这个工具模拟出来的。

KVM和QEMU-KVM通信是通过/dev/kvm实现的。

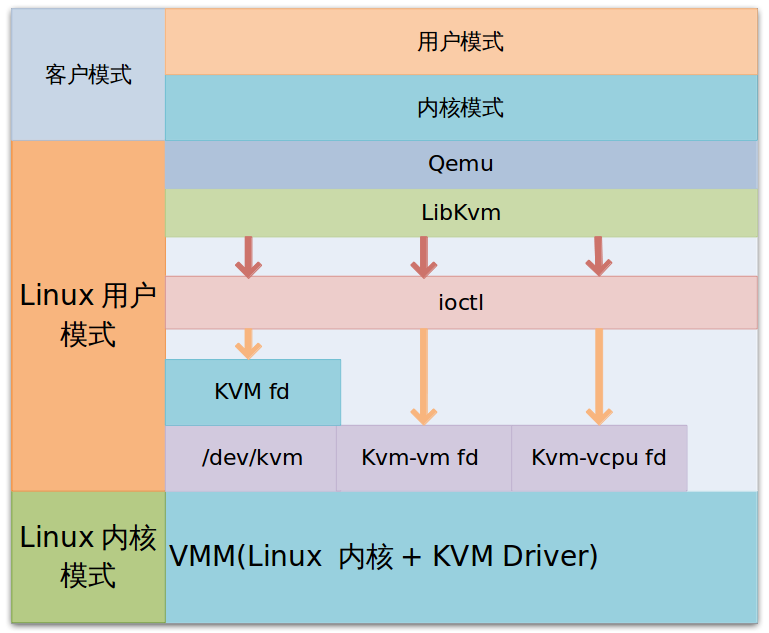
libvirt是用来管理KVM虚拟机的API，其命令为virsh。

## KVM架构

kvm基本结构有2个部分构成：

kvm 驱动，现在已经是linux kernel的一个模块了。其主要负责虚拟机的创建，虚拟内存的分配，VCPU寄存器的读写以及VCPU的运行。

另个组成是Qemu，用于模拟虚拟机的用户空间组件，提供I/O设备模型，访问外设的途径。



kvm 模块让Linux主机成为一个虚拟机监视器（VMM），并且在原有的Linux两种执行模式基础上，新增加了客户模式，客户模式拥有自己的内核模式和用户模式。在虚拟机运行时，三种模式的工作各为：

* 客户模式： 执行非I/O的客户代码，虚拟机运行在这个模式下。
* 用户模式：代表用户执行I/O指令，qemu运行在这个模式下。
* 内核模式：实现客户模式的切换，处理因为I/O或者其他指令引起的从客户模式退出（VM\_EXIT）。kvm 模块工作在这个模式下。

在kvm的模型中，每一个Gust OS都是作为一个标准的linux进程，都可以使用linux进程管理命令管理。

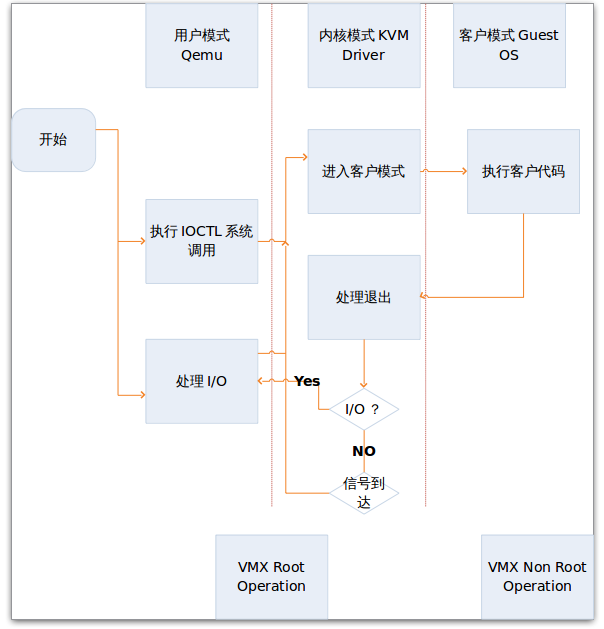
这里假如qemu通过ioctl发出KVM\_CREATE\_VM 指令，创建了一个VM后，qemu需要需要发送一些命令给VM，如KVM\_CREATE\_VCPU。这些命令当然也是通过ioctl发送的，用户程序中用ioctl发送KVM\_CREATE\_VM得到的返回值就是新创建的VM对应的fd(kvm\_vm)，fd是创建的指向特定虚拟机实例的文件描述符，之后利用这个fd发送命令给VM进行访问控制。kvm解析这些命令的函数是kvm\_vm\_ioctl。

