Proxmox VE Virtualization platform CLI

START HERE

`qm list`
VMs on this node

`pct enter <vmid>`
Root shell in a container

`pct list`
Containers on this node

`qm config <vmid>`
Show a VM's settings

`qm start <vmid>`
Boot a VM

`vzdump <vmid> --storage <storage>`
Back up a guest

`qm shutdown <vmid>`
Clean ACPI power-off

`pvesm status`
Storages and free space

CONCEPTS

qm manages QEMU/KVM VMs, pct manages LXC containers. Both share one ID space: every guest has a unique numeric VMID (100 and up) across the cluster.

Cluster pvecm, /etc/pve shared
- **Node** one PVE host
  - **Guest** VM (qm) or CT (pct), by VMID
    - **Volume** <storage>:<volume> on a storage

Disks, ISOs, templates and backups are addressed by volume ID, storage:name, e.g. local-lvm:vm-100-disk-0 or local:iso/<file>.iso.

Guest configs are plain files: /etc/pve/qemu-server/<vmid>.conf and /etc/pve/lxc/<vmid>.conf. Prefer qm set / pct set over editing them.

VM LIFECYCLE

`qm list`
List VMs with status, memory, PID

`qm start <vmid>`
Start a VM

`qm shutdown <vmid>`
Ask the guest to shut down
Sends ACPI; --timeout <s> --forceStop 1 kills it if it hangs.

`qm stop <vmid>`
Hard stop, like pulling the plug

`qm reboot <vmid>`
Clean reboot

`qm status <vmid>`
Running or stopped

`qm unlock <vmid>`
Clear a stale lock
For a lock left by a crashed backup, snapshot or migration.

`qm destroy <vmid> --purge`
Delete the VM and its disks
--purge also drops it from backup jobs, HA and replication.

VM CONSOLE & GUEST AGENT

`qm terminal <vmid>`
Attach to the serial console
Needs a serial port (qm set <vmid> --serial0 socket). Exit with Ctrl+O.

`qm set <vmid> --agent enabled=1`
Enable the QEMU guest agent
The guest must run qemu-guest-agent too.

VM CONSOLE & GUEST AGENT (cont.)

`qm guest cmd <vmid> network-get-interfaces`
Guest IP addresses via the agent

VM CONFIGURATION

`qm config <vmid>`
Show the VM's settings
--current shows the running values instead of pending ones.

`qm set <vmid> --memory <MiB>`
Set RAM

`qm set <vmid> --cores <n>`
Set CPU cores

`qm set <vmid> --onboot 1`
Start with the node

`qm set <vmid> --net0 virtio,bridge=vmbr0`
Add or change a NIC

`qm pending <vmid>`
Changes waiting for a restart

`qm create <vmid> --name <name> --memory 2048`
Create an empty VM
Add --net0 virtio,bridge=vmbr0 and a disk such as --scsi0 <storage>:32.

`qm disk resize <vmid> scsi0 +10G`
Grow a disk
Without + the size is absolute. Grow the partition inside the guest afterwards.

`qm disk import <vmid> <image> <storage>`
Import an image as a new disk
Lands as unusedN; attach it with qm set --scsi0 or use --target-disk.

TEMPLATES, CLONES & SNAPSHOTS

`qm snapshot <vmid> <name>`
Take a snapshot
Add --vmstate 1 to include RAM of a running VM.

`qm listsnapshot <vmid>`
List snapshots

`qm rollback <vmid> <name>`
Revert to a snapshot

`qm delsnapshot <vmid> <name>`
Delete a snapshot

`qm template <vmid>`
Turn a VM into a template
One way only; templates cannot be started.

`qm clone <vmid> <newid> --name <name>`
Clone a VM or template
From a template the default is a linked clone; --full 1 copies the disks.

CONTAINERS (PCT)

`pct list`
List containers

`pct create <vmid> <template> --rootfs <storage>:<GiB>`
Create a container
<template> is a volume ID like local:vztmpl/<file>. Add --hostname, --net0 and --unprivileged 1.

`pct start <vmid>`
Start a container

`pct enter <vmid>`
Root shell inside

`pct shutdown <vmid>`
Clean shutdown

`pct stop <vmid>`
Kill the container

`pct exec <vmid> -- <cmd>`
Run one command inside

`pct config <vmid>`
Show the container's settings

`pct set <vmid> --memory <MiB> --cores <n>`
Change resources

`pct set <vmid> -mp0 <storage>:<GiB>,mp=<path>`
Add a mount point volume

`pct push <vmid> <file> <dest>`
Copy a file into it

CONTAINER TEMPLATES (PVEAM)

`pveam update`
Refresh the template index

`pveam available --section system`
List downloadable OS templates

`pveam download <storage> <template>`
Download a template

STORAGE (PVESM)

`pvesm status`
Storages with type and usage

`pvesm list <storage>`
Volumes on a storage

BACKUP & RESTORE

`vzdump <vmid> --storage <storage>`
Back up a VM or container
Default mode is snapshot: live, no downtime.

`vzdump <vmid> --mode stop --compress zstd`
Consistent backup, stops the guest

BACKUP & RESTORE (cont.)

`pvesm list <storage> --content backup`
List backup archives

`qmrestore <archive> <vmid> --storage <storage>`
Restore a VM backup
Restoring onto an existing VMID needs --force 1. --unique 1 makes new MACs.

`pct restore <vmid> <archive> --storage <storage>`
Restore a container backup

MIGRATION

`qm migrate <vmid> <node>`
Move a stopped VM to another node

`qm migrate <vmid> <node> --online`
Live-migrate a running VM

CLUSTER & API

`pvecm status`
Quorum and membership

`pvecm nodes`
Cluster nodes and votes

`pvecm create <name>`
Create a cluster on this node

`pvesh get /cluster/resources --type vm`
Every guest in the cluster

`pveversion`
Proxmox VE version

RECIPES

Cloud-init VM template

```
qm create 9000 --name tpl --memory 2048 --net0 virtio,bridge=vmbr0 --scsihw virtio-scsi-pci
qm set 9000 --scsi0 local-lvm:0,import-from=/root/<cloudimg>.img
qm set 9000 --ide2 local-lvm:cloudinit --boot order=scsi0
qm set 9000 --serial0 socket --vga serial0
qm template 9000
```

New VM from the template

```
qm clone 9000 123 --name web1 --full 1
qm set 123 --sshkey ~/.ssh/id_rsa.pub
qm set 123 --ipconfig0 ip=10.0.10.123/24,gw=10.0.10.1
qm start 123
```

Debian container from scratch

```
pveam update
pveam download local <debian-template>.tar.zst
pct create 200 local:vztmpl/<debian-template>.tar.zst \
  --hostname ct1 --rootfs local-lvm:8 --unprivileged 1 \
  --net0 name=eth0,bridge=vmbr0,ip=dhcp \
  --password
pct start 200 && pct enter 200
```