# virsh Manage libvirt VMs

## 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.

## CONNECT

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

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

`virsh`
Interactive shell

## 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

`virsh reboot <vm>`
Reboot gracefully

`virsh destroy <vm>`
Force off immediately

`virsh autostart <vm>`
Start with the host

## DEFINE & INSPECT

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

`virsh edit <vm>`
Edit the XML in $EDITOR

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

## DEFINE & INSPECT (cont.)

`virsh undefine <vm>`
Remove the definition

## CONSOLE & NETWORK ACCESS

`virsh console <vm>`
Serial console

`virsh domifaddr <vm>`
IP addresses of the VM

## RESOURCES

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

## DISKS & DEVICES

`virsh domblklist <vm>`
Disks and image paths

`virsh attach-disk <vm> <path> <target> --persistent`
Hot-add a disk

## 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

## NETWORKS

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

`virsh net-start <net>`
Start a network

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

## STORAGE

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

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

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