## KVM 工作原理

### 3.1 kvm基本工作原理概述：

用户模式的qemu利用libkvm通过ioctl进入内核模式，kvm模块未虚拟机创建虚拟内存，虚拟CPU后执行VMLAUCH指令进入客户模式。加载Guest OS并执行。如果Guest OS 发生外部中断或者影子页表缺页之类的情况，会暂停Guest OS的执行，退出客户模式出行异常处理，之后重新进入客户模式，执行客户代码。如果发生I/O事件或者信号队列中有信号到达，就会进入用户模式处理。

### 3.2 KVM工作原理流程图



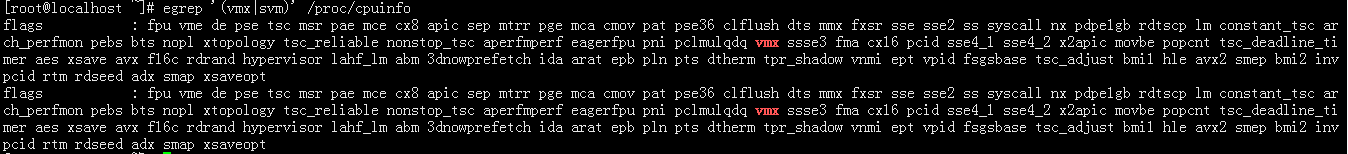
## KVM环境部署

CentOS 7.2 安装kvm环境

### 4.1 验证CPU是否支持KVM

如果结果中有vmx（Intel）或svm(AMD)字样，就说明CPU的支持的。

egrep '(vmx|svm)' /proc/cpuinfo



### 4.2 安装KVM及其依赖项

yum install virt-install

yum install libvirt

yum install bridge-utils

### 4.3 验证安装结果

lsmod | grep kvm



### 4.4 启动libvirtd

Systemctl start libvirtd

### 4.5 配置网卡

# cd /etc/sysconfig/network-scripts/

# vim ifcfg-br0

TYPE=Bridge

BOOTPROTO=static

NAME=br0

DEVICE=br0

ONBOOT=yes

IPADDR=172.30.127.252

NETMASK=255.255.255.0

GATEWAY=172.30.127.254

DNS1=114.114.114.114

# vim ifcfg-eth1

TYPE=Ethernet

BOOTPROTO=static

DEVICE=eth1

ONBOOT=yes

BRIDGE=br0

### 4.7 查看网桥状态

# brctl show

bridge name bridge id STP enabled interfaces

br0 8000.0050568c0f6d no eth1

virbr0 8000.525400b33ed2 yes virbr0-nic

### 4.8 命令行虚拟机

virt-install --name=test-kvm-vm01 --memory=1024,maxmemory=2048 --vcpus=1,maxvcpus=2 --os-type=linux --os-variant=rhel7 --location=/opt/CentOS-7-x86\_64-Minimal-1708.iso --disk path=/home/lixx/test-kvm-vm01,size=6 --bridge=br0 --graphics=none --console=pty,target\_type=serial --extra-args="console=tty0 console=ttyS0"

