

Table of Contents

01	qemu-img	Create and manage virtual machine disk images
02	virsh	KVM/libvirt virtual machine management CLI
03	virt-install	Create and install KVM virtual machines

Part 1 of 1 - Explore all 232+ Linux commands at dargslan.com/learn/linux-commands

Each command includes syntax, options, practical examples with output, and pro tips.

\$ qemu-img

Advanced

Create and manage virtual machine disk images

The `qemu-img` command creates, converts, and modifies virtual machine disk images. It supports numerous image formats including `qcow2` (QEMU native, with snapshots and compression), `raw` (maximum performance), `VMDK` (VMware), `VHD/VHDX` (Hyper-V), and `VDI` (VirtualBox).

`qemu-img` is essential for KVM/QE...

Options & Flags

<code>create</code>	Create a new disk image
<code>convert</code>	Convert between image formats
<code>resize</code>	Resize a disk image
<code>info</code>	Show image metadata and size info
<code>snapshot</code>	Manage internal snapshots
<code>check</code>	Check image for errors
<code>compare</code>	Compare two disk images
<code>-f FORMAT</code>	Source image format (raw, qcow2, vmdk, vhd)
<code>-O FORMAT</code>	Output format for conversion
<code>-o OPTIONS</code>	Format-specific options

Practical Examples

Example: Create a qcow2 disk image

```
$ qemu-img create -f qcow2 /var/lib/libvirt/images/vm-disk.qcow2 50G
Formatting 'vm-disk.qcow2', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=53687091200
```

Create a 50GB `qcow2` image. Thin-provisioned - actual file size starts near zero and grows with usage.

Example: Check image info

```
$ qemu-img info /var/lib/libvirt/images/vm-disk.qcow2
image: vm-disk.qcow2\nfile format: qcow2\nvirtual size: 50 GiB\ndisk size: 2.1 GiB
```

Show format, virtual size, actual disk usage, cluster size, and snapshot info.

Example: Convert VMware to KVM

```
$ qemu-img convert -f vmdk -O qcow2 -p vmware-disk.vmdk kvm-disk.qcow2
```

Convert a VMware VMDK image to `qcow2` for KVM. `-p` shows progress during conversion.

Example: Convert KVM to Hyper-V

```
$ qemu-img convert -f qcow2 -O vhdx kvm-disk.qcow2 hyperv-disk.vhdx
```

Convert a `qcow2` image to VHDX format for Microsoft Hyper-V.

Example: Resize a disk image

```
$ qemu-img resize /var/lib/libvirt/images/vm-disk.qcow2 +20G
Image resized.
```

Add 20GB to the virtual disk. Guest OS needs to extend the partition: `growpart + resize2fs` (or `xfs_growfs`).

Tips & Best Practices

Pro Tip: Always use `qcow2` for KVM: `qcow2` supports thin provisioning, snapshots, compression, and encryption. Use `raw` only when maximum I/O performance is critical and you do not need these features.

Warning: Stop VM before resize: Always shut down the VM before resizing its disk image. Online resize is possible but riskier. After resizing, boot the VM and extend the guest filesystem.

Note: Thin provisioning: qcow2 images are thin-provisioned: a 50GB image may only use 2GB on disk. Use `qemu-img info` to see actual vs virtual size.

Pro Tip: Backing files for dev environments: Create one base image, then use backing files for clones. Each VM only stores its differences - perfect for creating many test VMs from a single template.

\$ virsh

Advanced

KVM/libvirt virtual machine management CLI

The virsh command is the primary command-line tool for managing virtual machines through the libvirt virtualization API. It controls KVM/QEMU virtual machines - creating, starting, stopping, migrating, and monitoring VMs on Linux servers.

libvirt provides a unified management layer for multiple ...

Options & Flags

<code>list --all</code>	List all VMs (running and stopped)
<code>start VM</code>	Start a virtual machine
<code>shutdown VM</code>	Gracefully shut down a VM
<code>destroy VM</code>	Force stop a VM (like pulling power)
<code>console VM</code>	Connect to VM serial console
<code>define XML</code>	Define a VM from XML configuration
<code>snapshot-create-as</code>	Create a named snapshot
<code>snapshot-revert</code>	Revert VM to a snapshot
<code>dominfo VM</code>	Show VM information (CPU, RAM, state)
<code>migrate</code>	Live migrate VM to another host

Practical Examples

Example: List all virtual machines

```
$ virsh list --all
 Id   Name                               State\n-----\n 1    webserver                          running\n -    database                            shut off\n -    testvm
```

Show all defined VMs with state (running, shut off, paused).

Example: Start and connect to VM

```
$ virsh start webserver && virsh console webserver
```

Start a VM and attach to its serial console. Exit console with Ctrl+] (right bracket).

