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.

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

CONNECT (cont.)

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

virsh
Interactive shell

virsh -r list
Read-only connection

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 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

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

virsh undefine *<vm>*
Remove the definition

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

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

CONSOLE & NETWORK ACCESS

virsh console *<vm>*
Serial console

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

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

RESOURCES

virsh setmem *<vm> <size>*
Change memory of a running VM

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

virsh detach-disk *<vm> <target>* **--persistent**
Remove 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

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

NETWORKS

virsh net-list --all
List virtual networks

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

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

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

NETWORKS (cont.)

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

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

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

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
```

Delete a VM and its disks

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