命令virt-install 参数：

--name=test-kvm-vm01 # 虚拟机名称

--memory=1024,maxmemory=2048 # 最小内存为1024M 最大为2048M

--vcpus=1,maxvcpus=2 # 最小最大CPU

--os-type=linux # 系统类型

--os-variant=rhel7 # 版本

--location=/opt/CentOS-7-x86\_64-Minimal-1708.iso # 镜像位置

--disk path=/kvm\_data/test-kvm-vm01.qcow2,size=10 # 虚拟机磁盘路径和大小

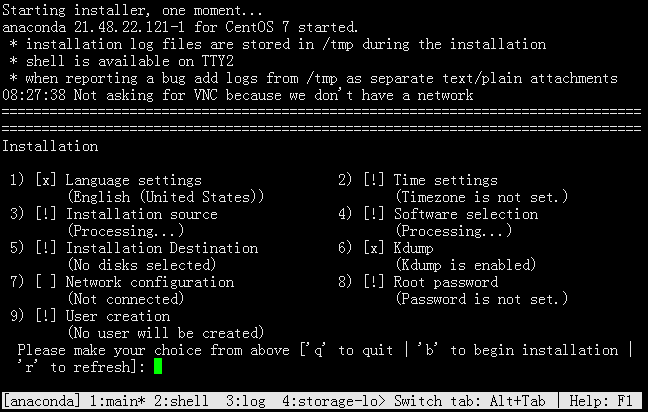
--bridge=br0 # 桥接网卡

--graphics=none # 是否为图形界面（图形界面建议使用vnc软件连接）

--console=pty,target\_type=serial # 终端的属性

--extra-args="console=tty0 console=ttyS0"

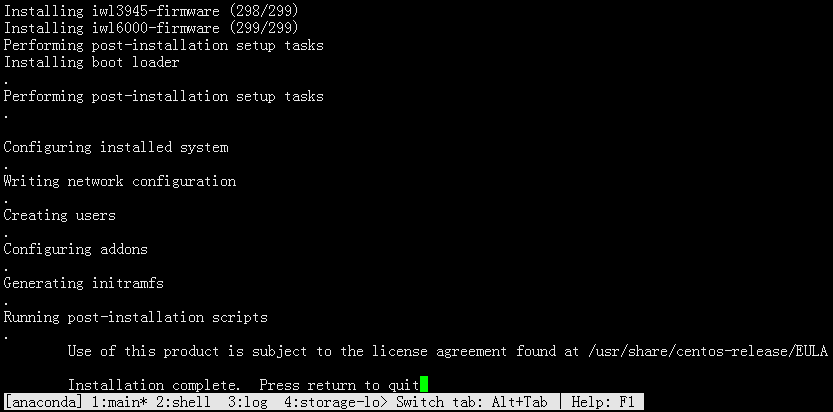
经过一段时间的硬件检查，然后到了如下这个操作界面：



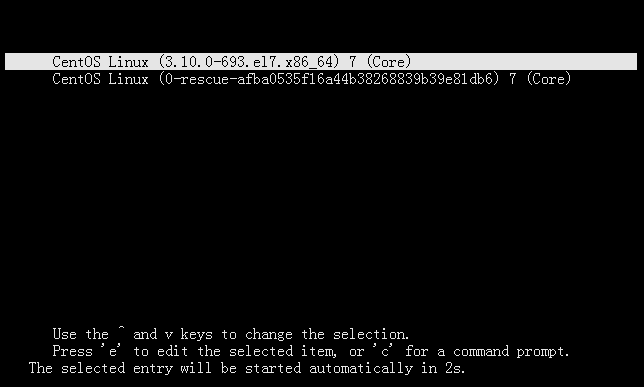
输入对应菜单的编号回车 即可进入设置。等设置完每一项，在每一项的前面对应的方括号内都会变成 [X] 注意磁盘设置的时候是选择LVM还是标准分区。

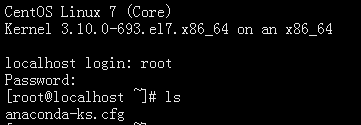


等待全部设置完毕，按 b 开始安装：



安装完成后，重启





## KVM管理

Virsh命令行

版本

# virsh -v

4.5.0

帮助

# virsh -h

virsh [options]... [<command\_string>]

virsh [options]... <command> [args...]

  options:

    -c | --connect=URI      hypervisor connection URI

    -d | --debug=NUM        debug level [0-4]

    -e | --escape <char>    set escape sequence for console

    -h | --help             this help

    -k | --keepalive-interval=NUM

                            keepalive interval in seconds, 0 for disable

    -K | --keepalive-count=NUM

                            number of possible missed keepalive messages

    -l | --log=FILE         output logging to file

    -q | --quiet            quiet mode

    -r | --readonly         connect readonly

    -t | --timing           print timing information

    -v                      short version

    -V                      long version

         --version[=TYPE]   version, TYPE is short or long (default short)

  commands (non interactive mode):

......