Example: Create VM from command line

```
$ virt-install --name testvm --ram 2048 --vcpus 2 --disk size=20 --os-variant ubuntu22.04 --cdrom /iso/ubuntu-22.04.iso --network 1
```

Create and install a new VM using virt-install (companion to virsh). Starts installation automatically.

Example: Create a snapshot

```
$ virsh snapshot-create-as webserver "pre-upgrade" "Before kernel upgrade" && virsh snapshot-list webserver
```

Create a named snapshot before making changes. Revert if something goes wrong.

Example: Revert to snapshot

```
$ virsh snapshot-revert webserver pre-upgrade && virsh start webserver
```

Roll back VM to the pre-upgrade snapshot. VM state is restored to the exact point of the snapshot.

Tips & Best Practices

Warning: destroy is NOT delete: virsh destroy only force-stops the VM (like pulling the power cord). It does NOT delete the VM or its disk. Use virsh undefine --remove-all-storage to actually delete a VM.

Pro Tip: Use autostart for critical VMs: Enable auto-start: `virsh autostart webserver`. The VM will start automatically after host reboot.

Note: XML configuration: `virsh dumpxml VM` shows the complete XML config. Edit with `virsh edit VM` (opens in `$EDITOR`). XML defines CPU, memory, disks, NICs, and all VM hardware.

Pro Tip: Monitor VM performance: Use `virsh domstats VM` for detailed performance metrics, or `virt-top` for a top-like view of all VMs resource usage.

\$ virt-install

Advanced

Create and install KVM virtual machines

The virt-install command creates new KVM/QEMU virtual machines and begins the guest OS installation process. It provisions VM resources (CPU, memory, disk, network), connects installation media (ISO, network, PXE), and starts the installer - all from the command line.

virt-install is the standar...

Options & Flags

<code>--name NAME</code>	Set the VM name
<code>--ram SIZE</code>	Set memory in MB (or --memory)
<code>--vcpus N</code>	Set number of virtual CPUs
<code>--disk</code>	Define storage (size, path, format)
<code>--cdrom FILE</code>	Attach ISO for installation
<code>--os-variant NAME</code>	Optimize for specific OS
<code>--network</code>	Configure network (bridge, NAT, etc.)
<code>--graphics</code>	Set display type (vnc, spice, none)
<code>--noautoconsole</code>	Do not automatically open console
<code>--import</code>	Import existing disk (skip installation)

Practical Examples

Example: Create VM from ISO

```
$ virt-install --name webserver --ram 2048 --vcpus 2 --disk size=20,format=qcow2 --cdrom /iso/ubuntu-22.04-server.iso --os-variant ubuntu22.04
```

Create a 2GB RAM, 2 vCPU VM with 20GB disk. Boots from Ubuntu ISO with VNC console for installation.

Example: Create from cloud image

```
$ virt-install --name cloudvm --ram 2048 --vcpus 2 --import --disk path=/images/ubuntu-cloud.qcow2 --os-variant ubuntu22.04 --network bridge=br0
```

Import a pre-built cloud image with cloud-init for initial configuration. No installation needed.

Example: Minimal headless server VM

```
$ virt-install --name dbserver --ram 4096 --vcpus 4 --disk size=50 --location https://mirror.example.com/ubuntu/dists/jammy/main/installer-amd64/iso/iso.img
```

Network install with serial console only - fully headless. Perfect for server VMs.

Example: Automated install with kickstart

```
$ virt-install --name rhelvm --ram 2048 --vcpus 2 --disk size=30 --location /iso/RHEL-9.iso --os-variant rhel9.0 --initrd-inject /kickstart.ks
```

Unattended RHEL installation using a kickstart file. VM installs itself without user interaction.

Example: VM with multiple disks

```
$ virt-install --name fileserver --ram 4096 --vcpus 2 --disk size=20 --disk path=/images/data.qcow2,size=100 --cdrom /iso/ubuntu.22.04-server.iso
```

Create VM with two disks: 20GB for OS and 100GB for data storage.

Tips & Best Practices

Pro Tip: Always use --os-variant: os-variant optimizes VM hardware for the guest OS (virtio drivers, bus types, timer settings). List options: `osinfo-query os` or `virt-install --osinfo list`

Note: Cloud images save time: Use pre-built cloud images with `--import` instead of installing from ISO. Download from cloud-images.ubuntu.com or similar. Combined with `cloud-init`, VMs are ready in seconds.

Warning: Bridge vs NAT networking: Use `--network bridge=br0` for VMs that need to be accessible from the network. Default NAT (`--network default`) only allows outbound connections from the VM.

Pro Tip: Use `--noautoconsole` for scripting: Add `--noautoconsole` when running `virt-install` in scripts. Otherwise the command blocks waiting for you to interact with the console.

Ready for more? Explore 200+ professional IT eBooks

Go deeper with comprehensive guides, hands-on projects, and real-world examples

dargslan.com/books