virsh Manage libvirt VMs

START HERE

`virsh list --all`
All VMs and their state

`virsh destroy <vm>`
Pull the plug

`virsh start <vm>`
Boot a VM

`virsh dumpxml <vm>`
Show the VM definition

`virsh console <vm>`
Attach to serial console

`virsh snapshot-create-as <vm> <name>`
Take a snapshot

`virsh shutdown <vm>`
Ask the guest to power off

`virsh domifaddr <vm>`
Find the VM's IP

CONCEPTS

As a normal user virsh talks to qemu:///session; system VMs live under qemu:///system, so add -c qemu:///system (or set LIBVIRT_DEFAULT_URI).

Connection -c qemu:///system
- **Domain (VM)** disks, NICs, CPU, RAM
  - **Network, pool, volume** shared host resources

destroy only stops a VM like a power cut; the definition stays. undefine removes the definition.

Changes to a running VM need --live, --config (next boot) or --persistent (both); without them behaviour depends on VM state.

CONNECT

`virsh -c qemu:///system <command>`
Run against the system hypervisor

`virsh -c qemu:///session <command>`
Run against your user session

`virsh -c qemu+ssh://<user>@<host>/system list --all`
Manage a remote host over SSH

`virsh`
Interactive shell
Type commands without the virsh prefix; quit exits.

`virsh -r list`
Read-only connection

`virsh uri`
Show the connection URI in use

`virsh version`
Show library and hypervisor versions

LIFECYCLE

`virsh list`
Running VMs

`virsh list --all`
Running and stopped VMs

`virsh start <vm>`
Start a defined VM

`virsh shutdown <vm>`
Graceful ACPI shutdown
Needs guest cooperation; --mode agent uses the guest agent.

`virsh reboot <vm>`
Reboot gracefully

LIFECYCLE (cont.)

`virsh destroy <vm>`
Force off immediately
Like pulling the power; the guest can lose data.

`virsh suspend <vm>`
Pause the VM in memory

`virsh resume <vm>`
Continue a paused VM

`virsh start --console <vm>`
Start and attach to console

`virsh autostart <vm>`
Start with the host
Add --disable to turn it off.

`virsh managedsave <vm>`
Save RAM to disk and stop
The next start resumes from the saved state.

`virsh list --name --state-running`
Names only, for scripting

DEFINE & INSPECT

`virsh dumpxml <vm>`
Print the VM's XML

`virsh edit <vm>`
Edit the XML in $EDITOR
Applies at the next boot.

`virsh define <file.xml>`
Create a VM from XML

`virsh undefine <vm>`
Remove the definition
Add --remove-all-storage to delete its disks; --nvram for UEFI VMs.

`virsh dominfo <vm>`
CPU, memory, state summary

`virsh dumpxml --inactive <vm>`
XML used at next start

`virsh domrename <vm> <new>`
Rename a stopped VM

`virsh domstats <vm>`
Runtime counters

`virsh create <file.xml>`
Start a transient VM from XML

CONSOLE & NETWORK ACCESS

`virsh console <vm>`
Serial console
Leave with Ctrl+]. The guest needs a getty on the serial port.

`virsh domifaddr <vm>`
IP addresses of the VM
Add --source agent or --source arp if the DHCP lease source is empty.

`virsh domiflist <vm>`
NICs with MAC and network

`virsh vncdisplay <vm>`
VNC address and port

`virsh domdisplay <vm>`
Graphics connection URI

RESOURCES

`virsh setmem <vm> <size>`
Change memory of a running VM
Size in KiB by default; 2G also works. Cannot exceed setmaxmem.

`virsh setmaxmem <vm> <size> --config`
Raise the memory ceiling

`virsh setvcpus <vm> <n> --config`
Set vCPU count for next boot

`virsh setvcpus <vm> <n> --maximum --config`
Raise the vCPU ceiling

`virsh cpu-stats <vm>`
CPU time used

DISKS & DEVICES

`virsh domblklist <vm>`
Disks and image paths

`virsh attach-disk <vm> <path> <target> --persistent`
Hot-add a disk
Target is the guest name such as vdb; add --subdriver qcow2 for qcow2 images.

`virsh detach-disk <vm> <target> --persistent`
Remove a disk

`virsh attach-device <vm> <file.xml> --persistent`
Add a device from XML

`virsh detach-device <vm> <file.xml> --persistent`
Remove a device from XML

`virsh change-media <vm> <target> <iso>`
Swap the CD image

`virsh attach-interface <vm> network <net> --persistent`
Add a NIC on a network

SNAPSHOTS

`virsh snapshot-create-as <vm> <name>`
Create a snapshot

`virsh snapshot-list <vm>`
List snapshots

`virsh snapshot-revert <vm> <name>`
Roll back to a snapshot

`virsh snapshot-delete <vm> <name>`
Delete a snapshot

`virsh snapshot-list <vm> --tree`
Show the snapshot hierarchy

`virsh snapshot-create-as <vm> <name> --disk-only`
Disk-only external snapshot

NETWORKS

`virsh net-list --all`
List virtual networks

`virsh net-start <net>`
Start a network
If default is inactive, VMs on it will not start.

`virsh net-autostart <net>`
Start network at boot

`virsh net-dhcp-leases <net>`
Leases handed out

`virsh net-info <net>`
Autostart, bridge, state

`virsh net-edit <net>`
Edit network XML

`virsh net-destroy <net>`
Stop a network

`virsh net-define <file.xml>`
Define a network

STORAGE

`virsh pool-list --all`
List storage pools

`virsh pool-info <pool>`
Capacity and state

`virsh vol-list <pool>`
Volumes in a pool

`virsh vol-create-as <pool> <name> 20G --format qcow2`
Create a disk image

`virsh vol-path <vol> --pool <pool>`
Path of a volume

`virsh vol-clone <vol> <new> --pool <pool>`
Copy a volume

`virsh vol-resize <vol> <size> --pool <pool>`
Grow a volume

RECIPES

Find a VM's IP and log in

```
virsh -c qemu:///system
domifaddr <vm>
ssh <user>@<ip>
```

Clean snapshot before an upgrade

```
virsh shutdown <vm>
virsh snapshot-create-as <vm>
pre-upgrade
virsh start <vm>
# if it goes wrong:
virsh snapshot-revert <vm> pre-
upgrade
```

Add a data disk

```
virsh vol-create-as default
<vm>-data 10G --format qcow2
virsh attach-disk <vm> /var/
lib/libvirt/images/<vm>-data \
  vdb --subdriver qcow2 --
persistent
```

Delete a VM and its disks

```
virsh destroy <vm>
virsh undefine <vm> --remove-
all-storage
```