  (specify help <group> for details about the commands in the group)

  (specify help <command> for details about the command)

|  |  |  |
| --- | --- | --- |
| Domain Management (help keyword 'domain'): | | |
| attach-device | attach device from an XML file | 从一个XML文件附加装置 |
| attach-disk | attach disk device | 附加磁盘设备 |
| attach-interface | attach network interface | 获得网络界面 |
| autostart | autostart a domain | 自动开始一个域 |
| blkdeviotune | Set or query a block device I/O tuning parameters. | 设定或者查询块设备 I/O 调节参数。 |
| blkiotune | Get or set blkio parameters | 获取或者数值 blkio 参数 |
| blockcommit | Start a block commit operation. | 启动块提交操作。 |
| blockcopy | Start a block copy operation. | 启动块复制操作。 |
| blockjob | Manage active block operations | 管理活跃块操作 |
| blockpull | Populate a disk from its backing image. | pasting 使用其后端映像填充磁盘。 |
| blockresize | Resize block device of domain. | 创新定义域块设备大小 |
| change-media | Change media of CD or floppy drive | 更改 CD 介质或者软盘驱动器 |
| console | connect to the guest console | 连接到客户会话 |
| cpu-stats | show domain cpu statistics | 显示域 cpu 统计数据 |
| create | create a domain from an XML file | 从一个 XML 文件创建一个域 |
| define | define (but don't start) a domain from an XML file | 从一个 XML 文件定义（但不开始）一个域 |
| desc | show or set domain's description or title | 显示或者设定域描述或者标题 |
| destroy | destroy (stop) a domain | 销毁（停止）域 |
| detach-device | detach device from an XML file | 从一个 XML 文件分离设备 |
| detach-device-alias | detach device from an alias |  |
| detach-disk | detach disk device | 分离磁盘设备 |
| detach-interface | detach network interface | 分离网络接口 |
| domdisplay | domain display connection URI | 域显示连接 URI |
| domfsfreeze | Freeze domain's mounted filesystems. |  |
| domfsthaw | Thaw domain's mounted filesystems. |  |
| domfsinfo | Get information of domain's mounted filesystems. |  |
| domfstrim | Invoke fstrim on domain's mounted filesystems. | 在域挂载的文件系统中调用 fstrim。 |
| domhostname | print the domain's hostname | 输出域主机名 |
| domid | convert a domain name or UUID to domain id | 把一个域名或 UUID 转换为域 id |
| domif-setlink | set link state of a virtual interface | 设定虚拟接口的链接状态 |
| domiftune | get/set parameters of a virtual interface | 获取/设定虚拟接口参数 |
| domjobabort | abort active domain job | 忽略活跃域任务 |
| domjobinfo | domain job information | 域任务信息 |
| domname | convert a domain id or UUID to domain name | 将域 id 或 UUID 转换为域名 |
| domrename | rename a domain |  |
| dompmsuspend | suspend a domain gracefully using power management functions | 使用电源管理功能挂起域 |
| dompmwakeup | wakeup a domain from pmsuspended state | 从 pmsuspended 状态唤醒域 |
| domuuid | convert a domain name or id to domain UUID | 把一个域名或 id 转换为域 UUID |
| domxml-from-native | Convert native config to domain XML | 将原始配置转换为域 XML |
| domxml-to-native | Convert domain XML to native config | 将域 XML 转换为原始配置 |
| dump | dump the core of a domain to a file for analysis | 把一个域的内核 dump 到一个文件中以方便分析 |
| dumpxml | domain information in XML | XML 中的域信息 |
| edit | edit XML configuration for a domain | 编辑某个域的 XML 配置 |
| event | Domain Events | Domain Events |
| inject-nmi | Inject NMI to the guest | 在虚拟机中输入 NMI |
| iothreadinfo | view domain IOThreads |  |
| iothreadpin | control domain IOThread affinity |  |
| iothreadadd | add an IOThread to the guest domain |  |
