QEMU Machine emulator and virtualizer

CONCEPTS

One binary per CPU architecture (qemu-system-x86_64, -aarch64, ...); without -accel kvm the guest is emulated and slow.

Host Linux, /dev/kvm

qemu-system-* one process per VM

Machine -machine, -cpu, -m, -smp

Devices -drive, -netdev, -device

The guest's disk is a file on the host; qcow2 grows on demand and supports snapshots and backing files.

RUN A VM

qemu-system-x86_64 -hda *<disk>*
Boot from a disk image

-accel kvm
Hardware acceleration (Linux)

-cpu host
Pass the host CPU model through

-m *<size>*
Guest RAM, e.g. 2G

-smp *<n>*
Number of virtual CPUs

-machine q35
Pick a modern chipset

DISKS AND BOOT

-drive file=*<disk>***,format=qcow2,if=virtio**
Attach a virtio disk

-cdrom *<iso>*
Attach an optical image

-boot d
Boot from the CD first

-boot order=*<cdn>*
Set boot device order

-snapshot
Discard disk writes on exit

-drive if=pflash,format=raw, readonly=on,file=*<OVMF_CODE.fd>*
UEFI firmware (OVMF)

DISPLAY AND CONSOLE

-nographic
No window; serial on the terminal

-serial mon:stdio
Serial and monitor on the terminal

NETWORK

-nic user
Simple NAT, no setup (default)

-nic user,hostfwd=::*<host>*-:*<guest>*
Forward a host port to the guest

-nic tap,script=no,ifname=*<tap>*
Attach to a host tap device

QEMU-IMG

qemu-img create -f qcow2 *<disk>* *<size>*
New disk image

qemu-img info *<disk>*
Show format and sizes

qemu-img convert -O qcow2 *<in>* *<out>*
Convert between formats

qemu-img resize *<disk>* **+***<size>*
Grow an image

qemu-img create -f qcow2 -b *<base>* **-F qcow2** *<overlay>*
Overlay on a backing file

Never run qemu-img write commands on an image a running VM uses; info -U is the read-only exception.

MONITOR

Ctrl+a c
Switch console / monitor (mon:stdio)

Ctrl+a x
Exit the emulator

info status
Running or paused?

savevm *<tag>*
Save full VM state (needs qcow2)

loadvm *<tag>*
Restore VM state