| iothreaddel | delete an IOThread from the guest domain |  |
| send-key | Send keycodes to the guest | 向虚拟机发送序列号 |
| send-process-signal | Send signals to processes | 向进程发送信号 |
| lxc-enter-namespace | LXC Guest Enter Namespace | LXC 虚拟机进入名称空间 |
| managedsave | managed save of a domain state | 管理域状态的保存 |
| managedsave-remove | Remove managed save of a domain | 删除域的管理保存 |
| managedsave-edit | edit XML for a domain's managed save state file |  |
| managedsave-dumpxml | Domain information of managed save state file in XML |  |
| managedsave-define | redefine the XML for a domain's managed save state file |  |
| memtune | Get or set memory parameters | 获取或者数值内存参数 |
| perf | Get or set perf event |  |
| metadata | show or set domain's custom XML metadata |  |
| migrate | migrate domain to another host | 将域迁移到另一个主机中 |
| migrate-setmaxdowntime | set maximum tolerable downtime | 设定最大可耐受故障时间 |
| migrate-getmaxdowntime | get maximum tolerable downtime | 获取最大可耐受故障时间 |
| migrate-compcache | get/set compression cache size | 获取/设定压缩缓存大小 |
| migrate-setspeed | Set the maximum migration bandwidth | 设定迁移带宽的最大值 |
| migrate-getspeed | Get the maximum migration bandwidth | 获取最长迁移带宽 |
| migrate-postcopy | Switch running migration from pre-copy to post-copy |  |
| numatune | Get or set numa parameters | 获取或者数值 numa 参数 |
| qemu-attach | QEMU Attach | QEMU 附加 |
| qemu-monitor-command | QEMU Monitor Command | QEMU 监控程序命令 |
| qemu-monitor-event | QEMU Monitor Events | QEMU Monitor Events |
| qemu-agent-command | QEMU Guest Agent Command | QEMU 虚拟机代理命令 |
| reboot | reboot a domain | 重新启动一个域 |
| reset | reset a domain | 重新设定域 |
| restore | restore a domain from a saved state in a file | 从一个存在一个文件中的状态恢复一个域 |
| resume | resume a domain | 重新恢复一个域 |
| save | save a domain state to a file | 把一个域的状态保存到一个文件 |
| save-image-define | redefine the XML for a domain's saved state file | 为域的保存状态文件重新定义 XML |
| save-image-dumpxml | saved state domain information in XML | 在 XML 中保存状态域信息 |
| save-image-edit | edit XML for a domain's saved state file | 为域保存状态文件编辑 XML |
| schedinfo | show/set scheduler parameters | 显示/设置日程安排变量 |
| screenshot | take a screenshot of a current domain console and store it into a file | 提取当前域控制台快照并保存到文件中 |
| set-lifecycle-action | change lifecycle actions |  |
| set-user-password | set the user password inside the domain |  |
| setmaxmem | change maximum memory limit | 改变最大内存限制值 |
| setmem | change memory allocation | 改变内存的分配 |
| setvcpus | change number of virtual CPUs | 改变虚拟 CPU 的号 |
| shutdown | gracefully shutdown a domain | 关闭一个域 |
| start | start a (previously defined) inactive domain | 开始一个（以前定义的）非活跃的域 |
| suspend | suspend a domain | 挂起一个域 |
| ttyconsole | tty console | tty 控制台 |
| undefine | undefine a domain | 取消定义一个域 |
| update-device | update device from an XML file | 从 XML 文件中关系设备 |
| vcpucount | domain vcpu counts | 域 vcpu 计数 |
| vcpuinfo | detailed domain vcpu information | 详细的域 vcpu 信息 |
| vcpupin | control or query domain vcpu affinity | 控制或者查询域 vcpu 亲和性 |
| emulatorpin | control or query domain emulator affinity | 控件和查询域仿真器关联 |
| vncdisplay | vnc display | vnc 显示 |
| guestvcpus | query or modify state of vcpu in the guest (via agent) |  |
| setvcpu | attach/detach vcpu or groups of threads |  |
| domblkthreshold | set the threshold for block-threshold event for a given block device or it's backing chain element |  |
| Domain Monitoring (help keyword 'monitor'): | | |
| domblkerror | Show errors on block devices | 在块设备中显示错误 |
| domblkinfo | domain block device size information | 域块设备大小信息 |
| domblklist | list all domain blocks | 列出所有域块 |
| domblkstat | get device block stats for a domain | 获得域设备块状态 |
| domcontrol | domain control interface state | 域控制接口状态 |
| domif-getlink | get link state of a virtual interface | 获取虚拟接口链接状态 |
| domifaddr | Get network interfaces' addresses for a running domain |  |
| domiflist | list all domain virtual interfaces | 列出所有域虚拟接口 |
| domifstat | get network interface stats for a domain | 获得域网络接口状态 |
| dominfo | domain information | 域信息 |
| dommemstat | get memory statistics for a domain | 获取域的内存统计 |
| domstate | domain state | 域状态 |
| domstats | get statistics about one or multiple domains |  |
| domtime | domain time |  |
| list | list domains | 列出域 |
| Host and Hypervisor (help keyword 'host'): | | |
| allocpages | Manipulate pages pool size |  |
| capabilities | capabilities | 性能 |
| cpu-baseline | compute baseline CPU | 计算基线 CPU |
| cpu-compare | compare host CPU with a CPU described by an XML file | 使用 XML 文件中描述的 CPU 与主机 CPU 进行对比 |
| cpu-models | CPU models |  |
| domcapabilities | domain capabilities |  |
| freecell | NUMA free memory | NUMA可用内存 |
| freepages | NUMA free pages |  |
| hostname | print the hypervisor hostname | 打印管理程序主机名 |
| hypervisor-cpu-baseline | compute baseline CPU usable by a specific hypervisor |  |
| hypervisor-cpu-compare | compare a CPU with the CPU created by a hypervisor on the host |  |
| maxvcpus | connection vcpu maximum | 连接 vcpu 最大值 |
| node-memory-tune | Get or set node memory parameters | 获取或者设定节点内存参数 |
| nodecpumap | node cpu map | 节点 cpu 映射 |
| nodecpustats | Prints cpu stats of the node. | 输出节点的 cpu 状统计数据。 |
| nodeinfo | node information | 节点信息 |
| nodememstats | Prints memory stats of the node. | 输出节点的内存状统计数据。 |
| nodesuspend | suspend the host node for a given time duration | 在给定时间段挂起主机节点 |
| sysinfo | print the hypervisor sysinfo | 输出 hypervisor sysinfo |
| uri | print the hypervisor canonical URI | 打印管理程序典型的URI |
| version | show version | 显示版本 |
| Interface (help keyword 'interface'): | | |
| iface-begin | create a snapshot of current interfaces settings, which can be later committed (iface-commit) or restored (iface-rollback) | 生成当前接口设置快照，可在今后用于提交 (iface-commit) 或者恢复 (iface-rollback) |
| iface-bridge | create a bridge device and attach an existing network device to it | 生成桥接设备并为其附加一个现有网络设备 |
| iface-commit | commit changes made since iface-begin and free restore point | 提交 iface-begin 后的更改并释放恢复点 |
| iface-define | define an inactive persistent physical host interface or modify an existing persistent one from an XML file |  |
| iface-destroy | destroy a physical host interface (disable it / "if-down") | 删除物理主机接口（启用它请执行 "if-down"） |
| iface-dumpxml | interface information in XML | XML 中的接口信息 |
| iface-edit | edit XML configuration for a physical host interface | 为物理主机界面编辑 XML 配置 |
| iface-list | list physical host interfaces | 物理主机接口列表 |
| iface-mac | convert an interface name to interface MAC address | 将接口名称转换为接口 MAC 地址 |
| iface-name | convert an interface MAC address to interface name | 将接口 MAC 地址转换为接口名称 |
| iface-rollback | rollback to previous saved configuration created via iface-begin | 恢复到之前保存的使用 iface-begin 生成的更改 |
| iface-start | start a physical host interface (enable it / "if-up") | 启动物理主机接口（启用它请执行 "if-up"） |
| iface-unbridge | undefine a bridge device after detaching its slave device | 分离其辅助设备后取消定义桥接设备 |
| iface-undefine | undefine a physical host interface (remove it from configuration) | 取消定义物理主机接口（从配置中删除） |
| Network Filter (help keyword 'filter'): | | |
| nwfilter-define | define or update a network filter from an XML file | 使用 XML 文件定义或者更新网络过滤器 |
| nwfilter-dumpxml | network filter information in XML | XML 中的网络过滤器信息 |
| nwfilter-edit | edit XML configuration for a network filter | 为网络过滤器编辑 XML 配置 |
| nwfilter-list | list network filters | 列出网络过滤器 |
| nwfilter-undefine | undefine a network filter | 取消定义网络过滤器 |
| nwfilter-binding-create | create a network filter binding from an XML file |  |
| nwfilter-binding-delete | delete a network filter binding |  |
| nwfilter-binding-dumpxml | network filter information in XML |  |
| nwfilter-binding-list | list network filter bindings |  |
| Networking (help keyword 'network'): | | |
| net-autostart | autostart a network | 自动开始网络 |
| net-create | create a network from an XML file | 从一个 XML 文件创建一个网络 |
| net-define | define an inactive persistent virtual network or modify an existing persistent one from an XML file |  |
| net-destroy | destroy (stop) a network | 销毁（停止）网络 |
| net-dhcp-leases | print lease info for a given network |  |
| net-dumpxml | network information in XML | XML 中的网络信息 |
| net-edit | edit XML configuration for a network | 为网络编辑 XML 配置 |
| net-event | Network Events |  |
| net-info | network information | 网络信息 |
| net-list | list networks | 列出网络 |
| net-name | convert a network UUID to network name | 把一个网络UUID 转换为网络名 |
| net-start | start a (previously defined) inactive network | 开始一个(以前定义的)不活跃的网络 |
| net-undefine | undefine a persistent network |  |
| net-update | update parts of an existing network's configuration | 更新现有网络配置的部分 |
| net-uuid | convert a network name to network UUID | 把一个网络名转换为网络UUID |
| Node Device (help keyword 'nodedev'): | | |
| nodedev-create | create a device defined by an XML file on the node | 根据节点中的 XML 文件定义生成设备 |
| nodedev-destroy | destroy (stop) a device on the node | 销毁（停止）节点中的设备 |
| nodedev-detach | detach node device from its device driver | 将节点设备与其设备驱动程序分离 |
| nodedev-dumpxml | node device details in XML | XML 中的节点设备详情 |
| nodedev-list | enumerate devices on this host | 这台主机中中的枚举设备 |
| nodedev-reattach | reattach node device to its device driver | 重新将节点设备附加到他的设备驱动程序中 |
| nodedev-reset | reset node device | 重置节点设备 |
| nodedev-event | Node Device Events |  |
| Secret (help keyword 'secret'): | | |
| secret-define | define or modify a secret from an XML file | 定义或者修改 XML 中的 secret |
| secret-dumpxml | secret attributes in XML | XML 中的 secret 属性 |
| secret-event | Secret Events |  |
| secret-get-value | Output a secret value | secret 值输出 |
| secret-list | list secrets | 列出 secret |
| secret-set-value | set a secret value | 设定 secret 值 |
| secret-undefine | undefine a secret | 取消定义 secret |
| Snapshot (help keyword 'snapshot'): | | |
| snapshot-create | Create a snapshot from XML | 使用 XML 生成快照 |
| snapshot-create-as | Create a snapshot from a set of args | 使用一组参数生成快照 |
| snapshot-current | Get or set the current snapshot | 获取或者设定当前快照 |
| snapshot-delete | Delete a domain snapshot | 删除域快照 |
| snapshot-dumpxml | Dump XML for a domain snapshot | 为域快照转储 XML |
| snapshot-edit | edit XML for a snapshot | 编辑快照 XML |
| snapshot-info | snapshot information | 快照信息 |
| snapshot-list | List snapshots for a domain | 为域列出快照 |
| snapshot-parent | Get the name of the parent of a snapshot | 获取快照的上级快照名称 |
| snapshot-revert | Revert a domain to a snapshot | 将域转换为快照 |
| Storage Pool (help keyword 'pool'): | | |
| find-storage-pool-sources-as | find potential storage pool sources | 找到潜在存储池源 |
| find-storage-pool-sources | discover potential storage pool sources | 发现潜在存储池源 |
| pool-autostart | autostart a pool | 自动启动某个池 |
| pool-build | build a pool | 建立池 |
| pool-create-as | create a pool from a set of args | 从一组变量中创建一个池 |
| pool-create | create a pool from an XML file | 从一个 XML 文件中创建一个池 |
| pool-define-as | define a pool from a set of args | 在一组变量中定义池 |
| pool-define | define an inactive persistent storage pool or modify an existing persistent one from an XML file |  |
| pool-delete | delete a pool | 删除池 |
| pool-destroy | destroy (stop) a pool | 销毁（删除）池 |
| pool-dumpxml | pool information in XML | XML 中的池信息 |
| pool-edit | edit XML configuration for a storage pool | 为存储池编辑 XML 配置 |
| pool-info | storage pool information | 存储池信息 |
| pool-list | list pools | 列出池 |
| pool-name | convert a pool UUID to pool name | 将池 UUID 转换为池名称 |
| pool-refresh | refresh a pool | 刷新池 |
| pool-start | start a (previously defined) inactive pool | 启动一个（以前定义的）非活跃的池 |
| pool-undefine | undefine an inactive pool | 取消定义一个不活跃的池 |
| pool-uuid | convert a pool name to pool UUID | 把一个池名称转换为池 UUID |
| pool-event | Storage Pool Events |  |
| Storage Volume (help keyword 'volume'): | | |
| vol-clone | clone a volume. | 克隆卷。 |
| vol-create-as | create a volume from a set of args | 从一组变量中创建卷 |
| vol-create | create a vol from an XML file | 从一个 XML 文件创建一个卷 |
| vol-create-from | create a vol, using another volume as input | 生成卷，使用另一个卷作为输入。 |
| vol-delete | delete a vol | 删除卷 |
| vol-download | download volume contents to a file | 将卷内容下载到文件中 |
| vol-dumpxml | vol information in XML | XML 中的卷信息 |
| vol-info | storage vol information | 存储卷信息 |
| vol-key | returns the volume key for a given volume name or path | 为给定密钥或者路径返回卷密钥 |
| vol-list | list vols | 列出卷 |
| vol-name | returns the volume name for a given volume key or path | 为给定密钥或者路径返回卷名 |
| vol-path | returns the volume path for a given volume name or key | 为给定密钥或者路径返回卷路径 |
| vol-pool | returns the storage pool for a given volume key or path | 为给定密钥或者路径返回存储池 |
| vol-resize | resize a vol | 创新定义卷大小 |
| vol-upload | upload file contents to a volume | 将文件内容上传到卷中 |
| vol-wipe | wipe a vol | 擦除卷 |
|  |  |  |
| Virsh itself (help keyword 'virsh'): | | |
| cd | change the current directory | 更改当前目录 |
| echo | echo arguments | echo 参数 |
| exit | quit this interactive terminal | 退出这个非交互式终端 |
| help | print help | 打印帮助 |
| pwd | print the current directory | 输出当前目录 |
| quit | quit this interactive terminal | 退出这个非交互式终端 |
| connect | (re)connect to hypervisor | 连接（重新连接）到 hypervisor |